



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

## **PRIMOS® Commands Reference Guide**

**Revision 23.3**

**DOC3108-8LA**



# PRIMOS<sup>®</sup> Commands Reference Guide

*Eighth Edition*

**Douglas Gilbert**

*This manual documents the software operation of the PRIMOS<sup>®</sup> operating system on 50 Series computers and their supporting systems and utilities as implemented at Master Disk Revision Level 23.3 (Rev. 23.3).*

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.

Copyright © 1992 by Prime Computer, Inc. All rights reserved.

PRIME, PRIME, PRIMOS, and the Prime logo are registered trademarks of Prime Computer, Inc. 50 Series, 400, 750, 850, 2250, 2350, 2450, 2455, 2550, 2655, 2755, 2850, 2950, 4050, 4150, 4450, 5310, 5320, 5330, 5340, 6150, 6350, 6450, 6550, 6650, 9650, 9655, 9750, 9755, 9950, 9955, 9955II, DISCOVER, PRIME EXLNET, Prime INFORMATION CONNECTION, PRIME/SNA, PRIME EXL, PRIME EXL MBX, INFO/BASIC, MIDAS, MIDASPLUS, PERFORM, PERFORMER, PRIFORMA, PRIMAN, Prime INFORMATION, INFORM, PRISAM, Prime INFORMATION PLUS, PRIMELINK, PRIMIX, PRIMENET, PRIMEWAY, PRODUCER, PRIMEWORD, Prime INFORMATION EXL, Prime INFORMATION/pc, PRIME TIMER, RINGNET, SIMPLE, PT25, PT45, PT46, PT65, PT200, PT250, and PST 100 are trademarks of Prime Computer, Inc. PrimeService is a service mark of Prime Computer, Inc.

Avant Garde, Bookman, Zapf Chancery, and Zapf Dingbats are registered trademarks of International Typeface Corporation (ITC).

Ethernet is a registered trademark of Xerox Corp.

Helvetica, Palatino, and Times are registered trademarks of Linotronic Corporation.

HP and LaserJet are registered trademarks of Hewlett-Packard Company.

IBM is a registered trademark of International Business Machines Corp.

PostScript is a registered trademark of Adobe Systems, Inc.

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

## RESTRICTED RIGHTS LEGEND

Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013.

First Edition (FDR3108-101) January 1979 for Revision 16.3

Second Edition (FDR3108-101) May 1980 for Revision 17.2

Third Edition (FDR3108-101) January 1981 for Revision 18.1

Fourth Edition (FDR3108-101) July 1982 for Revision 19.0

Fifth Edition (DOC3108-5LA) March 1985 for Revision 19.4

Sixth Edition (DOC3108-6LA) July 1987 for Revision 21.0

Seventh Edition (DOC3108-7LA) October 1988 for Revision 22.0

Updated by PRIMOS Commands Reference Guide Release Notes (RLN3108-71A)

July 1989 for Revision 22.1

Updated by PRIMOS User's Release Document (DOC10316-1PA) June 1990 for Revision 23.0

Eighth Edition (DOC3108-8LA) April 1992 for Revision 23.3

*Design:* Douglas Gilbert

*Editorial:* Mary Skousgaard, Judy Gordon

*Illustration:* Elizabeth Wahle, Douglas Gilbert

*Production:* Judy Gordon

## How to Order Technical Documents

To order copies of documents, or to obtain a catalog and price list

- United States customers call Prime Telemarketing, toll free, at  
**1-800-827-7463**

Monday through Thursday, 8:30 a.m. to 8:00 p.m., and  
Friday, 8:30 a.m. to 6:00 p.m. (EST).

- International customers contact your local Prime subsidiary or distributor.

## PrimeService<sup>SM</sup>

To obtain service for Prime systems

- United States customers call toll free at  
**1-800-800-PRIME**
- International customers contact your Prime representative.

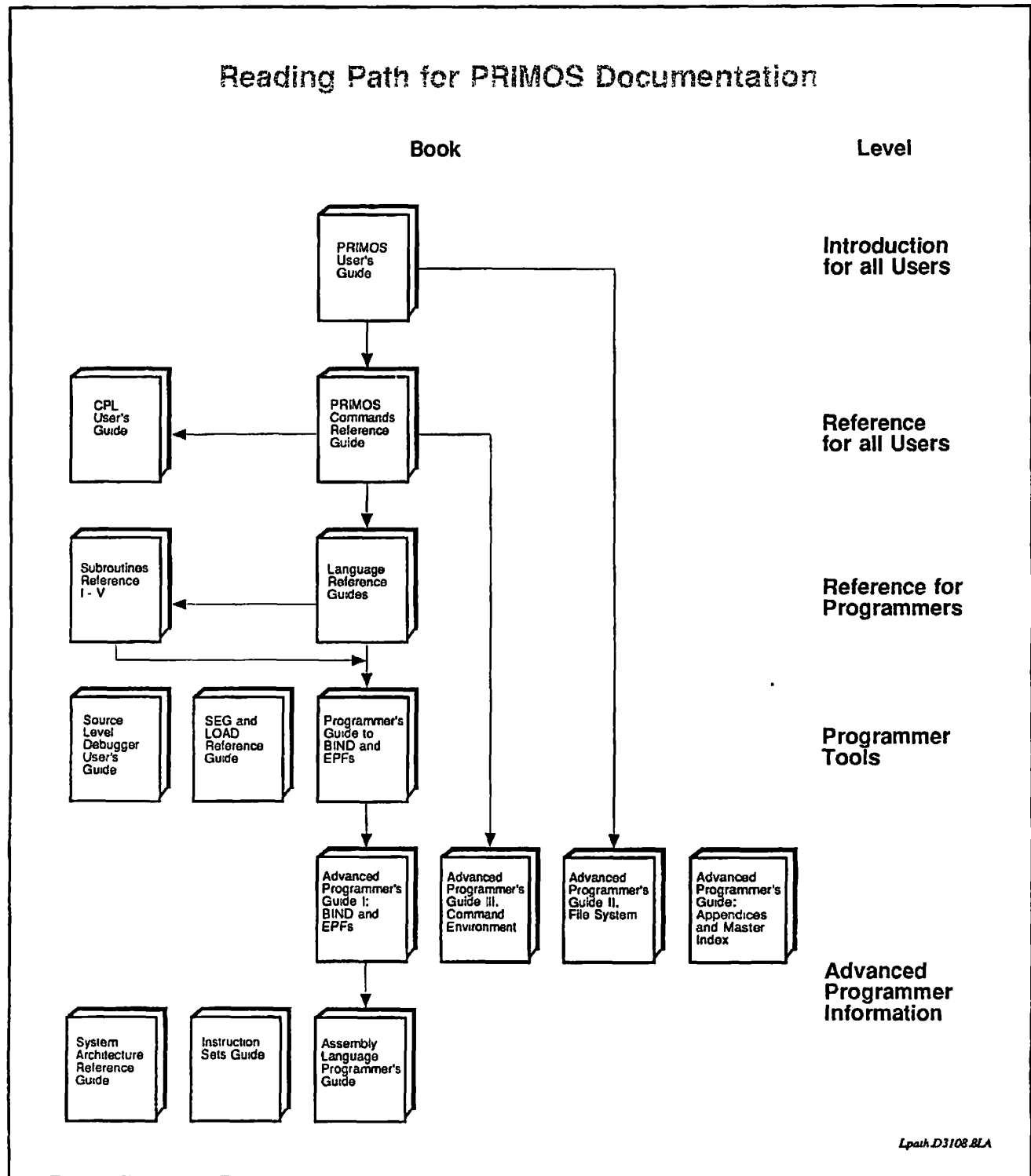
## Surveys and Correspondence

Please comment on this manual using the Reader Response Form provided in the back of this book. Address any additional comments on this or other Prime documents to

Technical Publications Department  
Prime Computer, Inc.  
500 Old Connecticut Path  
Framingham, MA 01701



# Reading Path for PRIMOS Documentation



Lpath.D3108.8LA

# Contents



## About This Book

### 1 PRIMOS Commands Overview

- PRIMOS Command-line Format . . . 1-1
  - Using Pathnames and Entrynames as Arguments . . . 1-2
  - Length of Command Lines . . . 1-2
  - Case of Command-line Words . . . 1-2
- Advanced Command-line Functionality . . . 1-2
- The Root Directory (Rev. 23.0 and Higher) . . . 1-3
  - Common File System Name Space . . . 1-3
- Pathnames . . . 1-4
  - Absolute Pathname . . . 1-4
  - Full Pathname . . . 1-5
  - Relative Pathname . . . 1-5
  - Pathname Qualification . . . 1-5
  - Entryname . . . 1-6
  - Basename and Suffix . . . 1-6
- Internal and External PRIMOS Commands . . . 1-7
- List of User Commands . . . 1-9
- Summary of User Commands by Function . . . 1-11
  - Establishing Your Identity and Accessing the System . . . 1-12
  - Managing Directories . . . 1-12
  - Protecting Directories . . . 1-13
  - Managing Files . . . 1-13
  - Editors and Text Handlers . . . 1-15
  - Compilers, Translators, and Interpreters . . . 1-15
  - Linking and Executing Programs . . . 1-16
  - Debuggers . . . 1-17
  - Job Processors . . . 1-17
  - Setting Your Terminal's Characteristics . . . 1-17
  - Defining Your Command Environment . . . 1-18
  - Displaying System Information . . . 1-19
  - Information on Your Current Environment . . . 1-19
  - Information on Program and Command Execution . . . 1-20
  - Sending Messages . . . 1-21

- Terminal I/O Handling . . . 1-21
- I/O Device Handling . . . 1-21
- Communications . . . 1-22
- Data Management . . . 1-23
- OAS (Office Automation System) . . . 1-25
- CAD/CAM . . . 1-25
- System Settings . . . 1-25
- Summary of Command Functions . . . 1-26
  - Arithmetic Functions . . . 1-26
  - File System Functions . . . 1-26
  - String-handling Functions . . . 1-27
  - Miscellaneous Functions . . . 1-27

## **2 Dictionary of PRIMOS User Commands**

## **3 Dictionary of Command Functions**

## **4 Command-line Features**

- Introduction . . . 4-1
- Global Variables . . . 4-2
  - Supplying Long Pathnames or Arguments . . . 4-3
  - Supplying Variables to Command Functions . . . 4-3
- Multiple Commands . . . 4-3
- Iteration . . . 4-4
  - Multiple Iteration Lists . . . 4-5
  - Iteration Lists as Parts of Arguments . . . 4-5
  - Cross-product Iteration Lists . . . 4-6
- Wildcarding . . . 4-6
  - Wildcard Matching . . . 4-7
  - Examples of Wildcard Names . . . 4-8
  - Inverted Matching . . . 4-9
  - Wildcard Options . . . 4-9
  - Type-designation Options . . . 4-11
  - Date-selection Options . . . 4-11
  - Verification Options . . . 4-13
- Treewalking . . . 4-15
  - Examples of Treewalking . . . 4-16
  - Treewalking Options . . . 4-18
  - Examples of Treewalking Using Options . . . 4-19
- Name Generation . . . 4-20
  - Requirements for Name Generation . . . 4-21
  - Source Pathname . . . 4-21

Generation Patterns . . .	4-21
Examples of Name Generation . . .	4-22
Adding Components . . .	4-23
Deleting Components . . .	4-23
Combining Additions, Substitutions, and Deletions . . .	4-23
Combining Command-line Features . . .	4-24
Combining Iteration With Other Features . . .	4-24
Wild Characters and Name Generation . . .	4-24
Syntax Suppression . . .	4-24

## **5 Command-line Processing**

Introduction . . .	5-1
Command-line Processing Order . . .	5-1
Expanding Abbreviations . . .	5-3
Suppressing Syntax . . .	5-4
Processing Multiple Commands . . .	5-4
Evaluating Variables and Functions . . .	5-5
Identifying Iteration Lists . . .	5-5
Implementing Treewalking . . .	5-6
Matching Wildcard Names . . .	5-6
Matching Name Generation Patterns . . .	5-7
Executing the Command . . .	5-7
Example of Command-line Processing . . .	5-7
Processing Steps . . .	5-8
Terminal Display . . .	5-15

## **6 Command-line Editing (ECL)**

Introduction . . .	6-1
Control Characters . . .	6-1
ECL Command Syntax . . .	6-2
The Kill Ring . . .	6-2
The Search Ring . . .	6-2
Suspending the PRIMOS Break Character . . .	6-2
ECL as an EPF Command Function . . .	6-3
ECL and Command Output (COMO) Files . . .	6-3
Automatic Pathname Completion . . .	6-3
How ECL Completes Pathnames . . .	6-3
Replacing Pathnames . . .	6-5
Treewalk Wildcard Expansion . . .	6-5
Example Session . . .	6-5
ECL Key Bindings . . .	6-9
Syntax of the Bindings File . . .	6-9
sequence Argument . . .	6-9

command Argument . . .	6-10
program Argument . . .	6-10
terminal_list Argument . . .	6-10
Key Binding Sequences . . .	6-11
Control Sequences With Printing Characters . . .	6-11
Other Nonprinting Character Sequences . . .	6-11
Commands Useful for ECL Key Bindings . . .	6-12
Programmable Function Commands . . .	6-13
How the PF Commands Work . . .	6-14
Multiple Command Sequences . . .	6-14
Self-terminating Function Keys . . .	6-15
Sample Bindings File . . .	6-17
Bindings for Terminals . . .	6-18
Compiling, Saving, and Restoring the Bindings File . . .	6-25
Compiling the Bindings File . . .	6-25
Reestablishing Fundamental Bindings . . .	6-26
Saving a Compiled File . . .	6-26
Restoring a Compiled File . . .	6-26
Saving and Restoring Simultaneously . . .	6-27
What Are the Current Bindings? . . .	6-27
Function Key Codes . . .	6-28
Display Manager . . .	6-30
Terminal Requirements for the Display Manager . . .	6-30
Specifying Terminal Type . . .	6-31
Prompt Handling . . .	6-31
ECL Uncensored Prompts . . .	6-32
The TERMINFO Database . . .	6-33
Using TERMINFO . . .	6-35
Backspace and Wrapping . . .	6-35
Using the Clearing Capabilities . . .	6-35
Optimization Using Insert Modes . . .	6-36
Optimization Using Delete Modes . . .	6-36
Tab Stops . . .	6-36
Changing the Video Display . . .	6-36
Disabling a Linefeed . . .	6-37
ECL Commands Reference . . .	6-37
ECL Commands by Function . . .	6-45
Cursor Movement . . .	6-45
Insertion . . .	6-46
Deletion . . .	6-46
Deletion Recall . . .	6-46
Regions . . .	6-47
Repetition . . .	6-47
Twiddle . . .	6-47

Submit and Abort . . . 6-48  
Case Conversion . . . 6-48  
Pathname and Abbreviation Expansion . . . 6-48  
History and Refresh . . . 6-49  
Keyboard Macros . . . 6-49  
Binding and PFkey Support . . . 6-49  
Miscellaneous . . . 6-50

## Appendices

### **A RVEC Parameters**

Meaning of RVEC Parameters . . . A-1  
Slash Convention . . . A-2  
Supplying RVEC Parameters . . . A-2  
Meaning of the Keys Parameter . . . A-2  
Keys (SR) Format . . . A-2  
Keys (VI) Format . . . A-3

### **B DUMP\_STACK Format**

DUMP\_STACK Command . . . B-1  
Format of Stack Frames . . . B-1  
Format of a Fault Frame . . . B-2  
Format of a Condition Frame . . . B-3

### **C Obsolete Commands**

### **D ECS and EBCDIC Character Sets**

Specifying Prime ECS Characters . . . D-1  
Direct Entry . . . D-1  
Octal Notation . . . D-1  
Character String Notation . . . D-2  
Special Meanings of Prime ECS Characters . . . D-2  
Prime Extended Character Set Table . . . D-2

## Glossary

## Index



# About This Book

The *PRIMOS User Commands Reference Guide* is intended for the user or programmer who is working on a 50 Series™ computer and who needs information about a particular PRIMOS® command. This guide is not intended as an introduction to PRIMOS, the Prime operating system. Introductory material to PRIMOS is supplied in the *PRIMOS User's Guide*.

All commands typically or exclusively used by the System Operator or System Administrator have been removed from this edition. See the *Operator's Guide to System Commands* for complete descriptions of their functions and further references.

This guide provides the following types of information:

- Detailed information on PRIMOS commands and command functions which are available to the user and programmer.
- A brief description of other PRIMOS commands (such as those that invoke separately priced products) and references to detailed information about those commands.

## Organization

This book contains six chapters, four appendices, a glossary, and a quick reference tab showing commonly used acronyms:

- Chapter 1, PRIMOS Commands Overview, provides a brief review of the PRIMOS command line and PRIMOS pathnames, and summarizes all PRIMOS user commands.
- Chapter 2, PRIMOS User Commands Dictionary, provides an alphabetical listing of commands. Each entry contains either detailed instructions for the use of the command or references to the books in which detailed descriptions exist.
- Chapter 3, PRIMOS Command Functions Dictionary, provides an alphabetical listing of command functions. Each entry describes the function's format and return values.





## Changes Since Rev. 22.1

The following four sections list user and programmer commands that are new, enhanced, or have become obsolete since the last edition of this guide at Rev. 22.0 (October 1988).

Commands that have only been changed to support the new singly-rooted file system structure introduced at Rev. 23.0 are not listed.

### *Changes at Revision 23.0*

#### **New Commands**

##### **LIST\_CONTIGUOUS\_BLOCKS**

Provides information about space available on a partition. This command is most useful on partitions with CAM files.

##### **LIST\_MOUNTS**

Lists mounted disk partitions.

##### **LIST\_REGISTERED\_EPF**

Lists registered EPFs.

#### **Enhanced Commands**

##### **EDIT\_COMMAND\_LINE**

Several additional commands have been added, as well as new support for non-displaying characters.

##### **HELP**

An entirely new menu driven on-line help facility has been added.

##### **LIST\_EPF**

The **-REG** option has been provided to display only registered EPFs and their current states.

##### **LIST\_LIBRARY\_ENTRIES**

The **-REG** option has been provided to display only registered libraries.

##### **STATUS**

The **ALL**, **DISKS** and **USERS** options have been added to list only a specific type of information.

### *Changes at Revision 23.1*

#### **New Commands**

##### **SHOW**

Authorizes the viewing of your terminal's input and output stream by another user. Used with the **WATCH** command.

- TALK** Provides a screen driven facility for a realtime two-way conversation between two users.
- WATCH** Monitors another user's terminal input and output stream. Used with the SHOW command.

### Enhanced Commands

- CNAME** The **-REPORT** option has been provided to inform the user of a successful name change.
- CREATE** The **-REPORT** option has been provided to inform the user of a successful directory creation.
- LD** Two new options, **-RWLOCK**, and **-TOTAL** have been added to provide further control of output contents.
- RDY** The command has been enhanced to support a set of dynamic and expandable prompt variables.
- SET\_ASYNC** Options have been added to allow/disallow logins on async lines based on the presence of a Carrier Detect signal.
- STATUS** The **SYSTEM** argument has been added to display the PRIMOS Revision currently running.
- TALK** The screen of the user not terminating the session remains until any key is pressed. TALK provides a "busy" signal if you try to talk to a user with an active TALK session. TALK will be rejected by a user who has set **MESSAGE -REJECT** or **MESSAGE -DEFER**.
- USAGE** The **-SYSTEM** option has been provided to list only system metering information.

### New Command Functions

- ATTACH\_POINT** Returns either your current or origin attach point.
- COMO\_INFO** Returns the current state of an active como file.
- GROUP\_LIST** Returns a list of the groups to which you belong, delimited by spaces.
- REVERSE** Returns the characters in the argument string in reverse order.
- SYSTEM\_INFO** Returns system information.
- SYSTEM\_USAGE** Returns system usage information.
- USER\_INFO** Returns information about a user process.

**USER\_USAGE** Returns usage information about a user process.

**VALIDATE** Validates the character contents of a string. Useful for validating the string returned by the CPL RESPONSE function as a pathname, date, password, etc.

### ***Changes at Revision 23.2***

#### **New Commands**

**CHANGE\_PROJECT** Changes your project assignment without having to log out.

#### **Enhanced Commands**

**COMOUTPUT** The **-QUERY** option has been added to allow the user to choose whether or not to overwrite or append to an existing file.

**SIZE** Output from the **SIZE** command now includes the names of open files, even though their size cannot be determined.

**SPOOL** New features and options were added for PostScript® printer support.

**STATUS** Invalid options will now cause the command to issue an error message instead of listing all categories.

### ***Changes at Revision 23.3***

#### **New Commands**

**LIST\_USERS** Lists information about various types of processes (users) currently running on a local or remote system.

**PDEV** Calculates a physical device number based on user input.

#### **Enhanced Commands**

**ASSIGN** The **-FORMAT** option has been added to permit the exchange of tapes between the Exabyte EXB-8500 and EXB-8200 tape drives.



### ***Commands Retired Since 22.0***

The following commands have become obsolete since Rev. 22.0. They are documented in Appendix C:

The following BRMS commands have been superseded by MAGSAV and MAGRST:

**ARCHIVE**  
**ARCHIVE\_RELEASE**  
**ARCHIVE\_RESTORE**  
**GENERATE\_CATALOG**  
**LISTF**  
**LIST\_CATALOG**  
**LIST\_TAPE**  
**TRANSPORT**  
**TRANSPORT\_RELEASE**  
**TRANSPORT\_RESTORE**

The following commands are interfaces to outmoded output devices or have been superseded by the SPOOL command:

**CPMPC**  
**CRMPC**  
**PRMPC**

## Prime Documentation Conventions

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

<i>Convention</i>	<i>Explanation</i>	<i>Example</i>
Uppercase	In command formats, words in uppercase bold indicate the names of commands, options, statements, and keywords. Enter them in either uppercase or lowercase. PRIMOS converts all lowercase characters to uppercase before processing.	<b>LIST_GROUP</b>
Italic	In command formats and text, characters in lowercase italic indicate variables for which you must substitute a value. In messages, variables are indicated by lowercase italic.	<b>DUMP <i>username1</i></b>  Supply a value for <i>x</i> between 1 and 10.
Abbreviations in format statements	If a command or option has an abbreviation, the abbreviation is printed in red. If no abbreviation is possible, the entire command is printed in red. Characters in variables that cannot be omitted are also printed in red.	<b>SET_QUOTA</b>
User input in examples	In examples, user input is in red; system prompts and output are not.	<b>OK, LIST_GROUP</b>
Brackets	Brackets enclose a list of one or more optional items. Choose none, one, or several of these items.	<b>STATUS [ ALL COMM ]</b>
Braces	Braces enclose a list of items. Choose one and only one of these items.	<b>DEVICE_ACLS { -ON -OFF }</b>
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.	<b>-SLOG [ { <i>node</i> <i>nodegroup</i> } ]</b>
Monospace	Identifies system output, prompts, messages, and examples.	<b>Process suspended</b>
Hyphen	Wherever a hyphen appears as the first character of an option, it is a required part of that option.	<b>SPOOL -NOTIFY</b>

<i>Convention</i>	<i>Explanation</i>	<i>Example</i>
Ellipsis	An ellipsis indicates that you have the option of entering several items of the same kind on the command line.	<i>pdev1 ... pdev9</i>
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 <sub>8</sub>

*Replace this page with the tab page labeled*

**1 PRIMOS Commands Overview**



# PRIMOS Commands Overview

## PRIMOS Command-line Format

The general format of the PRIMOS command line is

**COMMAND** [*argument*] [**-OPTION** *argument* [ . . . **-OPTION** *argument* ]]

**COMMAND** specifies one of the PRIMOS commands in Chapter 2. The command must be the first word on the command line.

*argument* is a value used by the command or option. Arguments must immediately follow the command or option they are associated with, separated from the command or option by a space. Command and option arguments are generally pathnames of directories or files, identifying names such as user IDs or job names, or parameter values.

An argument may be several space-separated values enclosed in parentheses. The following command will delete three files:

```
DELETE (file1 file2 file3)
```

**-OPTION** specifies a command option (some commands do not have options). All options are preceded by a hyphen.

As an example of a general command format, look at the COPY command:

**COPY** *source-pathname* [*target-pathname*] [*options*]

In this example, *source-pathname* and *target-pathname* are both arguments to the command. *source-pathname* tells PRIMOS the name of the file to be copied. *target-pathname*, which is optional (indicated by square brackets), provides a new pathname for the file.

*options* are one or more additional instructions of the form **-OPTION** concerning the copy procedure.

The example below shows the COPY command using the `-DELETE` option:

```
COPY STATUS.Q2 SALES>QUARTER_RPT -DELETE
```

The COMMAND is COPY, which takes two arguments, STATUS.Q2 and SALES>QUARTER\_RPT, and the option selected is `-DELETE`.

### ***Using Pathnames and Entrynames as Arguments***

When a command format calls for a pathname, you can usually use a full or relative pathname instead of an absolute pathname. If the file system object is in your current directory, you can simply use the entryname. Pathnames and entrynames are described in detail below.

In the COPY command above, for example, STATUS.Q2 is an entryname (a filename in this case) because it is in the current directory. SALES>QUARTER\_RPT is a full pathname because the copy is to be placed in another directory.

### ***Length of Command Lines***

Command lines can be a maximum of 160 characters long. Command lines that exceed that length are rejected by the command processor, with the following error message:

```
Command-line longer than 160 characters. (listen_)  
ER!
```

### ***Case of Command-line Words***

Unless specified otherwise in the command description, you can enter the command line in either uppercase or lowercase characters. PRIMOS converts all characters to uppercase before execution.

## **Advanced Command-line Functionality**

The PRIMOS command line supports the following features:

- User-defined abbreviations
- PRIMOS command functions
- User-defined global variables
- Multiple commands on one line

- Command iteration
- Wildcard names
- Treewalk pathnames
- Name generation patterns
- Command-line syntax suppression

User-defined abbreviations are explained in the discussion of the ABBREV command in Chapter 2. Command functions are described in Chapter 3. The other features are discussed in full in Chapters 4 and 5.

---

**Note**

Not all commands support all the features listed above. The general rule is that if a feature is not useful in connection with a particular command, the command does not support it.

---

## The Root Directory (Rev. 23.0 and Higher)

The PRIMOS file system, beginning at Rev 23.0, is a singly-rooted file system name space. All file system objects stem from a single root directory symbolized by the less than symbol (<). A file system object is a file, directory (either top-level directory or subdirectory), segment directory, or access category.

The root directory is a special directory maintained in memory. It contains special directories, known as root entrynames. A root entryname is a directory which serves as a mount-point pathname for the top-level directory of a disk partition, also called a Master File Directory (MFD) for consistency with the pre-Rev. 23.0 naming convention. A root entryname can be up to 32 characters long, and does not need to match the disk partition name.

Mount points are not restricted to root entrynames. Disk partitions may be mounted on any directory. Mounting disk partitions is a System Administrator function.

### ***Common File System Name Space***

If your system is part of a network and the System Administrator has configured the Name Server, your system is included in a collection of machines that share what is known as a common file system name space. All the machines in the collection use a single shared version of the root directory. Since each machine mounts its disk partitions on the shared root directory, all disk partitions on the specified collection of machines are visible to every machine in the collection.

Within a common file system name space, all fully-qualified pathnames of file system objects are unique. The file system looks the same regardless of where you log on because the root directory is the same on every machine which is included in



### ***Full Pathname***

A **full pathname** begins with a top-level directory. The root entryname is omitted and is assumed to be the root entry in which PRIMOS finds the top-level directory. A full pathname may also contain one or more subdirectories.

Here is an example of a full pathname:

```
OAK>BRANCH>LEAF
```

OAK is a top-level directory, BRANCH is a subdirectory of OAK, and LEAF is the entryname. The root entryname (in this case, FOREST) is not part of the full pathname.

### ***Relative Pathname***

A **relative pathname** begins with the current directory, which is represented by an asterisk (\*). The asterisk means that the part of the pathname down to and including the current directory is implied. For example, if <FOREST>OAK>BRANCH>LEAF were the absolute pathname of the object LEAF, then the following would be LEAF's relative pathname if OAK were the current directory:

```
*>BRANCH>LEAF
```

If BRANCH were the current directory, then the relative pathname of LEAF would be

```
*>LEAF
```

You can also use an asterisk to stand for the current root entryname. For example, if two disks had top-level directories named OAK and you wanted to attach to the OAK directory in your current root entryname, you would use the following relative pathname to make certain you arrived at the correct directory:

```
OK, ATTACH <*>OAK
```

### ***Pathname Qualification***

Pathnames are defined as **fully-qualified** or **unqualified**. A fully-qualified pathname is a pathname which is unique in the name space. An unqualified pathname is a pathname that is missing the root entryname component. When you use an unqualified pathname, the system determines the root entryname where the file system object is located by prepending pathnames listed in the ATTACH\$ search rule until a valid pathname is found (see the ATTACH command in Chapter 2 of this guide).



<b>CPL</b>	CPL source file
<b>DPTCFG</b>	DPTCFG input file
<b>EFASL</b>	EMACS fastload file
<b>EM</b>	EMACS extension file
<b>ERROR</b>	Error file
<b>ENV</b>	Printer environment file
<b>F77</b>	FORTRAN 77 source file
<b>FBIN</b>	FORMS binary file
<b>FORM</b>	FORMS file
<b>FTN</b>	FORTRAN IV source file
<b>HELP</b>	Help file directory
<b>LIST</b>	Listing file created by a compiler
<b>PASCAL</b>	Pascal source file
<b>PL1</b>	PL/I source file
<b>PLIG</b>	PL/I Subset G source file
<b>PMA</b>	PMA source file
<b>RUN</b>	EPF runfile created by the BIND linker
<b>SAVE</b>	Static R-mode runfile created by the LOAD loader
<b>SEG</b>	Segment directory (static V-mode runfile) created by SEG
<b>SR</b>	Search rules file
<b>VRPG</b>	RPG II source file (used for the V-mode VRPG compiler)

For further information on suffix-naming conventions, see the *PRIMOS User's Guide*.

## Internal and External PRIMOS Commands

Each PRIMOS command is either an internal command or an external command. **Internal commands** are part of PRIMOS itself. **External commands** are actually programs that are stored in a special top-level directory named `CMDNCO`. Some external commands invoke separately priced software products that may be on your system. Moreover, System Administrators may add or remove external commands to meet the needs of their particular systems. For these reasons, not every system recognizes all the external commands listed in this book.





## List of User Commands

ABBREV	DBG	HPSD
ADD_REMOTE_ID	DBUTL	INFO
ASSIGN	DEFINE_GVAR	INITIALIZE_COMMAND
ATM	DELAY	_ENVIRONMENT
ATTACH	DELETE	INPUT
AVAIL	DELETE_RBF	IPSD, IPSD0, IPSD16
BASIC	DELETE_VAR	JOB
BASICV	DELSEG	KBUILD
BATCH	DIAG	KIDDEL
BATGEN	DISCOVER	LAB
BINARY	DPTXMTR	LABEL
BIND	DROPDTR	LATE
CBL	DUMP_STACK	LD
CBLDML	ED	LEM
CBLSUBS	EDIT_ACCESS	LISTING
CC	EDIT_BINARY	LIST_ACCESS
CHANGE_PASSWORD	EDIT_CMD_LINE	LIST_CONTIGUOUS_BLOCKS
CHANGE_PROJECT	EMACS	LIST_EPF
CHAP	EXPAND_SEARCH_RULES	LIST_GROUP
CLOSE	F77	LIST_LIBRARY_ENTRIES
CLUP	F77DML	LIST_LIMITS
CMPF	F77SUBS	LIST_MINI_COMMANDS
CNAME	FAP	LIST_MOUNTS
CN_RBF	FAU	LIST_PRIORITY_ACCESS
COBOL85	FDL	LIST_QUOTA
COMINPUT	FDML	LIST_RBF
COMOUTPUT	FED	LIST_REGISTERED_EPF
CONCAT	FILMEM	LIST_REMOTE_ID
COPY	FILVER	LIST_SCHEDULER_ATTRIBUTES
COPY_RBF	FSUBS	LIST_SEARCH_RULES
CPL	FTN	LIST_SEGMENT
CREATE	FTR	LIST_SERVER_NAMES
CREATK	FTS	LIST_SESSIONS
DATE	HDXSTAT	LIST_USERS
DBASIC	HELP	LIST_VAR

PRIMOS Commands Reference Guide

LOAD	PMA	SET_ASYNC
LOGIN	PRERR	SET_DELETE
LOGOUT	PRIMAN	SET_QUOTA
LON	PRIME/SNA	SET_RBF
MAGNET	PRIMON	SET_SEARCH_RULES
MAGRST	PROP	SET_VAR
MAGSAV	PROTECT	SHOW
MDUMP	PRTDSC	SIZE
MEDCONFIG	PSD, PSD20	SLIST
MEDUSA	PT45DSC	SNADSC
MEDUTIL	PT46DSC	SORT
MESSAGE	PTDSC	SPOOL
MONITOR_NET	PTELE	SPY
MONITOR_SEARCH_RULES	RDY	START
MPACK	REENTER	STATUS
MPLUSCLUP	RELEASE_LEVEL	SVCSW
MGRF	REMOVE_EPF	SYNCSORT
NETLINK	REMOVE_REMOTE_ID	TALK
NSED	RESTOR	TCF
NTS_LINE	REST_RBF	TERM
NTS_LIST_ASSOCIATE	RESUME	THEMIS
NUMBER	REVERT_PASSWORD	TIME
OAS	RJE	TRAMLC
OPEN	RJQ	TYPE
ORIGIN	RSTERM	UNASSIGN
OWLDSC	RUNOFF	UPCASE
PASCAL	RWLOCK	USAGE
PASSWD	SAVE	USERS
PHANTOM	SAVE_RBF	UX_TAPE
PDEV	SCHDEC	VPSD, VPSD16
PL1	SCHED	VRPG
PL1G	SCHEMA	VRTSSW
PLOT	SEG	WATCH
PM	SET_ACCESS	WORD



### ***Establishing Your Identity and Accessing the System***

<b>ADD_REMOTE_ID</b>	Specifies a user ID to be used for remote file access. (Internal)
<b>ATTACH</b>	Moves your location from one directory to another. (Internal)
<b>CHANGE_PASSWORD</b>	Replaces your login password with another login password of your choice. (Internal)
<b>LIST_GROUP</b>	Lists the access groups to which you belong. (Internal)
<b>LIST_REMOTE_ID</b>	Displays our current set of remote IDs. (Internal)
<b>LOGIN</b>	Begins a terminal session by identifying the user to the system and establishing the initial contact between system and user. (Internal)
<b>LOGOUT</b>	Ends a terminal session, including closing all files and releasing the PRIMOS process to another user. (Internal)
<b>ORIGIN</b>	Moves your location back to your origin directory (the directory to which you were attached when you logged in). (Internal)
<b>REMOVE_REMOTE_ID</b>	Cancels one of your remote IDs. (Internal)

### ***Managing Directories***

<b>CNAME</b>	Changes the name of a directory. (Internal)
<b>COPY</b>	Copies a directory. (External)
<b>CREATE</b>	Creates and names a new directory. (Internal)
<b>DELETE</b>	Deletes a directory or segment directory. (External)
<b>LD</b>	Lists the contents of a directory. (External)
<b>LIST_QUOTA</b>	Lists current disk quota and storage information for a specified directory. (Internal)
<b>SET_DELETE</b>	Sets the delete switch on a directory so that it cannot be deleted accidentally. (External)
<b>SET_QUOTA</b>	Defines the maximum number of records a directory and its subtree may use. (Internal)
<b>SIZE</b>	Displays the number of entries in a directory. (External)

## ***Protecting Directories***

### **With Access Control Lists:**

<b>EDIT_ACCESS</b>	Changes protection on an ACL-protected directory by modifying its ACL. (Internal)
<b>LIST_ACCESS</b>	Displays access protection for a specified directory. (Internal)
<b>LIST_PRIORITY_ACCESS</b>	Displays the priority access in effect on a specified disk partition. (Internal)
<b>SET_ACCESS</b>	Sets access control protection on a directory. If the directory is a password-protected directory, converts it to an ACL-protected directory. (Internal)

### **With Passwords:**

<b>PASSWD</b>	Replaces existing passwords on the current directory with new passwords. (Internal)
<b>PROTECT</b>	Defines rights of others to access the user's directory. (External)
<b>REVERT_PASSWORD</b>	Converts the current ACL-protected directory to a password-protected directory. (External)

## ***Managing Files***

### **General Management of Normal Files:**

<b>CNAME</b>	Changes the name of a file. (Internal)
<b>COPY</b>	Makes a copy of a file. (External)
<b>DELETE</b>	Deletes a file. (External)
<b>LIST_CONTIGUOUS_BLOCKS</b>	Provides information about space available on a partition. This command is most useful on partitions with CAM files. (External)
<b>PLOT</b>	Plots a metafile or device-specific plot file. (External)
<b>SIZE</b>	Displays the size of a file. (External)
<b>SLIST</b>	Displays the contents of a file on your terminal. (External)
<b>SPOOL</b>	Prints a file on the system printer or plotter. (External)

### General Management of ROAM Files:

<b>CN_RBF</b>	Changes the name of an active ROAM master or slave file. (External)
<b>COPY_RBF</b>	Makes a copy of a ROAM file. (External)
<b>DELETE_RBF</b>	Deletes an active or inactive ROAM file. (External)
<b>LIST_RBF</b>	Lists the attributes of a ROAM master or slave file. (External)
<b>REST_RBF</b>	Restores an inactive ROAM file. (External)
<b>SAVE_RBF</b>	Archives a ROAM file to disk. (External)
<b>SET_RBF</b>	Defines the recovery and concurrency attributes of a ROAM file. (External)

### Opening and Closing Files:

<b>BINARY</b>	Opens a file for writing on PRIMOS File Unit 3, usually as a binary output file for use by a compiler or assembler. (Internal)
<b>CLOSE</b>	Closes a file. (Internal)
<b>COMINPUT</b>	Opens or closes a command input file, usually on File Unit 6. (Internal)
<b>COMOUTPUT</b>	Opens or closes a command output file for recording interactive terminal input and output. (Internal)
<b>INPUT</b>	Opens a source file for reading on File Unit 1. (Internal)
<b>LISTING</b>	Opens a file for writing on File Unit 2. (Internal)
<b>OPEN</b>	Opens a file unit for reading, writing, or updating. (Internal)

### Comparing and Modifying Files:

<b>CMPF</b>	Compares as many as five ASCII files and lists any discrepancies among them. (External)
<b>CONCAT</b>	Combines a number of ASCII files into one file; accepts formatting commands for spooling the file. (External)
<b>FILVER</b>	Compares two files and lists the differences between them (used primarily for runfiles). (External)
<b>MRGF</b>	Merges as many as five ASCII files, allowing the user to resolve conflicts among them. (External)



<b>CBL</b>	Compiles a COBOL program, using the Prime COBOL 74 compiler. (External)
<b>COBOL85</b>	Compiles a COBOL program, using the Prime COBOL85 compiler. (External)
<b>DBASIC</b>	Invokes an interpretive BASIC with double-precision arithmetic capabilities. (External)
<b>F77</b>	Compiles a FORTRAN IV or FORTRAN 77 source program, using the FORTRAN 77 compiler. (External)
<b>FTN</b>	Compiles a FORTRAN IV program. (External)
<b>PASCAL</b>	Compiles a Pascal program. (External)
<b>PL1</b>	Compiles a PL/I program. (External)
<b>PL1G</b>	Compiles a PL/I, Subset G, program. (External)
<b>PMA</b>	Assembles a program written in the Prime Macro Assembler language (PMA). (External)
<b>VRPG</b>	Compiles an RPG II program, using the newer Prime V-mode RPG compiler. (External)

### ***Linking and Executing Programs***

<b>BIND</b>	Links V-mode and I-mode programs to create Executable Program Format (EPF) dynamic runfiles (.RUN files). (External)
<b>DELSEG</b>	Frees previously used segments. (Internal)
<b>FILMEM</b>	Fills memory locations 100 through 32K with zeroes (used before invoking LOAD). (External)
<b>LOAD</b>	Creates R-mode static runfiles. (External)
<b>REENTER</b>	Reenters a subsystem following a CONTROL-P or an error condition. (Internal)
<b>REMOVE_EPF</b>	Removes a mapped EPF from your address space. (Internal)
<b>RESTOR</b>	Restores a runfile from disk to memory. (Internal)
<b>RESUME</b>	Executes a CPL program or an R-mode program. (Internal)
<b>SAVE</b>	Saves (in a file) the contents of a specified location in memory. (Internal)
<b>SEG</b>	Creates and/or executes a V-mode or I-mode static runfile (.SEG files). (External)





<b>LATE</b>	Forces the terminal to ignore commands until a specified time. (External)
<b>SHOW</b>	Authorizes the viewing of your terminal's input and output stream by another user. Used with the WATCH command. (External)
<b>TERM</b>	Defines the terminal's erase and kill characters and duplex mode and enables (or disables) the operation of the CONTROL-P (or BREAK), CONTROL-S, and CONTROL-Q key sequences. (External)

### ***Defining Your Command Environment***

<b>ABBREV</b>	Defines abbreviations for PRIMOS commands and their arguments. (Internal)
<b>CHAP</b>	Changes your execution priority level. (Internal)
<b>CHANGE_PROJECT</b>	Changes your project assignment without having to log out. (External)
<b>DEFINE_GVAR</b>	Creates, activates, or deactivates a global variable file. (Internal)
<b>DELETE_VAR</b>	Deletes one or more variables from a global variable file. (Internal)
<b>EXPAND_SEARCH_RULES</b>	Provides the full pathname of a specified file system object or search list. (External)
<b>INITIALIZE_COMMAND_ENVIRONMENT</b>	Resets your command environment to the state it was in at your login. (Internal)
<b>LIST_VAR</b>	Lists variables from a global variable file. (Internal)
<b>LON</b>	Enables or disables phantom logout notification. (Internal)
<b>RDY</b>	Selects the prompt message you want displayed at your terminal. (Internal)
<b>RELEASE_LEVEL</b>	Frees space by discarding unwanted stack history. (Internal)
<b>SET_SEARCH_RULES</b>	Sets a search list to user-specified values. (Internal)
<b>SET_VAR</b>	Adds a variable to a global variable file. (Internal)
<b>SHOW</b>	Authorizes the viewing of your terminal's input and output stream by another user. Used with the WATCH command. (External)



<b>LIST_ACCESS</b>	Lists user access on your current directory. (Internal)
<b>LIST_CONTIGUOUS_BLOCKS</b>	Provides information about space available on a partition. This command is most useful on partitions with CAM files. (External)
<b>LIST_GROUP</b>	Lists the access groups to which you belong. (Internal)
<b>LIST_LIMITS</b>	Lists the various attributes that limit your command environment. (Internal)
<b>LIST_MINI_COMMANDS</b>	Lists the PRIMOS commands that you can use when you reach mini-command level. (Internal)
<b>LIST_PRIORITY_ACCESS</b>	Lists the priority ACL on your current disk. (Internal)
<b>LIST_QUOTA</b>	Gives information on records used and maximum records available in your current directory. (Internal)
<b>LIST_REMOTE_ID</b>	Lists your current set of remote IDs. (Internal)
<b>LIST_SERVER_NAMES</b>	Lists the active server names on the local system. (Internal)
<b>LIST_SESSIONS</b>	Lists the active ISC sessions on the local system. (Internal)
<b>LIST_USERS</b>	Lists information about various types of processes (users) currently running on a local or remote system. (External)
<b>LIST_VAR</b>	Lists the contents of an active global variable file. (Internal)
<b>SIZE</b>	Displays the size of files and the number of entries in directories, segment directories, and access categories. (External)
<b>STATUS ME</b>	Lists your user ID, user number, line number, and assigned devices. (Internal)

***Information on Program and Command Execution***

<b>BATCH</b>	Provides information on progress of user's batch jobs. (External)
<b>DUMP_STACK</b>	Produces a call/return trace of the user's stacks. (Internal)
<b>LIST_EPF</b>	Lists information on your EPFs. (Internal)
<b>LIST_LIBRARY_ENTRIES</b>	Lists the entrypoints in library EPFs. (Internal)
<b>LIST_REGISTERED_EPF</b>	Lists registered EPFs. (External)

<b>LIST_SEARCH_RULES</b>	Lists the contents of all of your search lists. (Internal)
<b>LIST_SEGMENT</b>	Lists the segments you are using. (Internal)
<b>PM</b>	Lists the contents of the RVEC vector. (Internal)
<b>PRERR</b>	Gives locations and messages from PRIMOS' error vector, ERRVEC. (Internal)
<b>TIME</b>	Displays accounting information: time since login, CPU time used, and I/O time used. (Internal)

### ***Sending Messages***

<b>MESSAGE</b>	Sends a message from one terminal to another. (Internal)
<b>TALK</b>	Engage in a two-way on-screen conversation with another user. (External)

### ***Terminal I/O Handling***

<b>RSTERM</b>	Empties the terminal's input and/or output buffers. (Internal)
<b>TYPE</b>	Prints text at the terminal or into a command output file. (Internal)
<b>WATCH</b>	Monitors another user's terminal input and output stream. Used with the SHOW command. (External)

### ***I/O Device Handling***

#### **General Commands:**

<b>ASSIGN</b>	Gives the user at the terminal control of a magnetic tape unit or other peripheral device. (Internal)
<b>SET_ASYNC</b>	Configures an asynchronous line connected to an AMLC, ICS, or LTS controller. (External)
<b>UNASSIGN</b>	Releases control of a previously assigned peripheral device. (Internal)

#### **Magnetic Tapes:**

<b>LABEL</b>	Creates an ANS COBOL level 1 volume label, or an IBM 9-track EBCDIC, or a 7-track BCD label on a magnetic tape. (External)
--------------	--



<b>PT45DSC</b>	Allows a PT45™ terminal to emulate an IBM 3277 Model 2 display station on systems where DPTX/DSC is running. (External)
<b>PT46DSC</b>	Allows a PT46™ terminal to emulate an IBM 3277 Model 2 display station on systems where DPTX/DSC is running. (External)
<b>PTDSC</b>	Allows a PST 100™ or a PT200™ terminal to emulate an IBM 3277 Model 2 display station on systems where DPTX/DSC is running. (External)
<b>RJQ</b>	Invokes the Remote Job Entry utility to transmit files to remote computer sites. (External)
<b>SNADSC</b>	Invokes the PRIME/SNA™ Interactive Terminal Emulation program that allows a supported Prime terminal to access applications on a remote IBM host. (External)
<b>TCF</b>	Invokes DPTX/TCF on a system where DPTX/TSF and DPTX/OSC are running, allowing you to access a remote IBM host computer. (External)
<b>TRAMLC</b>	Transmits or receives a file over an assigned asynchronous line between two Prime computer systems. (External)

***Data Management***

**DBMS Subsystems:**

<b>CBLDML</b>	Invokes the COBOL 74 data manipulation language (DML). (External)
<b>CBLSUBS</b>	Invokes the COBOL 74 DBMS subschema. (External)
<b>CLUP</b>	Invokes the DBMS Cleanup Processor. (External)
<b>DBUTL</b>	Invokes a database dump utility that allows you to monitor the contents of a database schema and shared user table. (External)
<b>DISCOVER</b>	Invokes the DBMS and PRISAM™ query language and report writer. (External)
<b>F77DML</b>	Invokes the FORTRAN 77 DML preprocessor. (External)
<b>FDML</b>	Invokes the FORTRAN DML preprocessor. (External)





**DISCOVER**                    Invokes the DBMS and PRISAM query language and report writer. (External)

**FAU**                            Invokes a utility that creates, deletes, and manipulates PRISAM files. (External)

***OAS (Office Automation System)***

**ATM**                            Logs you in to the OAS Advanced Text Management Option Selection Menu. (External)

**OAS**                            Starts up the Prime Office Automation System. (External)

**PTELE**                        Invokes the OAS Telephone Inquiry function. (External)

**WORD**                        Invokes the PRIMEWORD™ Word Processing System. (Internal)

***CAD/CAM***

**MEDUSA**                        Brings up a PRIME MEDUSA™ workstation. (External)

**MEDCONFIG**                    Configures a PRIME MEDUSA installation. (External)

**MEDUTIL**                        Runs PRIME MEDUSA utility programs. (External)

**PLOT**                            Plots a metafile or device-specific plot file. (External)

**THEMIS**                        Invokes a logic simulation program that allows you to interactively examine digital circuit models. (External)

***System Settings***

**SVCSW**                        Controls handling of SVC instructions. (Internal)

**VRTSSW**                        Sets the virtual sense switches. (Internal)

## Summary of Command Functions

In the following summary, the PRIMOS command functions are ordered according to their type. The functions are described in greater detail in Chapter 3. Command functions and their arguments must be enclosed in square brackets when specified.

### ***Arithmetic Functions***

<b>CALC</b>	Evaluates arithmetic or logical expressions.
<b>HEX</b>	Converts a hexadecimal number to its decimal equivalent.
<b>MOD</b>	Divides one number by another and returns the remainder.
<b>OCTAL</b>	Converts an octal number to its decimal equivalent.
<b>TO_HEX</b>	Converts a decimal number to its hexadecimal equivalent.
<b>TO_OCTAL</b>	Converts a decimal number to its octal equivalent.

### ***File System Functions***

<b>ATTRIB</b>	Returns information (type, length, or date last modified) about a specified file or directory.
<b>DIR</b>	Returns the directory portion of a pathname.
<b>ENTRYNAME</b>	Returns the entryname (final component) portion of a pathname.
<b>EXISTS</b>	Determines whether a file system object exists and whether its file type matches the type specified.
<b>GVPATH</b>	Returns the pathname of an active global variable file.
<b>OPEN_FILE</b>	Opens a file for reading or writing.
<b>PATHNAME</b>	Returns the full pathname of a file system object.
<b>READ_FILE</b>	Reads a line from an ASCII file.
<b>WILD</b>	Produces a list of all names within a directory that match one or more wildcard names.
<b>WRITE_FILE</b>	Writes a line of text into an ASCII file.

### ***String-handling Functions***

<b>AFTER</b>	Returns the portion of a string that appears after some specified character(s).
<b>BEFORE</b>	Returns the part of a string that precedes some specified character(s).
<b>INDEX</b>	Returns the starting position of a specified substring within a string.
<b>LENGTH</b>	Returns the number of characters in a given string.
<b>NULL</b>	Tests for null strings.
<b>QUOTE</b>	Places a pair of quotation marks around a string, and doubles any quotation marks appearing within the string.
<b>REVERSE</b>	Reverses the order of a given string.
<b>SEARCH</b>	Compares two strings and returns the position of the first character in the first string that matches any character in the second string.
<b>SUBST</b>	Replaces text in one string with text from another.
<b>SUBSTR</b>	Returns a substring (specified by length and starting position) of a string.
<b>TRANSLATE</b>	Replaces character(s) in one string with character(s) from another.
<b>TRIM</b>	Removes characters from the left, right, or both sides of a specified string.
<b>UNQUOTE</b>	Removes outer quotation marks from around a specified text string and changes double quotation marks within the string to single quotation marks.
<b>VALIDATE</b>	Validates input characteristics from a CPL RESPONSE function.
<b>VERIFY</b>	Compares two strings and returns the position of the first character in one that does not match any character in the other.

### ***Miscellaneous Functions***

<b>ABBREV</b>	Expands the value of an abbreviation or returns the pathname of the abbreviation file.
<b>ATTACH_POINT</b>	Returns either the user's current attach point or origin attach point.



*Replace this page with the tab page labeled*

**2 User Commands  
A-D**

# Dictionary of PRIMOS User Commands

**A**                    *See* ATTACH.

## **ABBREV**

The **ABBREV** command allows users to create and use abbreviations for PRIMOS commands and their arguments.

A user abbreviation is an abbreviation that you create for one or more PRIMOS commands and their arguments. You can then use these abbreviations during interactive sessions and in CPL programs that use the **&EXPAND** directive.

To use abbreviations, you must follow these procedures:

1. Create an abbreviation file.
2. Define abbreviations. The abbreviations are stored in the abbreviation file, which can hold approximately 200 abbreviations.
3. Activate the abbreviation file to use the abbreviations stored in it. You can have more than one abbreviation file, but only one can be active at any given time.
4. If you are in a multiuser environment, deactivate the abbreviation file at the end of your work session.

When an abbreviation file is active, the PRIMOS abbreviation preprocessor scans each command line you enter at your terminal before passing it on to the standard command preprocessor (you cannot use abbreviations in a command input file). The abbreviation preprocessor checks each space separated character string (that is, each word or number) against the abbreviation names in the file. When the preprocessor finds a match, it substitutes its full form and then passes it to the standard command preprocessor for execution.

---

**Note**

You can use abbreviations only if the System Administrator has enabled the abbreviation preprocessor.

---

**Format**

**ABBREV** { *abbrev-pathname* [*options*] }  
          *options*

**Arguments and Options**

*abbrev-pathname*

The pathname of an abbreviation file. In most circumstances, you do not need to specify an abbreviation file if you already have one active. In this case, the active file becomes the default abbreviation file. The only exception is when you use the **-CREATE** option, in which case you must specify a pathname. If you do specify a pathname that is different from an already active file, the **ABBREV** command activates the new file and deactivates the old file.

**-ADD** *name value*

Adds the abbreviation *name* to the current file with the meaning *value*. If *name* already exists, you are asked if it should be replaced. The abbreviation is expanded when used in either the command position or argument position of a command line.

**-ADD\_ARGUMENT** *name value*

Adds the abbreviation *name* to the current file with the meaning *value*. If *name* already exists, you are asked if it should be replaced. The abbreviation is expanded only when used in the argument position of a command line.

**-ADD\_COMMAND** *name value*

Adds the abbreviation *name* to the current file with the meaning *value*. If *name* already exists, you are asked if it should be replaced. The abbreviation is expanded only when used in the command position of a command line.

- CHANGE *name-1* [... *name-n*]** Changes the specified abbreviations so that they are expanded anywhere on the command line.
- CHANGE\_ARGUMENT**  
***name1* [... *name-n*]** Changes the specified abbreviations to argument abbreviations, so that they are expanded only in the argument position on the command line.
- CHANGE\_COMMAND**  
***name1* [... *name-n*]** Changes the specified abbreviations to command abbreviations, so that they are expanded only in the command position of the command line.
- CHANGE\_NAME**  
***old-name new-name*** Changes the name of the abbreviation from *old-name* to *new-name*.
- CREATE** Creates and activates an empty abbreviation file. You must supply a pathname or filename. For example, ABBREV *pathname* -CREATE. If the file already exists, the command activates that file.
- DELETE *name1* [... *name-n*]** Deletes the specified abbreviations from the abbreviation file.
- EXECUTE *rest-of-line*** Passes the rest of the command line to the command processor for execution without first expanding it. Must be the last option on the command line because all text that follows (except comments) is assumed to be the argument.
- EXPAND *rest-of-line*** Expands the rest of the command line and displays it at the terminal, but does not execute it. Must be the last option on the command line because all text that follows (except comments) is assumed to be the argument.
- EXPAND\_EXECUTE**  
***rest-of-line*** Expands the rest of the command line and then passes it to the command processor for execution. Must be the last option on the command line because all text that follows (except comments) is assumed to be the argument.





active abbreviation file when you issued the command, that file is deactivated and the new file is activated.

The example below shows the creation of an abbreviation file in the current ACL directory:

```
OK, ABBREV RBJ.ABBREV -CREATE
Creating new abbreviation file: <DPT2>RBJ>RBJ.ABBREV (ab_file_)
OK,
```

If the directory in which the file is created has a password, the password must be specified after the directory name and both must be enclosed in single quotation marks, as shown here:

```
OK, AB 'DM READY'>DM.ABBREV -CREATE
Creating new abbreviation file: <STD2>DM READY>DM.ABBREV (ab_file_)
OK,
```

In this example, READY is the password for the directory DM. The abbreviation file, DM.ABBREV, is placed in DM.

### Defining Abbreviations

To define an abbreviation, or replace an existing one, use this command format:

$$\text{ABBREV [pathname] } \left\{ \begin{array}{l} \text{-ADD\_ARGUMENT} \\ \text{-ADD\_COMMAND} \\ \text{-ADD} \end{array} \right\} \text{ name value}$$

You do not have to specify *pathname* (the name of the abbreviation file) if the file is active. If you specify *pathname*, the command activates the file.

*name* is the name of the abbreviation and *value* is what it stands for. Each abbreviation must have a name and a value. The parameters for these two arguments and the use of the three options are described in the following sections.

### Abbreviation Names

Abbreviation names can be as many as eight ASCII characters in length, but cannot use spaces, single quotation marks ('), commas, angle brackets (>), and vertical bars (|). You should not begin names with hyphens because of possible confusion with PRIMOS command options. You can specify the name in lowercase or uppercase characters because the abbreviation preprocessor converts all lowercase characters to uppercase.



the abbreviation processor substitutes the string 23 for the variable %1%, the string 25 for %2%, and the string 26 for %3%. Your command becomes

```
SPOOL -CANCEL 23 25 26; SPOOL -LIST
```

In effect, you have created a new command to delete files from the spool queue, and then show you what the queue contains after your deletions. Your command has the format

**CAN** *file1 file2 file3*

If you supply fewer words or numbers than there are variables, the extra variables are ignored. For example, using the abbreviation above, the command

```
OK, CAN 4
```

is expanded to

```
OK, SPOOL -CANCEL 4 ; SPOOL -LIST
```

Variables %2% and %3% are ignored.

If you supply more words or numbers than there are variables, the extra words or numbers are appended to the end of the command line after expansion. This is likely to cause unexpected results or errors.

**Global Variables and Command Functions:** You can include global variables and command functions in an abbreviation definition. When you use a global variable or a command function, you must be sure that PRIMOS interprets it as you intend. When you want to include the variable name instead of its current value in the abbreviation definition, place a tilde (~), the PRIMOS syntax suppression character, before the ABBREV command on the command line. When PRIMOS encounters a tilde first on the command line, it does not evaluate any global variables or functions before executing the command. When an abbreviation is created with the tilde, evaluation occurs each time the command is executed. If the tilde is omitted, the variables and commands in the abbreviation are evaluated at creation and the results are included in the abbreviation. To create an abbreviation that contains a global variable name or command function, the format is

**~ABBREV** *add-option name value*

To include a global variable, enclose the variable name in % signs, the way you do for positional variables. To use the .TERMINAL\_TYPE\$ variable, for example:

```
OK, ~ABBREV -AC EM EMACS -TTP % .TERMINAL_TYPE$ %
```



command line (that is, immediately after the OK, prompt). Thus, you can use command abbreviations as if they were PRIMOS commands.

The two examples above (WHO and CAN) are command abbreviations. Here is another example of defining a command abbreviation:

```
OK, ABBREV -ADD_COMMAND CP PASCAL %1% -NO_MAP -LISTING
OK,
```

CP is the name of the abbreviation. Entering CP and a filename at the PRIMOS command line (for example, CP PROG.PASCAL) is equivalent to entering the PRIMOS command

```
OK, PASCAL PROG.PASCAL -NO_MAP -LIST
```

When you list the contents of an abbreviation file with the ABBREV -LIST command, command abbreviations are preceded by a (C) symbol, as in the next example:

```
OK, ABBREV -LIST
Abbreviation file: <DPT2>RBJ>RBJ.ABBREV
Abbreviations: 3

(C) CAN      SPOOL -CANCEL %1% %2% %3%
(C) CP       PASCAL %1% -NO_MAP -LISTING
(C) WHO      STATUS USERS
```

```
OK,
```

The (C) that appears in the first column indicates that the abbrev is a command abbrev and will only be valid if it appears first on a command line.

### **-ADD Option**

The -ADD option creates an abbreviation that is expanded regardless of where you use it on the command line. Therefore, it is possible to define an abbreviation that you could use either as a command or as an argument to a command.

It is recommended that you use the -ADD option only when the -ADD\_COMMAND or -ADD\_ARGUMENT options will not suffice, since the use of the more specific options will avoid errors that can occur if you mistakenly place a command abbreviation in the argument position.

### Listing an Abbreviation File

Use the `-LIST` option to display the contents of an abbreviation file. The format is

```
ABBREV [pathname] -LIST [name1 [ name2 ... name16]]
```

*pathname* is the name of an abbreviation file. The argument, if used, activates the specified abbreviation file and then lists its contents. Otherwise the currently active file is listed.

*name1* is the name of an abbreviation in the specified file. Up to sixteen names may be specified. If the abbreviation is not in the specified list an error message is displayed.

#### **-LIST Example**

Listing the active abbreviation file:

```
OK, ABBREV -LIST
Abbreviation file: <TPUBS8>DRG>LOGIN.ABBREV
Abbreviations: 16

(C) .AC      abbrev -add_command
(C) .AA      abbrev -add_argument
(C) .CN      ABBREV -CHANGE_NAME
(C) .D       ABBREV -DELETE
(C) .L       ABBREV -LIST
(C) BD      BATCH -DISPLAY
(C) EM      EMACS %1% -NOXOFF
(A) HOME    <USERS>MYDIR
(C) ME      STATUS ME
(A) REPORTS <USERS>MYDIR>DEPARTMENT>REPORTS
(C) SPELL   X.SPELL
(C) SPL     SPOOL -LIST
```

If (C) appears in the first column, the abbrev is a command abbrev and will only be valid if it appears first on a command line.

If (A) appears in the first column, the abbrev is an argument abbrev and will not be valid if used as a command.

No letter will appear in the first column if the abbreviation was added with the `-ADD` option.

---

#### **WARNING**

The abbreviation file is NOT a text file. Do not try to view it with `SLIST`, edit it with a text editor like `ED` or `EMACS`, or spool it to a printer. Editing an abbreviation file with a text editor can make it unusable.

---







Create two abbrevs called U and SP:

```
OK, ABBREV -ADD_ARGUMENT U <FOREST>BEECH>%1%
OK, ABBREV -ADD_COMMAND SP SPOOL %1% -ATT PRINTA %2% %3% -LIST
```

List all the abbrevs with the .L abbrev:

```
OK, .L
Abbreviation file: <FOREST>BEECH>LOGIN.ABBREV
Abbreviations: 4
```

- (C) .A            ABBREV -AC
- (C) .L            ABBREV -LIST
- (C) SP            SPOOL %1% -ATT PRINTA %2% %3% -LIST
- (A) U             <FOREST>BEECH>%1%

OK,

### Example of a Nested Abbreviation

For the sake of clarity, the following example uses the verify mode of ABBREV (which you enable by entering ABBREV -VERIFY so that PRIMOS echos the expanded abbreviations before using them). The verify line begins with (listen\_).

The example uses two of the abbrevs created above in a single command line to show how nested variable substitution takes place:

```
OK, SP U UPDATE.MEMO -DEFER 2200
(listen_) "SPOOL <FOREST>BEECH>UPDATE.MEMO -ATT PRINTA -DEFER 2200 -LIST"
[SPOOL Rev. 23.3.0 Copyright (c) 1992, Prime Computer, Inc.]
Request 21 added to queue. 1 records : <FOREST>BEECH>UPDATE.MEMO
```

System PRINTA						
Request	Time	User	File	No	Size	State
21	1450	BEECH	UPDATE.MEMO	1	1	Defer

OK,

U is expanded first because it is the inner abbreviation. It requires one variable (%1%) so it takes UPDATE.MEMO as that variable. When SP is expanded, it takes the expanded value of U (which is <FOREST>BEECH>UPDATE.MEMO) as its variable %1% and then takes -DEFER and 2200 as its variables %2% and %3%.

For more information on the ABBREV command and user abbreviations, see the *PRIMOS User's Guide*. For more information about the command processor and the command environment and the *Advanced Programmer's Guide III: Command Environment*.

See Chapter 3 for the use of ABBREV as a command function.



If you add a remote ID to a system that already has one, the new remote ID replaces the old ID. The remote IDs exist until you log out or until you remove them from the list of remote IDs with the REMOVE\_REMOTE\_ID command.

### **Example of Using ADD\_REMOTE\_ID**

In the following example, a person with a user ID of LAURA on SYS1, her local system, has a valid remote ID of LAURA2 on SYS2. The remote ID has a password of BLUEJAY and is affiliated with the project GRAND. To activate the remote ID LAURA2, LAURA would type the following from her terminal on SYS1:

```
ADD_REMOTE_ID LAURA2 BLUEJAY -ON SYS2 -PROJECT GRAND
```

For further information on remote IDs, see the *User's Guide to Prime Network Services*. See also LIST\_REMOTE\_ID, LOGIN, and REMOVE\_REMOTE\_ID.

## **ARID**

See ADD\_REMOTE\_ID.

## **ASSIGN**

ASSIGN obtains complete control of a disk, line, or peripheral device from the user terminal.

### **Format**

```
ASSIGN device [-WAIT]
```

### **Argument**

*device* is an available device of the type listed below. The ASYNC, DISK, MT/MTX, and SMLC arguments are described in greater detail in following section.

<i>Device</i>	<i>Description</i>
ASYNC -LINE <i>n</i> [-TO <i>m</i> ]	Asynchronous communications line. See the section called Assigning Asynchronous Lines below for explanation of options.
CARDR	Serial card reader.
CENPR	Serial printer.



---

**Note**

The terminal from which you invoked the ASSIGN command cannot be used until you press the Ctrl-P or BREAK keys or until the assigned device is available again for assignment.

---

For Example, to assign the serial line printer and queue the assignment if the printer is already assigned, enter the command

OK, ASSIGN CENPR -WAIT

If you do not assign a device and attempt to perform I/O to or from the device, the following error message is displayed:

Device not assigned.  
ER!

### ***Assigning Magnetic Tapes***

For magnetic tape, the ASSIGN command either indicates by number which physical tape drive you want, or provides a description of a tape drive that meets your requirements. In addition, you can make special requests to the system operator with an ASSIGN option (for example, removing the write ring or mounting a particular tape). These requests are useful primarily for batch jobs.

The command format for assigning magnetic tapes is

$$\text{ASSIGN } \left\{ \begin{array}{l} \text{MT}pdn \\ \text{MTX -ALIAS MT}ldn \end{array} \right\} [\text{options}]$$

### **Arguments**

- |                      |   |
|----------------------|---|
| <b>MT</b> <i>pdn</i> | Assigns the magnetic tape drive specified by <i>pdn</i> . <i>pdn</i> is the physical device number assigned to each drive at system startup. Values are from 0 to 7, inclusive. For example, MT0 specifies magnetic tape device 0. Note that there is no space between the MT and the number. |
| <b>MTX</b>           | Assigns any available drive. Must be accompanied by the -ALIAS <i>MT</i> <i>ldn</i> option. The drive assigned depends on the other options specified on the command line.  |

ASSIGN

MT and MTX Device Options

-ALIAS MT*ldn* Specifies a tape drive with a logical device number. *ldn* is from 0 to 7, inclusive. A logical device number is a user-specified number assigned to a particular physical drive unit. When used with the MT*pdn* argument, *ldn* is mapped into *pdn* in subsequent magnetic tape operations.

-DENSITY *bpi* Specifies the tape density in bytes per inch. *bpi* must be one of the following: 800, 1600, 3200, or 6250. This option sets tape density automatically on Version 3 tape drives (drives that can handle 6250 bpi), but requires operator intervention on Version 0, 1, and 2 drives so that density can be set manually.

-FORMAT { 4598 }  
                  { 4601 } Specifies the data format to use when writing to a Model 4601 cartridge tape drive that you assign. The -FORMAT option takes effect only when the heads are positioned at the beginning of a tape (BOT). When adding to an existing tape, the Model 4601 automatically writes in the same format as the data already on the tape cartridge. When you ASSIGN a Model 4601 cartridge tape drive the default format is 4601.

-FORMAT 4598 specifies the format used by Model 4595, 4596, or 4598 cartridge tape drives. About 2GB of data can be written on the tape using this format. Once you assign a Model 4601 drive to write in 459x format, it continues to write every new tape in that format until you issue another ASSIGN command. An ASSIGN command without the -FORMAT option restores the standard Model 4601 format.

-FORMAT 4601 specifies the Model 4601 format which stores up to 5GB of data on each cartridge. Tapes written in Model 4601 format cannot be read on a Model 4595, 4596, or 4598 cartridge tape drive.

-MOUNT Instructs the operator to mount a new tape reel. This option is used in the middle of a mag tape procedure (such as MAGSAV or MAGRST) when one reel of tape is exhausted and another is needed. It is usually accompanied by the -TPID option. Unlike other uses of the ASSIGN command, use of the -MOUNT option requires that the drive has been assigned already by a previous ASSIGN command.

- RETENSION Causes the tape to be fast-forwarded to the end of the tape and then rewound to the beginning of the tape. This procedure stabilizes the tape-to-head pressure and stacks the tape evenly on the reel. Used only with cartridge tape drives (Version 5).
- RINGOFF Instructs the operator to remove the write ring from the tape so that it may be read but not written.
- RINGON Instructs the operator to place the write ring on the tape so that it may be both read and written.
- SPEED  $\left\{ \begin{array}{c} 25 \\ 50 \\ 100 \end{array} \right\}$  Sets the tape speed at 25, 50, or 100 inches per second (ips).
- TPID *id* Instructs the operator to mount a particular reel of tape described by *id*. *id* is a list of tape identifiers (arguments) that refers to a specific reel of tape and/or type of tape drive (name, number, and so on). Identifiers cannot begin with a hyphen (-) and cannot contain the following delimiters: commas, spaces, .NL., and /\*. Requires operator intervention.
- WAIT Indicates the user will wait until the requested drive is available.
- 7TRK Specifies a 7-track tape drive. (Default is 9-track.) Like the -9TRK option, this option is usually used with the MTX argument. Requires operator intervention.
- 9TRK Specifies a 9-track tape drive. (Default) Like the -7TRK option, this option is usually used with the MTX argument. Requires operator intervention.

### Examples of Assigning Magnetic Tapes

The following four examples illustrate the use of the ASSIGN command to assign tape drives.

**Example 1:** Assign magnetic tape drive MT3 (3 is the physical device number).

```
OK, AS MT3
Device MT3 assigned.
OK,
```

Magnetic tape drive MT3 is assigned.



ASSIGN

**Example 2:**

```
OK, ASSIGN MT4 -ALIAS MT0
Device MT4 assigned.
OK,
```

Magnetic tape drive MT4 is assigned by means of a logical device number. Henceforth, MT4 is referred to as logical MT0. The physical-to-logical number correspondence can be listed with the STAT DEV command:

```
OK, AS MT4 -ALIAS MT0
Device MT4 assigned.
OK, STAT DEV
Device User name           Usrnum Ldevice
MT4      HENRY              67     MT0
```

**Example 3:**

```
OK, ASSIGN MTX -ALIAS MT4
Device MT2 Assigned.
OK,
```

The operator is requested to assign any available tape drive as logical device 4. In response to the command, the operator sends a message to your terminal that indicates which physical drive has been assigned. In this example, the operator assigned magnetic tape drive MT2 in response to the request.

**Example 4:**

```
OK, AS MTX -ALIAS MT3 -TPID POWER -9TRK -PINGOFF -DENSITY 6250
Device MT0 Assigned.
OK,
```

The operator is requested to mount a 9-track tape named POWER on any drive that can handle 6250 bpi. In addition, the user wants write protection and an alias of MT3 to whatever device the operator chooses. The operator assigned tape drive MT0 in response to the command. If the request could not be handled by the operator, the following message would be displayed instead:

```
MagTape Assignment Request Aborted (ASSIGN)
ER!
```

**Operator Intervention**

The System Administrator uses the SETMOD command to determine the operator's role in tape assignments. Three modes are possible:

- Each user can assign a tape drive from any terminal, and operator intervention is necessary only for processing special requests. This is the default mode.

- Each user must send all assignment requests through the operator, who controls all access to tape drives. The operator then sends messages to the user terminal indicating the status of the assignment request.
- Tape-drive assignment from any user terminal is strictly forbidden. This feature is used to restrict access to tape drives in security-conscious environments, or to warn users when the operator is not available to process requests. In this mode, any attempt by a user to assign a magnetic tape drive results in the following message:

```
No MagTape Assignment Permitted. (AS)
ER!
```

### Assigning Disks

To assign a disk, use the command format

```
ASSIGN DISK pdev
```

*pdev* is the physical device number. (Use the STATUS DISKS command to list the physical device numbers of your system's disks.) The disk cannot be any of the following: the paging disk, a disk already assigned to another user, or a disk not in the Assignable Disks Table.

Before you can assign a disk, you must ask the operator to add the disk to the Assignable Disks Table. The operator prepares this table by using the DISKS command from the supervisor terminal. This restriction provides a degree of system integrity because it prevents users from assigning a disk without the system operator's knowledge, or from assigning disks or partitions the operator wishes to reserve for special use.

To assign a disk that has been started by ADDISK, the disk must first be shut down at the supervisor terminal by the SHUTDN command.

The maximum number of disk partitions that may be assigned to all users at any one time is 64. If an attempt is made to assign too many disks, the following message is displayed:

```
ASSIGN TABLE FULL
```

The following example assigns to the user the disk with the physical device number 2260:

```
OK, ASSIGN DISK 2260
```

For information about assigning disks for operators, see the *Operator's Guide to System Commands*. For more information about disks in the PRIMOS file system see the *Operator's Guide to File System Maintenance*.

## Assigning Asynchronous Lines

You can assign an asynchronous line only if it has been configured to be assigned and if it is not assigned to another user.

At Rev. 21.0, the ASSIGN ASYNC command (decimal notation) replaced the ASSIGN AMLC command (octal notation). ASSIGN ASYNC allows you to assign either a line or a range of consecutive lines on one command line. To assign an asynchronous line, use the following command format:

```
ASSIGN ASYNC -LINE n [-TO m]
```

### ASYNC Device Options

- LINE *n* Specifies the decimal line number of the line to be assigned. *n* is valid line number. If you are assigning more than one line, *n* is the first value in a series of line numbers.
- TO *m* Assigns a consecutive number of lines: *n* is the first number and *m* is the last number in the series of line numbers that you are assigning. The value of *m* must be greater than *n*.

To set assignable line characteristics, including terminal line characteristics and Network Terminal Service (NTS) lines, use the SET\_ASYNC command, described later in this chapter. For further information on asynchronous lines, see the *System Administrator's Guide, Volume II: Communication Lines and Controllers*.

## Assigning Synchronous Lines

To assign an SMLC line, use the following command format:

```
ASSIGN SMLCnn [-WAIT]
```

*nn* is an synchronous line number ranging from 00 through 07, inclusive. For details on configuring synchronous lines and maintaining synchronous controllers, see the *System Administrator's Guide, Volume II: Communication Lines and Controllers*.

## Assigning NTS Lines

You can assign Network Terminal Service (NTS) lines provided that they have been associated by the system operator or the Network Administrator. The PRIMOS line number must be specified in decimal notation. NTS lines assigned with the ASSIGN command can be unassigned only with the UNASSIGN command. For further information, see the *NTS User's Guide*. For more information on NTS lines, see the *NTS Planning and Configuration Guide*.

## ATM

ATM displays the Advanced Text Management Option Selection Menu of the Prime Office Automation System (OAS).

### **Format**

ATM

### **Usage**

For detailed information, see the *OAS Advanced Text Management Guide*.

## ATTACH

ATTACH moves the user's attach point from the current directory to another directory. Your **current directory** is the directory to which you are attached at any given moment.

### **Format**

ATTACH *pathname*

### **Argument**

*pathname* is the name of the directory to become the new current attach point. When no *pathname* is specified, your attach point remains the same.

Pathnames are specified as follows:

- To attach to a specific subdirectory, use its absolute (fully-qualified) pathname. For example,

OK, ATTACH <FOREST>OAK>ACORN

- To attach to a subdirectory of your current directory, begin the pathname with an asterisk (\*). For example:

OK, ATTACH \*>BRANCH5



---

**Note**

An obsolete form of the ATTACH command also uses logical disk (ldisk) and key parameters as arguments. This version of ATTACH is documented in Appendix C, Obsolete Commands.

---

**Protection**

To attach to an ACL directory, you must have Use (U) access to all directories in the pathname.

To attach to a password directory, you must supply the appropriate password following the directory name. For example, the following command attaches you to directory BEECH that has SECRET as its password:

```
OK, ATTACH BEECH SECRET
```

**Search Order**

When you do not specify a root entryname as part of the pathname (i.e., an unqualified pathname), all the disks to which you have access are searched. The disks are searched in the following order:

1. All local disks are searched in logical device (ldev) number order, starting with the lowest number. The command STATUS DISKS displays the available local disks and their ldev numbers.
2. If your system is on a network, remote disks are searched in optimized order based on the ATTACH\$ search rules. PRIMOS searches the directory or list of directories specified by the first line of the ATTACH\$ search rules file, then the second and so on, until it finds a matching directory name. Within a list specified on a line of the search rules file, PRIMOS will optimize the search based on system considerations. You can control the search sequence with the line order in the search rules file. See SET\_SEARCH\_RULES in this chapter for further explanation.

If an error message is returned following an ATTACH command (for example, if an MFD is not found), you remain attached to the directory in which you were working when you issued the command.

If there are duplicate top-level directory names, you may be attached to a directory you did not intend. The following two examples illustrate this.



or using an unqualified pathname:

OK, A SHAKESPEARE>PLAYS

If you use the command:

ATTACH PLAYS

PRIMOS will attach you to a top-level directory called PLAYS if there is one or give you the error message:

Top-level directory not found or inaccessible. PLAYS (ATTACH)

If you try to attach to a directory that doesn't exist:

OK, A \*>HISTORIES

Not found. HISTORIES (ATTACH)

ER! A \*>6RAGEDIES

Illegal name. 6RAGEDIES (ATTACH)

If you try to attach to a file:

OK, A \*>FILEA

Not a directory. FILEA (ATTACH)

You may still use the old form of the ATTACH command. See ATTACH in Appendix C.

See also the related command function ATTACH\_POINT in Chapter 3.



.....  
AVAIL

## AVAIL

AVAIL displays information on disk usage.

The information displayed for each disk consists of the diskname, the total number of records used, the number of records available for use, and the percentage of records used. The default measurements are in physical records (1 record = 2048 bytes), but can be given in normalized form (1 normalized record = 880 bytes). In either case, the numbers are in decimal.

Use LIST\_MOUNTS to list the partition names and mount-point pathnames for all disks. Use the STATUS DISKS command to list the names and logical device numbers of the disks connected to your system.

---

### Note

The AVAIL command will fail if the System Administrator restricts access to disk information.

---

### Format

```
AVAIL [ *  
      -LDEV n  
      partition-name  
      pathname ] [-NORM]
```

### Options and Arguments

If you do not specify a disk using one of the arguments listed below, information is displayed for the disk to which you are attached, provided that you have sufficient access rights.

- |                |  |
|----------------|--|
| *              | Displays information on all the local disks which have been started (if your System Administrator has configured the SYSTEM>DISCS file).   |
| -LDEV <i>n</i> | Displays information on the local disk with the logical device number <i>n</i> , where <i>n</i> is a decimal number. To obtain the logical device number of a disk, use the STATUS DISKS command. Logical device numbers are listed in octal, use the TYPE [OCTAL <i>n</i> ] command to convert the octal number to decimal. |
| -NORM          | Displays disk usage in normalized form, expressed in decimal. One normalized record equals 880 bytes.  |

*partition-name* The name of the partition. The maximum length of *partition-name* is 6 characters. If you do not have the partition name, you may obtain it from the LIST\_MOUNTS display. Use *pathname*, rather than *partition-name* if the partition is mounted on a subdirectory.

*pathname* The pathname of any directory on the desired partition, including the mount-point pathname. If the partition is mounted on a subdirectory, the partition name cannot be used. Use the fully-qualified mount-point pathname or the fully-qualified pathname of any other directory in the partition.

### Examples

For a local disk partition named MARKET with logical disk number 3 to which you are currently attached:

```
OK, AVAIL
  Volume  MARKET
  81477   total records
   4074   records available
  95.0%   full
```

```
OK, AVAIL -LDEV 3
  Volume  MARKET
  81477   total records
   4074   records available
  95.0%   full
```

```
OK, AVAIL MARKET
  Volume  MARKET
  81477   total records
   4074   records available
  95.0%   full
```

OK,

AVAIL

For a remote disk partition named USERS1, mounted on root entryname <USER\_LOGINS, to which you are not attached:

```
OK, AVAIL USERS1
  Volume USERS1
  128896 total records
  79226 records available
  38.5% full
```

```
OK, AVAIL <USER_LOGINS
  Volume USERS1
  128896 total records
  79226 records available
  38.5% full
```

```
OK, AVAIL <USER_LOGINS>MYDIR
  Volume USERS1
  128896 total records
  79226 records available
  38.5% full
```

```
OK, AVAIL USERS1 -NORM
  Volume USERS1
  299976 total records (normalized)
  184371 records available (normalized)
  38.5% full
```

OK,

If your System Administrator has configured the SYSTEM>DISCS file you can check the status of all local started disks using the asterisk (\*) argument. PRIMOS reads the SYSTEM>DISCS file and displays the current record utilization for all disks listed in the file. The Comments column displays optional comments entered by the Operator or System Administrator. In the following example the Comment column displays the logical and physical device numbers of each disk.

OK, AVAIL \*

Volume ID	Total recs	Free recs	% Full	Comments
SHIP	140733	3924	97.2	0 4463
MARKET	59256	7080	88.1	1 32060
ACCNTS	37035	21503	41.9	2 71060

OK,

If the System Administrator has not set up the SYSTEM>DISCS file, the following error message is displayed when you issue the AVAIL \* command:

```
NO DISCS FILE IN UFD SYSTEM
```

Similarly, if the System Administrator restricts access to disk information, all versions of AVAIL will fail.

See also STATUS.

B

See BINARY.

## BASIC

BASIC loads the older Prime BASIC language interpreter.

### **Format**

```
BASIC [pathname]
```

*pathname* is an ASCII file containing BASIC programming code.

### **Usage**

When you invoke BASIC with a *pathname*, BASIC loads and runs the contents of *pathname* and then returns to PRIMOS command level. Single-precision arithmetic is standard.

BASIC invoked without a *pathname* starts the BASIC interpreter, which displays a greater-than symbol (>) prompt and waits for a BASIC command.

For further information on BASIC, see the *Interpretive BASIC Programmer's Guide*.

To invoke the newer BASIC language interpreter, use BASICV.

See also DBASIC and NUMBER.

.....  
**BASICV**

**BASICV**

BASICV invokes the newer Prime BASIC language interpreter. The BASICV interpreter is a virtual-memory subsystem, and is the recommended interpreter for executing BASIC programs.

**Format**

**BASICV [-MIN] [-PRX] [pathname]**

**Argument and Options**

- pathname** Specifies an ASCII file containing BASIC programming code.
- MIN** Checks that the program meets the American National Standard for MINIMAL BASIC.
- PRX** Forces the PRINT statement of BASICV to display a maximum of 14 digits of precision. Use it with or without the -MIN option.

**Usage**

If you do not specify a pathname, BASICV enters Interactive mode and displays a greater-than symbol (>) prompt. You must then enter a BASICV subcommand. The following example illustrates the Interactive mode:

```
OK, basicv
[BASICV Rev. T3.0-23.0 Copyright (c) 1990, Prime Computer, Inc.]
>old bas.test
>list
bas.test      WED, SEP 04 1991      08:51:46

10 PRINT 'HELLO THERE'
20 PRINT 'SURE I AM WORKING'
30 END
>runh
HELLO THERE
SURE I AM WORKING
>quit
OK,
```

For further information on BASICV, see the *BASICVM Programmer's Guide*. See also NUMBER.

# BATCH

BATCH, as a user command, displays information on active batch jobs. As an operator command, BATCH controls the processing of users' jobs.

## Format

```
BATCH { -DISPLAY
        -STATUS }
```

## Options

- DISPLAY Provides detailed information in two tables. The first table lists the number of jobs waiting, deferred, and held in each queue. The second table lists the number of jobs currently executing and identifies each by user ID, job ID, phantom user number, and queue name.
- STATUS Gives a one-line summary that includes the total number of waiting, deferred, and held jobs, the number of queues with waiting, deferred, and held jobs, and the number of executing jobs. The total number of active jobs is also given if there are waiting, deferred, and held jobs in addition to executing jobs. If there are no active jobs, the message `No batch jobs` is displayed.

BATCH accepts other options that are used only by the operator to control the processing of users' jobs.

## Examples

```
OK, BATCH -DISPLAY
[BATCH Rev. 23.3.0 Copyright (c) 1992, Prime Computer, Inc.]
Number of waiting, deferred, and held jobs:
```

```
Queue      Jobs
-----
Normal-1   1
Normal-2   3

Total=     4 (2 queues)
```

2 currently running jobs:

```
  User      Jobid#    #    Queue
-----
PANCHO     #10023   114   Normal-2
CISCO     #00127   117   Normal-1
OK,
```



### -DISPLAY Option Example

```
OK, BATGEN -DISPLAY EXPRESS
[BATGEN Rev. 23.3.0 Copyright (c) 1992, Prime Computer, Inc.]
```

```
Queue name = express, unblocked, uncapped.
Active window = FULL;
Default cptime=121, etime=6, priority=9;
Maximum cptime=120, etime=5; Funit=6;
Delta rlevel=0; Timeslice=99;
```

OK,

The following table explains the meaning of each parameter in the display.

<i>Parameter</i>	<i>Meaning</i>
Queue name	The name of the queue, followed by its status.
Active window	The queue's time window of daily activity, within which its jobs are executed. The time window is specified in 24-hour format, hh:mm-hh:mm, or by the key word FULL if the queue is active at all times.
cptime	Specifies the maximum amount of CPU time (in seconds) allotted to the queue. The job aborts if it exceeds the time limit. NONE places no time limit on the queue.
etime	Specifies (in minutes) the elapsed time allowed before the job is aborted. Details are the same as for cptime.
Funit	Specifies the file unit used for command input. The default depends on queue parameters but is usually 6.
Delta rlevel	The number of levels a job's priority is lowered, at runtime, from the priority of the Batch monitor. A value of 0 (the minimum value) means a job runs at the same priority level as the monitor, while a greater value (7 is the highest numerical value) lowers the job priority by that many levels. A value of IDLE (the maximum value and lowest priority level) means that a job runs only when the system is otherwise idle.
priority	The job's priority within its queue. The value of <i>n</i> ranges from 0 to 9, inclusive, with 9 being the highest (most favored) priority. The default is queue-dependent.
Timeslice	The amount of time (in tenths of seconds) a job execution receives before PRIMOS services the next user or process.



.....

## BINARY

### -STATUS Option

The following example illustrates the -STATUS option:

```

OK, BATGEN -STATUS
[BATGEN Rev. 23.3.0 Copyright (c) 1992, Prime Computer, Inc.]
Queue:          Status:
-----
express         unblocked  uncapped
normal-1        unblocked  uncapped
normal-2        blocked    capped (inactive)
background-1    unblocked  uncapped (inactive)
background-2    unblocked  capped

```

OK,

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

See also BATCH; JOB.

## BINARY

BINARY opens a file for writing on PRIMOS File Unit 3, usually as a binary output file for use by a compiler or assembler.

### Format

BINARY *pathname*

### Usage

The file opened by BINARY is assigned the name *pathname*. If *pathname* is an entryname, the file is opened in the current directory.

This command has the same effect as the following OPEN command:

```
OPEN pathname 3 2
```

The CC, CBL, COBOL85, F77, FTN, PASCAL, PL1, PL1G, and VRPG compilers, and the PMA assembler, normally open a file named *basename*.BIN (if the source filename has the appropriate suffix) or *B\_filename* (if the source filename has no suffix) as the binary output file. Use the BINARY command to send the output of several compilations to a single file.

## **BIND**

BIND creates an Executable Program Format (EPF) runfile, either dynamic V-mode or I-mode, from binary files. The input files for BIND are produced by the following compilers: CC, CBL, COBOL85, F77, FTN, PASCAL, PL1, PL1G, PMA, and VRPG. EPFs are executed with the RESUME command.

You can issue BIND subcommands on the command line or interactively at the BIND prompt.

### **Format**

$$\text{BIND} \left\{ \begin{array}{l} [\textit{pathname}] [\textit{options}] \\ -\text{HELP} \left[ \begin{array}{l} \textit{subcommand} \\ -\text{LIST} \end{array} \right] \end{array} \right\}$$

### **Options and Arguments**

**pathname**      The name you wish to give the EPF file generated by BIND. BIND appends a .RUN suffix to the name. If you do not specify *pathname*, the resulting EPF file is given the same name as the input file, with a .RUN suffix replacing the .BIN suffix.

**options**        One or more BIND subcommands and their arguments.

### **Usage**

When issued without any options, BIND invokes the utility as an interactive program. It displays a colon (:) prompt and waits for a BIND subcommand

```
OK, BIND
[BIND Rev. T3.0-23.0 Copyright (c) 1990, Prime Computer, Inc.]
:
```

Enter one BIND subcommand per line. Enter the HELP subcommand to display BIND's online help facility.

When you invoke BIND with subcommands as options, the entire linking session is performed from the command line. With few exceptions, you can specify a BIND subcommand as a command line option if you precede it with a hyphen. For example, LIBRARY is a subcommand and -LIBRARY is a command line option.

The operation and options of BIND and information about EPFs are documented in *Programmer's Guide to BIND and EPFs*. Further discussion of these topics can be found in the *Advanced Programmer's Guide I: BIND and EPFs*.

.....

C

C                    See CLOSE.

CBL

CBL invokes the COBOL 74 compiler.

**Format**

CBL *pathname* [*options*]

**Argument**

*pathname* is the name of the source file to be compiled. Use a filename if the file is in your current directory. It is recommended that you give *pathname* a .CBL suffix.

**Usage**

See the *COBOL 74 Reference Guide*.

See also COBOL85.

CBLDML

CBLDML invokes the COBOL 74 Data Manipulation Language (DML) preprocessor.

The DML preprocessor translates embedded DML statements in the source file into a form that is usable by the COBOL 74 compiler. If you do not specify an output file, the preprocessor automatically adds a .CBL suffix to the entryname part of source file.

**Format**

CBLDML *in-pathname* [[-OUTPUT] *out-pathname*] [[-ERROR] *err-pathname*]  
[-DYNAMIC] [-NO\_LINE\_NUMBER]

**Arguments and Options**

*in-pathname*                    Source file containing embedded DML statements.  
*in-pathname* be the first argument following the command name.

<i>out-pathname</i>	The name you want for the file that contains the translated DML statements. The DML preprocessor appends a .CBL suffix to <i>out-pathname</i> . <i>out-pathname</i> must follow <i>in-pathname</i> on the command line unless you use the <code>-OUTPUT</code> option.
<i>err-pathname</i>	The name you want for the file that contains the error listing. <i>err-pathname</i> must follow <i>out-pathname</i> on the command line unless you use the <code>-ERROR</code> option.
<code>-DYNAMIC</code>	Allows programs to invoke a schema at runtime by a schema name rather than by the schema number obtained at compile time. The schema name is used at runtime to dynamically resolve the schema number.
<code>-ERROR</code>	Use this option to specify <i>err-pathname</i> in any position after <i>in-pathname</i> in the command syntax.
<code>-NO_LINE_NUMBER</code>	Suppresses the generation of line numbers in the output file.
<code>-OUTPUT</code>	Use this option to specify <i>out-pathname</i> in any position after <i>in-pathname</i> in the command syntax.

### Usage

See the *DBMS Data Manipulation Language Reference Guide*.

## CBLSUBS

CBLSUBS invokes the COBOL 74 Subschema Data Definition Language (DDL) compiler. The CBL Subschema DDL compiler translates the source file into the subschema table.

### Format

CBLSUBS [*in-pathname*] [`-OUTPUT` *out-pathname*] [`-LIST` *list-pathname*]

### Argument and Options

<i>in-pathname</i>	Source file containing the subschema definition. <i>in-pathname</i> , if used, must be the first argument following the command name. If you do not specify <i>in-pathname</i> , CBLSUBS prompts for it.
--------------------	--

■ ■ ■ ■ ■ ■ ■ ■ ■ ■  
CC

- OUTPUT *out-pathname***     *out-pathname* is the name of the directory where you want the subschema table to be located. If you do not specify *out-pathname*, CBLSUBS places the subschema table in the current directory.
- LIST *list-pathname***     *list-pathname* is the name you want for the file that contains the source listing, error messages, and a map of the User Work Area. If *list-pathname* is not specified, a file named *in-pathname*.LIST is put in the same directory as *out-pathname*.

### **Usage**

See the *DBMS Data Description Language Reference Guide*.

CC

CC invokes the Prime C compiler and produces an object program from an ASCII source file.

### **Format**

CC *pathname* [*options*]

### **Argument**

*pathname* is an ASCII file containing C programming code. It is recommended that you give *pathname* a .C or .CC suffix.

### **Usage**

See the *C User's Guide*.

## CHANGE\_PASSWORD

CHANGE\_PASSWORD changes your login password.

### Format

```
CHANGE_PASSWORD { old-password }  
                 -PROMPT
```

### Argument and Option

- |                     |  |
|---------------------|--|
| <i>old-password</i> | Your current password.   |
| -PROMPT             | Makes the system prompt you for your old password so that you do not have to enter it on the command line. When you type the old password, it is not echoed on the screen. The rest of the procedure is the same as without this option. |

### Usage

After you enter the old password, the system prompts you twice for the new password. For security reasons, the new password is not echoed at the terminal. Your password is changed only if you enter the old password correctly and if both new passwords match.

If you do not have a login password (that is, if you have a null password), you can create a password by typing CHANGE\_PASSWORD followed by a carriage return, and then entering the new password in response to the system prompts.

### Errors While Changing Passwords

The list below describes errors you may commit while changing your password:

- You enter the old password incorrectly.
- You enter the second new password differently from the first during the confirmation dialogue. Both new passwords must match.
- You enter a new password that is shorter than the minimum length allowed by your system. Your System Administrator determines the minimum length of a password.
- You attempt to create a new password that is the same as the old password.
- You attempt to create a null password on a system that does not allow them. Your System Administrator determines if null passwords are allowed.
- You attempt to change your password to one of sixteen passwords you previously used (if the System Administrator has configured password history).

CHANGE\_PASSWORD

- You attempt to create a null password on a system that does not allow them. Your System Administrator determines if null passwords are allowed.
- You attempt to change your password to one of sixteen passwords you previously used (if the System Administrator has configured password history).

These errors cause the CHANGE\_PASSWORD command to fail and produce appropriate error messages.

If any of these errors occur, your old password remains in effect.

### Examples of Changing Passwords

The next three examples illustrate the CHANGE\_PASSWORD command.

**Example 1:** The following example illustrates CHANGE\_PASSWORD without the -PROMPT option. You must enter the old password (ROSEBUD) on the command line. The new password is not echoed at the terminal:

```
OK, CHANGE_PASSWORD ROSEBUD
New password: Enter a new password.
Reenter new password for confirmation: Enter new password again.
OK,
```

**Example 2:** When you use of the -PROMPT option enter the old password at the first CHANGE\_PASSWORD prompt, instead of on the command line. Neither the old password nor the new password is echoed at the terminal during the dialogue:

```
OK, CPW -PROMPT
Old password: Enter your old password.
New password: Enter the new password.
Reenter new password for confirmation: Enter new password again.
OK,
```

**Example 3:** The following example illustrates the error message produced by a mismatch of new passwords:

```
OK, CHANGE_PASSWORD BARON
New password? Enter the new password MARS.
Reenter new password for confirmation: You enter MARX instead of
MARS.
Passwords do not match! (change_password)
ER!
```

---

#### Note

If you forget your current password, you must contact the System Administrator because only the System Administrator can create a new password for you.

---

## CHANGE\_PROJECT

At login, you are assigned to a project determined by the System Administrator. The CHANGE\_PROJECT command is used to change to a new project without logging out and logging back in again. The new project assignment continues for the duration of your login session, or until you change it again. You can only access projects to which you belong. Only one project can be assigned at a time. When you change your project, your access groups, command environment attributes, and the origin directory, become those of the new project. The system console (User 1) and phantom processes such as batch processes cannot change projects.

You can use a password to restrict access to a project. This is a security feature designed to safeguard unattended terminals.

---

### Note

The ability to change projects is an optional facility which must be activated on your system by your System Administrator.

---

For information on the use of CHANGE\_PROJECT as an operator command see *Operator's Guide to System Commands*.

### Format

$$\text{CHANGE\_PROJECT} \left\{ \begin{array}{l} [\textit{project}] \left[ \left\{ \begin{array}{l} \text{-PASSWORD } \textit{password} \\ \text{-PROMPT} \end{array} \right\} \right] \\ \text{-ENABLE\_PASSWORD} \\ \text{-DISABLE\_PASSWORD} \left\{ \begin{array}{l} \text{-PASSWORD } \textit{password} \\ \text{-PROMPT} \end{array} \right\} \\ \text{-HELP} \end{array} \right\}$$

### Arguments and Options

*project*

Specifies the name of the project to which to change. You must have been assigned to the project by the System Administrator in order to access it. The STAT PROJECTS command lists the current project. If a project ID is not specified, you will be assigned to the default login project, if one exists on your system.



**CHANGE\_PROJECT**

- DISABLE\_PASSWORD** Specifies that subsequent use of **CHANGE\_PROJECT** will not require a password in order to change projects. The change will be in effect for the duration of the current login session or until you use the **-ENABLE\_PASSWORD** option. Use this option only if the **-ENABLE\_PASSWORD** option was used, since by default, no password is required. For security reasons, you must enter a password to disable the requiring of passwords. Use either the **-PASSWORD** option and the password, or the **-PROMPT** option and the appropriate password response with the **-DISABLE\_PASSWORD** option.
- ENABLE\_PASSWORD** Specifies that subsequent use of **CHANGE\_PROJECT** will require a password in order to change projects. The change will be in effect for the duration of the current login session or until you use the **-DISABLE\_PASSWORD** option. By default no password is required. This option is commonly used in a **LOGIN.CPL**.
- PASSWORD password** Uses *password* to gain access to a restricted project. *password* is the same as your login password. The **-PASSWORD** option is required only if you have enabled password protection.
- PROMPT** Interactively specifies the password. The **-PROMPT** option directs the system to prompt for the password. Respond with your current login password. The advantage of the prompt is that the password is not echoed on the screen, thereby increasing security.
- HELP** Displays an online list of the **CHANGE\_PROJECT** command options.

---

**Note**

When changing projects, specify the **CHANGE\_PROJECT** command on its own command line or as the last command on a command line. PRIMOS cannot execute a command that follows a **CHANGE\_PROJECT** command on the same command line.

---

## CHAP

CHAP changes the execution priority level of the current process within a range assigned by the System Administrator. For example, if your priority level is 2, you can vary the level between 0 and 2. If your priority level is 1, you can lower it to 0 and then raise it again to 1, but you cannot raise it above 1.

### Format

```
CHAP {
  DEFAULT
  DOWN
  IDLE
  LOWER n [tenths]
  -TIMESLICE milliseconds
  UP
}
```

### Arguments and Option

DEFAULT

Resets your priority level to your current default level. Especially useful to return a process from the IDLE priority level.

DOWN

Lowers your priority level by one level (but no lower than your lowest assigned level). If you are already at your lowest level (usually 0), the command does nothing. You cannot use DOWN to move a process into the IDLE or SUSPEND level. CHAP DOWN is equivalent to the CHAP LOWER 1 command.

IDLE

Places a user's process at the IDLE priority level. You can issue this command format only from a phantom process, not from your terminal. Use CHAP DEFAULT to return to your default priority level.

**LOWER *n [tenths]***

Lowers your priority level by *n* levels (but no lower than your lowest assigned level). *n* is an integer from 0 to 7, inclusive. The optional *tenths* argument assigns you a new timeslice value. (Your timeslice value determines the amount of time that PRIMOS gives to your process before it goes on to the next user.) *tenths* is an octal integer that specifies tenths of a second (for example, octal 24 is decimal 20-tenths of a second, which is two seconds.) The value of *tenths* must be less than your current timeslice value (that is, you can only shorten your timeslice value, not lengthen it). You cannot use LOWER to move a process into the IDLE or SUSPEND level.

**-TIMESLICE *milliseconds***

Specifies a new timeslice value in milliseconds (thousandths of a second). Valid timeslices are between 4 and 32767 *milliseconds* in decimal (not octal, as with the tenths argument above). Only the system operator can increase your timeslice. If *milliseconds* is 0 or missing, your timeslice is reset to the default value for your machine.

**UP**

Raises your priority level by one (but no higher than your highest assigned level). CHAP UP does nothing if you are already at your upper level or if your process is at the IDLE priority level.

**Priority Levels**

The operator assigns each user or process (including phantom processes) on a Prime system a range of execution priority levels. The general-purpose levels range from 0 (lowest priority) through 3 (highest priority). The higher your priority level, the more attention you receive from PRIMOS. If no range is assigned, the default is 1 for the upper level and 0 for the lower level.

There are two special-purpose priority levels, IDLE and SUSPEND. A user or process which has been assigned an IDLE level is serviced by PRIMOS only when no other users or processes at any other priority are eligible to run (that is, when the system is idle). You can assign an IDLE level to your phantom processes but not to yourself. An IDLE process remains at that level until it is changed with CHAP to another priority level.

A user or process assigned a SUSPEND level cannot be serviced by PRIMOS until the operator changes the level with CHAP to another priority level. Only the operator can suspend the priority level of a user or process.

As an operator command, CHAP changes the priority level and timeslice of one or all users. For more information on CHAP as an operator command see *Operator's Guide to System Commands*.

### Determining Priority Levels

To determine your current priority levels issue a STATUS ME command. The priority level is indicated at the end of each line of output, in parentheses. If nothing is displayed you are running at the default level (level 1, if the system operator has not changed it). The following example shows the output of the STATUS ME command for user DRG. DRG's login process is at the default priority level, phantom process 263 is running at level 0, and phantom process 264 is running at level IDLE.

OK, STATUS ME

User	User No (In Decimal)	Line No	Devices (AL in Decimal)
DRG	153	1030	<DISK12>
DRG	263	phant	<DISK12> <SYSONE> (0)
DRG	264	phant	<DISK12> <SYSONE> (IDLE)

OK,

### Setting Timeslices

You can decrease your timeslices in either of two ways: by using the priority and tenths arguments, or by using the -TIMESLICE milliseconds option. If you use the tenths argument, the number you enter will be interpreted as an octal quantity of tenths of seconds; if you use the -TIMESLICE milliseconds option, the number you enter will be interpreted as a decimal quantity of milliseconds. You can not increase your timeslice values, only the system operator may do that.

If you specify tenths or milliseconds as zero, or if you specify -TIMESLICE and omit milliseconds, you reset the timeslice to that system's default value. If you specify priority, but omit tenths, the timeslice is unchanged. Finally, if you issue the CHAP command without the -TIMESLICE option and without the priority and tenths arguments, your priority is reset to 1 (the level at which users normally run), and the timeslice is reset to the default value for that system.

.....

## CLOSE

### CHAP Examples

To change the priority of a process currently running at level 1 to level 0, use one of the two following forms:

```
OK, CHAP DOWN
OK, CHAP LOWER 1
```

To change the priority of a process currently running at level 0 to level 1, enter:

```
OK, CHAP UP
```

For a table of major and minor timeslices for most PRIMOS systems, see the CHAP and ELIGTS commands in the *Operator's Guide to System Commands*, respectively.

## CLOSE

CLOSE closes one or more files.

The CLOSE command closes files in one of three ways: by pathname, by file unit number (*funit*), or by closing all files (except a command output file) at once. In all cases, issuing CLOSE for a file that is already closed does not result in an error.

### Format

```
CLOSE { pathname
        funit1 [ ... funit16 ]
        -ALL }
```

### Arguments and Options

- pathname* Specifies the name of the file. If *pathname* cannot be found, an error message is displayed and you are returned to PRIMOS command level.
- funit1* [...*funit16*] Closes up to 16 files specified by the space-separated list of file unit numbers *funit1* through *funit16*. File unit numbers range from 1 to 32,762, depending on the configuration of the system.

**-ALL**

Closes all open files except for a command output file. You must close command output files by pathname or by using the COMOUTPUT -END command. The -ALL option ensures that buffers are retrieved properly and that the state of the file system is reset. The option is particularly useful when you are uncertain as to the state of the files in the current directory since it has no effect if there are no open files.

If you stop a program by pressing the BREAK key or issuing the Ctrl-P key sequence, you should issue a CLOSE -ALL command. Otherwise an error message may result when you enter a subsequent command. After CLOSE -ALL has been given, the stopped program cannot be continued (started).

---

**Caution**

If you use CLOSE -ALL from within a command or CPL file, the file itself will be closed and its execution terminated.

---

See also OPEN.

**CLUP**

CLUP invokes the ROAM Command Processor Clean-up Program.

**Format**

CLUP [-USERNO *user-number*]

**Option**

**-USERNO *user-number*** Performs a remote cleanup for a phantom user, a batch job, or a user who has logged out. *user-number* is the user's terminal number or phantom number.

**Usage**

To perform the cleanup procedure at your terminal, use CLUP without options. See the *ROAM Administrator's Guide* or the *DBMS Administrator's Guide*.

.....  
CMPF

## CMPF

CMPF compares lines in a maximum of five ASCII files and shows textual differences. The files to be compared can be specified by pathnames or by entrynames if the files are in your current directory. The CMPF command, along with the MRGF command, is particularly useful for parallel software development.

### Format

CMPF *fileA fileB [fileC ... fileE] [options]*

### Arguments and Options

<i>fileA fileB [fileC ... fileE]</i>	<i>fileA</i> is the original file against which the other files are compared and is treated as the common ancestor of those files. <i>fileB</i> through <i>fileE</i> are the files to be compared.
<b>-BRIEF</b>	Displays only the file identification letter and the line number of a line that differs, but not the text itself.
<b>-MINL</b> <i>number</i>	Sets the minimum number of lines that must match (after a difference in the files being compared is found) in order to resynchronize file comparison. (Default is 3.)
<b>-REPORT</b> [ <i>pathname</i> ]	Adds a brief header including the date and names of files compared. If you specify <i>pathname</i> , CMPF creates and directs the output to a file named <i>pathname</i> instead of to the terminal.
<b>-STOP</b>	Stops the comparison process as soon as one difference among the files is discovered. If a difference exists, displays a FILES DO NOT MATCH message.

### CMPF Operation

The CMPF output lists text lines from specified files that were added, changed, or deleted from an original file.

When a difference is found between *fileA*, the pathname passed as the first argument, and the other files specified, CMPF attempts to resynchronize the files for further comparison. Resynchronization is accomplished only when a certain minimum number of lines (specified in **-MINL**) match in all the files being compared. After resynchronization, lines that differ from *fileA* are displayed at the terminal or written to the report file. The comparison process then continues until another difference is found.

When line differences are reported, CMPF identifies each line from *fileA* with the letter A and a line number. The corresponding lines of the other files are indicated in the same manner, using letters B through E, respectively.

**CMPF Example**

For this example, two files, named FOX\_ONE and FOX\_TWO, are used. FOX\_ONE is *fileA*, the original file.

FOX_ONE	FOX_TWO
The	The
quick	swift
brown	red
fox	fox
jumps	jumps
over	over
the	the
lazy	the
dog	dog

A CMPF comparison of these two files works as follows:

```
OK, CMPF FOX_ONE FOX_TWO
[ CMPF Rev. 23.3.0 Copyright (c) 1992, Prime Computer, Inc.]
```

```
A2    quick
A3    brown
CHANGED TO
B2    swift
B3    red

A8    lazy
DELETED BEFORE
B8    dog
```

```
COMPARISON FINISHED.
2 DISCREPANCIES FOUND.
```

```
OK,
```

---

**Note**

CMPF compares compressed lines of any length and assumes that files of common ancestry contain lines compressed in an identical fashion. It is therefore possible for a mismatch to occur between two lines that appear identical but were compressed differently.

---

See also FILVER; MRGF.



.....  
CNAME

## CNAME

CNAME changes the name of a file system object. The file system object must be closed before you can change its name.

To use CNAME, you must have Delete (D) and Add (A) rights for an ACL directory or owner status for a password directory. You cannot change the name of a directory while attached to that directory. However, you can change a directory name while attached to one of its subdirectories.

### Format

CNAME *old-name new-name* [-REPORT]

### Arguments and Option

<i>old-name</i>	The name of the file, directory, segment directory, or access directory you want changed.
<i>new-name</i>	Replaces only the entryname of the pathname. <i>new-name</i> cannot be specified as a path.
-REPORT	Causes a brief message reporting the success of the command to be displayed.

### Usage

The following command changes file named STUDENTS.PASCAL to CLASS.PASCAL:

```
OK, CNAME STUDENTS.PASCAL CLASS.PASCAL  
OK,
```

The following command changes the name of the top-level directory JONES to WILSON:

```
OK, CNAME <ADMIN>JONES WILSON  
OK,
```

If a file system object called *new-name* already exists you get an error message:

```
OK, CN FILEX SAVEFILE -REPORT  
Already Exists. SAVEFILE (CNAME)  
ER!
```

You can use the `-REPORT` option to verify a correct change, as follows:

```
OK, CNAME FILEA SAVEFILE -REPORT
"FILEA" name changed to "SAVEFILE".
OK,
```

---

**Note**

Do not change the names of special directories such as `CMDNCO`.

---

## CN\_RBF

CN\_RBF changes the name of an active ROAM master or slave file.

The new name assigned by CN\_RBF appears in the user file directory and in the system recovery table (RCVTAB) file. For DBMS files, this command changes the name of the entire segment directory as well as the schema name.

### **Format**

CN\_RBF *old-name new-name* [-ALL]

### **Arguments and Option**

<i>old-name</i>	Name of the file you want changed.
<i>new-name</i>	Specifies the new name of the file and must be an entryname, not a pathname.
-ALL	Changes the names of the master segment directory and of all slave segment directories to <i>new-name</i> . This option is the default for DBMS files.

### **Usage**

See the *ROAM Administrator's Guide*.



## COMINPUT

COMINPUT executes a command input file or controls its command flow.

A command input file (also called a COMI file) contains a series of commands, each on a separate line. These commands can be any legal PRIMOS command, utility subcommand, or dialog response. When you execute the command input file, PRIMOS takes its commands from the file instead of from your terminal. The commands are processed as if you had entered them at your terminal.

You construct command input files using the ED or EMACS text editors. Command input files are especially useful for repetitive processes such as compiling and loading a series of programs, building libraries, and running production jobs.

### Format

$$\text{COMINPUT } \left\{ \begin{array}{l} \textit{pathname} \textit{ [funit] } \\ \textit{option} \end{array} \right\}$$

### Arguments and Options

After PRIMOS begins to process a command input file, you can control the file's command flow with the COMINPUT options listed below. You can use only one option at a time.

<i>pathname</i>	Specifies the command input file from which input is to be read by PRIMOS. If the file is in your current directory, you can specify a filename instead of a pathname.
<i>funit</i>	Specifies the PRIMOS file unit number, in octal, on which <i>pathname</i> is opened. If you omit <i>funit</i> , File Unit 6 is the default.
-CONTINUE [ <i>funit</i> ]	Resumes reading commands from a command input file after a COMINPUT -PAUSE command or an error. <i>funit</i> , which is an octal number, must be specified if the file is opened on a file unit other than the default File Unit 6.
-END	Closes the command input file and returns to the terminal for input.
-PAUSE	Returns to the terminal for input but does not close the command file.

.....

## COMINPUT

- START** [*funit*]            Restarts processing of a command file after **BREAK**, **Ctrl-P**, a warm start of **PRIMOS**, or a subsystem error. *funit*, which is an octal number, must be specified if the file is opened on a file unit other than the default File Unit 6.
- TTY**                        Closes the command input file and returns to the terminal for input. (Same as **-END**.)

The **-END**, **-TTY**, and **-PAUSE** options should be used from within a command input file, not from your terminal. Conversely, you should use the **-START** option only at your terminal. You can use the **-CONTINUE** option from your terminal or from within the file.

### Usage

Use the following command format to execute a command input file:

**COMINPUT** *pathname* [*funit*]

Here are some guidelines for using **COMINPUT**:

- If you do not specify a pathname or an option when issuing **COMINPUT**, the following error message is displayed:

```
OK, COMINPUT
Pathname or option must be specified. (COMINPUT)
ER!
```

- After a **COMINPUT -PAUSE** command, you can invoke other commands, or use the following form to start another command file on another unit:

**COMINPUT** *pathname funit*

- To resume processing the original command file, use the command format

**COMINPUT -CONTINUE** *funit*

- Do not use the **-START** option if you entered **PRIMOS** command level from a command file process by means other than a **BREAK** or **Ctrl-P**, such as by an **ER!** error message. In these instances, use the following command format to continue processing a command file:

**COMINPUT -CONTINUE** [*funit*]

You should use **COMINPUT -START** if the quit is from inside a processing program, and **COMINPUT -CONTINUE** if an error in a command line in the command file causes an automatic **COMINPUT -PAUSE**.

- Any error message that occurs during the processing of a command file causes the command input stream to be switched to the terminal. However, the command input file remains open, which allows you to reinvoke the command that caused the error message and, subsequently, to resume the command file at the command following the one that caused the error.
- You can insert comments in a command input file by using the format

`/ text`

The slash and asterisk characters (`/*`) begin the comment, which ends at the end of the line. A comment may also be appended to a command line within the file, as in the following example:

```
LIST BENCH MAP MAPPING MAP POLP
```

Comments may also be inserted at subcommand level for certain utilities such as the SEG LOAD subprocessor.

- You can chain command files with the `-CONTINUE` option. The last command file in the chain must be terminated with a `COMINPUT -TTY` or `COMINPUT -END` to ensure that files opened in the process are closed and that the chain is terminated properly. You must keep track of which file units are being opened and closed as the chain progresses, especially those file units that are opened for command input files.
- Do not use the command `CLOSE -ALL` within a command file because it closes the command file with the following error message:

```
End of file. Cominput. (Input from terminal.)
```

- The `COMI` command must appear as the only command on a command line.

For further information on `COMINPUT` and command input files, see the *PRIMOS User's Guide*.

See also `CPL`.

COMOUTPUT

# COMOUTPUT

COMOUTPUT opens a command output file and sends to it the output produced by the user at the terminal or by a command file.

A command output file (a DAM type file also called a COMO file) can store everything that you see on your terminal. This includes all commands that you (or a command file) enter and all output produced by those commands.

You can therefore use a command output file as a record of your terminal transactions. After you close the file, you can display its contents at your terminal with the SLIST command or print it with the SPOOL command.

### Format

COMOUTPUT { *pathname* [*options*]  
*options* }

### Arguments and Options

- pathname*                      The name of the command output file. You can use a filename instead of a pathname to place the file in your current directory. If *pathname* does not exist, the command creates it. If *pathname* already exists, the file is overwritten, unless you specify the -CONTINUE option.
- CONTINUE                      Appends the PRIMOS output to a command output file. If the file is not open *pathname* must precede this option. If the file is open but was suspended with -PAUSE, this option does not take a pathname because it starts sending PRIMOS output again to the suspended file.
- END                              Stops sending command output to a file and closes it. This is the recommended method of closing a COMOUTPUT file.
- NTTY                            Turns off output to the terminal (that is, does not display the prompt or any characters you type), but does continue to save output in the open command output file. Terminal output is resumed when one of the following occurs: you press the BREAK or Ctrl-P keys, an error occurs, or -TTY is specified in a subsequent COMOUTPUT command.
- PAUSE                           Suspends command output to the command output file, but does not close the file. You must subsequently issue a COMOUTPUT -CONTINUE or -END command.





.....  
**COMOUTPUT**

Use the **STATUS UNITS** command to find out if you have a command output file opened, as shown below:

```
OK, COMO COMOFILE
OK, STATUS UNITS
```

```
User GARY                               SYS6
File   File      Open   File
Unit   Position  Mode   Type  RWlock  Treename
COMO   000000030   W      DAM   NR-1W  <USERS>GARY>COMOFILE
```

OK,

### ***Closing a Command Output File***

Use the following command to close a command output file:

```
COMOUTPUT -END
```

You can also use the **CLOSE** command if you specify the filename.

### ***COMOUTPUT Examples***

The following series of **COMOUTPUT** commands illustrates the effects of the options:

The following command opens a command output file named **REPORT** and writes to it any subsequent terminal output:

```
OK, COMO REPORT
```

Commands and echoed responses are also displayed at the terminal. **REPORT** is overwritten if it already exists.

Use the **-QUERY** option to ensure that a previously written **COMO** file is preserved:

```
OK, COMO REPORT -QUERY
"<USERS>MYDIR>REPORT" already exists, do you wish
  to overwrite it? n
OK,
```

The following command stops writing command output to **REPORT**, but does not close it:

```
OK, COMO -PAUSE
```

Output continues to be displayed at the terminal.

To resume writing command output to REPORT, use the following command:

OK, COMO -CONTINUE

Use the following command to close the REPORT file:

OK, COMO -END  
OK,

The following command reopens an existing file named DATA.COMO and positions it to end-of-file, but does not start sending terminal output to the file because of the -PAUSE option:

OK, COMO DATA COMO -CONTINUE -PAUSE  
OK,

If you need to verify the existence of DATA.COMO, use the -QUERY option with the previous command:

OK, COMO DATA COMO -CONTINUE -QUERY -PAUSE  
"<USERS>MYDIR>DATA.COMO" already exists, do you wish  
to append to it?   
OK,

Because the -PAUSE option was used you need the following command to start writing terminal output to DATA.COMO:

OK, COMO -C  
OK,

The following command continues to send terminal output to DATA.COMO, but the output does not appear on the terminal screen because the -NTTY option turns off terminal display:

OK, COMO -C -NTTY

The OK, prompt is not displayed and no characters are echoed because of the previous -NTTY option.

To turn the terminal display back on, use the following command (not echoed on the terminal):

OK, COMO -C -NTTY  
OK,

.....

## CONCAT

Close the file with the following command:

```
OK; COMO -END
OK,
```

For further information on COMOUTPUT and command output files, see the *PRIMOS User's Guide*.

See also the related command function COMO\_INFO in Chapter 3.

## CONCAT

CONCAT concatenates a number of input files into one output file suitable for spooling.

### Format

```
CONCAT [out-pathname] [options]
```

### Argument

*out-pathname*

The name of the output file. Use an entryname to place the file in your current directory. You can omit *out-pathname* if you previously opened a file (with the OPEN command). File Unit 2 is the default output file unit, but you can specify another output file unit with the -OUNIT option. If *out-pathname* is omitted and no file is open, CONCAT aborts and displays the following error message:

```
Output file not open (CONCAT)
ER!
```

### Options and Subcommands

CONCAT instructions are either options or subcommands. Options can be given only on the command line and must be preceded by a hyphen. Subcommands can be given only in Command mode and cannot be preceded by a hyphen.

Many options can be given as subcommands if the initial hyphen is omitted. For example, -HEADER is a command line option, whereas HEADER is a subcommand.

## Command-line Options

The options below can be given only on the command line. You may enter them in any order.

- APPEND** Preserves the contents of *out-pathname* (which is an existing file) and appends the input files to its end.
- CLOSE** Truncates and closes the output file on exit. (Default)
- COMMAND** Begins CONCAT in Command mode with the greater-than symbol (>) prompt.
- INSERT** Begins CONCAT in Insert mode with the colon (:) prompt. (Default)
- JUNIT *n*** Specifies the file unit on which an input file is open. (Default input unit is 1.)
- OPEN** Leaves the output file open on exit but does not truncate it.
- OUNIT *n*** Specifies the file unit on which the output file is open. (File Unit 2 is the default.) If *out-pathname* is omitted from the command line, the file open on unit *n* is used for output.
- OVERWRITE** Writes over *out-pathname* (which is an existing file), erasing its contents.
- TRUNCATE** Truncates the output file on exit but leaves it open.
- VERIFY** If *out-pathname* already exists, asks you (with the OK TO MODIFY OLD prompt) if the file should be modified. If you answer YES or OK, then asks you (with the OVERWRITE OR APPEND prompt) how to modify the file. (Default)

## Options/Subcommands

The following instructions may be given as options on the command line (if they are preceded by a hyphen) or as subcommands in Command mode (without a hyphen). If specified, the **-BANNER** option must be the last option on the command line.

- BANNER [*banner-line*]** Generates both titles and banner pages. Banner pages are inserted between input files. A banner page consists of two lines, each containing as many as 14 large characters. *banner-line* specifies the first line and is read as raw text. (Spaces are therefore accepted.) If you omit *banner-line* the first line will be blank. The second line is the entryname (last component) of the input file pathname. Titles are generated as in HEADER.

.....  
**CONCAT**

<b>DELETE</b>	Deletes input file after copying it to the output file. This option has no abbreviation.
<b>EJECT</b>	Generates a page eject between input files and suppresses both titles and banner pages.
<b>HEADER</b>	Generates titles but not banner pages. (Default) If the first line of an input file is a title (that is, begins with octal 1 in the left byte), then the line appears only as the title for the file. Otherwise, the line appears both as the title line and as the first line of the file.
<b>NDELETE</b>	Does not delete input file after copying it. (Default)
<b>NHEADER</b>	Suppresses both titles and banner pages. The input files are copied to the output file without modification.
<b>NRESETP</b>	Does not reset spooler page numbering between input files. (Default)
<b>RESETP</b>	Resets spooler page numbering between input files.

### **Subcommands**

The following three subcommands can be given only in Command mode and not on the command line.

<b>INSERT</b> [ <i>file-list</i> ]	If you omit <i>file-list</i> , CONCAT enters Insert mode to accept the names of the files to be concatenated. Enter one filename per line. To exit from Insert mode, press RETURN without typing anything. If you specify a list of files, the files are concatenated into the output file without entering Insert mode. A maximum of 40 files may be specified on one line, separated by spaces or commas. Pathnames with embedded spaces (that is, passwords) must be enclosed in quotes. If an error exists in the line, the rest of the line after the error is ignored.
<b>QUIT</b>	Exits from CONCAT. This is the only way to exit from CONCAT without leaving one or more file units open.
<b>TITLE</b> [ <i>new-title</i> ]	Uses <i>new-title</i> as the banner page for the next input file. <i>new-title</i> is read as raw text, so that spaces are accepted. If <i>new-title</i> is omitted, the filename is used.

### **CONCAT Modes**

CONCAT begins its process in either Insert mode (which is the default mode) or in Command mode (if you specify the **-COMMAND** option on the command line). A

colon prompt (:) means that CONCAT is in Insert mode; a greater-than symbol prompt (>) identifies Command mode.

### Insert Mode

If you do not specify the `-COMMAND` option on the command line, CONCAT enters Insert mode. The next step depends on whether the output file is a new file or an existing one.

#### Using a New Output File

If *out-pathname* is a new file, CONCAT displays the colon prompt so that you can enter the names of the input files

```
Enter filenames, one per line:
:
```

Enter only one filename (in either lowercase or uppercase) at each colon prompt. To finish specifying filenames, enter a blank line (that is, press RETURN without typing anything). A blank line changes CONCAT to Command mode, where you can enter any CONCAT subcommand (including INSERT to return to Insert mode or QUIT to return to PRIMOS command level).

The following example illustrates a simple CONCAT operation. `OUT_FILE` is the name of the output file:

```
Enter filenames, one per line:
: ALPHA
: BETA
: OMEGA
:
                                     (entering Command mode)
> QUIT
OK,
```

The input files ALPHA, BETA, and OMEGA are now concatenated into the output file `OUT_FILE`. The original input files, however, have not been altered.

**Using an Existing Output File:** If *out-pathname* is an existing file and you do not specify either the `-APPEND` or the `-OVERWRITE` option, CONCAT first asks you how to modify the file and then requests the names of the input files.

The following example illustrates this process, using `RESULT` as the name of the output file.

```
OK TO MODIFY OLD RESULT? YES
OVERWRITE OR APPEND: A
```

```
Enter filenames, one per line:
:
```

.....  
**COPY**

Answering N or NO to the first prompt aborts CONCAT and returns you to PRIMOS command level. Answering Y, YES, O, or OK continues the CONCAT procedure.

At the second prompt (OVERWRITE OR APPEND), either answer OVERWRITE (or O) to replace the old file with the new one, or answer APPEND (or A) to preserve the contents of the old file and add the new input files at the end of the old file.

All answers can be entered either in lowercase or uppercase. The rest of the CONCAT operation is the same as described above.

### **Command Mode**

CONCAT enters Command mode if you specify the -COMMAND option on the command line or if you enter a blank line at the Insert mode colon (:) prompt.

The Command mode prompt is a greater-than symbol (>). At this prompt, you can enter any CONCAT subcommand listed below. (Any subcommand, except QUIT, used after the INSERT subcommand has no effect on the output file.) Enter one subcommand per line. Blank lines are ignored, causing another greater-than symbol prompt to be displayed. Any text preceded by a slash and an asterisk (/\*) is taken to be a comment and is also ignored.

The INSERT subcommand inserts the desired files but does not cause CONCAT to leave Command mode. To exit Command mode, issue the QUIT subcommand. QUIT ends CONCAT and returns you to PRIMOS command level.

## **COPY**

COPY copies file system objects either from one directory to another or within a directory.

The copy operation does not alter the source object in any way, nor does it remove it from its original directory, unless you specify the -DELETE option.

### **Format**

**COPY** *source-pathname* [*target-pathname*] [*options*]

### **Arguments**

<i>source-pathname</i>	The pathname of the object to be copied (source object).
<i>target-pathname</i>	The pathname to be given to the copied object (target object).

See the sections below for details on specifying the source and target pathnames

## Options

You may specify two or more options if they do not conflict.

- ADD** Copies the source object to the target directory under the name *target-pathname*. This option works only if an object named *target-pathname* does not already exist in the target directory. **-ADD** is incompatible with **-MERGE**, **-REPLACE**, and **-INCREMENTAL**.
- CAM** Converts all copied DAM or SAM files to CAM files. CAM (Contiguous Access Method) files are files that contain groups of contiguous blocks. The default is preservation of the original file type. The **-CAM** option is used to copy one directory level, it does not copy subentries.
- COPY\_ALL** Copies all the attributes of the source object, except its Access Category (ACAT), to *target-pathname*. (To copy an ACAT, specify that ACAT by itself on the COPY command line.) If possible, a specific ACL is placed on *target-pathname* so that its protection is the same as the source object. If you do not specify this option or another attribute-copying option (**-DTM**, **-PROTECT**, **-QUOTA**, or **-RWLOCK**), the target objects attributes are set to the protection defaults of the target's directory, and no access categories in a subdirectory are copied to the target directory. To use this option on ACL directories, you must have Protect (P) access.
- DAM** Converts all copied SAM or CAM files to DAM files. The default is preservation of the original file type. The **-DAM** option is used to copy one directory level, not subentries.
- DELETE** Deletes the source object after it has been copied. (Default is no deletion.) If the source object is delete-protected, this option does not work unless you also specify the **-FORCE** option. You must have Delete (D) access on the source directory to use this option.
- DTM** Copies the date/time modified stamp of the source object to the target object (including subentries in a copied directory). (Default is to reset the date/time modified stamp to the current date/time.) To use this option with ACL directories, you must have Protect (P) access.



■ ■ ■ ■ ■ ■ ■ ■ ■ ■  
*COPY*

- FORCE** Deletes a source object (when used with the **-DELETE** option) or a target object (when used with the **-REPLACE** option) that is delete-protected (by the **SET\_DELETE** command). If you do not specify **-FORCE**, **COPY** asks for your confirmation before deleting a delete-protected object, unless you specified the **-NO\_QUERY** option (in which case the protected object is not deleted). The **-FORCE** option is particularly useful for overwriting a directory tree that contains delete-protected objects.
- INCREMENTAL** Copies only source objects whose dump bit is off: that is, objects that have not been dumped to tape. (Default is to copy objects regardless of their dump bit settings.) This option works on or in directory objects and is propagated throughout a tree copy. (The **-INC** option checks the dump bit of the directory before checking individual files during a tree copy. If that directory's dump bit is on, no file objects in that directory are examined. Therefore, you should use the **-SUFD** option in conjunction with the **-INC** option when performing incremental backups.)
- LEVELS [n]** Copies down to the level indicated by *n* when you are copying a directory tree. *n* is a decimal integer from 0 to 999. If you omit **-LEVEL**, the entire tree is copied. If you omit *n*, 0 levels (the default value) are copied. That is, if you omit *n*, only the top-level directory is copied; none of the directory's subdirectories are copied.
- MERGE** Merges the source object with the target object. The **-MERGE** option may be used with directory and nondirectory file system objects. See the **-MERGE** Option section below for more detailed information.
- MXL n** Allows you to set the extent length of a CAM file on a per-file basis. *n* is in the range 0 through 32768. If you set this value so that it is not 0, the default algorithm for allocating CAM files is no longer used for that file.
- NO\_CHECK** Specifies that **COPY** should not check that the source and target objects are the same object. In addition, **-NO\_CHECK** omits the validation that the directory object pathnames are not nested. This option is intended to be used only when performance is a major consideration; if you use the **-NO\_CHECK** option, you must check to ensure that the source and target objects are not nested and are not the same object.

- NO\_CMLV** Prevents the COPY command from invoking a new command level when the conditions “disk full” or “maximum quota exceeded” are encountered during the copy operation. If either condition occurs, the copy operation ends and an error message is displayed. The **-NO\_CMLV** option is useful primarily for allowing a CPL program to handle a return error properly.
- NO\_QUERY** Does not ask you to verify whether a copy procedure should be carried out during an unexpected or potentially dangerous situation. (Two examples of such requests are whether to overwrite an existing object with the same name as *target-pathname* and whether to copy a directory or access directory.) If you specify this option, COPY carries out the copy procedure if it can. (Default is **-QUERY**.)
- PROTECT** Copies the protection attributes (protection keys, passwords, and ACLs) of the source object to the target object. (Default is to use the default ACLs of the target directory and not to copy access categories in a source subdirectory.) To use this option on ACL directories, you must have Protect (P) access.
- QUERY** Specifies that you must be asked to resolve any unexpected or potentially dangerous situations during a copy procedure. (Default)
- QUOTA** Sets the maximum quota of a copied directory and its subdirectories the same as the source object. (Default is to place no quota on the target directory.) To use this option on ACL directories, you must have Protect (P) access.
- REPLACE** Deletes *target-pathname* and then copies the source object to the target directory under the name *target-pathname*. The copied object replaces the existing target object, but retains the name of the replaced object. This option works only if an object named *target-pathname* already exists in the target directory. See the **-REPLACE** Option section below for more detailed information.
- REPORT** Reports the results of each successful copy operation.
- RWLOCK** Sets the concurrency lock setting of the target object the same as that of the source object. (Default is to set the read/write locks to the system default.) To use this option on ACL directories, you must have Protect (P) access.

- |           |  |
|-----------|--|
| -SAM      | Converts all copied DAM or CAM files to SAM files. The default is preservation of the original file type. The -SAM option is used to copy one directory level, not subentries.   |
| -SAVE_UFD | Performs incremental copy operations in conjunction with the -INC option. This option instructs COPY always to copy directories, whether or not they have been modified. See the -SAVE_UFD Option section below for more detailed information. |
| -HELP     | Displays a list of COPY options and their usage.   |

### **Required Access Rights**

To use COPY, you must have the following rights:

- Read (R) access to both the source directory (the directory that contains the object you want copied) and to the source object.
- Add (A) access to the target directory (the directory that will contain the copied object).
- Delete (D) access to the target directory if an object with the same name as the target object (the copied object) exists. You must also have Delete access to the source directory to use the -DELETE option.
- Protect (P) access if you are working with an ACL directory and use the -COPY\_ALL, -DTM, -PROTECT, -QUOTA, or -RWLOCK options.

### **Specifying a Source Pathname**

*source-pathname* is the pathname of the object to be copied (source object). The source object can be a file (including an EPF file), directory, segment directory, or access category.

If the source object is in your current directory, you need only specify its entryname. For example, to copy the file DAISY from your current directory FLOWER to the directory GATSBY, you issue the command

```
OK, COPY DAISY GATSBY>DAISY
```

Your file FLOWER>DAISY now also exists as GATSBY>DAISY.

### Specifying a Directory or Access Category

If *source-pathname* is a directory or access category, PRIMOS asks for your confirmation before copying it, as in the following example:

```
OK, COPY <BIRD>SPARROWS <FISH>TROUT>RAINBOW
OK to copy directory "<BIRD>SPARROWS" to "<FISH>TROUT>RAINBOW"? YES
OK,
```

Use the `-NO_QUERY` option to copy the directory or access category without the verification query.

When copying a directory tree, you can use the `-LEVELS` option to copy only a specified number of levels in the tree. For example, the command below copies the top three levels of the directory tree ORCHARD to the directory FOREST:

```
COPY ORCHARD FOREST -LEVELS 3
```

See the section below, Copying With Password Directories, for details on using COPY with password directories.

### Specifying a Segment Directory

COPY allows the use of segment directory file offset numbers in the source or target pathname. PRIMOS does not query you for verification before proceeding with the copy operation:

```
OK, COPY <USERS>MYDIR>MY.FILE>6 <USERS>YOURDIR>YOUR.FILE>2
OK,
```

### Source Object Errors

If *source-pathname* is a file that is open, the file cannot be copied and the following error message is returned:

```
OK, COPY TOM>CAT MANX
File in use. Unable to open for reading "TOM>CAT" (copy)
ER!
```

If *source-pathname* does not exist, the COPY procedure aborts and returns an error message, as in the following example:

```
OK, COPY EREWHON>NO_EXIST
Not found. "EREWON>NO_EXIST" (copy)
ER!
```

### Specifying a Target Pathname

*target-pathname* is the pathname that is given to the copied object (target object). Use an entryname to place the target object in your current directory:

```
COPY MONEY>DOLLAR PESO
```

In this example, the source object DOLLAR in the directory MONEY is copied to your current directory and given the name PESO.

### Specifying an Existing Target Pathname

If the target directory already contains an object with the same name as *target-pathname*, COPY asks you if it should overwrite the existing target object with *source-pathname*. If you answer Y or YES (in either uppercase or lowercase), the existing target object is deleted and the copied object replaces it. The name of the copied object will be the same as the deleted object.

In the following example, COPY overwrites the file MONTH.DATA in the current directory with the file JUNE in the subdirectory SUMMER:

```
OK, COPY *>SUMMER>JUNE MONTH.DATA
"MONTH.DATA" already exists, do you wish to overwrite it? YES
OK,
```

If the source object is a directory or access category, COPY first asks for your verification to continue the copy procedure. If you answer YES, COPY then asks you if you wish to overwrite the existing target object, as in the above example.

When overwriting an existing target object, you can use the NO\_QUERY option to suppress all queries and replace the object. (The exception is if the target object was delete-protected with the SET\_DELETE -PROTECT command. In this case, you also have to specify the -FORCE option.)

---

#### Caution

A common error when copying a source file into a directory is to specify only the directory's name as the target pathname, rather than as a part of the copied object's pathname. This could result in the deletion of the directory and its replacement by the copied object. For example, the correct command to copy your file GOLD into the directory METAL is

```
COPY GOLD METAL>GOLD
```

If you use the incorrect command COPY GOLD METAL, the following query prompt appears:

```
"*>METAL" (dir) already exists, do you wish to overwrite it
with "*>GOLD" (sam)?
```

If you answer YES or Y, COPY overwrites the directory METAL with the file GOLD, which it subsequently renames METAL.

To correctly copy the file GOLD to the directory METAL, you can also use the double-equal (=) name generation symbol, as follows:

```
COPY GOLD METAL>==
```

See Chapter 4 of this guide for more information on name generation.

---

### Nested Directory Copying

COPY prevents nested directory copying, that is, copying a portion of a directory to the same portion of the directory. The following example shows the error message returned in this situation:

```
OK, COPY MYUFD>MY.DIR MYUFD>MY.DIR>MY.SUB.DIR
Source and target have duplicate/nested paths. (copy)
OK,
```

Use the `-NO_CHECK` option to suppress this check for nested directory copying.

### Copying to an Open Target File

The COPY command cannot work if *target-pathname* is an open file. The following example shows the error message returned in this situation:

```
OK, COPY WEBB WEBB2
"WEBB2" already exists, do you wish to overwrite it? YES
File open on delete. Unable to delete file "WEBB2" (copy)
ER!
```

The exception to this rule is if you are using the COPY command to replace an open EPF file with another EPF file. For this procedure, see the section below, Replacing a Mapped EPF File.

### Omitting the Target Pathname

If you do not specify a target pathname, the source object is copied into your current directory with the entryname (the final element) of *source-pathname* as its name. The following example shows this format:

```
OK, COPY COLORS>BLUE
OK,
```

In this example, the source object BLUE in the directory COLORS is copied into your current directory and given the name BLUE.

### Copying to an Open (Mapped) EPF

As described above in the section Copying to an Open Target File, the COPY command normally does not work if the target object is an open file. The exception is if the target file is an open Executable Program Format (EPF) file. (An open EPF file is also called a mapped EPF. See the LIST\_EPF command for definitions of mapped and unmapped EPFs.)

If *source-pathname* and *target-pathname* are EPF files with .RUN suffixes and *target-pathname* is in use, COPY will rename the existing .RUN file and then copy *source-pathname* to *target-pathname*.

After you have replaced the EPF, any user who invokes it gets the new version. The old version, however, remains mapped in the address space of anyone who was using it when you replaced it. The renamed EPF will not disappear when it is no longer in use, however. You should inform all users of your change and suggest that they use the REMOVE\_EPF command and then invoke the new version.

---

#### Note

Do not confuse the replace-EPF operation with the -REPLACE option. Unlike the -REPLACE option, the replace-EPF operation does not delete the replaced EPF but instead renames it.

---

### Specifying EPF Source and Target Pathnames

*source-pathname* is the EPF that replaces the mapped EPF file. *target-pathname* is the mapped EPF file that you want replaced. *target-pathname* must have a .RUN suffix.

The following example illustrates a command line for a replace-EPF operation:

```
OK, COPY MYLIB>NEW_EPF.RUN LIBRARIES*>OLD_EPF.RUN
```

OLD\_EPF (*target-pathname*) is the mapped EPF file you want to replace and NEW\_EPF (*source-pathname*) is the file you are putting in its place.

### Replace-EPF Operation

The replace-EPF operation works as follows:

1. The name of the mapped EPF file (*target-pathname*) is changed by having its .RUN suffix replaced by a .R*n* suffix (where *n* is a digit from 0 to 9, inclusive). The file is not deleted, but becomes a REPLACE file. The digit

to be assigned to `.RPn` depends on how many REPLACE files exist. (Only 10 REPLACE files can exist at any one time.) In the above example, `OLD_EPF.RUN` might be renamed `OLD_EPF.RP2`.

2. The source EPF file (*source-pathname*) is then copied to *target-pathname*. In the example, `NEW_EPF.RUN` is copied to `LIBRARIES*>OLD_EPF.RUN`.

By default, COPY asks you if it should replace *target-pathname*. If you answer YES, COPY informs you when the operation is completed and then displays the new name of the replaced file. These messages are shown in the following example:

```
OK, COPY TIM>FORTRAN_IO_LIBRARY.RUN LIBRARIES*>===
Ok to replace EPF file LIBRARIES*>FORTRAN_IO_LIBRARY.RUN? YES
New version of EPF file FORTRAN_IO_LIBRARY.RUN now in place.
Old version of active EPF file now named FORTRAN_IO_LIBRARY.RP0.
```

Use the `-NO_QUERY` option to suppress the verification query and replace the EPF automatically.

If 10 REPLACE files already exist, COPY first asks you whether it can delete one of the REPLACE files that is not currently mapped. If you answer YES, the operation continues, as shown below:

```
OK, COPY TIM>LD.RUN CMDNC0>LD.RUN
Ok to replace EPF file CMDNC0>LD.RUN? YES
Ok to delete EPF file CMDNC0>LD.RP2? YES
New version of EPF file LD.RUN now in place.
Old version of active EPF file now named LD.RP2.
```

Use the `-NO_QUERY` option to suppress both prompts.

If all 10 REPLACE files are mapped when you issue the command, the replace-EPF operation cannot be completed, as the following example illustrates:

```
OK, COPY TIM>LD.RUN CMDNC0>LD.RUN
Ok to replace EPF file CMDNC0>LD.RUN? YES
EPF replace files are all in use.
Unable to complete file copy. (copy)
```

For more information on EPFs and on using the COPY command to replace mapped EPFs, see the *Programmer's Guide to BIND and EPFs*.

### **`-REPLACE` Option**

The `-REPLACE` option replaces an existing target object with a source object without asking for your verification to overwrite it. The operation of this option, therefore, is similar to specifying the `-NO_QUERY` option in the overwrite procedure described in the section above, Specifying an Existing Target Pathname.





```

OK, SLIST FILE.1
This is File #1
OK, SLIST FILE.2
This is File #2
OK, COPY FILE.1 FILE.2 -MERGE
OK, SLIST FILE.2
This is File #1
OK,

```

If the target file does not exist, the source file is simply copied.

If the target object type is different from the source object type (SAM, DAM, or CAM) and the target object is being changed during the copy, no merge takes place. The target object is first deleted, and the source object is then copied. To use the `-MERGE` option, you must have at least `ALURW` rights to the target object, since the target object (in most cases) is not deleted initially.

---

#### Note

Merging segment directories is similar to merging directories and file objects. Be aware that merging a source segment directory containing objects 1,2,3 with a target segment directory containing objects 2,3,4 results in a segment directory containing objects 1,2,3,4; the original target objects 2 and 3 are overwritten. When you perform a merge operation in which the source and target segment directories have the same name, the target segment directories are overwritten; you are not prompted to verify the operation.

---

#### ***-SAVE\_UFD Option***

The `-SAVE_UFD` option is used in conjunction with the `-INC` option to perform incremental copy operations. This option instructs `COPY` always to copy directories, whether or not they have been modified.

Modifying a file directly affects the dump bit switch on the file's parent directory as well as the dump bit switch on the file itself. For instance, modifying the file `BOOKS>TECH>FIRST` causes the dump bit switch to be reset on both `BOOKS>TECH` and `BOOKS>TECH>FIRST`. The following command line requests an incremental copy operation on the file's grandparent (top-level) directory:

```
COPY BOOKS <BACKUP>BOOKS -INC
```

No copy takes place unless the dump bit switch on `BOOKS` is reset.

The `-SAVE_UFD` option disables the incremental check for directories only. This option causes the directory structure (its "skeleton") to be copied, even if no files within that directory are incrementally copied. In effect, the `-SAVE_UFD` option forces a treewalk of all directories; files that should be incrementally copied cannot be missed during an incremental copy.

The example below illustrates the SAVE\_UFD option:

```
OK, COPY <TDISK>T.UFD>@@ *>BACKUP.DIR>== -INC -SUFD
Ok to copy directory "<TDISK>T.UFD>T.DIR" to "*>BACKUP.DIR>T.DIR"? Y
"<TDISK>T.UFD>T.DIR" copied to "*>BACKUP.DIR>T.DIR".
OK,
```

### Using Command-line Features

Command-line processing features such as iteration, wildcards, and name generation are particularly useful with COPY. The example below uses three command-line features:

```
OK, COPY (@.LIST @.BIN) ARCHIV>(<=.OLDLIST =.OLDBIN)
```

This example does the following: copies all files with suffixes .LIST or .BIN in the current directory to the directory ARCHIV; replaces the suffix .LIST with .OLDLIST and the suffix .BIN with .OLDBIN; and preserves the rest of each entryname. For instance, a file called PAYROLL.BIN in the current directory is copied to the directory ARCHIV and is renamed PAYROLL.OLDBIN.

See Chapter 4 of this guide for further information on command-line features.

### Copying With Password Directories

Access requirements are different under password directories. In all cases, you must have owner access on the target object. To delete an object, Delete (D) access is required for the object.

The password protecting a source object is also copied if you specify `-PROTECT`. Protection attributes include protection keys (for files, directories, and segment directories) and passwords (for directories only). If you do not specify `-PROTECT`, the source object is copied with the system default rights of RWD NIL, which means that the owner has all rights and nonowners have none.

To copy the passwords of a directory, you must have owner rights in the source object. If you do not have owner rights, the copied directory acquires the system default passwords. For owner passwords, the default is blank. For nonowners, the default provides no password (that is, the password is null).

---

#### Note

COPY does not allow the MFD, BOOT, or DSKRAT files of an MFD to be overwritten or copied. To copy a boot file to an MFD, you must first restore the new boot to memory and then save it under the name BOOT. This restriction does not apply when these files exist somewhere other than in an MFD.

---

## COPY\_RBF

COPY\_RBF copies a ROAM file. Use COPY\_RBF to move or replicate a ROAM file. When you move a ROAM file, the original source file is deleted after a successful copy. When you replicate a file, both source and duplicate files remain.

**Format**

**COPY\_RBF** *source-pathname target-pathname [options]*

**Arguments and Options**

<i>source-pathname</i>	The name of the existing master or slave segment. If you are moving the file, this name can be a subfile name. If you are replicating the file, <i>source-pathname</i> must be the master segment directory.
<i>target-pathname</i>	The pathname of the new location for the source file.
<b>-CAM</b>	Creates all the subfiles of the target segment directory as contiguous files. Cannot be used with the <b>-DAM</b> option. If neither <b>-CAM</b> nor <b>-DAM</b> is specified, the subfiles of the target segment directory assume the same file type as those of the source segment directory.
<b>-DAM</b>	Creates all the subfiles of the target segment directory as DAM files. Cannot be used with the <b>-CAM</b> option. If neither <b>-DAM</b> nor <b>-CAM</b> is specified, the subfiles of the target segment directory assume the same file type as those of the source segment directory.
<b>-DELETE</b>	Deletes the source file after a successful copy. The target file retains the ROAM file identifier of the original source file. If <b>-DELETE</b> is not specified, the new file is assigned a new ROAM file identifier. <b>-DELETE</b> cannot be used if <i>target-pathname</i> is on a remote system.
<b>-PROTECT</b>	Assigns to the target file the same ACL protection that the source file possesses.
<b>-REPORT</b>	Reports each RBF subfile as it is copied or deleted. The default is not to report successful operations on individual subfiles. With or without the <b>-REPORT</b> option, COPY_RBF reports the successful copy of the entire RBF.



## CREATE

CREATE creates a new directory.

### Format

CREATE *pathname* [*options*]

### Argument and Options

<i>pathname</i>	The name of the new directory. If <i>pathname</i> is an entryname, the directory is created in the current directory. If the directory already exists, the command fails with an error message.
-CATEGORY <i>name</i>	Protects the directory with the access category named <i>name</i> . The access category must reside in the same directory as the newly created directory.
-MAX <i>n</i>	Creates the directory as a quota directory and sets the quota to <i>n</i> . <i>n</i> is a decimal integer between 0 and the maximum number of records on the disk.
-PASSWORD	Creates the directory as a password directory. If you do not specify -PASSWORD, the directory is the same type as its parent directory.
-REPORT	Directs the system to display the name(s) of any directories created.

You cannot use the -PASSWORD option with the -MAX and -CATEGORY options.

### Rules for Creating Directories

- You must have Add (A) rights to create a new directory within an ACL directory.
- If the new directory is created subordinate to an ACL directory, the new directory becomes an ACL directory. It is protected by default protection taken from its parent directory.
- To create a password directory subordinate to an ACL directory, use the -PASSWORD option.
- If the new directory is created subordinate to a password directory, it is automatically created as a password directory, whether or not you specify the -PASSWORD option. The directory is created with a blank owner password and a null nonowner password. (Any password will match it.) The protection keys

.....  
**CREATK**

are set to RWD NIL, which allow Read, Write, and Delete rights to the owner and no rights to nonowners.

- An ACL directory cannot be subordinate to a password directory.

**CREATK**

CREATK invokes an interactive program that creates, modifies, and monitors keyed-index or direct-access files.

**Format**

CREATK

**Usage**

CREATK is a part of the MIDASPLUS subsystem. When invoked, CREATK sets up a dialogue that asks you questions about the template structure. For further information, see the *MIDASPLUS User's Guide*.

**DATE**

DATE displays the current calendar date and clock time.

**Format**

DATE {*options*}

**Options**

Output format is controlled with one of the following options. The sample outputs all use September 4, 1991 as the date.

<i>Option</i>	<i>Sample Output</i>
-AMPM	10:44 AM
-CAL	September 4, 1991
-DAY	4

```

-DOW      Wednesday
-FTAG     910904.104400
-FULL     91-09-04.10:44:00.Wed
-MONTH    September
-TAG      910904
-TIME     10:44:00
-USA      09/04/91
-UFULL    09/04/91.10:44:00.Wed
-VFULL    04 Sep 91 10:44:00 Wednesday
-VIS      04 September 91
-YEAR     1991

```

### **Usage**

If DATE is invoked with no option, it uses the date format shown in the following example.

```

OK, DATE
04 Sep 91 10:44:00 Wednesday
OK,

```

DATE is particularly useful to date command output files.  
See Chapter 3 for the use of DATE as a command function.

## **DBASIC**

DBASIC invokes the Prime version of interpretive BASIC that has double-precision arithmetic capabilities.

### **Format**

DBASIC [*options*]

### **Usage**

The operation of the DBASIC command is the same as the operation of the BASIC command.





## DEFINE\_GVAR

DEFINE\_GVAR creates and manipulates global variable files.

You must create a global variable file before you can define global variables. You may have more than one global file, but only one can be active at any time.

You can use the DEFINE\_GVAR command at command level or inside a CPL program.

### Format

```
DEFINE_GVAR { pathname [-CREATE] }
             -OFF
```

### Argument and Options

- |                 |   |
|-----------------|---|
| <i>pathname</i> | Activates the global variable file named <i>pathname</i> when used without the -CREATE option.  |
| -CREATE         | Creates a new empty global variable file named <i>pathname</i> . The command also activates the new file.   |
| -OFF            | Deactivates the current global variable file. Global variable files are also deactivated when you log out or when you activate a new global variable file with DEFINE_GVAR. |

### Creating a Global Variable File

To create a global variable file, use the format

```
DEFINE_GVAR pathname -CREATE
```

The following example creates and activates an empty global variable file named GLOBAL.VARS:

```
OK, DEFINE_GVAR GLOBAL.VARS -CREATE
```

### Activating an Existing Global Variable File

You must activate the global variable file at the beginning of any work session during which you want to use the global variables it contains. To activate the file, use the format

```
DEFINE_GVAR pathname
```



## Usage

If a carriage return is output at some point within a line, the time delay is proportional to the number of characters typed. The defaults given are for a 30 characters-per-second (cps) terminal. These defaults are assumed if you issue DELAY with no parameters.

The following example illustrates the DELAY command:

```
OK, DELAY 0 10 100
```

To turn off delay padding, use the following command:

```
OK, DELAY 0 0 1
```

You can issue the DELAY command before you log in. DELAY can also be issued from the supervisor terminal if the terminal is designated to be User 1 (by using either the USRASR operator command or the VCP command MO USER).

DELAY is particularly useful for a terminal with a nonstandard line speed. In this case, the command DELAY 10 should set the terminal to function in the Prime computer configuration.

The DELAY command works with terminals on Network Terminal Service (NTS) lines as well as with those on local lines. You can use DELAY before you log in to a terminal with an NTS line, provided that the NTS line was connected previously.

## DELETE

DELETE deletes file system objects.

### Format

```
DELETE pathname [options]
```

### Argument and Options

You may specify one or more options in any order on the command line following *pathname*.

*pathname*

The name of the file, directory, segment directory, or access category you want to delete. You can specify an entryname if the object is in your current directory.

.....

## DELETE

<b>-FORCE</b>	Deletes an object that is delete-protected by the SET_DELETE command. Use -FORCE to delete a directory that may contain delete-protected objects. If you do not specify -FORCE, PRIMOS asks you to confirm the deletion of a delete-protected object.
<b>-NO_QUERY</b>	Instructs PRIMOS to resolve any unexpected situations during a deletion procedure. If you specify this option for a delete-protected object but do not specify -FORCE, the object is not deleted.
<b>-QUERY</b>	Instructs PRIMOS to ask you to resolve any unexpected situations during a deletion procedure. (Default)
<b>-REPORT</b>	Reports the results of each successful deletion.

### **DELETE Permissions**

To use DELETE on an ACL directory, you must have Delete (D) access on the target object's parent directory. To delete an object from a password directory, you must have Delete (D) access on *pathname* or owner access on the directory.

If *pathname* is a directory or an access category, PRIMOS first asks you to verify the object's deletion, as in the following example:

```
OK, DELETE CASE
Ok to delete directory "CASE"? YES
OK,
```

Use the -NO\_QUERY option to delete the object without the verification query.

### **Using Command-line Features With DELETE**

Some command-line features, such as wildcarding and iteration, are particularly useful with the DELETE command.

#### **Using Wildcards With DELETE**

When your pathname includes wildcards, PRIMOS asks you to verify the deletion of each file system name that matches the wildcard name, even if you specified the -NO\_QUERY option. The reason for this is that -NO\_QUERY is a DELETE option and is overridden by the verification procedure of wildcards.

To suppress wildcard verification queries, use the wildcard -NO\_VERIFY option (abbreviated -NVFY) with the command, as in the following example:

```
OK, DELETE TEST -NO_VERIFY
```

With `-NO_VERIFY`, PRIMOS queries you about the deletion of subdirectories and access categories, but not about files and segment directories.

To suppress all queries about deletions, specify both the `-NO_QUERY` and `-NO_VERIFY` options, as in the following example:

```
OK, DELETE TEGG -NO_VERIFY NO_QUERY
```

### Using Iteration Lists With DELETE

To delete two or more specific files, substitute an iteration list for the pathname, as follows:

```
OK, DELETE (PEAR PLUM *>CARROT>ROOT)
```

This example deletes the objects PEAR and PLUM from your current directory and the object ROOT in the subdirectory CARROT.

For more information on command line features, see the *PRIMOS User's Guide* and Chapter 4 of this guide.

### Examples of Using DELETE

The following examples illustrate the DELETE command.

Deleting a delete-protected file without options:

```
OK, DELETE JONES.REPORT
"JONES.REPORT" protected, ok to force delete? YES
OK,
```

Deleting a delete-protected file with the `-NO_QUERY` option:

```
OK, DELETE BROWN.MEMO -NO_QUERY
File is delete-protected. Unable to delete
"BROWN.MEMO" (delete)
ER!
```

Deleting a delete-protected file with the `-FORCE` and `-REPORT` options:

```
OK, DELETE SMITH.RPT -FORCE -REPORT
"SMITH.RPT" deleted.
OK,
```

.....  
**DELETE\_RBF**

Deleting files with a wildcard name but without the `-NO_VERIFY` option:

```
OK, DELETE RE@@
(std$cp) Verify wildcard selections for "RE@@":"REBATE"? YES
"REPORT.MAY"? YES
"RESULTS" (dir)? YES
Ok to delete directory "RESULTS"? YES
OK,
```

Same as previous example, but using the `-NO_VERIFY` option:

```
OK, DELETE RE@@ -NO_VERIFY
Ok to delete directory "RESULTS"? YES
```

Same as previous example, but also using the `-NO_QUERY` option:

```
OK, DELETE RE@@ -NO_QUERY -NO_VERIFY
OK,
```

---

**Note**

DELETE cannot delete the MFD, BOOT, or DSKRAT files in an MFD. These files can be deleted only if they exist somewhere other than in an MFD.

---

**DELETE\_RBF**

DELETE\_RBF deletes an active or inactive ROAM file (master and slaves).

**Format**

```
DELETE_RBF pathname [options]
```

**Argument and Options**

<i>pathname</i>	The name of the file to be deleted. It should be the name of the master segment directory.
<code>-NO_QUERY</code>	Deletes the file without asking for your verification.
<code>-REPORT</code>	Reports each RBF subfile as it is deleted. The default is not to report successful deletions of individual subfiles. With or without the <code>-REPORT</code> option, DELETE_RBF reports the successful deletion of the entire RBF.

**Usage**

See the *ROAM Administrator's Guide*.

## DELETE\_VAR

DELETE\_VAR deletes one or more global variables from an active global variable file.

Before you can use DELETE\_VAR, you must activate the global variable file with the DEFINE\_GVAR command.

### **Format**

DELETE\_VAR *variable1* [ ... *variable-n*]

### **Arguments**

*variable1* [ ... *variable-n*] One or more names of global variables. Wildcards are supported. When you specify more than one global variable, you must separate the names with a space or a comma.

### **DELETE\_VAR Examples**

This example activates the global variable file GLOBAL.VARS and deletes the variable .ALPHA:

```
OK, DEFINE_GVAR GLOBAL.VARS
OK, DELETE_VAR .ALPHA
OK,
```

This example deletes three variables from the active global variable file:

```
OK, DELETE_VAR .LEO .VIRGO .PISCES
OK,
```

See also LIST\_VAR; SET\_VAR.



.....  
DELSEG

## DELSEG

DELSEG frees (deletes) segments.

### Format

$$\text{DELSEG } \left\{ \begin{array}{l} \text{segment number [-TO segment number]} \\ \text{ALL} \end{array} \right\}$$

### Argument and Option

*segment number* [-TO *segment number*]

Specifies the segment number of the segment to be freed. Use the -TO option to specify a range of segments to be deleted. Segment numbers must be specified in octal and must be 4000<sub>8</sub> or above for users. The DELSEG command can be used to delete segment 4000<sub>8</sub> but the results are unpredictable. Therefore, using DELSEG to delete segment 4000<sub>8</sub> is not recommended.

ALL

Deletes all segments belonging to the user at that terminal.

### Usage

To delete segment number 4003 use the command:

```
OK, DELSEG 4003
```

To delete segment numbers 4003 through 4050, use the command:

```
OK, DELSEG 4003 -TO 4050
```

A BAD PARAMETER message is displayed if you specify an illegal segment number. Deleting a nonexistent segment has no effect.

DIAG

DIAG invokes the PRISAM File Diagnostic Utility. The File Diagnostic Utility is an interactive program that verifies the structural validity of both relative and indexed PRISAM files.

**Format**

DIAG

**Usage**

When invoked, the File Diagnostic Utility displays a `DIAG>` prompt and waits for you to enter a DIAG subcommand. Type `HELP` at the `DIAG>` prompt for a list of the DIAG subcommands.

See the *PRISAM User's Guide*.

DISCOVER

DISCOVER invokes the DBMS and PRISAM query language and report writer.

Use this query language, in conjunction with DBMS and PRISAM, to retrieve information from DBMS databases and PRISAM indexed files and to update PRISAM indexed files without writing application programs. You can retrieve information by typing nonprocedural statements, and you can also format the retrieved information by using the DISCOVER Report Generator if the default display format is not what you want.

**Format**

DISCOVER [-CLUP]

**Option**

`-CLUP` Activates a cleanup utility that allows you to recover manually from a software error; `-CLUP` cleans up any internal inconsistencies and subsequently returns control to command level.

DMSTK

### Usage

After you issue the DISCOVER command, the program displays a greater-than symbol (>) prompt and waits for a DISCOVER command.

Type HELP at the greater-than symbol (>) prompt to display information on DISCOVER commands and on the Report Generator.

For detailed information on DISCOVER, see the *DISCOVER Reference Guide* and the *DISCOVER User's Guide*.

DMSTK            See DUMP\_STACK.

### DPTXMTR

DPTXMTR displays information about the Distributed Processing Terminal Executive (DPTX) queues and events on the communication lines. DPTX provides a communication network between Prime and IBM systems, supporting existing IBM software applications.

### Format

$$\text{DPTXMTR} \left[ \begin{array}{l} \{-\text{QUEUE} [-\text{FREQUENCY } \textit{seconds}] \} \\ \{-\text{TOTALS} [-\text{FREQUENCY } \textit{minutes}] \} \\ \{-\text{FREQUENCY } \textit{minutes} \} \end{array} \right]$$

### Options

**-FREQUENCY**  
*seconds / minutes*            Used by itself, displays the DPTX status screen containing averages during *minutes*. Used with the **-QUEUE** option, specifies how often (in seconds) to display the DPTX queues. Used with the **-TOTALS** option, specifies how often (in minutes) to display the DPTX status screen.

**-QUEUE**                        Starts the queue monitor program, displaying the DPTX interprocess communication queues and the free pool queues.

**-TOTALS**                      Displays a screen containing the total values since Emulator and Traffic Manager startup.

**Usage**

For detailed information, see the *Distributed Processing Terminal Executive Guide*.

**DROPDTR**

DROPDTR drops the DTR (Data Terminal Ready) signal associated with an asynchronous line.

**Format**

DROPDTR

**Usage**

You cannot issue the DROPDTR command while logged in. The DROPDTR command is intended for use only in the following situation:

1. A user has been communicating with a Prime computer over a dialup asynchronous line, using a port selector or modem.
2. The user logs out.
3. The user now wants to disconnect from the current line and reconnect to a new line. (For example, the user may wish to log in to a different node on a network.) To force the disconnect, the logged-out user gives the DROPDTR command.

Normally, the DTR is dropped following a grace period of a maximum of 10 minutes. The length of the grace period is set by the System Administrator, using the AMLTIM or DTRDRP configuration directives. These commands are described in the *System Administrator's Guide I: System Configuration*.

.....  
**DUMP\_STACK**

**DUMP\_STACK**

DUMP\_STACK displays the addresses of stack frames allocated for your program.

**Format**

**DUMP\_STACK** [*options*]

**Options**

The DUMP\_STACK options, which may be given in any order, specify how the dump is to be done. Addresses are always printed in octal.

- FROM** *n* Begins the dump with the frame from which DUMP\_STACK was called. (Default is to begin the dump with the most recent condition frame.)
- BRIEF** Specifies a short format dump, omitting condition frames and fault frames. If you do not specify **-BRIEF**, the dump is printed in full format, which is the default.
- AMES** *n* Specifies that only *n* frames of the stack are to be dumped. *n* must be a positive decimal integer. The default is to dump the entire stack.
- FROM** *n* Begins the dump with frame *n*. (The frame from which DUMP\_STACK is called is frame 1.) If you do not specify **-FROM**, the **-ALL** option determines the starting point for the dump.
- NITS** Produces a list of on-units established by each frame that is dumped.

**Usage**

For information on the format of the stack dump, see Appendix B. For an explanation of how to use the dump stack on EPFs, see the *Advanced Programmer's Guide I: BIND and EPFs*. For more on how to use the dump stack to analyze program failures, see the *Advanced Programmer's Guide III: Command Environment*.

*Replace this page with the tab page labeled*

**E-L**

**ECL**

See EDIT\_CMD\_LINE.

**ED**

ED invokes EDITOR, the Prime line-oriented text editor.

**Format**

ED [*pathname*]

**Argument**

*pathname*      The name of the file to edit.

**Usage**

If you specify *pathname*, that file is loaded into EDITOR's text buffer and EDITOR is started in Edit mode. Use a filename if the file is in your current directory. EDITOR automatically opens and closes files and file units.

If you omit *pathname*, EDITOR is started in Input mode with an empty text buffer.

If you accidentally return control to PRIMOS (for example, by pressing the BREAK or Ctrl-P keys), you can restart EDITOR without losing any of the text buffer by issuing either of the following commands: START 1000 (continue from the break) or START 1001 (resume in Edit mode).

For details of EDITOR's operation, see the *New User's Guide to EDITOR and RUNOFF*.

.....  
EDIT\_ACCESS

## EDIT\_ACCESS

EDIT\_ACCESS modifies an existing Access Control List (ACL).

### Format

EDIT\_ACCESS *target-pathname* *ACL* [-NO\_QUERY]

### Arguments and Option

*target-pathname*

The name of a file, directory, or segment directory protected by a specific ACL, or the name of an access category. To use EDIT\_ACCESS, you must have Protect (P) access on *target-pathname*.

The ACL of *target-pathname* should have been created with the SET\_ACCESS command. (Use the LIST\_ACCESS command to display the contents of a specific ACL or access category.)

If *target-pathname* is an access category, you do not have to specify the .ACAT suffix unless there is another object in the directory with the same name as the unaffixed access category.

*ACL*

Specifies the access rights for an individual or a group. The format for *ACL* is a list of names and access rights in the following format:

*identifier1:rights* [ . . . *identifier-n:rights* ]

*identifier* is a user ID, a group name (which must begin with a period), or the special identifier \$REST.

*rights* are one or more of the mnemonic codes listed under the SET\_ACCESS command. *identifier* and *rights* are separated by a colon without any blank spaces. For a complete description of identifiers and rights, see the SET\_ACCESS command.

-NO\_QUERY

Suppresses any queries and instructs PRIMOS to resolve any problems during the editing procedure.

### Adding or Changing an Identifier

To add an identifier to a specific ACL or access category, or to change the access rights of an existing identifier, specify the identifier and the access rights as







## EDIT\_BINARY

EDIT\_BINARY is an interactive tool for creating, editing, and combining binary libraries.

### Format

```
EDIT_BINARY [outfile] [
    -DAM
    -NO_QUERY
    -NO_TRUNCATE -CXX
    -SKIP
    -HELP
]
```

### Argument and Options

<i>outfile</i>	Specifies the pathname of the output binary library file to be created. The file is automatically assigned the .BIN suffix; specifying the suffix on the command line is optional. If you specify only an entryname, the file is created in the current directory. If <i>outfile</i> is omitted, you are prompted for one when you issue a FILE subcommand within EDIT_BINARY.
-DAM	Creates a DAM (Direct Access Method) <i>outfile</i> . The default file type is SAM.
-NO_QUERY	Suppresses queries that occur when you attempt to quit without saving the file you created, and when the FILE subcommand will overwrite an output file of the same name.
-NO_TRUNCATE -CXX	Suppresses truncation of C++ entry names greater than 32 characters. Both switches must be used together.
-SKIP	Includes bypass information with each module loaded. This speeds up linking by allowing BIND (or SEG) to skip unneeded routines instead of reading and discarding all unwanted object text.
-HELP	Displays the command syntax and options. EDIT_BINARY is not started.

### Usage

EDIT\_BINARY is an interactive program. It displays a colon (:) prompt and waits for an EDIT\_BINARY subcommand, as follows:

```
OK, EDIT_BINARY
[EDIT_BINARY Rev. T3.1-23.0 Copyright (c) 1991, Prime Computer, Inc.]
:
```

.....

## EDIT\_CMD\_LINE

Enter one EDIT\_BINARY subcommand per line. Enter the HELP subcommand to display EDIT\_BINARY's online help facility.

The subcommands and operation of EDIT\_BINARY are documented in the *Advanced Programmer's Guide I: BIND and EPFs*.

## EDIT\_CMD\_LINE

EDIT\_CMD\_LINE (ECL) is a command-line editor that allows you to actively manage most aspects of terminal input.

ECL allows you to

- Edit terminal input after you have typed it in, but before you have actually submitted it for processing.
- Bind any key sequence to any command.
- Bind and program a maximum of 512 programmable function keys.
- Keep a command history of the last 200 commands issued. You may save and edit this command history.

ECL references ECL\$LIB, a process-class library EPF that contains the actual command-line processing routines. When you invoke ECL, you actually replace the PRIMOS routine that reads and processes terminal input; thus, you should have some familiarity with the ECL facility before using it.

Chapter 6 in this volume is devoted to ECL and its command-line characteristics. For more information about the use of ECL, see the *PRIMOS User's Guide*.

### **Format**

EDIT\_CMD\_LINE [*options*]

### **ECL Command-line Options**

Table 2-1 lists all of the ECL options in separate categories by function. Refer to the table to find the category of the desired option, then find that option in the appropriate section in the pages that follow.

Some options act as toggle switches; that is, one option turns a condition off and an accompanying option turns the condition on. These options are listed together, separated by a slash (/) character, in the table. If a default value for an option is applicable, that value is listed in the right column.





succession. See the discussion in the *PRIMOS User's Guide* for more information on prompt usage and specification. ECL uses the standard PRIMOS RDY brief prompt set by default. ECL converts to its own internal set if any of the above prompt options are used to set an internal prompt. The # character is meaningful only in an internal prompt; it is treated as just another printable character within a RDY-specified prompt.

- VERSION                      Displays ECL's full internal version and copyright notice, which are normally suppressed.
- HELP                         Displays all of ECL's available options.

### Terminal Bindings and Characteristics

The options described below help you define your terminal environment in ECL.

- BIND\_TERM  
  [*bindings\_file*]  
  [*term\_alias*]                Specifies the file containing command bindings. This option is ignored if terminal bindings are successfully restored via the -RESTORE\_TERM option. Otherwise, ECL first establishes the fundamental set of bindings, then reads the bindings file to supplement the fundamental set. Terminal-specific bindings are normally coupled with the terminal name provided by the -TERMINAL\_TYPE option. You can override this setting by specifying a terminal name alias in *term\_alias*, allowing slightly different capabilities or bindings to be combined. If no *bindings\_file* is specified, the fundamental bindings are simply reestablished. If the -BIND\_TERM option is not specified, current bindings are untouched, even if -TERMINAL\_TYPE was specified. The format of the bindings file is discussed in Chapter 6. To disable bound sequences in the fundamental set, rebind them to the *unbound* command.





- TERMINAL\_TYPE**  
[*type*]
- Specifies the terminal type of your terminal to ECL. This option is not case-sensitive. **-BIND\_TERM** uses the terminal type specified by **-TTP** to set up a bindings file and **-RESTORE\_TERM** verifies the terminal type when you restore a compiled bindings file. (These two options are discussed elsewhere in this section.) If you are not restoring or the restore operation is not successful, ECL searches the TERMINFO database for the entry *type* and initializes terminal capabilities. If *type* is not specified, ECL checks the global variable **.TERMINAL\_TYPE\$** for a match; if it does not find one, ECL establishes capabilities for a generic 24-line by 80-column terminal.
- WALLPAPER**  
[*bindings\_file*]
- Displays the current command bindings, including the function key programs. If *bindings\_file* is specified, ECL redirects the display to the specified file instead of the terminal. The display format conforms to the bindings file format discussed in Chapter 6. This means that files produced with this option can be reloaded using the **-BIND\_TERM** option.
- WIDTH [n]**
- Specifies the terminal width in *n* columns. This option overrides the width specified as **cols** in the TERMINFO database or the default width of 80 columns for the generic terminal. Omitting *n* or specifying it as 0 tells ECL that you have a terminal incapable of wrapping.
- XOFF / -NO\_XOFF**
- Overrides the PRIMOS TERM command XON/XOFF setting within the ECL command environment. **-XOFF** overrides any bindings to Ctrl-S/Ctrl-Q and uses them to stop and start output to the terminal, respectively. **-XOFF** is the default. **-NO\_XOFF** enables Ctrl-S/Ctrl-Q to execute the ECL commands to which they are bound. ECL reverts to the original **-XOFF** duplex setting when you invoke a PRIMOS command. This means that you can start and stop terminal output with Ctrl-S/Ctrl-Q while a PRIMOS command is executing, even if you have specified **-NO\_XOFF**. Once the command has finished executing, **-NO\_XOFF** is again in effect. If your terminal must rely on Ctrl-S/Ctrl-Q for output control, then always specify **-XOFF**.

## Controlling Terminal I/O

This section describes the options that control certain terminal I/O characteristics.

- CLEAN\_COMO /**  
**-NO\_CLEAN\_COMO** Controls ECL terminal output that is captured by a command output (COMO) file. **-CLEAN\_COMO** prevents the capture of all but the submitted commands and any ECL prompts displayed. **-NCCOMO** captures all character output within ECL. Be aware that **-CLEAN\_COMO** only cleans COMO files opened by the local system. **-CLEAN\_COMO** is the default.
- NO\_CLEAR\_LINE /**  
**-CLEAR\_LINE** Controls ECL feature which clears out the area on a terminal when needed for input. **-CLEAR\_LINE** turns feature on, **-NO\_CLEAR\_LINE** turns feature off. **-NO\_CLEAR\_LINE** is the default.
- NO\_EDIT\_COMI /**  
**-EDIT\_COMI** Controls the editing of input from a command input (COMI) file. **-NO\_EDIT\_COMI** passes back each line of a COMI file as uninterpreted without recording it within the command history. **-EDIT\_COMI** treats COMI file input as terminal input; all characters are passed through the ECL command interpreter and the submitted commands are incorporated into the command history. **-NO\_EDIT\_COMI** is the default.
- OBEY\_ERKL /**  
**-NO\_OBEY\_ERKL** Controls the use of the PRIMOS erase and kill characters. **-OBEY\_ERKL** always monitors and obeys the erase and kill characters as defined by PRIMOS, overriding any ECL binding associated with these characters. **-OBEY\_ERKL** enforces the erase character as being bound to *rubout\_char* and the kill character as being bound to *kill\_region* (with a count of 4). **-NO\_OBEY\_ERKL** allows you to define your own erase and kill characters within ECL. **-OBEY\_ERKL** is the default. For more information on the *rubout\_char* and *kill\_region* commands, see Chapter 6 and also see the *PRIMOS User's Guide*.
- ROW\_MAJOR /**  
**-COL\_MAJOR** Sets the display characteristics for automatic pathname completions. **-ROW\_MAJOR** displays the list in alphabetical order within rows (as the PRIMOS LD command does). **-COL\_MAJOR** orders the display alphabetically within columns. **-ROW\_MAJOR** is the default, although **-COL\_MAJOR** is more natural to read.

**-SILENT**

Instructs ECL to display only the most serious error messages, such as improper package installation or bad command-line syntax. **-SILENT** disables the display of most error, warning, or informational messages. This option is useful within CPL interfaces to ECL; in this case, the CPL program can examine the returned SEVERITY code and produce a message of its own.

**Command History**

This section describes the options that allow you to manipulate the command history.

**-NO\_STACK / -STACK**

Controls the logical structure of the command history as either a ring or a stack. **-NO\_STACK** specifies the structure as a ring. **-STACK** specifies the structure as a stack. In a ring structure, all submitted commands replace their ring entry; thus, changing a previous command replaces its original form. With a stack, each new command, even if it's only a modified previous command, is pushed onto the top of the stack and the history pointer is set to the top. While a stack minimizes lost commands, it is more difficult to execute a sequence of previous commands than with a ring. **-NO\_STACK** is the default.

**-NO\_STICK / -STICK**

Controls the positioning of the history pointer after a previous command has been recalled. **-STICK** does not reposition the history pointer; the pointer *sticks* to the position after the recalled command. **-NO\_STICK** moves the history pointer back to the top of the stack (if you have used the **-STACK** option) or repositions the pointer after the last *new* ring entry (if you have used the **-NO\_STACK** option). Making the history pointer stick allows a sequence of previous commands to be executed in a row. If you use **-STICK** within a ring history (by using the **-NO\_STACK** option), subsequently entering a new command overwrites the corresponding hidden history command. The **-SHOW\_HIDDEN** option discussed below informs you of this history command loss by revealing hidden commands automatically when used in conjunction with **-STICK**. **-NO\_STICK** is the default.

**-NO\_SHOW\_HIDDEN /  
-SHOW\_HIDDEN**

Controls the automatic display of the underlying hidden command in the command history at a new prompt (while the current event is within past history). **-NO\_SHOW\_HIDDEN** does not automatically reveal the hidden command, but instead provides a blank prompt line. **-SHOW\_HIDDEN** reveals the hidden command at the new prompt and the cursor is placed at the *beginning* of this command. **-SHOW\_HIDDEN** facilitates the rapid replay of previous consecutive commands in the command history without having to recall them explicitly by using, for example, the *prev\_line* or *next\_line* command. Also, the revealed command is automatically erased if the very next character sequence is not one intended to edit the hidden command (that is, when you start typing a new command). Note that **-SHOW\_HIDDEN** is not useful if **-NO\_STICK** is enabled because **-NO\_STICK** prevents sticking within past history.

**-RESTORE\_HISTORY  
historyfile**

Restores the command history from *historyfile*, which must be specified. The search ring and kill ring are restored along with the command history. Use this feature to save your command history (with the **-SAVE\_HISTORY** option) just before logging out or reinitializing your command environment, and then to restore the history upon logging back in. Note that ECL need not be enabled (with the **-ON** option) for these options to work, it needs only to be installed.

**-SAVE\_HISTORY  
historyfile**

Saves the command history to *historyfile*, which must be specified (unless you use **-SHIST** on the same command line as **-RHIST historyfile**). The search ring and kill ring are saved along with the command history. Use this feature to save your command history just before logging out or reinitializing your command environment, and then to restore the history (with the **-RESTORE\_HISTORY** option) upon logging back in. Note that ECL need not be enabled (with the **-ON** option) for these options to work, it needs only to be installed.

## Pathname Completion

This section describes the options that allow you to control automatic pathname completion and abbreviation expansion.

- |  |  |
|--|--|
| <p><b>-NO_WILD_ABBREV /<br/>-WILD_ABBREV</b></p>       | <p>Controls automatic abbreviation expansion as the first step in automatic pathname completion. (Automatic pathname completion is done with the <i>expand_wild</i> command.) <b>-WILD_ABBREV</b>, after extracting the partial pathname from the command line for completion, first references the PRIMOS abbreviation mechanism to expand any argument-type abbreviations that may exist within the specified pathname. Global variable references are also expanded. As in the PRIMOS command environment, expansion is not performed if the first character in the command line is a tilde (~). (Note that expansions are sensitive to cursor position within the pathname while the <b>-WILD_TAIL</b> option is in effect.) The default, <b>-NO_WILD_ABBREV</b>, disables automatic abbreviation expansion.</p> |
| <p><b>-WILD_DIRECTORY /<br/>-NO_WILD_DIRECTORY</b></p> | <p>Controls the automatic appending of the angle bracket (&gt;) character when the last element of an automatic pathname completion is a directory. The default, <b>-WILD_DIRECTORY</b>, causes &gt; to always be appended. <b>-NO_WILD_DIRECTORY</b> specifies that you must explicitly type the right angle character when you want to reference a file system object within that directory. This option is useful because many PRIMOS commands, such as ATTACH, do not allow pathnames that end with the right-angle character. The default is <b>-WILD_DIRECTORY</b>.</p>  |
| <p><b>-WILD_MENU /<br/>-NO_WILD_MENU</b></p>           | <p>Controls the numbering of potential pathname completions displayed in the menu list generated by the <i>expand_wild</i> command. The default, <b>-WILD_MENU</b>, numbers each potential completion for selection by means of the <i>expand_wild_menu</i> command. <b>-NO_WILD_MENU</b> displays an unnumbered menu list.</p>  |



## Examples

In the first example, ECL is enabled for use at PRIMOS command level:

```
OK, ECL -ON
Editing command input with ECL$LIB (US-English Prime-ECS)
OK,
```

In the next example, a new RDY prompt is defined:

```
OK, ECL -RB '>> '
>>
```

In the next example, all of the RDY prompts are customized:

```
>> ECL -RB 'YES! ' -EB 'NO! ' -WB 'FLAG ' -ON
YES!
```

In the next example, the number-sign character (#) in the prompt substitutes the current command history entry number in place of the # character:

```
YES! ECL -RB '#: '
004:
```

In the next example, a common way to set up the command history is shown. This method allows you to recall a sequence of commands from the command history, but if you modify one of those commands, the modified command is added to the end of the history rather than replacing the original command in the history.

```
004: -ECL -STICK -SHOW_HIDDEN -STACK
005:
```

In the next example, the default bindings file is established, restored and saved on one command line. This is the recommended method for using a bindings file.

```
005: ECL -TTP PT200 -BTERM -RTERM COMPILED.FILE -STERM
Established PT200 terminal capabilities (9600 baud).
(EDIT_CMD_LINE)
Established default bindings (92% available). (EDIT_CMD_LINE)
Saved PT200 terminal capabilities and bindings (9600 baud).
(EDIT_CMD_LINE)
006:
```





## EXPAND\_SEARCH\_RULES

EXPAND\_SEARCH\_RULES provides the fully-qualified pathname of a specified file system object or search list.

### Format

EXPAND_SEARCH_RULES	[	<i>objectname</i> [-LIST_NAME <i>listname</i> ] -ACCESS_CATEGORY -DIRECTORY -FILE -REFERENCING_DIR <i>pathname</i> -SEGMENT_DIRECTORY -SUFFIX <i>suffix</i>	]
---------------------	---	---	---

### Argument

*objectname*

The entryname of the file system object to expand to a fully-qualified pathname.

-LIST\_NAME *listname*

Permits you to specify the name of the search list that PRIMOS should use to locate the named object. If you do not specify -LIST\_NAME, PRIMOS selects the search list (ATTACH\$ or COMMAND\$) based on the suffix of the objectname. If you specify -LIST\_NAME as COMMAND\$, PRIMOS supplies suffixes to the objectname in the following sequence: .RUN, .SAVE, .CPL, no suffix.

### Options

-ACCESS\_CATEGORY

Specifies that the file system object sought is an access category. This option allows you to limit the search to access categories.

-DIRECTORY

Specifies that the file system object sought is a directory. This option allows you to limit the search to directories.

-FILE

Specifies that the file system object sought is a file. This option allows you to limit the search to files.

.....  
**EXPAND\_SEARCH\_RULES**

- REFERENCING\_DIR *pathname*** Permits you to specify a search rule that PRIMOS substitutes for the REF-ERENCING\_DIR entries in the search list. EXPAND\_SEARCH\_RULES then uses the search list to find the file system object.
- SEGMENT\_DIRECTORY** Specifies that the file system object sought is a segment directory. This option allows you to limit the search to segment directories.
- SUFFIX *suffix*** Specifies suffixes that PRIMOS appends to the objectname to conduct the search. The suffixes must begin with a period (for example, .RUN). You can specify a maximum of eight suffixes following a -SUFFIX option. PRIMOS searches for suffixes in the order listed. If no match is found with all listed suffixes, PRIMOS searches for the object with no suffix.

**Usage**

EXPAND\_SEARCH\_RULES uses the Search Rules facility to expand an objectname to a fully qualified pathname. *objectname* is used to specify the objectname to be expanded. *objectname* must include all suffixes. PRIMOS either returns the fully-qualified pathname to the terminal screen, or issues a message indicating that the requested object could not be found.

EXPAND\_SEARCH\_RULES uses search lists to determine the fully-qualified pathname. You can specify the appropriate search list by using the LIST\_NAME option. If you do not specify a LIST\_NAME option, EXPAND\_SEARCH\_RULES selects the appropriate search list based on the suffix of the objectname.

PRIMOS selects the following search lists by default:

<i>Suffix</i>	<i>Search List</i>
.RUN	COMMAND\$
.SAVE	COMMAND\$
.CPL	COMMAND\$
Other/no suffix	ATTACH\$

Note that because search lists are used, sub-directories are not searched unless listed individually as a search rule in the specified or default list.

## Examples

To expand the name of a command:

```
OK, EXPAND_SEARCH_RULES PROG.RUN
Pathname: <SYSTEM>CMDNC0>PROG.RUN
OK,
```

or

```
OK, EXPAND_SEARCH_RULES PROG -LIST_NAME COMMANDS
Pathname: <SYSTEM>CMDNC0>PROG.RUN
OK,
```

If you do not include the .RUN suffix or the -LIST\_NAME option, EXPAND\_SEARCH\_RULES uses the ATTACH\$ search list with the following result:

```
OK, EXPAND_SEARCH_RULES PROG
Top-level directory not found or inaccessible.  PROG (ESR)
OK,
```

The same command line works if PROG is a top-level directory:

```
OK, EXPAND_SEARCH_RULES PROG
Pathname: <USERS>PROG
OK,
```

You can create a search list in your directory (see SET\_SEARCH\_RULES in this chapter) and direct EXPAND\_SEARCH\_RULES to use it with the -LIST\_NAME option. If you create a search list called MYRULES containing the line

```
<USERS>MYHOME>MEMOS>DECEMBER
```

and the file CHRISTMAS\_PARTY is in that directory, you get the fully-qualified pathname by entering:

```
OK, EXPAND_SEARCH_RULES CHRISTMAS_PARTY -LIST_NAME MYRULES
Pathname: <USERS>MYHOME>MEMOS>DECEMBER>CHRISTMAS_PARTY
```

You can also invoke EXPAND\_SEARCH\_RULES as a CPL function. When invoked as a CPL function, EXPAND\_SEARCH\_RULES returns the fully-qualified pathname to a variable in the CPL program (see the EXPAND\_SEARCH\_RULES function in Chapter 3).

For further details concerning the search rules facility, see the *Advanced Programmer's Guide II: File System*.

See also LIST\_SEARCH\_RULES, and SET\_SEARCH\_RULES.

See Chapter 3 for the use of EXPAND\_SEARCH\_RULES as a command function.

.....  
F77

F77

F77 loads the Prime FORTRAN 77 compiler and compiles the object program from the ASCII file specified by *pathname*.

### **Format**

**F77 *pathname* [*options*]**

### **Usage**

It is recommended that you give *pathname* an .F77 suffix. FORTRAN programs can be compiled in V mode or I mode but not in R mode. The F77 compiler can compile programs written in FORTRAN IV.

For detailed information on the operation and options of the F77 compiler, see the *FORTRAN 77 Reference Guide*, and the *Translator Family Software Release Document, Release T3.0-23.0*.

F77DML

F77DML invokes the FORTRAN 77 Data Manipulation Language (DML) preprocessor.

### **Format**

**F77DML [[-INPUT] *in-pathname* [[-OUTPUT] *out-pathname*]  
[[-ERROR] *err-pathname*] [-DYNAMIC] [-NO\_LINE\_NUMBER]**

### **Arguments and Options**

<b>-INPUT</b> <i>in-pathname</i>	<i>in-pathname</i> must be present, with or without the -INPUT option tag. If the -INPUT tag is omitted, <i>in-pathname</i> must appear as the first argument.
<b>-OUTPUT</b> <i>out-pathname</i>	<i>out-pathname</i> may be given with or without the -OUTPUT option tag. If the -OUTPUT tag is omitted, <i>out-pathname</i> must appear immediately following <i>in-pathname</i> , also given without an option tag.

<b>-ERROR</b> <i>err-pathname</i>	<i>err-pathname</i> may be given with or without the <b>-ERROR</b> option tag. If the <b>-ERROR</b> tag is omitted, <i>err-pathname</i> must appear as the third argument, following <i>in-pathname</i> and <i>out-pathname</i> , both given without option tags. To specify <i>err-pathname</i> but not <i>out-pathname</i> , the <b>-ERROR</b> option must be used.
<b>-DYNAMIC</b>	Allows programs to invoke a schema at runtime by a schema name rather than by the schema number obtained at compile time. The schema name is used at runtime to dynamically resolve the schema number.
<b>-NO_LINE_NUMBER</b>	Suppresses the generation of line numbers in the output file.

### Usage

The F77 DML translates embedded DML statements in the source file specified by *in-pathname* into a form that is usable by the FORTRAN 77 compiler. The DML will write the translated DML statements into *out-pathname* when specified, otherwise the output will be written to a file of the form *in-pathname.F77* (when the source file has a *.F77DML* suffix) or *D\_in-pathname*, in the current directory. Any errors will be written to *err-pathname* when specified, otherwise DML writes errors to *in-pathname.ERROR*.

Example of the command line when used without the **-INPUT**, **-OUTPUT**, and **-ERROR** option tags.

```
F77 DML in-pathname [out-pathname [err-pathname]] [options]
```

When omitting option tags, the pathnames must appear in the order shown. If option tags are used, the options specified may appear in any order.

For further information, see the *DBMS Data Manipulation Language Reference Guide*.

.....  
F77SUBS

## F77SUBS

F77SUBS invokes the FORTRAN 77 Subschema Data Definition Language (DDL) compiler.

### **Format**

F77SUBS [*in-pathname*] [-OUTPUT *out-pathname*] [-LIST *list-pathname*]

### **Usage**

*in-pathname* is a source file containing the subschema definition. The CBL Subschema DDL compiler translates the source file into the subschema table. The subschema table is placed in the same directory as the schema table. Use the -OUTPUT option to place the subschema table in another directory.

The compiler also produces an output listing file that contains the source listing, error messages, and a map of the User Work Area. The file is called *in-pathname*.LIST. The listing file is placed in the directory that contains the schema table. Use the -LIST option to specify another directory for the listing file or a different name for the file.

For more information on F77SUBS, see the *DBMS Data Description Language Reference Guide*.

## FAP

FAP invokes the FORMS Administrative Processor.

### **Format**

FAP

### **Usage**

The FORMS Administrative Processor creates and maintains the form definition catalog, configures new terminals and new device drivers into the FORMS system, and obtains system status information. When invoked, FAP displays an asterisk (\*) prompt and waits for you to enter a FAP command.

For detailed information on FAP, see the *FORMS Programmer's Guide*.

See also FDL; FED; FORMS.

## FAU

FAU invokes the PRISAM File Administrator Utility (FAU).

### **Format**

FAU

### **Usage**

FAU is an interactive, menu-driven utility that manipulates PRISAM files. For details, see the *PRISAM User's Guide*.

## FDL

FDL invokes the FORMS Definition Language (FDL) compiler.

### **Format**

FDL *pathname* [*options*]

### **Usage**

*pathname* is an FDL source file. It is recommended that you give *pathname* a .FORM suffix. For detailed information on the operation and options of FDL, see the *FORMS Programmer's Guide*.

See also FAP; FED; FORMS.





Example of the command line when used without the `-INPUT`, `-OUTPUT`, and `-ERROR` option tags.

`FDML in-pathname [out-pathname [err-pathname]] [options]`

When omitting option tags, the pathnames must appear in the order shown. If option tags are used, the options specified may appear in any order.

For further information, see the *DBMS Data Manipulation Language Reference Guide*.

## FED

FED invokes the FORMS Editor, FED.

### *Format*

`FED [-PROFILE profile-pathname] [-CATALOG catalog-pathname]`

### *Usage*

Use FED to create and maintain form definitions. For further information, see the *FED User's Guide*.

See also FAP; FDL; FORMS.

## FILMEM

FILMEM clears the contents of user address space to zeros for unsegmented (R-mode) programs.

### *Format*

`FILMEM [ALL]`

### *Usage*

Under PRIMOS, FILMEM with no argument fills memory locations 100g through 77777g with zeros. FILMEM ALL clears 100g through 77777g.

.....  
FILVER

FILVER

FILVER compares two files and reports any differences.

### Format

FILVER [*pathname1* *pathname2*]

### Arguments

*pathname1* *pathname2* Pathnames of the files you want to compare, separated by a blank. If you do not specify the pathnames, FILVER prompts you for them.

### Usage

*pathname1* and *pathname2* are the two files you want to compare. If you do not specify the pathnames, FILVER prompts you for them:

```
OK, FILVER  
FILE 1: PROG1  
FILE 2: PROG2
```

The two files are compared for equivalence. FILVER compares every halfword in *pathname1* against the equivalent halfword in *pathname2* (One halfword equals two bytes.) The differences are reported in a form useful for comparing runfiles. You should have listings of both files to make use of the comparison information printed by FILVER.

If both files are exactly the same, the following message is displayed:

```
FILES ARE EQUAL
```

If the files are not the same, each difference is displayed with the word DIFF followed by four six-digit numbers in octal:

**DIFF *w x y z***

*w* and *x* describe the position of the file, *w* being a sector number and *x* being the offset within the sector. A sector is 2048 bytes or 2000<sub>8</sub> (1024 decimal) halfwords long. (You must take into account the 18-byte header in runfiles and any offset from a sector boundary in the starting location of the runfile.) *y* is the value of the differing halfword in *pathname1*; *z* is the value of the halfword in *pathname2*.

If one file is longer than the other, FILVER also lists the lengths, in halfwords, of both files. (Halfwords are shown as "WORDS".)

If the FILVER output is more than one full screen, the prompt CONTINUE= is displayed. To display the next screen, enter YES, OK, or AYE in uppercase letters. To stop the output, enter NO or NAY in uppercase. Any other response produces a WHAT? error prompt.

The following example illustrates how FILVER operates. The files to be compared are named ALPHA and BETA.

```
OK, SLIST ALPHA
These are the times that try men's souls.
OK, SLIST BETA
These are the chimes that fry hen's soles.
OK, FILVER ALPHA BETA
```

```
DIFF 000000 000007 172351 161750
DIFF 000000 000010 166745 164755
DIFF 000000 000011 171640 162763
DIFF 000000 000012 172350 120364
DIFF 000000 000013 160764 164341
DIFF 000000 000014 120364 172240
DIFF 000000 000015 171371 163362
DIFF 000000 000016 120355 174640
DIFF 000000 000017 162756 164345
DIFF 000000 000020 123763 167247
DIFF 000000 000021 120363 171640
DIFF 000000 000022 167765 171757
DIFF 000000 000023 166363 166345
DIFF 000000 000024 127212 171656
FILE LENGTHS DIFFER AS FOLLOWS:
FILE 1: 21 WORDS
FILE 2: 22 WORDS
```

```
FILES NOT EQUAL
ER!
```

See also CMPF.

.....  
FSUBS

## FSUBS

FSUBS invokes the FORTRAN Subschema Data Definition Language (DDL) compiler.

### Format

**FSUBS** *in-pathname* [-OUTPUT *out-pathname*] [--LIST *list-pathname*]

### Usage

*in-pathname* is the name of the source file containing the subschema definition. The FORTRAN subschema compiler translates the subschema DDL into the subschema table. The subschema table is placed in the same directory as the schema table, unless you specify the -OUTPUT option to place the subschema table in another directory.

The compiler also produces a listing file (named *in-pathname*.LIST) that includes the DDL source text, error messages, and a map of the User Work Area. The listing file is placed in the directory that contains the schema table, unless you use the -LIST option to place it elsewhere.

If you enter FSUBS without arguments, you are prompted for the name of the source file.

For details, see the *DBMS Data Description Language Reference Guide* and the *DBMS Administrator's Guide*.

## FTN

FTN loads the Prime FORTRAN IV compiler and compiles the object program from the ASCII file specified by *pathname*.

### Format

**pathname** [*options*]

### Usage

*pathname* should have the .FTN suffix, although this form is not mandatory. For detailed information on the options and operation of FTN, see the *FORTRAN Reference Guide*.

See also F77.

FTR

FTR invokes the File Transfer Request (FTR) utility.

The File Transfer Request utility allows you to request and manage file transfers interactively between Prime computers connected by PRIMENET. FTR is part of the File Transfer Service (FTS), which is a separately priced product.

You can use FTR between machines that have been configured using the FTGEN command. For information on transferring files to or from sites that are not configured, see the *User's Guide to Prime Network Services*.

FTR offers two distinct functions: a transfer request function and a transfer management function. The first is used to submit a transfer request and the second to manage requests that have been submitted. Each function has its own FTR command-line format.

After you request a file transfer, FTR in turn submits the request to FTS, which then queues it for transfer. (Because FTS queues all requests on the local computer, you can make requests even when a communications link or remote computer is down.) After your request is queued, you can use the second FTR management function to display, modify, suspend, release, abort, or cancel the request.

**Format**

$$FTR \left\{ \begin{array}{l} \textit{source-pathname} \textit{ [destination-pathname] [submit-options]} \\ \textit{manage-option} \textit{ [request-name] [-QUEUE queue-name] \left\{ \begin{array}{l} \textit{-QUERY} \\ \textit{-NO\_QUERY} \end{array} \right\}} \end{array} \right\}$$

**Arguments and Options**

- source-pathname* Specifies the name of the file to be transferred. If you specify a filename rather than a pathname, the file must be in the directory to which you are currently attached.
- destination-pathname* Specifies the name the file is given at the destination site. Do not specify *destination-pathname* if you use the *-DEVICE* submit option.

**Submit Options**

Use one or more options from the following list to define the parameters to use for the transfer.

- COPY* Transfers a copy of the file, not the original. (This mode is used by FTR by default. To transfer an original use the *-NO\_COPY* option.) This option does not affect files being transferred from a remote site.

- DEFER *date.time*** Specifies a date and time when the request is to be submitted for processing. If this option is not specified, the request is submitted immediately. *date.time* may be in any of the following formats:

*yy-mm-dd.hh:mm:ss*  
*mm/dd/yy.hh:mm:ss*  
'*dd mon yy hh:mm:ss*'  
'*dd mon yy.hh:mm:ss*'

The last two formats contain spaces and therefore must be enclosed within single quotation marks. *mon* stands for a month's first three characters — JAN, FEB, MAR, etc. Double letters represent one- or two-digit numbers. Leading zeros are not required. The *hh* field uses 24-hour notation. Omitted date fields are replaced by current date information; omitted time fields are replaced by zeros.

For more information about the format of *date.time*, see the section Wildcard Options in Chapter 4.
- DELETE** Deletes the local source file after it has been transferred successfully. (Default is **-NO\_DELETE** only if you use **-NO\_COPY**; otherwise, the copy is deleted.) This option does not affect files being transferred from a remote site.
- DEVICE LP** Transfers the file directly to the default line printer of a remote site. Only DAM and SAM files can be used with this option. Do not specify *destination-pathname* if you specify this option. (**-DEVICE** and **LP** must be separated by a space.)
- DSTN\_FILE\_TYPE *type*** Specifies the destination file type. *type* is one of the following: SAM, DAM, SEGSAM, or SEGDM. (Default is the source file type when sending and null when fetching.)
- DSTN\_NTFY** Notifies the destination user when a file transfer starts and ends. Use **-DSTN\_USER** to specify the destination user.
- DSTN\_SITE *site-name*** Specifies the name of the site to which the file is being transferred. (Default is local site.) Required when sending a file to a remote site.
- DSTN\_USER *user-name*** Specifies the owner of the file at the destination site. (Default is the login user ID for a local site and null for a remote site.)

- HOLD** Holds the request in a queue so that it is not initiated until you release the request with the **-RELEASE** management option. The **-HOLD** option is also a management option. See the following section listing management options. Using the **-HOLD** option when you submit a request *assures* that the transfer is not initiated until you or the administrator release it. If you submit a request when the queue is empty, the transfer may be initiated immediately, therefore a subsequent FTR **-HOLD** command will not arrive in time.
- LOG *pathname*** Specifies a log file that records the progress of the request, including whether the transferral was successful. If *pathname* exists, new log entries are appended to it; otherwise, a new log file is created.
- MESSAGE\_LEVEL *level*** Specifies the level of detail entered in a log file. *level* is one of the following: **NORMAL** (minimum detail, which is the default); **DETAILED** (logs all events); **STATISTICS** (same as **DETAILED** but with statistics); or **TRACE** (same as **STATISTICS** but with trace information). To use this option, you must also specify the **-LOG** option.
- NAME *external-name*** Specifies the external name of the request. (The default external name is the name of the source file.)
- NO\_COPY** Transfers the original source file, not a copy of it.
- NO\_DELETE** Does not delete the local source file after it has been transferred successfully. (Default)
- NO\_DSTN\_NTFY** Does not notify the destination user when a transfer starts and ends. (Default)
- NO\_QUERY** Does not query you when you submit or modify a file transfer request. This option is useful when you are running FTS from a CPL file.
- NO\_SRC\_NTFY** Does not notify the source user when a transfer starts and ends. (Default)
- PRIORITY *value*** Sets the relative priority of a transfer request on the request queue. This means that you are no longer bound to the first in, first out queuing scheme that FTR uses by default. If you do not specify the **-PRIORITY** option, the default priority of a request is set by the Network Administrator using FTGEN. *value* can be 1–9, excluding 7 and 8, which are reserved for administration.
- QUERY** Asks you to confirm any request questions. (Default)

- QUEUE *queue-name*** Specifies the name of the file transfer request queue in which a request is to be placed. (Default is the configured remote site queue, or OPEN\_SYSTEM if that has been configured.) Normally, you use this option when sending a file to a remote site that has not been configured by FTGEN, in which case you must specify a queue that has been configured with FTGEN.
- SRC\_FILE\_TYPE *type*** Specifies the source file type. *type* is one of the following: DAM, SAM, SEGDAM, or SEGSAM. (Default is the type of the local source file; otherwise, the default is null.)
- SRC\_NOTIFY** Notifies the source user when a transfer starts and ends.
- SRC\_SITE *site-name*** Specifies the site from which the file is being transferred. (Default is the local site.) Required when transferring a file from a remote site.
- SRC\_USER *user-name*** Specifies the owner of the file at the source site. (Default is the login user ID for a local source and null for a remote source.)
- HELP [*option*]** Provides detailed information about the command and its options.

### Manage Options

Use only one option from the following list to specify the type of action you want to take on the specified request(s).

- request-name*** Identifies the file transfer request. You can use either the external or internal name as *request-name*. The **external name** is either the name of the file to be transferred or a specific name that you assign to the request when you submit it. You usually refer to the request by its external name. FTS assigns the **internal name**, always a number, which you may use to distinguish between two requests that have the same external name. If you do not specify *request-name*, the action of the command is performed on all your requests.
- ABORT** Aborts a transfer in progress.
- CANCEL** Cancels a file transfer request.
- DISPLAY** Displays a detailed status of the request.



<b>-HOLD</b>	Delays a file transfer until you or the administrator releases the request with the <b>-RELEASE</b> option. Only waiting requests will be held by this option; requests in other states, such as in progress, already held, and aborting, are ignored.
<b>-MODIFY</b> [ <i>submit-options</i> ]	Changes the characteristics of a queued request. <i>submit-options</i> are the request submittal options listed above.
<b>-RELEASE</b>	Releases a request held with the <b>-HOLD</b> option.
<b>-STATUS</b>	Displays information about the current status of the request.
<b>-STATUS_ALL</b>	Displays information about all the requests on all the configured transfer queues.

### **Requesting a File Transfer**

To request the transfer of a file, use the following FTR command format:

**FTR** *source-pathname* [*destination-pathname*] [*submit-options*]

### **Source and Destination Sites**

File transfers take place between sites. A site is a single computer that is identified by a unique site name. (Prime sites normally use their PRIMENET node names as site names.) Files are transferred from a source site to a destination site. One of these sites must be your local site. The other site is usually a remote site, but it can be a different directory on the same local site. FTR cannot transfer files between two remote sites.

Files are transferred initially to a temporary file to avoid accidental access to the destination file during the transfer process. The temporary file is renamed to the required destination filename upon successful completion of the transfer. For further details, see the *User's Guide to Prime Network Services*.

### **Sending a File**

The basic FTR command format for sending a file from your local site to another Prime computer is:

**FTR** *source-pathname destination-pathname -DSTN\_SITE sitename* [*submit-options*]

*submit-options* are the options listed above.

The following example illustrates sending a file:

```
FTR JONES>NEWS SMITH>MAY.MEMO -DS SYS3 -LOG JONES>NEWS.LOG
```



The `-NO_QUERY` option suppresses the verification of your action on each request. By default, verification is enabled.

For example, the following cancels all your requests in all queues without verification:

```
FTR -CANCEL -NQ
```

### **-MODIFY Manage Option**

The `-MODIFY` option modifies the parameters originally specified by the submit options of a request after the request has been queued. Any request that has been initiated or aborted is ignored. The command format for modifying parameters is:

```
FTR -MODIFY [request-name] [new-options]
```

You can modify any request parameter, with the following exceptions:

```
-COPY / -NO_COPY
-DSTN_SITE
-DSTN_FILE_TYPE
-HOLD
-QUEUE
-SRC_SITE
-SRC_FILE_TYPE
```

Note that the `-QUEUE` option can not be used to specify the queue where the request is located in the manner consistent with the other management options. When `-QUEUE` follows `-MODIFY` on the command line, an error is reported.

If you don't specify any options, the following modifications are made to the requests:

- Date and time of last retry is set to zero
- Number of retries is set to zero

If the requests are waiting, this makes them eligible for immediate retry by the file transfer server without waiting until the next 30 minute retry period has expired. This can be useful, such as when a previously inoperative site becomes operational and the you would like your pending request to be retried as soon as possible.

### **FTR Help Screen**

To display an online help screen listing the syntax and meanings of FTR options, issue the FTR command without an argument or issue the command:

```
FTR -HELP USAGE
```

For complete information on FTR, see the *User's Guide to Prime Network Services*.

See also FTS.

.....  
FTS

## FTS

FTS is the Prime communications File Transfer Service, a separately priced product that transfers files between computers connected by PRIMENET. FTS is not a command. The commands used with FTS are

- The FTR command, which provides the user interface to FTS
- The FTOP command, which provides operator control of FTS
- The FTGEN command, which allows the System Administrator to define the FTS environment

For a detailed description of FTS, see the *User's Guide to Prime Network Services*. See also FTR.

## HDXSTAT

HDXSTAT displays the current status of all lines and sites of a half-duplex (HDX) network configuration.

### *Format*

HDXSTAT

### *Usage*

When a network connection is defined as half duplex, communication between two Prime systems lasts only for the duration of the phone call.

The HDXSTAT display shows each defined HDX site, each defined HDX line, the association (where one exists) between individual sites and lines, the state of the telephone connection, and the status of the link. The link status is one of the following:

<b>assigned</b>	The line has been reserved for use by HDX PRIMENET, but is not yet set to receive or initiate calls.
<b>awaiting call</b>	A line has been set to receive a call from any remote site, but no remote site has yet called in.
<b>not assigned</b>	The line is not in use by HDX PRIMENET.
<b>offline</b>	No connection to the site exists.
<b>running</b>	The network connection to the remote site is working.
<b>trying to establish</b>	A line has been set to initiate calls to a remote site, but contact with the site has not been made.

For further information on HDXSTAT, see the *User's Guide to Prime Network Services*.

## HELP

The HELP command invokes a menu-based help facility that may be used to read PRIMOS Help files.

Each PRIMOS command has a Help file that explains the syntax and options for the command. In most cases, the Help file also refers the user to the manual that contains the most comprehensive information on that command.

The Help menu interface provides:

- Search and selection of commands, topics, and character strings, including the use of abbreviations and wildcarding.
- Hierarchical organization of Help files.
- Ability to navigate through a given database from entry to entry and from level to level within multilevel Help files.
- Backward and forward movement within text displays.
- Direct access to cross referenced material.
- Multiple database access.
- Display of both system- and user-created databases.
- Use of ACLs to restrict access to database entries.

**Format**

$$\text{HELP } \left[ \begin{array}{l} \{ \text{command-name} \\ \text{topic} \\ \text{search-string} \} \left[ \text{-OPTIONS } \left\{ \begin{array}{l} \text{option-name} \\ \text{subtopic} \end{array} \right\} \right] \end{array} \right]$$

-DATABASE *database-name*  
-NOHEADER  
-TERMINAL\_TYPE *termtyp* ..  
-HELP

**Arguments and Options**

<i>command-name</i>	The name of the PRIMOS command you want Help with.
<i>topic</i>	The name of a special Help file containing a list of PRIMOS commands that relate to that topic.

- search-string*                      A wildcarded string. The Help system will open a menu listing all PRIMOS commands and topics that match *search-string*. For example, if LI@@ is the search string, the Help menu you see lists all the PRIMOS commands and topic files that begin with LI, such as LISP, LIST, LIST\_MOUNTS.
  
- DATABASE**  
*database-name*                      Specifies an alternate Help text database for the Help facility. See Using Alternate Help Files below .
  
- NOHEADER**                              Informs the Help system that the Help files in the specified database do not have the standard 3-line PRIMOS copyright header. This option allows you to read any type of help text using HELP. If you do not use the **-NOHEADER** option the first 3 lines of the help file are ignored.
  
- OPTIONS**    { *option-name* }              Enters the Help file for the selected command at a screen explaining *option-name* or listing the options related to *subtopic*.  
                  { *subtopic*                      }
  
- TERMINAL\_TYPE**  
*termtype*                              Specifies your terminal type. Use this option if the .TERMINAL\_TYPE\$ global variable is not set, or if you wish to override its value. *termtype* must be a type defined in the CURSES database. See Specifying a Terminal Type below.
  
- HELP**                                      Provides a description of the HELP command syntax.

### Specifying a Terminal Type

In order for the Help facility to work properly you must correctly specify your terminal type.

At Rev 23.3, Help uses CURSES for its screen-oriented functionality. Any terminal type defined for CURSES can take advantage of the screen interfaces to Help. If your terminal type is not defined, you can create a definition file using CURSES for HELP to use.

Help finds out what kind of terminal you have in one of two ways. If the **-TERMINAL\_TYPE** (**-TTP**) option is present, Help checks that the terminal type you specify is valid for CURSES. 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 CURSES. If it is, Help opens the Main Menu.

If your terminal type is not defined for CURSES, 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 to display the first menu. Note that Help uses an ASCII screen format when CURSES can not be used, therefore screen features such as reverse video are not available.

### Starting the HELP Facility

PRIMOS Help files may be opened at any level directly from the PRIMOS command prompt, or by selection from the Main Menu. The Main Menu provides the means to open any of the Help files, list command names, and search for commands that match wildcarded strings.

To get to the Main Menu, enter HELP on the PRIMOS command line:

OK, HELP

The main menu appears, as shown:

```

MAIN MENU                                     Online HELP Utility
Field Selections:

-A Action
-D Database
-K Keyword
-Q Quit

Press RETURN for Defaults

Keyword or Topic: [                               ]
Database:          [PRIMOS                        ]

Action:
(-Q)uit (-R)efresh (-C)ontinue (-K)eyword (-D)atabase (-H)elp
  
```

For information on using the Main Menu , type HELP HELP on the PRIMOS command line.

From the command line, you can open the Help facility at a specific command by specifying the command as an argument. The first screen of the COPY command can be viewed by entering the following:

OK, HELP COPY





HPSD

HPSD loads a version of PSD (the Prime Symbolic Debugger) that is stored in the upper portions of memory. For details, see the *Assembly Language Programmer's Guide*.

**Format**

HPSD

**Usage**

When the HPSD program starts, it displays a dollar-sign prompt (\$) and waits for a debugging command.

See also DBG; PSD; IPSD; VPSD.

ICE

See INITIALIZE\_COMMAND\_ENVIRONMENT.

INFO

INFO invokes the Prime INFORMATION system, a separately priced product.

**Format**

INFO

**Usage**

Prime INFORMATION is documented in numerous books. For a complete list of documents see the Prime INFORMATION section of the *Guide to Prime User Documents*.

.....  
INITIALIZE\_COMMAND\_ENVIRONMENT

## INITIALIZE\_COMMAND\_ENVIRONMENT

INITIALIZE\_COMMAND\_ENVIRONMENT resets a user's command environment to the state it was in at login.

Before using this command, you may solve the problem without reinitializing your environment by issuing the CLOSE -ALL command followed by the RELEASE\_LEVEL -ALL command. If this does not work, then use INITIALIZE\_COMMAND\_ENVIRONMENT.

### *Format*

INITIALIZE\_COMMAND\_ENVIRONMENT [-SERVER]

### *Option*

-SERVER When you use INITIALIZE\_COMMAND\_ENVIRONMENT with the -SERVER option, all the operations described in the next section called ICE Operation are performed. In addition, the -SERVER option performs the following operations:

- Terminates all of the server's InterServer Communication (ISC) sessions.
- Terminates all PRIMIX child processes that are a part of the server. The server is the server to which the caller's process belongs.
- Deletes all synchronizers and timers.

-SERVER is available only to a terminal or phantom process; it is not available to a child process.

### *ICE Operation*

The INITIALIZE\_COMMAND\_ENVIRONMENT command performs the following operations:

- Closes all open files, including the COMOUTPUT file
- Returns you to your origin directory (Initial Attach Point)
- Executes your login file
- Resets search rules to system defaults
- Resets terminal erase and kill characters
- Forces terminal output on



.....

*INPUT*

## **INPUT**

INPUT opens a source file on File Unit 1 for reading.

### **Format**

**INPUT** *pathname*

### **Argument**

*pathname*      The pathname of the file to be opened.

### **Usage**

INPUT has the same effect as the command

**OPEN** *pathname* 1 1

For PMA and FTN, the source filename is usually provided with the command that starts assembly or compilation.

## IPSD, IPSD0, IPSD16

IPSD loads and initializes an extended version of the Prime Symbolic Debugger (PSD) that supports S-mode, R-mode, V-mode, and I-mode programs. Unlike the VPSD debugger, IPSD is able to map in the programs to be debugged, and can be used on both static-mode (SEG) programs and EPF programs.

IPSD0 loads and starts the I-mode version of the Prime Symbolic Debugger (PSD). IPSD0 is loaded at location '0. Its commands are identical to those for IPSD.

IPSD16 loads and starts the I-mode version of the Prime Symbolic Debugger (PSD). IPSD16 is loaded at location '160000. Its commands are identical to those for IPSD.

### **Format**

```
IPSD [pathname] [program_command_line] [-FCN]
IPSD0 [pathname] [program_command_line] [-FCN]
IPSD16 [pathname] [program_command_line] [-FCN]
```

### **Arguments and Option**

<i>pathname</i>	The pathname of the EPF or static-mode program to be mapped in and executed. The .RUN, .SEG, and .SAVE suffixes are optional.
<i>program_command_line</i>	The command line argument(s) expected by the program.
-FCN	Executes the user program as an EPF command function.

### **Usage**

For more information, see the *Assembly Language Programmer's Guide*.

See also DBG; HPSD; PSD; VPSD.

.....

JOB

JOB

JOB submits and manages individual jobs run in the Batch subsystem.

### Format

$$\text{JOB} \left\{ \begin{array}{l} \textit{pathname} \textit{ [submit-options]} \\ \textit{jobid} \textit{ manage-option} \\ \textit{[jobid]} \textit{ monitor-option} \left[ \left\{ \begin{array}{l} \text{ALL} \\ \text{TODAY} \end{array} \right\} \right] \end{array} \right\}$$

### Arguments

*pathname*

Specifies the pathname of either a CPL file with a .CPL suffix or a command input file (.COMI) . If you specify no options, the Batch monitor first searches for the CPL file named *pathname*. If no such file exists, the monitor looks in your working directory for the entryname part of *pathname* and assumes that it is a command input file.

*jobid*

Specifies the job number or the job name.

The job number is the 5-digit number preceded by a number sign (#) that is assigned to a job when it is placed in a queue. Because each job number is unique, you should use the job number rather than the job name to identify a particular job. The format of the job number is

**#qnnnn**

# is the number character as shown, *q* is the number of the queue and *nnnn* is the job's number within the queue. #10125 is an example of a job number. The queue is 1 and the queued file is 0125.

The job name is the name of the file being run. If the job was submitted as a pathname (for example, JOB PRISM>COLOR>BLUE), its job name is the final element of the pathname (BLUE, in this example). More than one active job can have the same job name if the same file was submitted more than once as a batch job. When managing a submitted job, you should use the job name only if you have one active job of that name.

## Options

For the user or programmer JOB has three groups of options. Each of these groups of options are described in the following sections:

<i>submit-options</i>	Define the parameters for the program you are submitting to the Batch subsystem.
<i>manage-options</i>	Action you wish to perform on jobs that you submitted to the Batch subsystem.
<i>monitor-options</i>	Request information about jobs in Batch queues.

### Submit Options

<b>-ACCOUNTING</b> <i>text</i>	Specifies accounting information for the job. <i>text</i> cannot be longer than 80 characters and cannot be an explicit register setting (octal number) or be preceded by a hyphen without single quotation marks. If <i>text</i> contains spaces, commas, or comment designators (/*), the text must be enclosed within single quotation marks. The information is included in job displays, but is not used in running the job.
<b>-ARGS</b> <i>cpl-arguments</i>	Passes CPL arguments to the job being processed. JOB does not read the CPL arguments but only passes them to the CPL file when execution of the file begins. <b>-ARGS</b> must be the last option on a command line because everything that follows (except comments) is assumed to be the CPL arguments being passed.
<b>-COMOUTPUT</b> <i>pathname</i>	Opens a command output file (also called a COMO file) with the specified pathname at the beginning of job execution. If you specify an entryname rather than a pathname, the COMO file is opened in the directory to which you are attached at the time of job submittal. If you specify the <b>-COMOUTPUT</b> option for a job that itself issues a <b>COMOUTPUT</b> command, the file opened for the <b>-COMOUTPUT</b> option is closed when the <b>COMOUTPUT</b> command internal to the job is executed; if the two files share the same name, the file opened for the <b>-COMOUTPUT</b> option is overwritten.
<b>-CPL</b>	Runs <i>pathname</i> as a CPL file, regardless of its name. <b>-CPL</b> is not needed if <i>pathname</i> has a .CPL suffix or if you use the <b>-ARGS</b> option.

**-CPTIME** { *seconds*  
NONE }

Specifies the maximum amount of CPU time (in seconds) allotted to the job. The job aborts if it exceeds the time limit. NONE places no time limit on the job. If the **-CPTIME** specification conflicts with the **-CPTIME** limit of a queue requested with the **-QUEUE** option, you are asked to specify a new limit. If you do not use the **-QUEUE** option and the specification conflicts with all queues, the error message

No queue available for job is displayed.

**-DEFER** *date.time*

Defers job execution until the date and time specified. At job submittal, the acknowledgement message gives the date and time of deferral. If the specified date and time have already passed, you receive a warning message, and the job becomes eligible for immediate execution. If the system goes down and the specified date and time have passed when the system comes back up, the job becomes eligible for immediate execution. If a job is deferred until a time that falls outside its queue's active window, the job is executed at the beginning of the first active window after the specified date and time.

Valid formats for *date.time* are:

*mo/dd/yy.hh:mm:ss*  
*yy-mo-dd.hh:mm:ss*  
'*dd mon yy.hh:mm:ss*'  
'*dd mon yy hh:mm:ss*'

The last two formats contain spaces and therefore must be enclosed within single quotation marks. *mon* stands for a month's first three characters — JAN, FEB, MAR, etc. Double letters represent one- or two-digit numbers. Leading zeros are not required. The *hh* field uses 24-hour notation. Omitted date fields are replaced by current date information; omitted time fields are replaced by zeros.

For more information about the format of *date*, see the section Wildcard Options in Chapter 4.

**-ETIME** { *minutes*  
NONE }

Specifies (in minutes) the elapsed time allowed before the job is aborted. Details are the same as for **-CPTIME**.

**-FUNIT** *number*

Specifies the file unit used for command input. (Default depends on queue parameters but is usually 6.) Cannot be used for CPL jobs.



- ▪ ▪ ▪ ▪ ▪ ▪ ▪ ▪ ▪
- HOME *pathname*** Specifies the directory in which a job executes. *pathname* cannot be either a null specification or a relative pathname (that is, the pathname cannot begin with \*>) and cannot exceed 80 characters in length. -HOME has the same effect as providing an ATTACH command as the first line of the command file.
- NOTIFY** Notifies you at job termination that your job has either completed or aborted. If you previously specified MESSAGE -DEFER, notification is deferred until you return to PRIMOS command level. If you previously specified MESSAGE -REJECT or are not logged in when the job terminates, you do not receive notification.
- NO\_COMOUTPUT** Cancels a request made with the -COMOUTPUT option for a COMO file. If no COMO file was requested, -NO\_COMOUTPUT has no effect. The -NO\_COMOUTPUT option usually is used with the -CHANGE option or to override a -COMOUTPUT option on the \$\$ command line.
- NO\_DEFER** If you deferred job execution with the -DEFER option, -NO\_DEFER makes the job eligible for immediate execution. If you did not request job deferral, -NO\_DEFER has no effect. The -NO\_DEFER option usually is used with the -CHANGE option or to override a -DEFER option on the \$\$ command line.
- NO\_NOTIFY** Does not notify you at job termination. If you did not request notification with the -NOTIFY option, -NO\_NOTIFY has no effect. The -NO\_NOTIFY option usually is used with the -CHANGE option or to override a -NOTIFY option on the \$\$ command line.
- PRIORITY *n*** Sets the job's priority within its queue. The value of *n* ranges from 0 to 9, inclusive, with 9 being the highest (most favored) priority. (Default is queue-dependent.)
- PROJECT *project\_id*** Runs a job under a project ID different from the project ID under which you logged in. A corollary to this feature is that you can issue the ORIGIN command within a batch job; the attach point is your Initial Attach Point in the project under which the job is running.
- QUEUE *queuename*** Places the job in a specific queue. (Use the BATGEN -DISPLAY command to obtain the queue names and characteristics.) (Default is the first queue available.) If a conflict results between -QUEUE and another option (for example, -CPTIME), you are asked to resubmit the job with different options.

**-RESTART** { YES }  
                  { NO }                   Determines if a job can be restarted after an abort or system shutdown. (Default is YES.)

### Manage Options

Use only one option at a time.

**-ABORT**                   Terminates the execution of a running job or cancels a waiting, deferred, or held job.

**-CANCEL**                   Prevents the execution of a waiting, deferred, or held job. Does not halt the execution of a running job but does mark it as not restartable.

**-CHANGE**                   Modifies the parameters of a waiting job. You may change the values of any of the following submit options:  
  [*submit-options*]

- |                   |                      |
|-------------------|----------------------|
| <b>-ACCT</b>      | <b>-HOME</b>         |
| <b>-ARGS</b>      | <b>-NOTIFY</b>       |
| <b>-COMOUTPUT</b> | <b>-NO_COMOUTPUT</b> |
| <b>-CPTIME</b>    | <b>-NO_DEFER</b>     |
| <b>-DEFER</b>     | <b>-NO_NOTIFY</b>    |
| <b>-ETIME</b>     | <b>-PROJECT</b>      |
| <b>-FUNIT</b>     | <b>-RESTART</b>      |

The possible new values for each option are the same as those specified above in the section Submitting a Job. **-CPL**, **-QUEUE**, and **-PRIORITY** cannot be changed.

**-RESTART**                   Aborts an executing job and then restarts the job if it is restartable. Normally used after a **JOB -CHANGE** command.

### Monitor Options

Use only one option at a time.

**-DISPLAY**                   Displays the same information about a job as the **-STATUS** option, plus the home directory of the job and the values (both user-specified and queue defaults) for all **JOB** and **\$\$** command options.

**-STATUS**                   Displays the job's number, its execution state (waiting, deferred, held, or executing), its name, and the name of its queue.

## Suboptions to Monitor Options

<b>ALL</b>	Displays information about all your jobs, regardless of state. Like the TODAY option, the display includes jobs that have been completed, cancelled, or aborted. By default only running, waiting, and deferred jobs are displayed.
<b>TODAY</b>	Displays information about all your jobs submitted during the current day, regardless of state. Like the ALL option, the display includes jobs that have been completed, cancelled, or aborted. By default only running, waiting, and deferred jobs are displayed.
<b>-QUEUE <i>queuename</i></b>	Displays your jobs in the specified queue.
<b>-USER <i>userid</i></b>	This option is available to unprivileged users, but unless you have priority access rights, only your own user ID is valid.

## Submitting a Job

Use the following format for submitting a batch job:

**JOB *pathname* [*submit-options*]**

The monitor submits the job by performing the following steps:

1. Assigns your login name to the job.
2. Places the job in an appropriate Batch queue.
3. Assigns the job a unique job number.
4. Adds the queue's default options and values.

---

### Note

If you submit a job from a password directory, you must use the **-HOME** option with the password included in the pathname of the home directory and you must enclose the pathname in single quotation marks. For example,

```
JOB REPORTS>PRODUCE_REP -HOME 'RPDB SECREP'
```

Otherwise, the job aborts because the batch phantom cannot attach to the directory.

---

## Using the \$\$ Command to Submit a Job

You can specify most of the above options within the command input file itself by using the \$\$ command as the first noncomment line of the file. The command format is as follows:

```
$$ JOB { userid
        * } [submit-options]
```

*userid* is the name under which you logged in. If you specify an asterisk (\*) instead of a user ID, any user can submit the file to Batch. Otherwise, only a user logged in as *userid* can submit it.

*submit-options* are any of the job submittal options listed above, except for -ARGS and -CPL.

The following example shows a command line using the \$\$ command:

```
$$ JOB JONES -HOME JONES>REPORTS -CPTIME NONE -ETIME NONE
```

Any job can specify options with the JOB command, the \$\$ command, or both. If an option is specified twice, the JOB option overrules the \$\$ option. For example, if \$\$ called for -CPTIME 20 and JOB called for -CPTIME 10, the job would be allowed only 10 seconds of CPU time before being aborted.

---

### Note

If *pathname* resides in or is executed from a password directory, you must use the -HOME option with the password included in the pathname and the pathname enclosed in single quotation marks:

```
JOB pathname -HOME 'directoryname password'
```

Otherwise, the job aborts because the Batch monitor cannot attach to the directory.

---

## Managing an Active Job

After the job is in a Batch queue, use the following JOB command format to control the execution or characteristics of the job:

```
JOB jobid manage-option
```

*jobid* is either the job number or the job name, as described above in the Arguments and Options section. *manage-option* is one of the options described above in the Manage Options section.

## Modifying a Job In a Queue

To change the characteristics of a previously submitted job, first use the JOB `-CHANGE` command and then the JOB `-RESTART` command, as in the following example:

```
OK, JOB BLUE -CHANGE -DEFER 20:00 -NO_COMOUTPUT -RESTART YES
OK, JOB BLUE -RESTART
```

In this example, the first command makes changes in the job named BLUE and the second command aborts the job and restarts it with its new characteristics.

---

### Note

Do not confuse the `-RESTART YES/NO` submit option with the `-RESTART` manage option. The first always takes YES or NO as an argument and determines whether the job can be restarted if it aborts. The second never takes an argument and actually aborts and restarts a changed job.

---

## Monitoring an Active Job

Use the format below to monitor one or all of your active jobs:

```
JOB [jobid] monitor-option { ALL
                             TODAY } { -QUEUE queue-name
                                         -USER userid }
```

*jobid* is either the job number or the job name, as described above in the section Managing an Active Job. *monitor-option* is one of the two options described above in the Monitor Options section. If you do not specify *jobid*, information is displayed on each of your active jobs.

You can select jobs from specific queues with the `-QUEUE` option, but unless you have priority access rights, you can only list your own jobs with the `-USER` option.

The following example illustrates the `-DISPLAY` option:

```
OK, JOB -DISPLAY
[JOB Rev. 23.3.0 Copyright (c) 1992, Prime Computer, Inc.]
Cpl job curly(#00029), user PANCHO deferred (queue normal-1).
Submitted today at 10:05:52 a.m., deferred until today at 8:00:00
p.m.
Priority=5, cpu limit=None, elapsed limit=None.
Project=FOOBAR, Notify=No, Como=<WWEST>PANCHO>TV>curly.como.
Home ufd=<WWEST>PANCHO>TV

Cpl job shemp(#10000), user PANCHO waiting (queue normal-2).
Submitted today at 9:03:14 a.m.
Priority=5, cpu limit=None, elapsed limit=None.
Project=DEFAULT, Notify=Yes.
Home ufd=<WWEST>PANCHO>TV

Cpl job moe(#30000), user PANCHO waiting (queue background-2).
Submitted today at 9:46:25 a.m.
Priority=5, cpu limit=None, elapsed limit=None.
Project=DEFAULT, Notify=Yes.
Home ufd=<WWEST>PANCHO>TV
OK,
```

The first line of the `-DISPLAY` option's output shows the job's name, number, user ID, execution state (waiting, deferred, held, or executing), and the name of its queue.

The second line shows the time of job submittal, the time of job initiation (if the job is currently executing), and the time of job deferral (if the `-DEFER` option was previously specified).

The third line shows the job's file unit number (unless the job is a CPL program), its priority within its queue, its CPU time limit, and its elapsed time limit.

The fourth line shows the job's default project ID or the project ID specified with the `-PROJECT` option, its notification status ("Yes" if the `-NOTIFY` option was specified; "No" otherwise), and the pathname of its command output file if the `-COMOUTPUT` option was specified.

The last line shows the job's home directory.

The following example illustrates the `-STATUS` option:

```
OK, JOB -STATUS
[JOB Rev. 23.3.0 Copyright (c) 1992, Prime Computer, Inc.]

Job status listing for user PANCHO:
```

Jobid#	State	External name	Queue
#00029	deferred	curly	normal-1
#10000	waiting	shemp	normal-2
#30000	waiting	moe	background-2

OK,

Using the ALL option with -STATUS lists all your jobs, including jobs that were completed at an earlier date:

```
OK, JOB -STATUS ALL
[JOB Rev. 23.3.0 Copyright (c) 1992, Prime Computer, Inc.]
```

Job status listing for user PANCHO:

Jobid#	State	External name	Queue
#00029	deferred	curly	normal-1
#11563	completed	larry	
#11615	completed	curly2	
#10000	waiting	shemp	normal-2
#30000	waiting	moe	background-2
#11615	cancelled	curly3	

OK,

For more information on the JOB command and batch jobs, see the *PRIMOS User's Guide*.

## KBUILD

KBUILD builds MIDASPLUS files.

### Format

KBUILD

### Usage

KBUILD is an interactive subsystem that builds MIDASPLUS data and index files from sequential disk files. When invoked, KBUILD asks you a series of questions about the file to be built.

KBUILD can process the following five types of files:

- ASCII text files (compressed)
- Binary files written from FORTRAN programs
- Binary files written by PRW\$\$
- COBOL-written text files (uncompressed)
- RPG-written files (uncompressed)

KBUILD is the easiest method of adding data to a new or existing MIDASPLUS file. For information on KBUILD and other methods of building MIDASPLUS data and index subfiles, see the *MIDASPLUS User's Guide*.

.....

KIDDEL

KIDDEL

KIDDEL deletes or cleans out MIDASPLUS files.

### **Format**

KIDDEL

### **Usage**

KIDDEL is an interactive MIDASPLUS utility that is used primarily to remove entries from the various parts of a MIDASPLUS file while retaining the template of the file for further use. KIDDEL can also delete the template entirely.

The KIDDEL options allow you to perform the following functions:

- Clean out entries in one or more secondary indexes.
- Clean out entries in all index and data subfiles.
- Delete an entire MIDASPLUS file.
- Delete one or more secondary index subfiles.

For detailed information on KIDDEL, see the *MIDASPLUS User's Guide*.



## LAB

LAB displays buffer sizes for asynchronous lines. It displays the current and initial buffer sizes for the user's own line or any lines that are assigned to the user.

For the network user, LAB displays buffer sizes for NTS lines and PRIMENET Remote login lines, as well as for terminal lines.

Use of the LAB command may be restricted by the DSM Administrator.

### Format

$$\text{LAB} \left\{ \begin{array}{l} \text{-LINE } n \text{ [-TO } m \text{ [-EXCEPT } num1 \text{ [... } num10 \text{ ]]}] \\ \left[ \begin{array}{l} \text{-NTSABF} \\ \text{-NTSBUF} \\ \text{-REMBUF} \end{array} \right] \\ \text{-HELP} \end{array} \right\} \left[ \begin{array}{l} \text{-ACTIVE} \\ \text{-ALL} \\ \text{-NO\_HEADER} \\ \text{-NO\_WAIT} \\ \text{-ON } node\text{-name} \end{array} \right]$$

### Options

- ALL** Requests information on the configuration of your lines on the system. Do not specify **-ALL** with the **-ACTIVE** option.
- ACTIVE** Requests information on all lines currently assigned to you. Do not specify **-ACTIVE** with the **-LINE** or **-ALL** options.
- LINE *n*** Requests information on a single line, where *n* is the required decimal line number, or when used with the **-TO** option, the first line number in a series. Valid line numbers range from 0 to 512 for direct-connect and from 1024 to 1536 for NTS lines.
- TO *m*** Requests information on a range of consecutively numbered lines, where *m* specifies the last number in a series beginning at the line number *n* given in **-LINE**. The value of *m* ranges from 1 to 511 and must be greater than *n*. You can use **-TO** with the **-EXCEPT** option, if desired.
- EXCEPT *num1* . . . *num10*** Excludes up to ten lines from a range specified with the **-TO *m*** option, where *num* is a valid decimal line number less than *m*. Separate each *num* with a single blank space in the command line.

LAB

- ON *node-name* Allows a privileged DSM user to display the buffer sizes on a remote system, where *node-name* is the node name of the remote system. If the node name is omitted, the LAB command is executed on the local system.
- NO\_HEADER Suppresses the output header.
- NO\_WAIT Suppresses the --More-- prompt and does not pause after every 23 lines of output. Output scrolls continuously.
- HELP Displays the format of the command and a list of valid command line options. When you select -HELP all other options are ignored.

**LAB Example**

The following example shows the use of the LAB -ALL option:

```
OK, LAB -ALL
[LAB Rev. 23.3.0 Copyright (c) 1992, Prime Computer, Inc.]

Node: NODE 1
-----
+
|          |          Current Buffers          ||          Default/Initial Buffers
|
|-----+-----+-----+-----+-----+-----+-----+-----|
| Line | Input | Output | DMQ || Input | Output | DMQ
|-----+-----+-----+-----+-----+-----+-----+-----|
|  REM | 2048, | 2048,  | **** || 2048, | 2048,  | ****
|-----+-----+-----+-----+-----+-----+-----+-----|
+

```

For more information see *System Administrator's Guide, Volume II: Communication Lines and Controllers*.

See also CAB.

## LABEL

LABEL reads or writes volume 1 (VOL1) level labels on magnetic tapes in any of the following formats:

- ANSI X3.27-1978 or X3.27-1987 standard 9-track ASCII
- IBM® 9-track EBCDIC or IBM 7-track BCD
- Prime format (nonstandard Level 1 volume labels followed by a dummy HDR1 label and two file markers)

Before using LABEL, you must assign a tape unit with the ASSIGN command and mount the correct reel on the tape drive.

### Format

```
LABEL MTn [
  -ACCESS char
  -INIT
  -OVERWRITE
  -OWNER owner
  -PARITY {EVEN
           ODD}
  -TYPE type
  -VOLUME volume-id
  -HELP
]
```

### Argument

**MTn** Specifies the tape drive on which the tape to be labeled is mounted. *n*, an integer in the range 0 through 7 inclusive, is the tape drive's number. This argument must be present and must be the first option on the command line. For example, magnetic tape unit 0 is MT0. (Note that there is no space between MT and 0.)

### Options

If you use LABEL without the -VOLUME option on a tape which is already labeled, it will read the existing label. (To read a BCD or EBCDIC label, you must specify the appropriate -TYPE option.) If you want to relabel a previously labeled tape, you must use the -INIT option.

**-ACCESS char** A single character which defines access rights to this tape. This character is ignored for Prime and IBM format labels. On ANSI labels, this character is copied as the access if present; otherwise, the ACCESS field is left blank.

.....  
LAC

- INIT** Tells LABEL that this tape is being written for the first time. This option must be used on unformatted tapes or on tapes whose labels should be overwritten.
- OVERWRITE** Tells LABEL to overwrite a BRMS tape. If you try to overwrite an ARCHIVE, BACKUP, or TRANSPORT BRMS tape, you must use the **-OVERWRITE** option.
- OWNER *owner*** Identifies the owner of the tape. *owner* is a string which contains, for ANSI labels, 1 through 14 characters; for IBM labels, 1 through 10 characters. If you specify a label which is shorter than the allowed maximum length, it is blank-padded on the right to the maximum length. If you omit *owner*, LABEL uses your login name as the default value.
- PARITY** { **EVEN**  
          **ODD** } Specifies EVEN or ODD parity for the label. This option may be used only with the **-TYPE B** option.
- TYPE *type*** Specifies what sort of label you want written. The legal types are shown below:
- |                   |   |
|-------------------|---|
| <b>ANSI87</b>     | ANSI X3.27-1987 standard label.   |
| <b>BCD</b>        | IBM label for 7-track BCD tapes.  |
| <b>EBCDIC</b>     | IBM label for 9-track EBCDIC tapes.   |
| <b>PRIME</b>      | PRIME ASCII label. This is the default. ANSI and A are synonyms for label type PRIME. |
| <b>STANDARD_1</b> | ANSI X3.27-1978 standard label.   |
- VOLUME *volume-id*** The volume number which uniquely identifies this tape reel. *volume-id* must be from one through six characters long; if it is shorter than six characters, it is blank-padded on the right to make six characters. **-VOLSER** and **-VOLID** are synonyms for **-VOLUME**. If this option is not present, LABEL attempts to read an existing label from the tape; if this option is present, LABEL writes a new label to the tape.

**-HELP** Prints a list of the command's options.

For a complete discussion of LABEL and tape labels, see the *Data Backup and Recovery Guide*.

LAC

See LIST\_ACCESS.

## LATE

LATE defers execution of the next command until a specified time.

### **Format**

LATE

### **Usage**

After you issue LATE, you are asked to enter the time at which the next command is to be executed. The prompt is as follows:

Time of day (HHMM) to execute next command:

Enter the time in the format HHMM, where HH is the hour (00 through 23) and MM is the minute (00 through 59). LATE responds to this input with the following message:

Next command will be executed at HH:MM

Respond to this prompt by entering a command. The command is executed at the time you designated at the first prompt.

If you specify a time that is earlier than the current time, execution of the command is deferred until the following day. For example, if LATE is issued at 12:00 noon, the following is displayed:

Time of day (HHMM) to execute next command: 0230

Next command will be executed at 02:30 tomorrow.

If you include the colon, the first two digits designate hours and the last two digits designate minutes. If you omit the colon, the last two digits are considered minutes. For example, 30 is interpreted as 00 hours and 30 minutes. If no time specification is made, the default is 00:00 (midnight).

After you issue LATE and the next command, you cannot use your terminal until the specified time, unless you use Ctrl-P to abort from LATE. LATE is particularly useful for running batch files during periods when the system load is light.

.....

LCB

LCB                    See LIST\_CONTIGUOUS\_BLOCKS.

LD

LD lists the contents of a directory and, optionally, the attributes of the entries.

### Format

LD [*pathname*] [*wild1* ... *wild15*] [*options*]

### Arguments

*pathname*                    Identifies the directory to be listed and contains the first wildcard name. The default pathname is your current directory and the default wildcard name is @@.

*wild1* ... *wild15*            The default @@ wildcard name can be replaced with a total of 15 wildcard names.

If you do not specify *pathname*, all entries in your current directory are listed because the LD command without arguments or options is equivalent to the command format

LD @@

A directory entry is listed if it matches the entry that is part of the pathname and/or one of the wildcard names. For example, the following command lists entries in the current directory whose names begin with SALE or end with .MEMO:

```
OK, LD SALE@@ @.MEMO
```

### Options

The LD options in this section are organized by function as follows:

- Listing by file system object type
- Listing by date and time
- Listing by access control permissions
- Listing by attribute
- Setting display characteristics
- Sorting

## Listing by file system object type

You may use more than one selection option at a time. Object types are always listed in the same order, regardless of where options appear on the command line, as follows:

Files { -SAM  
-DAM  
-CAM

Directories  
Segment Directories  
Access Categories

File type selection options are ineffective on disks on pre-Rev. 23.3.0 systems. For example, if you use the -SAM option on a pre-Rev. 23.3.0 system's disk, all files are displayed by type.

- ACCESS\_CATEGORY**      Selects access categories. When used in combination with other file system object type options, such as -SAM and -DIR, access categories are always listed last.
- 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.
- CAM**                    Selects CAM type files.
- DAM**                    Selects DAM type files.
- DIRECTORY**            Selects regular directories.
- FILE**                    Selects file system objects that are files. When used in combination with other file type options, files are listed first.
- SAM**                    Selects SAM type files.
- SEGMENT\_DIRECTORY**    Selects file system objects that are segment directories. When used in combination with other file type options, the default order of display prevails.

## Listing by date and time

These options require a *date.time* argument. The format may be any of the following:

```
yy-mm-dd.hh:mm:ss
mm/dd/yy.hh:mm:ss
'dd mon yy hh:mm:ss'
'dd mon yy.hh:mm:ss'
```

The last two formats must be enclosed within single quotation marks as shown. *mon* stands for a month's first three characters — JAN, FEB, MAR, etc. All the other letters represent one- or two-digit numbers. The *hh* field uses 24-hour notation. Omitted date fields are replaced by current date information; omitted time fields are replaced by zeros. Leading zeros are not required.

<b>-ACCESSED_AFTER</b> <i>date.time</i>	Lists only selected entries that were accessed after <i>date.time</i> .
<b>-ACCESSED_BEFORE</b> <i>date.time</i>	Lists only selected entries that were accessed before <i>date.time</i> .
<b>-AFTER</b> <i>date.time</i>	Lists only selected entries that were modified after <i>date.time</i> . Same functionality as the <b>-MODIFIED_AFTER</b> option.
<b>-BACKEDUP_AFTER</b> <i>date.time</i>	Lists only selected entries that were backed up after <i>date.time</i> .
<b>-BACKEDUP_BEFORE</b> <i>date.time</i>	Lists only selected entries that were backed up before <i>date.time</i> .
<b>-BEFORE</b> <i>date.time</i>	Lists only selected entries that were modified before <i>date.time</i> . Same functionality as the <b>-MODIFIED_BEFORE</b> option.
<b>-CREATED_AFTER</b> <i>date.time</i>	Lists only selected entries that were created after <i>date.time</i> .
<b>-CREATED_BEFORE</b> <i>date.time</i>	Lists only selected entries that were created before <i>date.time</i> .
<b>-MODIFIED_AFTER</b> <i>date.time</i>	Lists only selected entries that were modified after <i>date.time</i> .
<b>-MODIFIED_BEFORE</b> <i>date.time</i>	Lists only selected entries that were modified before <i>date.time</i> .



### Listing by access control permissions

- CATEGORY\_PROTECTED** Lists only the entries that are protected by the access category identified by *category-name*. If you do not specify *category-name*, all entries protected by any access categories are listed.  
[*category-name*]
- DEFAULT\_PROTECTED** Lists only the entries protected by default access control.
- SPECIFIC\_PROTECTED** Lists only the entries protected by specific ACLs.

### Listing by attribute

- DTA** Lists each entry's type and the date and time the entry was last accessed. If an entry has an undefined DTA value, displays **\*\* dta not set \*\***.
- DTB** Lists each entry's type and the date and time the entry was last backed up. If an entry has not been backed up, displays **\*\* dtb not set \*\***.
- DTC** Lists each entry's type and the date and time the entry was created. If an entry has an undefined DTC value, displays **\*\* dtc not set \*\***.
- DTM** Lists each entry's type and the date and time the entry was last modified.
- 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 can not be used with the **-SIZE** or the **-TOTAL** options.
- PROTECT** Lists the protection information for each entry. Protection information includes the access rights, the delete-protection, and the type of protection.
- RBF** Lists only Recovery Based Files (ROAM files).
- RWLOCK** Displays the read and write locks on files and segdirs in the current directory.
- SIZE** Lists the size of each entry and the quota for directories only. Displays a size of **-1** for entries to which you do not have Read (R) access and List (L) access.

- TOTAL Displays the total number of records of user data used by the listed items and lists the names of files and other objects in the directory. Totals displayed by LD -TOTAL do not include record overhead.
- 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. 23.3.0 systems, all selected file system objects will be listed, but only those files that were truncated will have trunc displayed in the trunc column.

**Setting display characteristics**

- BRIEF Limits the top header to a one-line format. Also eliminates the column labels for options (such as -DETAIL or -DTM) that normally produce them.
- DETAIL Lists all attributes for each entry you select. (Default lists only the entrynames.) The attributes are displayed on three lines following the entryname. See Example 5 below for an explanation of the output contents.
- DISPLAY\_MATCHES Suppresses directory header information for directories that do not match a supplied wildcard (@@). Using the -DISPLAY\_MATCHES option with no argument implies the use of the @@ wildcard.
- 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.
- HELP Displays LD's syntax and options. The -HELP usage list is also displayed if PRIMOS detects an error while parsing the command.
- NO\_COLUMN\_HEADERS Suppresses column labels.
- NO\_HEADER Suppresses the top header line and entry type headers. Also suppresses the type totals if specified with any sorting option. This option is most useful when combined with the -SINGLE\_COLUMN option.
- NO\_WAIT Suppresses the --More-- prompt and does not pause after every page of output. Output scrolls continuously.
- SINGLE\_COLUMN Lists each entry on a separate line. Use this option only with the default output format in which only names are listed.

**-WIDE** Lists output in lines wider than the 79-character line default. Uses 100-character lines for name-only output; otherwise, assumes name and all attributes fit on one line. This option is useful for output to printers that can handle wide paper.

**-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.

### Sorting

**-NO\_SORT** Does not sort entries by name or by type and does not display type headers. This option takes precedence over any other sorting option specified on the command line.

**-REVERSE** Reverses the sort order from its default, but does not affect the sort order of entry types.

**-SORT\_DTA** Sorts entries within their type by descending date/time last accessed information and lists the date and time each entry was last accessed. If you try to sort by DTA in a pre-Rev. 20.0 directory, no sorting occurs, and each object returns `** dta not set **`. When only some of the objects you are sorting have undefined DTA values, these objects appear at the end of the list. Cannot be used with any other sorting option.

**-SORT\_DTB** Sorts entries within their type by descending date/time backed-up information, and lists the date and time each entry was last backed up. If an entry has not been backed up, displays `** dtb not set **`. Cannot be used with any other sorting option.

**-SORT\_DTC** Sorts entries within their type by descending date/time created information and lists the date and time each entry was created. If you try to sort by DTC in a pre-Rev. 20.0 directory, no sorting occurs, and each object returns `** dtc not set **`. When only some of the objects you are sorting have undefined DTC values, these objects appear at the end of the list. Cannot be used with any other sorting option.

-SORT_DTM	Sorts entries within their type by descending date/time modified information, and lists the date and time each entry was last modified. Cannot be used with any other sorting option.
-SORT_NAME	Sorts entries alphabetically by name only, not within their type. Does not display type headers. Cannot be used with any other sorting option.
-SORT_SIZE	Sorts entries by descending size within each type, and lists sizes of entries (except for access categories) and quotas of directory entries. Cannot be used with any other sorting option.

---

**Note**

Using the `-DETAIL` option or the `-SIZE` option on an entry in a Rev. 20.0 or later directory updates the parent directory's DTA, not the entry's own DTA. However, using either the `-DETAIL` or the `-SIZE` option from a pre-Rev. 20.0 system on an entry in a Rev. 20.0 or later directory updates the DTA values of both the object and its parent directory.

---

You can also use any wildcard option (except for `-VERIFY` and `-NO_VERIFY`) with LD, even if you do not specify a wildcard name. For example, `LD -FILE` lists only the files in your current directory and `LD -DIR` lists only the directories. The wildcard options can be used in conjunction with the LD options (for example, `LD -FILE -SORT_SIZE`). See Chapter 4 of this guide for details on wildcard names and wildcard options.

### **LD Output**

By default, the LD command displays a header line followed by the directory's contents. The header line lists the following items:

- The directory's pathname
- Your access rights if you are in an ACL directory or your status (OWNER or NONOWNER) if you are in a password directory
- The number of records used by the directory and its files
- The quota for the directory
- At PRIMOS Revision 22.0, the word **robust**, if you are listing a directory on a robust partition. A **robust partition** is a new partition type, introduced at PRIMOS Revision 22.0. See the *Operator's Guide to File System Maintenance* for details.

The directory entries are grouped in the following order: file, segment directory, directory, access category. Each entry type has a header that identifies it and reports the number of entries of that type. The entries of each type are listed four across, sorted alphabetically.

## Brief Output

An example of an LD output is shown below:

```
OK, LD

<PUBS>CARVER>PROJECT (ALL access) Robust
45 records in this directory, 47 total records out of quota of 0.

11 Files.

BOOK_SCHED      LOAD          LOGIN.ABBREV    LOGIN.CPL
LOGIN.LAST      NETENX.CPL    NETLINK.CPL     PROG.BIN
PROG.PASCAL     TOLL.BIN      TOLL.CBL

2 Segment Directories.

PROG.SEG        TOLL.SEG

2 Directories.

MAIL            PERSONAL

1 Access category.

GUARD.ACAT
OK,
```

Use one or more of the LD options to produce a different output.

Unless the `-NO_WAIT` option is used, the `--more--` prompt is displayed after 23 lines or the number of lines specified with the `-HEIGHT` option.

To abort the output, enter Q, QUIT, N, or NO in uppercase or lowercase letters. To continue the listing, press RETURN or enter any other characters.

## Detailed Output

When you use the `-DETAIL` option, detailed information about file system objects is displayed. An example of the detailed listing is shown below, followed by a description of each entry.

OK, FD -DETAIL

<PUBS3>EDWARD (ALL access)

46 records in this directory, 48 total records out of quota of 0.

8 Files.

name	date/time modified	date/time backedup	date/time created
	date/time accessed	size rwlock	dump access delprot
	type rbf trunc	protected by	
LOGIN.ABBREV	17 Sep 91 11:52:56	** dtb not set **	16 Dec 90 17:51:00
	18 Sep 91 10:15:56	2 sys nodmp	ALL
	sam	(Default ACL)	
LOGIN.CPL	16 Jun 90 16:37:32	** dtb not set **	16 Dec 90 17:51:00
	18 Sep 91 10:15:56	1 sys nodmp	ALL
	sam	(Default ACL)	
TEST1	17 Sep 91 11:50:44	** dtb not set **	17 Sep 91 11:50:44
	18 Sep 91 10:19:20	7 sys nodmp	ALL
	dam	(Default ACL)	

2 Directories.

name	date/time modified	date/time backedup	date/time created
	date/time accessed	size quota	dump access delprot
	type rbf trunc	protected by	
OUTPUT	18 Sep 91 10:17:16	** dtb not set **	18 Sep 91 10:17:16
	18 Sep 91 10:17:16	1 0 nodmp	ALL
	dir	(Default ACL)	

1 Access category.

name	date/time modified	date/time backedup	date/time created
	date/time accessed	access type rbf	
NO.TOUCH.ACAT	18 Sep 91 10:18:24	** dtb not set **	18 Sep 91 10:18:24
	18 Sep 91 10:18:24	ALL acat	

OK,

The meaning of each item in the entry is as follows (listed alphabetically):

access	Access rights available to you; corresponding protection keys are listed for password-protected entries.
date/time accessed	Date and time entry was last accessed. Displays ** dtb not set ** for entries with undefined DTA values, for example, entries in pre-Rev. 20.0 directories.
date/time backedup	When entry was backed up to tape by the BACKUP command. Displays ** dtb not set ** if the entry has not yet been backed up or if BACKUP is not used on your system.
date/time created	When entry was created. Displays ** dtc not set ** for entries with undefined DTC values, for example, entries in pre-Rev. 20.0 directories.
date/time modified	When entry was last modified.

<b>delprot</b>	Delete-protection switch (dprot indicates the delete-protection switch was set with the SET_DELETE command; for files, directories, and segment directories only).
<b>dump</b>	Dump switch indicator (dmp indicates entry has been dumped, and nodmp indicates entry has not been dumped; for files, directories, and segment directories only).
<b>name</b>	Name of entry.
<b>protected by</b>	Name of access category protecting the entry (Specific ACL indicates protection by a specific ACL, and Default ACL indicates default protection; for files, directories, and segment directories only).
<b>rbf</b>	RBF means the file is a Recovery Based File, and a blank means it is not.
<b>rwlock</b>	Read/write concurrency lock setting, which specifies the number of users who can access the entry at the same time (files and segment directories only). The setting is one of the following: SYS (for system default); EXCL (for any number of readers or one writer); UPDT (for any number of readers and one writer); or NONE (for any number of readers and writers).
<b>size</b>	Size of entry in physical disk records (files, directories, and segment directories only). A size of -1 is shown for open COMO files and for entries to which you do not have Read (R) and List (L) access if the read/write concurrency lock setting is either SYS or EXCL; the actual record size is shown for open COMO files and for entries to which you do not have Read (R) and List (L) access if the read/write concurrency lock setting is either UPDT or NONE.  For directories this location lists the quota of the directory in physical disk records instead of the rwlock setting.
<b>trunc</b>	Truncate indicator (trunc means the entry was truncated by a FIX_DISK operation, and a blank means it was not; for files only).
<b>type</b>	Type of entry (sam, dam, or cam for files; sseg or dseg for segment directories; dir for directories; acat for access categories).

### Interrupting the LD Command

If you use LD when you are attached to a remote partition and the network experiences a momentary problem, PRIMOS may lose your attach point. In this case, the output of the LD command is as follows:

```
OK, LD
Virtual circuit was cleared. (current_directory) (ld)
ER!
```

To correct this problem, use the ATTACH command to reattach to the original partition, or to another partition. (Use the ICE command if the problem is still not corrected.)

Similarly, when you issue LD to list a directory to which you are not currently attached, PRIMOS temporarily attaches you to the directory that you are listing. If you allow LD to complete without interruption, your attach point is returned to the directory you were in when you issued the LD command (your current attach point). If you issue Ctrl-P (QUIT) after issuing LD, but before LD completes, the LD command may be interrupted before PRIMOS has a chance to return you to your current attach point. Type S to allow the LD command to finish and to return to your current attach point. Otherwise you may have to reset your attach point yourself.

### LD Examples

The following series of examples illustrate the LD command.

**Example 1:** Listing the date and time each object in the current directory was last accessed:

```
OK, LD -DIA

<PUBS3>EDWARD (ALL access)
46 records in this directory, 48 total records out of quota of 0.

5 Files.
name                date/time accessed  type  rbf
-----
LOGIN.ABBREV        18 Sep 91 10:15:56  sam
LOGIN.CPL           18 Sep 91 10:15:56  sam
TEST1               18 Sep 91 10:19:20  dam
TEST2               17 Sep 91 12:03:04  cam
WHERE.CPL           17 Sep 91 11:43:44  sam

1 Directories.
name                date/time accessed  type  rbf
-----
OUTPUT              18 Sep 91 10:17:16  dir

1 Access category.
name                date/time accessed  type  rbf
-----
NO.TOUCH.ACAT      18 Sep 91 10:18:24  acat

OK,
```



**Example 2:** Listing the date and time each object beginning with TEST (using wildcard tokens) was created:

OK, LD TEST?? -DTC

<PUBS3>EDWARD (ALL access) 46 records in this directory, 48 total records out of quota of 0.

5 Files.

name	date/time created	type	rbf
TEST1	17 Sep 91 11:50:44	dam	
TEST2	17 Sep 91 12:03:04	cam	
TEST3	17 Sep 91 12:05:12	sam	
TEST4	17 Sep 91 12:05:20	dam	
TEST5	18 Sep 91 10:19:16	cam	

OK,

**Example 3:** Sorting all objects in the current directory by the date and time each was last accessed:

OK, LD -SORT\_DTA

<PUBS3>EDWARD (ALL access) 46 records in this directory, 48 total records out of quota of 0.

5 Files.

name	date/time accessed	type	rbf
TEST1	18 Sep 91 10:19:20	dam	
LOGIN.CPL	18 Sep 91 10:15:56	sam	
LOGIN.ABBREV	18 Sep 91 10:15:56	sam	
TEST2	17 Sep 91 12:03:04	cam	
WHERE.CPL	17 Sep 91 11:43:44	sam	

2 Directories.

name	date/time accessed	type	rbf
OUTPUT	18 Sep 91 10:17:16	dir	

1 Access category.

name	date/time accessed	type	rbf
NO.TOUCH.ACAT	18 Sep 91 10:18:24	acat	

OK,

LD

**Example 4:** Sorting all objects beginning with TEST (and wildcard tokens) by the date and time each was created:

```
OK, LD TEST?? -SORT_DTC
```

```
<PUBS3>EDWARD (ALL access)
```

```
46 records in this directory, 48 total records out of quota of 0.
```

```
5 Files.
```

name	date/time created	type	rbf
TEST5	18 Sep 91 10:19:16	cam	
TEST4	17 Sep 91 12:05:20	dam	
TEST3	17 Sep 91 12:05:12	sam	
TEST2	17 Sep 91 12:03:04	cam	
TEST1	17 Sep 91 11:50:44	dam	

```
OK,
```

**Example 5:** Listing all objects in a user directory with the -RWLOCK option:

```
OK, LD -RWLOCK
```

```
<USERS>MORROW (ALL access)
```

```
133 records in this directory, 236 total records out of quota of 0.
```

```
5 Files.
```

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

```
3 Segment Directories.
```

name	rwlock	type	rbf
X.SEG	none	sseg	
Y.SEG	updt	sseg	
Z.SEG	sys	sseg	

```
1 Directories.
```

name	type	rbf
X.DIR	dir	

```
1 Access categories.
```

name	type	rbf
X.ACAT	acat	

**Example 6:** Listing all objects in a user directory created after September 17, 1991 at 2:00 pm, first without a sort option, and then sorted by date of creation:

OK, LD -CRA 9/17/91.14:00

<PUBS3>CARVER>PROJECT (ALL access)

45 records in this directory, 47 total records out of quota of 0.

8 Files.

BOOK_SCHED	LOGIN.ABBREV	LOGIN.CPL
LOGIN.LAST	NETENX.CPL	PROG.COMO

2 Directories.

MAIL                      PERSONAL

1 Access category.

GUARD.ACAT  
OK,

OK, LD -CRA 9/17/91.14:00 -SORT\_DATE

<PUBS3>CARVER>PROJECT (ALL access)

45 records in this directory, 47 total records out of quota of 0.

6 Files.

name	date/time created	type	rbf
BOOK_SCHED	17 Sep 91 16:19:20	sam	
LOGIN.CPL	18 Sep 91 10:15:56	sam	
LOGIN.ABBREV	18 Sep 91 10:15:56	sam	
PROG.COMO	21 Sep 91 11:43:44	dam	
NETLINK.CPL	21 Sep 91 12:03:04	cam	
LOGIN.LAST	30 Oct 91 07:05:20	sam	

2 Directories.

name	date/time created	type	rbf
PERSONAL	18 Sep 91 10:17:08	dir	
MAIL	18 Sep 91 10:17:16	dir	

1 Access category.

name	date/time created	type	rbf
GUARD.ACAT	18 Sep 91 10:18:24	acat	

OK,

LE

**Example 7: Using the -TRUNCATED option to list files truncated by FIX\_DISK:**

OK, LD -TRUNC

<USERCOM>ALBEE (ALL access)

46 records in this directory, 48 total records out of quota of 0.

1 File.

name	type	rbf	trunc
REPORT.COMO	dam		trunc

If the disk partition EXPORT is on a system running a PRIMOS revision before 23.3, the output for LD -TRUNC is as follows:

OK, LD -TRUNC

<EXPORT>EDWARD (ALL access)

46 records in this directory, 48 total records out of quota of 0.

5 Files.

name	type	rbf	trunc
LOGIN.ABBREV	sam		
LOGIN.CPL	sam		
TEST1	dam		trunc
TEST2	cam		
WHERE.CPL	sam		

OK,

The file TEST1 was truncated by FIX\_DISK. All the others were not.

LE

See LIST\_EPF.

LEM

The LEM command lists the extent map of a Contiguous Access Method (CAM) subfile on either Prime standard partitions or robust partitions.

**Format**

LEM { *filename*  
           *segdir>subfile-number* } [-NO\_WAIT]

**Arguments and Option**

- filename*                         Specifies the name of the CAM data file.
- segdir>subfile-number*         Specifies the segment directory pathname and subfile-number of the desired CAM segment directory subfile.
- NO\_WAIT                           Suppresses the --More-- prompt and does not pause after every 23 lines. Output scrolls continuously.

The extent map occupies one disk record and is the first record in a CAM file. It serves as an index to all other records in the file. The extent map records the number of extents in the file, the length and location of each extent, the extent length, and the physical and logical end of file. In a CAM file, the physical size is the number of disk records allocated to the file and the logical size is the number of disk records that contain data.

**Examples**

The following example shows the extent map for subfile Days1. Note that the term *record* (or *disk record*) represents 2048 bytes.

```
OK, LEM DAYS1
[LEM Rev. 23.3.0 Copyright (c) 1992, Prime Computer, Inc.]

Pathname: <SYS3>BSB>DAYS1.
Logical end of file:           1 (records)  443 (half-words).
Physical end of file:         1 (records).
Allocation size:              0 (records). (Using partition defaults)
Composed of 1 extents.
```

Extent	Address (octal)	Length (dec)
*****	*****	*****
1	52544	1

OK,

The following example shows the extent map for subfile 2.

```
OK, LEM MASTER.CAM.DBMS>2
[LEM Rev. 23.3.0 Copyright (c) 1992, Prime Computer, Inc.]

Pathname: <SYS1>JOAN.Q>MASTER.CAM.DBMS>2.
Logical end of file:      113 (records)  115712 (half-words).
Physical end of file:    118 (records).
Allocation size:         16 (records).
Composed of 3 extents.
```

Extent	Address (octal)	Length (dec)
*****	*****	*****
1	537	50
2	604	52
3	726	16

After displaying the file's pathname, the above output shows the extent map's header followed by information about each extent. A description of each line in the output follows.

The four lines of the header are described below.

<i>Line</i>	<i>Meaning</i>
Logical end of file	Offset of (distance to) the logical end of file in records and half-words (16 bits). In SAM and DAM files the logical end of file record equals the last physical record in the file; the logical end of a CAM file refers to where the data ends and not to empty disk records allocated to the file.
Physical end of file	Offset of (distance to) the physical end of file in records, including all disk records allocated to the file, whether used or unused.
Allocation size	The allocation size value is used by PRIMOS when additional space is required for a CAM file. When the allocation size is 0 (first example), PRIMOS uses the default disk partition minimum. If a specific allocation size is specified, PRIMOS allocates additional space equal to that value (second example).
Composed of <i>n</i> extents	Number of extents occupied by this file or subfile. Use the LIST_CONTIGUOUS_BLOCKS command to display the minimum and maximum extent sizes.

The second half of the display shows information about each extent. For each extent, this display contains the three columns described below.

<i>Column</i>	<i>Meaning</i>
Extent	Extent number. This number is not physically recorded in the table; in displaying the table's contents, the system automatically assigns a number for each extent entry according to its position in the table.
Address	The address of the first block in the extent. This address is an octal number.
Length	The number of blocks in the extent. This is a decimal number.

See also LIST\_CONTIGUOUS\_BLOCKS later in this chapter.

For more information on CAM files, see the *Operator's Guide to File System Maintenance*.

LG

See LIST\_GROUP.

LISTING

LISTING opens a file for writing on File Unit 2.

**Format**

LISTING *pathname*

**Usage**

A file opened by LISTING is usually an output file for a compiler or assembler. All subsequent compilation and assembly listings are written to this file until it is closed. *pathname* is the name of the listing file. If *pathname* is a simple filename, the file must be in your current directory. LISTING has the same effect as the following command:

OPEN *pathname* 2 2

.....  
LIST\_ACCESS

## LIST\_ACCESS

LIST\_ACCESS lists the access rights for any file system object.

### Format

LIST\_ACCESS [*pathname*]

### Argument

*pathname* Specifies the file, directory, segment directory, or access category whose rights you want to list.

### Access Rights

Table 2-1 lists the access rights you can use to make up the Access Control Lists (ACLs) for a file or directory. The symbols for the access rights are displayed in the left column.

Table 2-1. ACL Access Rights

Symbol	Right	Applies To	Meaning
R	Read	Files	File can be read or executed.
W	Write	Files	File can be modified.
X	Execute	Local EPF runfiles (no effect on remote EPF files or registered EPFs)	Executable Program Format (EPF) file can be executed, but cannot be copied with the standard file system utilities. Read (R) access automatically includes X access.
U	Use	Directories	User can attach to directory.
L	List	Directories	Directory contents can be listed.
A	Add	Directories	Directory entry can be added or re-named.
D	Delete	Directories	Directory entry can be deleted or re-named.
P	Protect	Directories	Access can be changed.
O	Owner	Files and directories	Owner can set all rights, except P and ALL, and can change RWLOCK.
ALL		Files and directories	All of the above rights.
NONE		Files and directories	No access allowed.



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

Access rights to registered EPFs cannot be limited by using ACLs.

### Examples

If you do not specify *pathname* on the command line, the system lists the access rights for the current directory. In all other cases, the ACL that protects the object you specify is listed.

Any priority ACL set for the disk partition on which the object resides is also listed, as shown in the first example. Note the header differs for top-level directories (first example) and sub-directories (second example).

```
OK, LIST_ACCESS
ACL protecting "<Current directory>":
    FTP:          DALURWX
    GARY:          ALL
    .TCP_FTP$:    ALURWX
    .TPUBS:        ALURWX
    $REST:        LURX
```

```
Priority ACL in effect for "<Current directory>":
    .GARY:        ALL
```

OK,

If you use the LIST\_ACCESS command on a top-level directory without specific ACLs, any existing priority ACLs are listed as shown in this example.

```
OK, LIST_ACCESS
"<Current directory>" protected by default ACL
  (from "<SBUPT2>OLDMAN>CURLW"):
    PMG:          ALL
    MARLEY:       ALL
    AUK:          ALL
    NAILS:        ALL
    SYSTEM:       ALL
    .SMGR:        ALL
    .SBUPT:       DALURWX
    .TYPOS:       ALL
    $REST:        LUR
```

If the object specified by *pathname* is an access category, the system lists its contents (the ACL) as shown in this example.

```
OK, LIST_ACCESS <SBUPT>EVOG>EVOG.ACAT
Access category "<SBUPT>EVOG>EVOG.ACAT":
    EVOG:         ALL
    .SBUPT:       NONE
    $REST:        NONE
```

---

**Note**

If the access rights on a default-protected directory are changed while you are attached to that directory, LIST\_ACCESS does not display the new access rights to you until you reattach to that directory. If the access rights on a directory protected by a specific ACL are changed while you are attached to that directory, LIST\_ACCESS immediately displays the new access rights to you.

---

See also LIST\_GROUP later in this chapter to display the ACL groups to which a user belongs. LIST\_ACCESS is also discussed in the *PRIMOS User's Guide*, the *Operator's System Overview*, and the *System Administrator's Guide, Volume III: System Access and Security*.

See also EDIT\_ACCESS; SET\_ACCESS.

## LIST\_CONTIGUOUS\_BLOCKS

LIST\_CONTIGUOUS\_BLOCKS returns information about the contiguous space available on a specified partition. This command is most useful on a partition containing CAM files because it determines the degree to which the partition is fragmented. CAM files require contiguous blocks of space.

### Format

```
LIST_CONTIGUOUS_BLOCKS [ { partition  
                        { pathname  
                        -LDEV ldev } }  
                        -DISPLAY n  
                        -FREE size  
                        -HELP
```

### Arguments and Options

The following options are provided:

*partition*

The name of a disk partition. The maximum length of *partition* is 6 characters. If you do not have the partition name, you may obtain it from the STATUS DISKS or LIST\_MOUNTS display. Use *pathname*, rather than *partition* if the partition was added to the system at a mount-point.

- pathname***                    The pathname of any directory on the desired partition, including the mount-point pathname. If a disk is added to the system with a mount-point pathname, that pathname or the pathname of any other directory in the partition must be used instead of the partition name. This pathname must be fully qualified, that is, it must begin with the root directory.
- LDEV ldev***                    Specifies the logical device number of the partition. *ldev* must be expressed in decimal format. The logical device number is expressed in octal format in the STATUS DISKS display. Use the TYPE command and the PRIMOS function [OCTAL] to convert octal to decimal. For example, if the logical device number is 70:
- ```
OK, TYPE [OCTAL 70]
56
OK,
```
- DISPLAY n***                    Displays the sizes of up to *n* fragments, or blocks, of free contiguous space larger than 32 records, or larger than a size specified by the *-FREE* option. The default number of blocks is 20 and the range of the number of blocks that can be specified is from 6 to 1024, inclusive. If *n* is outside this range, it is set to 20.
- FREE size***                    Specifies the minimum size, in records, of the largest blocks of free contiguous space to be displayed. The default minimum of *size* is 32 records and the range of *size* that you can specify is from 0 to 9999 records, inclusive. If *size* is outside this range, it is set to 32.
- HELP***                            Displays the command's usage and options.

If no partition name, pathname, or device number is specified, the MFD of the current attach point is used.

### ***LIST\_CONTIGUOUS\_BLOCKS Output***

LCB returns information about the

- Sizes of the 20 largest blocks of free space greater than 32 records
- Total number of fragments (or free contiguous blocks)
- Minimum and maximum CAM file extension sizes in effect for that partition
- Total number of records
- Total number of free records or available space
- Percent full (total records minus free records, divided by total records)

LIST\_EPF

The following is an example of LCB output for the partition INTEG.

```

OK, LIST_CONTEGUCUS_BLOCKS INTEG
[LCB Rev. 23.3.0 Copyright (c) 1992, Prime Computer, Inc.]
+-----+
! Partition: INTEG !
+-----+
! Largest 20 blocks of free          176    85    !
! contiguous space larger           175    84    !
! than 32 records.                   152    83    !
!                                   133    80    !
!                                   108    72    !
!                                   104    70    !
!                                   99     62    !
!                                   97     51    !
!                                   91     50    !
!                                   87     49    !
+-----+
!                                     !
! Number of fragments : 3958 ! Total records      : 125920 !
! Minimum extent size : 16  ! Free records       : 16497  !
! Maximum extent size : 32  ! Percentage full    : 86.80% !
!                                     !
+-----+

```

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

LIST\_EPF

LIST\_EPF displays information about a user's Executable Program Formats (EPFs).

The LIST\_EPF default output displays the type and pathname of each EPF. The pathnames of the EPFs within each type are listed alphabetically. Registered EPFs are displayed as entrienames.

**Format**

```
LIST_EPF [ pathname1 [ ... pathname8 ] ] [options]
```

**Arguments and Options**

If you issue LIST\_EPF with no pathnames and no options, information is displayed on all the EPFs currently mapped into your address space.

*pathname1* [...] *pathname8*] Specifies up to eight pathnames of EPFs. You do not have to specify the EPF's suffix, which is either .RUN or .RP*n* (where *n* is a single digit). Wildcards are supported.

Unless you specify the `-NOT_MAPPED` option, LIST\_EPF assumes that *pathname* is a mapped (active or nonactive) EPF. If *pathname* is not currently mapped into your address space or does not exist, the following message is displayed on your screen:

No entries selected.

Use the following options to vary the selection and output display of LIST\_EPF. You can specify more than one option if they do not conflict.

- `-ACTIVE` Selects active EPFs.
- `-COMMAND_PROCESSING` Displays the pathnames of the EPFs and the state of the command processing features for program EPFs. These features include the type of file system objects on which the EPF may operate; whether the command processor should process wildcarding, treewalking, or command iteration; and the name generation position for the EPF.
- `-DETAIL` Displays all attributes for each entry selected.
- `-EPF_DATA` Displays the type, status, and pathname of an EPF, plus the following BIND information: the version of BIND used to create the EPF; the date on which the EPF was bound; the program name; the user version; the contents of the EPF comment field; and the number of debugger segments. If the EPF was bound by a version of BIND unable to indicate the first five BIND items, the command returns the message EPF data not available.
- `-LIBRARY` Selects both program-class and process-class library EPFs.
- `-NOT_ACTIVE` Selects nonactive EPFs (that is, EPFs currently mapped into your address space, but not suspended program EPFs or in-use process-class library EPFs).
- `-NOT_MAPPED` Selects unmapped EPFs (that is, EPFs that exist in your file system but are not currently mapped into your address space). If you do not specify *pathname*, information is displayed on all the unmapped EPFs in your current directory.



- Mapped Nonactive. These are EPFs that are mapped into your address space but are neither suspended EPFs nor in-use EPFs. EPFs that have terminated but remain mapped are listed as (not active) by LIST\_EPF.
- Not Mapped. These are EPFs that are not mapped into your address space but are stored in a directory.

Mapped EPFs (active or nonactive) are also referred to as open EPFs.

The LIST\_EPF command works on any EPF, regardless of its type or status.

### **LIST\_EPF Examples**

The following examples illustrate the LIST\_EPF command:

#### **Example 1:**

```
OK, LIST_EPF1

Process-Class Library EPF.

(active)      <GRP3>LIBRARIES*>SYSTEM_LIBRARY.RUN

1 Program-Class Library EPF.

(not active) <GRP3>LIBRARIES*>FORTRAN_IO_LIBRARY.RUN

1 Program EPF.

(not active) <GRP3>CMDNC0>LD.RUN

OK,
```

#### **Example 2: Listing inactive EPFs**

```
OK, LIST_EPF -NOT_ACTIVE

1 Program-Class Library EPF.

(not active) <GRP3>LIBRARIES*>FORTRAN_10_LIBRARY.RUN

1 Program EPF.

(not active) <GRP3>CMDNC0>LD.RUN

OK,
```

**Example 3: Displaying detailed information about an EPF**

OK, LE CIRCLE.RUN -DETAIL

1 Program EPF.

```
(not active) <GRP3>BEECH>CIRCLE.RUN
 1 procedure segment:    +0:4775
 1 linkage area:        -2:4777(3)/3074
bind version:           21.0
date of binding:        91-03-12.10:26:20.Tue
program name:           CIRCLE user version:    (none)
comment:                (none) debug segments:  1
command options:        wldcrd,trwlk,iter file,dir,segdir,acat 1
OK,
```

**Example 4: Listing only the registered EPFs**

OK, LE -REG

3 Process-Class Library EPFs.

```
(registered) (Ready)    PRIMIX_IX_SYSTEM_LIBRARY.RUN
(registered) (Ready)    SYSTEM_LIB$PRC.RUN
(registered) (Ready)    TRANS_LIB$PRC.RUN
```

4 Program-Class Library EPFs.

```
(registered) (Ready)    CC_LIBRARY.RUN
(registered) (Ready)    PRIMIX_IX_CC_LIBRARY.RUN
(registered) (Ready)    PRIMIX_IX_LIBCOURSES.RUN
(registered) (Ready)    VRPG_LIBRARY.RUN
```

2 Program EPFs.

```
(registered) (Ready)    CSH.RUN
(registered) (Ready)    SH.RUN
```



**Example 5:** Listing segments; showing the actual address of each imaginary segment.

OK, LIST\_EPF -SEGMENTS

3 Process-Class Library EPFs.

```
(active)                <GRP3>LIBRARIES*>ECL$LIB.RUN
  1 procedure segment:   +0:5751
  1 linkage area:       -2:5750(3)/20
(not active)            <GRP3>LIBRARIES*>SIT_LIBRARY.RUN
  2 procedure segments: +0:5760                +2:5757
  1 linkage area:       (not allocated)
(active)                SYSTEM_LIB$PRC.RUN
  1 procedure segment:   +0:7777
  2 linkage areas:      -2:3000(0)/0            -4:6020(0)/0
```

4 Program-Class Library EPFs.

```
(not active)            APPLICATION_LIBRARY.RUN
  1 procedure segment:   +0:7777
  2 linkage areas:      -2:3005(0)/0            -4:6020(0)/105326
(not active)            <GRP3>LIBRARIES*>DSMLIB.RUN
  4 procedure segments: +0:5766                +2:5765
                       +4:5764                +6:5763
(not active)            <GRP3>LIBRARIES*>MAIL_LIBRARY.RUN
  1 procedure segment:   +0:5767
  1 linkage area:       (not allocated)
(not active)            <GRP3>LIBRARIES*>SP$LIB.RUN
  1 procedure segment:   +0:5752
  2 linkage areas:      (not allocated)
```

1 Program EPFs.

```
(not active)            <GRP3>CMDNCO>TERM.RUN
  1 procedure segment:   +0:5776
  1 linkage area:       -2:5777(3)/7506
```

For further information on LIST\_EPF and on EPFs, see the *Programmer's Guide to BIND and EPFs*. For further information on registered EPFs, see the *Advanced Programmer's Guide I: BIND and EPFs*.

See also LIST\_REGISTERED\_EPF; REMOVE\_EPF.

.....  
LIST\_GROUP

## LIST\_GROUP

LIST\_GROUP lists the ACL groups to which a user belongs.

### *Format*

LIST\_GROUP

### *Usage*

An ACL group is a number of users who have the access rights assigned to that group. All group names begin with a period (for example, .MANAGERS). ACL groups are created by the System Administrator or by a Project Administrator. Each user may be a member of as many as 32 groups.

If you are not a member of any group, LIST\_GROUP displays the following message:

```
OK, LIST_GROUP  
No groups.  
OK,
```

If you are a member of one or more groups, the command lists the group names, as in the following example:

```
OK, LIST_GROUP  
Groups are: .TEAM .ADMINISTRATORS .LEADERS  
OK,
```

For further information on ACLs and groups, see the *PRIMOS User's Guide*.

See also the GROUP\_LIST command function in Chapter 3.

## LIST\_LIBRARY\_ENTRIES

LIST\_LIBRARY\_ENTRIES displays the entrypoints in a library EPF.

The LIST\_LIBRARY\_ENTRIES command works on program-class library EPFs, process-class library EPFs, and Registered EPFs.

### Format

LIST\_LIBRARY\_ENTRIES [ *pathname1* [ ... *pathname8* ] ] [*options*]

### Arguments and Options

*pathname1* [ ... *pathname8*]

*pathname* is the pathname or filename of a library EPF. You can specify a maximum of eight pathnames. You do not have to specify the EPF's suffix, which is either .RUN or .RP*n* (where *n* is a single digit). You can use wildcard names. To specify registered library EPFs, specify only the filename of each library, and include the -REG option. If no pathnames are specified, information is displayed on all the library EPFs listed in your entrypoint search list (ENTRY\$.SR).

-ENTRYNAME *entryname*  
[ ... *entryname-7*]

Displays the entrypoint specified by *entryname*, which can be a wildcard name. You can specify a maximum of eight entrynames.

-NO\_WAIT

Suppresses the --More-- prompt and does not pause after every 23 lines of output. Output scrolls continuously.

-REG

Displays the list of entrypoints contained in the registered library EPFs. If omitted, registered library EPFs are not listed.

-HELP

Displays the syntax of the command. The Help screen is also displayed if PRIMOS detects an error while parsing the command.

### Usage

The default LIST\_LIBRARY\_ENTRIES output for each EPF displays a two-line header followed by an alphabetical listing of the entrypoints. The first line of the

.....

## LIST\_LIMITS

header lists the EPF's status (active, not active, or not mapped) and its pathname. The second line of the header lists the EPF's type (process-class or program-class), the total number of entrypoints, and the number of entrypoints currently selected for display.

The EPFs are listed alphabetically if there are two or more.

The example below illustrates the LIST\_LIBRARY\_ENTRIES command:

```

OK, LBENT <GRP>LIBRARIES>FORTRAN_IO_LIBRARY -ENTRYNAME R04
(active) <GRP>LIBRARIES*>FORTRAN_IO_LIBRARY.RUN
Program-Class Library EPF, 110 Total Entryoints, 6 Selected Entryoints
RDASC      RDBIN      RDLN$P      RDLN$P$I    RECLN$      RSTBL
OK,

```

For further information on LIST\_LIBRARY\_ENTRIES and on EPFs, see the *Programmer's Guide to BIND and EPFs*.

See also LIST\_EPF; LIST\_SEARCH\_RULES; REMOVE\_EPF; SET\_SEARCH\_RULES.

## LIST\_LIMITS

LIST\_LIMITS displays information on the various attributes of a user's command environment.

### Format

```

LIST_LIMITS { -SERVER
              -HELP }

```

### Options

|         |                                                                    |
|---------|--------------------------------------------------------------------|
| -SERVER | Additionally displays InterServer Communications (ISC) attributes. |
| -HELP   | Displays the command syntax.                                       |

### Usage

The System Administrator sets the command environment attributes of all users. The attributes are set in a user profile or on a systemwide basis. The limits of these attributes are important when you are programming with Executable Program Formats (EPFs).

The limits include the following:

- The number of command levels you can use
- The number of programs you can invoke at any command level
- The number of private dynamic segments you can use
- The number of private static segments you can use

The `-SERVER` option provides the following additional information:

- The number of ISC sessions you can start
- The number of ISC synchronizers you can allocate (usually 8 per session)
- The number of timers you can use to post notices on ISC synchronizers

The following example illustrates the `LIST_LIMITS` display:

```
OK, LIST_LIMITS
```

```
Maximum number of command levels: 10
Maximum number of program invocations: 10
Maximum number of private static segments: 100
Maximum number of private dynamic segments: 150
```

```
OK,
```

```
OK, LIST_LIMITS -SERVER
```

```
Maximum number of command levels: 10
Maximum number of program invocations: 10
Maximum number of private static segments: 100
Maximum number of private dynamic segments: 150
Maximum number of ISC sessions: 16
Maximum number of synchronizers: 128
Maximum number of timers: 16
```

```
OK,
```

For further information on `LIST_LIMITS` and on EPFs, see the *Advanced Programmer's Guide III: Command Environment*.

For further information on ISC, see *Subroutines Reference V*.

See also `INITIALIZE_COMMAND_ENVIRONMENT`, `LIST_SEGMENT`.

.....

## LIST\_MINI\_COMMANDS

### LIST\_MINI\_COMMANDS

LIST\_MINI\_COMMANDS lists the PRIMOS commands allowed at mini-command level.

The LIST\_MINI\_COMMANDS output gives the names and abbreviations of the PRIMOS commands you can use when you reach mini-command level, that is, when you exceed your maximum command level depth.

#### Format

LIST\_MINI\_COMMANDS [*command\_match*]

#### Argument

*command\_match* is a character string used as a pattern match for the commands allowed at mini-command level. The character string can contain wildcard characters, such as @@.

#### Usage

If you do not specify *command\_match*, all the commands allowed at mini-command level are listed alphabetically, as follows:

OK, LIST\_MINI\_COMMANDS

| Abbrev | Full name          | Abbrev | Full name                      |
|--------|--------------------|--------|--------------------------------|
| -----  | -----              | -----  | -----                          |
| C      | CLOSE              | COMO   | COMOUTPUT                      |
| DMSTK  | DUMP_STACK         | ICE    | INITIALIZE_COMMAND_ENVIRONMENT |
| LE     | LIST_EPF           | LL     | LIST_LIMITS                    |
| LMC    | LIST_MINI_COMMANDS | LS     | LIST_SEGMENT                   |
|        | LOGIN              | LO     | LOGOUT                         |
| P      | PM                 | PR     | PRERR                          |
|        | RDY                | REN    | REENTER                        |
| RLS    | RELEASE_LEVEL      | REMEPF | REMOVE_EPF                     |
| S      | START              |        |                                |

OK,

For further information on LIST\_MINI\_COMMANDS and on EPFs, see the *Programmer's Guide to BIND and EPFs*.

## LIST\_MOUNTS

LIST\_MOUNTS displays the name and location of disks and portals in your file system name space.

LIST\_MOUNTS displays the contents of the Global Mount Table (GMT) instead of the contents of the Local Disk Table, which is displayed with the STATUS DISKS and LIST\_DISKS commands. If the system is running the Name Server, STATUS DISKS and LIST\_DISKS no longer display remote disks unless they are manually added to the Disk Table with the ADDISK command. For more information on the ADDISK command see the *Operator's Guide to System Commands*.

### Format

```
LIST_MOUNTS [-DISK diskname
             -MOUNT_PATH pathname
             -NO_SORT
             -NO_WAIT
             -PORTAL [systemname diskname]
             -SYSTEM [systemname]
             -HELP
```

### Options

**-DISK *diskname***

Lists currently mounted disk partitions. The lists excludes portals. Specify *diskname* to display individual partitions. *diskname* may be a prefix or contain wildcards. (A prefix is any number of characters at the beginning of the partition name). If you use neither the -DISK nor the -PORTAL option, all disks and portals are displayed. The -PORTAL option cannot be used in conjunction with the -DISK option.

**-MOUNT\_PATH *pathname***

Lists mounts that were added with a mount-path *pathname* specified by *pathname*. *pathname* may be the pathname or a prefix. (A prefix is any number of characters at the beginning of the pathname). Wildcards are not accepted.

**-NO\_SORT**

Lists mounts in the order in which they are stored in the Global Mount Table (an order convenient for PRIMOS) instead of sorting alphabetically by mount-point pathname.





*Mount pathname*                      Indicates where the disk was added in the tree hierarchy. If the object is a portal, the name before the arrow shows the pathname of the portal; the name after the arrow is the name of the remote system and disk to which the portal provides a gateway.

## **LIST\_PRIORITY\_ACCESS**

*LIST\_PRIORITY\_ACCESS* lists the priority Access Control List (ACL) of the specified local disk.

### **Format**

*LIST\_PRIORITY\_ACCESS diskname*

### **Argument**

*diskname*                      Specifies the partition name of a local disk. Do not use angle bracket delimiters.

### **Usage**

The *LIST\_PRIORITY\_ACCESS* command is provided because a priority ACL can prevent you from accessing directories and from using the *LIST\_ACCESS* command. To display the names of the disk partitions (disknames) connected to your system, use the *STATUS DISKS* command.

The following example illustrates the use of *LIST\_PRIORITY\_ACCESS* on the partition named *GALAXY*:

```
OK, LIST_PRIORITY_ACCESS GALAXY
Priority ACL on partition "<GALAXY>":
    MERCURY:  ALL
    $REST:    NONE
```

.....  
LIST\_QUOTA

## LIST\_QUOTA

LIST\_QUOTA lists current disk quota and storage information for a directory.

A quota is the maximum number of disk records that a directory can hold. The System Administrator usually sets quotas on top-level directories. Users who have access rights can set quotas on directories with the SET\_QUOTA command. If a directory has a quota, it cannot obtain records that would cause it to exceed the quota. This restriction is enforced on the entire subtree. If various levels of a subtree have multiple quotas, then the most restrictive quota is enforced.

To use LIST\_QUOTA, you must have List (L) access to the target directory and Use (U) access to all higher-level directories.

### Format

LIST\_QUOTA [*pathname*] [-BRIEF]

### Argument and Option

|                 |                                                                                                                                                                         |
|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>pathname</i> | Specifies the directory for which you want quota information. If you do not specify <i>pathname</i> , information for your current directory is listed.                 |
| -BRIEF          | Displays the information on a single line. The name of the directory is displayed as the last entry on the line only if you specified <i>pathname</i> with the command. |

LIST\_QUOTA displays information for both quota and nonquota directories. (A nonquota directory has a quota setting of zero.) The information consists of the following three items. Unless you use the -BRIEF option, each item is displayed on a separate line.

- The maximum number of records allocated for use by the directory (the quota) or a message stating that the directory is not a quota directory.
- The total number of records currently used by the entire subtree, including the target directory. This number must be less than the quota.
- The number of records currently used by the target directory.

### Examples of Using LIST\_QUOTA

The following three examples illustrate the LIST\_QUOTA command.

**Example 1:** This example shows quota information for the subdirectory STATS in the directory REPORTS:

```
OK, LIST_QUOTA REPORTS>STATS

Maximum records allowed on "REPORTS>STATS" = 500
Total records used = 425.
Records used in this directory = 28.
OK,
```

The output shows that the quota for the directory STATS is 500 records. The directory tree is currently using 425 records, 28 of which are used by the entries in STATS itself.

**Example 2:** This example uses the same directory as Example 1, but uses the -BRIEF option to produce the same information in an abbreviated form:

```
OK, LIST_QUOTA REPORTS>STATS -BRIEF
MAX:      500, Used:      425, Records:      28, REPORTS>STATS
OK,
```

**Example 3:** This example is similar to Example 1, but assumes that STATS does not have a quota:

```
OK, LIST_QUOTA REPORTS>STATS
"REPORTS>STATS" is not a quota directory.
Total records used = 425.
Records used in this directory = 28.
OK,
```

The records used by both the subtree and the target directory are displayed whether or not there is a quota set.

For more information on directory quotas, see the *PRIMOS User's Guide*.

.....

LIST\_RBF

## LIST\_RBF

LIST\_RBF lists the attributes of a ROAM master or slave file.

The LIST\_RBF output includes the activation status of the file, the status of the recovery attributes, the date and time last saved, the usage of the file, and the ROAM file identifier.

### Format

LIST\_RBF [ { -DETAIL }  
          { -SIZE } ]

### Options

- SIZE           Displays the size, in halfwords, of the requested segment directory and of the complete ROAM file.
- DETAIL         Reserved for future use and has no effect at present.

### Usage

For further details on LIST\_RBF, see the *ROAM Administrator's Guide*.

## LIST\_REGISTERED\_EPF

LIST\_REGISTERED\_EPF lists the dependency list and/or the unresolved entrypoints for a specified EPF.

### Format

```
LIST_REGISTERED_EPF [epfname] [-DEPENDENCY_LIST
                                -UNRESOLVED_ENTRYPOINTS
                                -HELP]
```

### Argument and Options

If you do not specify any option, LIST\_REGISTERED\_EPF displays both the dependency list and the unresolved entrypoints.

|                                |                                                                                                                                                                                       |
|--------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>epfname</i>                 | Specifies the name of a specific registered EPF about which information is needed. If it is omitted, information about all registered EPFs is displayed.                              |
| <b>-DEPENDENCY_LIST</b>        | Lists the EPFs that are necessary for the specified EPF to run. This option shows which of the listed EPFs are Direct (explicitly linked) and which are Indirect (implicitly linked). |
| <b>-UNRESOLVED_ENTRYPOINTS</b> | Lists the unresolved entrypoints in the specified EPF.                                                                                                                                |
| <b>-HELP</b>                   | Displays command syntax.                                                                                                                                                              |

---

#### Note

Installation of the Translator Family Release T3.0-23.0 or greater is required.

---

For further information on LIST\_REGISTERED\_EPF and on EPFs, see the *Programmer's Guide to BIND and EPFs*. For further information on registered EPFs, see the *Advanced Programmer's Guide I: BIND and EPFs*.

See also LIST\_EPF; REMOVE\_EPF.

.....

## LIST\_REMOTE\_ID

## LIST\_REMOTE\_ID

LIST\_REMOTE\_ID displays the user's remote IDs.

Your remote IDs are established with ADD\_REMOTE\_ID commands and are kept on a remote-ID list. The list holds a maximum of 16 remote IDs, each for use on a different remote system. LIST\_REMOTE\_ID displays the contents of this list. The display includes all remote IDs, project IDs (if any), and the systems on which the remote IDs are used. Passwords are never listed.

### Format

LIST\_REMOTE\_ID [-ON *system*]

### Option

**-ON *system*** Limits the output to remote IDs on the system specified by *system*.

### Usage

The following example illustrates the LIST\_REMOTE\_ID command with and without the -ON option. The three remote IDs were set up with the ADD\_REMOTE\_ID command.

```

OK, LIST_REMOTE_ID
System   User id   Project id
-----
---
SYSB     LAURA.B
SYSC     LAURA.C OAKTREE
SYSF     LAURA.F GRAND

OK, LRID -ON SYSC
System   User id   Project id
-----
---
SYSC     LAURA.C OAKTREE
OK,

```

See also ADD\_REMOTE\_ID; REMOVE\_REMOTE\_ID.

## LIST\_SCHEDULER\_ATTRIBUTES

LIST\_SCHEDULER\_ATTRIBUTES displays the current scheduler status.

### *Format*

LIST\_SCHEDULER\_ATTRIBUTES

### *Example*

OK, LIST\_SCHEDULER\_ATTRIBUTES

|                             | Scheduler Status            |
|-----------------------------|-----------------------------|
| Short Job setting (0 to 4)  | : 4                         |
| Queues                      | : Lopriq, Eligq and Hipriq. |
| target ratio                | : 1 : infinite : infinite   |
| Priority levels             | : Priority 0 to 4           |
| target ratio                | : 1:2:4:8:16                |
| Eligibility time slice (ms) | : 101                       |
| Maximum scheduled jobs      | : 108                       |

OK,

LIST\_SEARCH\_RULES

LIST\_SEARCH\_RULES

LIST\_SEARCH\_RULES lists the contents of search list(s) for a user's process. The use of wildcards is supported.

By default, LIST\_SEARCH\_RULES lists the system-defined search rule files. Typically, these search lists are:

- ATTACH\$
- COMMANDS\$
- DEFINITION\$
- ENTRY\$
- INCLUDE\$
- BINARY\$

Note

If you have FS\_RECOVER installed on your system, the default search lists also contain the AUTOPSY and MAPS search lists.

When you either log in or otherwise initialize a process, PRIMOS establishes the special-purpose search lists noted above. (See the SET\_SEARCH\_RULES command, described later in this chapter, for information about modifying the contents of these search lists and creating additional search lists.) The LIST\_SEARCH\_RULES command allows you to display the contents of your active search lists, and shows the pathname of the search rules file used to create each search list.

Format

```
LIST_SEARCH_RULES [ { [ listname1 ... listname16 ] } ]
                  [ { --NO_WAIT } ]
                  [ { --HELP } ]
```

Argument and Options

*listname1 ... listname16* Specifies which search lists you wish to see. If you specify one or more list names, LIST\_SEARCH\_RULES displays those search lists in the order specified. You can specify a maximum of 16 list names. If you do not specify *listname*, LIST\_SEARCH\_RULES displays the five system-defined search lists mentioned above, plus all of the search lists that you have defined. The most recently set search list is displayed first.



- NO\_WAIT**                      Suppresses the `--More--` prompt and does not pause after every 23 lines of output. Output scrolls continuously.
- HELP**                         Displays the command syntax and options. The information is also displayed if PRIMOS detects an error while parsing the command.

### Usage

The following example illustrates the LIST\_SEARCH\_RULES command:

```
OK, LSR

List: ATTACH$
Pathname of template: <UPLAB>SEARCH_RULES*>ATTACH$.SR

    -added_disks

List: INCLUDE$
Pathname of template: <UPLAB>SEARCH_RULES*>INCLUDE$.SR

    [HOME_DIR]

List: BINARY$
Pathname of template: <UPLAB>SEARCH_RULES*>BINARY$.SR

    [HOME_DIR]

List: COMMAND$
Pathname of template: <UPLAB>SEARCH_RULES*>COMMAND$.SR

    cmdnc0

List: ENTRY$
Pathname of template: <UPLAB>SEARCH_RULES*>ENTRY$.SR

    -primos_direct_entries
    libraries*>system_library.run
    libraries*>fortran_io_library.run
    libraries*>application_library.run
    -static_mode_libraries

OK,
```

For more information on the Search Rules facility, see the *Advanced Programmer's Guide II: File System*. For further information on system and administrator default rules, see the *System Administrator's Guide, Volume III: System Access and Security*. See also LIST\_LIBRARY\_ENTRIES; SET\_SEARCH\_RULES.

.....

## LIST\_SEGMENT

### LIST\_SEGMENT

LIST\_SEGMENT lists the numbers and access rights for the DTAR2 private segments that you are currently using.

Segment numbers are displayed in ascending numerical order. The access rights codes used in the list and their meanings are as follows:

|      |                                 |
|------|---------------------------------|
| NULL | No access allowed               |
| GATE | Gate access                     |
| R    | Read access                     |
| RW   | Read and write access           |
| RX   | Read and execute access         |
| RWX  | Read, write, and execute access |

#### Format

**LIST\_SEGMENT** [*segno1* [ ... *segno8*]] [*options*]

#### Arguments and Options

You can specify more than one option if they do not conflict.

|                                     |                                                                                                                                                                                                                                                                                                                                                                                                  |
|-------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>segno1</b> [ ... <b>segno8</b> ] | Specifies the segment number (in octal) about which you want information. You can specify a maximum of eight segment numbers, separating each number with a space or comma. You cannot use wildcarding and iteration. If you do not specify any segment numbers, LIST_SEGMENT displays the numbers and access rights for each segment currently in use in the static and dynamic segment ranges. |
| <b>-BRIEF</b>                       | Displays only the total number of segments currently in use in each segment range.                                                                                                                                                                                                                                                                                                               |
| <b>-DYNAMIC</b>                     | Displays information only on private dynamic segments.                                                                                                                                                                                                                                                                                                                                           |

- NAME                    Displays the name of any EPF associated with the segment. This option is valid only for private dynamic segments. An EPF is associated with a segment if the procedure or linkage areas for that EPF are assigned to that segment. If more than one EPF is associated with a given segment (as may be the case if several EPFs have their linkage areas allocated within the same segment), the EPF pathnames are arranged alphabetically on separate lines. If a given EPF uses more than one segment, the EPF pathname appears alongside the segment number and access rights for each segment used. If a dynamic segment is not associated with an EPF, the message *none* appears instead.
  
- NO\_WAIT                Suppresses the *--More--* prompt and does not pause after every 23 lines of output. Output scrolls continuously.
  
- STATIC                 Displays information only on private static segments.
  
- HELP                    Displays the command's correct syntax. The HELP screen is also displayed if PRIMOS detects an error while parsing the command.

**Usage**

The following example illustrates LIST\_SEGMENT without options:

```
OK, LIST_SEGMENT

1 Private static segment.
segment access
-----
4000    RWX

5 Private dynamic segments.
segment access
-----
4152    RX
4153    RX
4154    RX
4155    RX
4377    RWX

OK,
```

For further information on LIST\_SEGMENT and on EPFs, see the *Advanced Programmer's Guide III: Command Environment*.

See also LIST\_EPF; LIST\_LIMITS.

.....

## LIST\_SERVER\_NAMES

### LIST\_SERVER\_NAMES

LIST\_SERVER\_NAMES lists server names on the local system together with the user ID and member process numbers for each server. The LIST\_SERVER\_NAMES command also displays the total number of servers on the local node.

#### Format

LIST\_SERVER\_NAMES [-HELP]

#### Usage

PRIMOS assigns a server name to every process (user processes as well as system processes) when you initialize the process. By default, PRIMOS places each terminal process or phantom process in its own server. A PRIMIX child process, however, is a member of its parent's server.

Starting with PRIMOS Revision 22.0, these server names are used by InterServer Communications (ISC). ISC is a PRIMOS facility that provides for message exchange between two concurrently running servers.

This command lists server name information as follows:

```
OK, LIST_SERVER_NAMES
[LIST_SERVER_NAMES Rev. 23.3.0 Copyright (c) 1992, Prime Computer, Inc.]
```

113 Servers for node S51

| Server Name   | UserID             | Member Process(es) |
|---------------|--------------------|--------------------|
| DSMSMSR\$     | SYSTEM_MANAGER     | 313                |
| DSMSR\$       | DSMSR              | 311                |
| DSM_LOGGER\$  | DSM_LOGGER         | 312                |
| ISCSNETSVR    | ISC_NETWORK_SERVER | 314                |
| NAME_SERVER\$ | NAME_SERVER        | 315                |
| NFS_MOUNT\$   | NFS_MOUNT          | 275                |
| .             | .                  | .                  |
| .             | .                  | .                  |
| TNDQCDTWHGGC  | JOSH               | 68                 |
| TNDQCDTWJDCV  | GPRPR              | 50                 |
| TNDQCDTWJGTF  | MACNE              | 155                |

OK,

Servers are listed in alphabetical order by server name.

For further information on server names, see the *Subroutines Reference V: Event Synchronization*.

See also LIST\_SESSIONS.

## LIST\_SESSIONS

LIST\_SESSIONS displays information about Interserver Communications (ISC) sessions. A session links two active servers.

Information may be displayed about

- a single ISC session
- all of the sessions belonging to a single ISC server
- about all ISC sessions

LIST\_SESSIONS can only display information about sessions on the local node.

If you specify no options, LIST\_SESSIONS displays overview information about all currently active ISC sessions. If there are no active sessions, LIST\_SESSIONS returns the message `No Sessions`.

### Format

```
LIST_SESSIONS [ -SERVER_NAME name [-SESSION number]
               -DETAIL
               -USER userid
               -HELP ]
```

### Options

The following LIST\_SESSIONS options are available:

|                                 |                                                                                                                                                                                |
|---------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-SERVER_NAME <i>name</i></b> | Lists session information for all sessions owned by <i>name</i> . <i>name</i> is the name of an active server on the local node. <i>name</i> cannot be the ISC Network Server. |
| <b>-SESSION <i>number</i></b>   | Lists session information for session <i>number</i> owned by server <i>name</i> . You cannot use the -SESSION option without the -SERVER_NAME option.                          |
| <b>-DETAIL</b>                  | Lists detailed information for the sessions selected by the other LIST_SESSIONS options.                                                                                       |
| <b>-USER <i>userid</i></b>      | Lists information about sessions owned by user <i>userid</i> .                                                                                                                 |
| <b>-HELP</b>                    | Displays command syntax.                                                                                                                                                       |

### LIST\_SESSIONS Output

By default, the LIST\_SESSIONS command displays overview information about each listed session. Sessions are sorted by server name, then by session number within each server name. If you display information for all servers, each local session is represented by two entries, one for each server. A remote session is represented by only one entry for the server that is on the local system.

LIST\_SESSIONS

OK, LIST\_SESSIONS  
 [LIST\_SESSIONS Rev 23.3.0 Copyright (c) 1992, Prime Computer, Inc.]

NODE SYS.01 SESSIONS:

| Server Name    | Ses# | Init? | Phase | User ID        | Mate Server Name |
|----------------|------|-------|-------|----------------|------------------|
| Mate User ID   |      |       |       | Mate Node      |                  |
| DSMSMSR\$      | 1    | yes   | DXFER | SYSTEM_MANAGER | DSM_LOGGER\$     |
| DSM_LOGGER     |      |       |       | local node     |                  |
| DSMSMSR\$      | 2    | yes   | DXFER | SYSTEM_MANAGER | DSM_LOGGER\$     |
| DSM_LOGGER     |      |       |       | local node     |                  |
| DSM_LOGGERS    | 8    | no    | DXFER | DSM_LOGGER     |                  |
| SYSTEM_MANAGER |      |       |       | local node     |                  |
| TNDNGWGNBBBC   | 1    | yes   | ESTAB | SYSTEM         | NAME_SERVER\$    |
| NAME_SERVER    |      |       |       | local node     |                  |
| TNDNGWGNBDWV   | 1    | yes   | DXFER | NM_SERVER      | DSMSMSR\$        |
| SYSTEM_MANAGER |      |       |       | local node     |                  |

The meaning of the listed parameters is as follows:

- Server Name            The 12-character name of the server that owns this session.
- Ses#                    The session number that this server uses to identify this session. The server on the other side of the session identifies the session with a different session number.
- Init?                   If yes, the server is the session initiator. If no, the server is the session recipient.
- Phase                   The current state (phase) of the session. The phase codes used and their meanings are as follows:
  - DXFER            Data transfer
  - ESTAB            Establishing
  - EXCPT            Exception pending
  - PEND             Pending
  - TERM             Terminating
  - UNDEF            Undefined
- User ID                The user ID of this server.
- Mate User ID           The user ID of the server on the other end of the session. If this user ID cannot be determined, this parameter displays a question mark (?) within parentheses.

**Mate Node**                    The node name of the server on the other side of the session. If both servers are on the same node, this parameter displays `local node`. If this node name cannot be determined, this parameter displays a question mark (?) within parentheses.

**Mate Server Name**        The server name of the server on the other end of the session. This server name is only displayed if `Init?=YES`; otherwise, this field is blank.

### Output Using the `-DETAIL` Option

If you specify the `-DETAIL` option, the `LIST_SESSIONS` command displays detailed information about each session. You can list detailed information about all sessions on the current node, or use the `-SERVER_NAME` or `-SERVER_NAME` and `-SESSION` options to select specific sessions for detailed display.

Session detail information is displayed as follows:

```
OK, LIST_SESSIONS -DETAIL -SERVER_NAME DSMSMSR$ -SESSION 1
[LIST_SESSIONS Rev. 23.3.0 Copyright (c) 1992, Prime Computer, Inc.]

Session # 1 of server DSMSMSR$
  Session detailed state:                ESTABLISHED
  Mate user ID:                          DSM_LOGGER
  Mate node name:                         local node

Normal service:      CONFIGURED      Expedited service: NOT CONFIGURED
Max ctl: 128 Max data: 32630      Max expedited size: 0

Normal Send          Normal Recv          Message area
-----
Queue size           31 Queue size           31 Block size           1024
Free slots           31 Msgs on Q             0 Number of blocks    128
Total sent           2 Total recd             0 Current % in use:   3
   Max % in use:       7

Exp Send             Exp Recv
-----
Queue size           0 Queue size           0
Free slots           0 Msgs on Q             0
Total sent           0 Total recd             0
```

The meaning of the listed parameters is as follows:

**Session #**                    The number by which this server identifies the session. The server on the other side of the session identifies the session with a different session number.

**Server**                        The 12-character name of the server that owns this session.

■ ■ ■ ■ ■ ■ ■ ■ ■ ■  
*LIST\_SESSIONS*

|                        |                                                                                                                                                                                                                                                                                                                                                        |
|------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Session detailed state | <p>The current state of the session. The possible states are as follows:</p> <p>ESTABLISHED<br/>EXCEPTION<br/>EXCEPTION BEFORE ACCEPT RECEIVED<br/>PENDING<br/>REQUEST RECEIVED<br/>SESSION ACCEPTED<br/>SESSION REJECTED<br/>SESSION REQUESTED<br/>TERMINATED<br/>TERMINATED BEFORE REPLYING<br/>TERMINATED PENDING<br/>TERMINATING<br/>UNDEFINED</p> |
| Mate user ID           | <p>The user ID of the server on the other end of the session. If this user ID cannot be determined, this parameter displays a question mark (?) within parentheses.</p>                                                                                                                                                                                |
| Mate node name         | <p>The node name of the server on the other side of the session. If both servers are on the same node, this parameter displays <code>local node</code>. If this node name cannot be determined, this parameter displays a question mark (?) within parentheses.</p>                                                                                    |
| Normal service         | <p>CONFIGURED indicates that normal message service is provided for this session; NOT CONFIGURED indicates that normal message service is not provided for this session.</p>                                                                                                                                                                           |
| Expedited service      | <p>CONFIGURED indicates that expedited message service is provided for this session; NOT CONFIGURED indicates that expedited message service is not provided for this session.</p>                                                                                                                                                                     |
| Max ctl                | <p>The maximum length (in bytes) of the control part of normal messages.</p>                                                                                                                                                                                                                                                                           |
| Max data               | <p>The maximum length (in bytes) of the data part of normal messages.</p>                                                                                                                                                                                                                                                                              |
| Max expedited size     | <p>The maximum length (in bytes) of expedited messages.</p>                                                                                                                                                                                                                                                                                            |
| Queue size             | <p>The queue length for normal and expedited message send and receive queues.</p>                                                                                                                                                                                                                                                                      |
| Free slots             | <p>The number of free message slots available on each of the send queues.</p>                                                                                                                                                                                                                                                                          |



|                  |                                                                                                                        |
|------------------|------------------------------------------------------------------------------------------------------------------------|
| Total sent       | The number of normal and expedited messages successfully sent.                                                         |
| Msgs on Q        | The number of messages on each receive queue.                                                                          |
| Total recd       | The number of normal and expedited messages successfully received.                                                     |
| Block size       | The length in bytes of the blocks that comprise the session's data part message area.                                  |
| Number of blocks | The number of blocks that comprise the session's data part message area.                                               |
| Current % in use | The percentage of the session's data part message area that is currently in use.                                       |
| Max % in use     | The maximum percentage of the session's data part message area that has been used since the beginning of this session. |

For further details on ISC sessions, see the *Subroutines Reference V: Event Synchronization*.

See also LIST\_SERVER\_NAMES; LIST\_LIMITS.

## LIST\_USERS

LIST\_USERS provides information about users currently logged into the system.

### Format

```
LIST_USERS { user } [ selection options ] [-HELP]
             { -ME }  [ display options ]
```

### Options

-HELP                      Displays the LIST\_USER syntax and options.

### Selection Options

*user*                      Selects users named *user*. Wildcarding is supported. Cannot be used with the -ME option.

-ALL\_DISKS                 Selects the disks in use by users.

-ASSIGNED\_DEVICES         Lists the devices assigned to users.

.....  
**LIST\_USERS**

- BATCH\_JOBS**                      Selects batch job processes.
- DISK *diskname***                Selects users on disk partition named *diskname*.  
*diskname* is the disk's partition name without enclosing angle brackets. Use the STATUS DISKS or LIST\_MOUNTS commands to list available partition names.
- CHILDREN**                        Selects child processes.
- IDS\_ONLY**                        Lists only the user IDs of users. Excludes user numbers and other attributes.
- ME**                                Selects processes belonging to you. Cannot be used if you specify *user*.
- ON *system***                      Lists users on a remote system named *system*.
- PHANTOMS**                        Selects phantom processes.
- PROJECTS**                        Lists current project assignments for all users.
- SERVERS**                         Selects server processes.
- SLAVES**                          Selects slave processes.
- SPOOLERS**                        Selects spooler processes. Not effective on systems running a pre-Rev. 23.3 PRIMOS version.
- USERS**                            Selects user processes. Excludes children, phantoms, servers, and slaves.

**Display Options**

- DETAIL**                            Lists all attributes for each user you select.
- 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.
- NO\_HEADER**                        Suppresses the top header line and process type headers. This option is most useful when combined with the -SINGLE\_COLUMN option.
- NO\_WAIT**                         Suppresses the --More-- prompt and does not pause after every 23 lines of output. Output scrolls continuously.
- SINGLE\_COLUMN**                    Lists each entry on a separate line. Useful with the -NO\_HEADER option to provide an uninterrupted list of all users.

**-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.

**LIST\_USERS Examples**

To list processes belonging to you, use the command

OK, LIST\_USERS -ME

One user:

149 DRG (me)

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 | 322 | ISC_NETWORK_SERVER | 323 | NAME_SERVER   |
| 324 | UBI_SERVER     | 325 | DSMASR             |     |               |

OK,

LIST\_USERS

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

OK, LIST\_USERS -ALL\_DISKS

5 users:

|    |        |          |          |
|----|--------|----------|----------|
| 3  | BDD    | <OPTRAC> |          |
| 4  | MAXXI  | <OPTRAC> |          |
| 18 | BRUSH  | <OPTRAC> |          |
| 29 | WWMP   | <OPTRAC> | <SYSONE> |
| 44 | BUDMAN | <OPTRAC> | <SYSONE> |

3 phantoms:

|     |               |          |  |
|-----|---------------|----------|--|
| 254 | DST_SERVER    | <SYSONE> |  |
| 255 | MAIL_SERVER   | <SYSONE> |  |
| 258 | BATCH_SERVICE | <SYSONE> |  |

1 child processes:

|     |      |          |          |
|-----|------|----------|----------|
| 288 | GIVE | <TPBSL1> | <OPTRAC> |
|-----|------|----------|----------|

10 servers:

|     |                |          |          |
|-----|----------------|----------|----------|
| 1   | SYSTEM         | <OPTRAC> | <SYSONE> |
| 314 | TIMER_PROCESS  | <SYSONE> |          |
| 315 | BUFFER_SERVER  | <SYSONE> |          |
| 316 | LOGOUT_SERVER  | <SYSONE> |          |
| 317 | LOGIN_SERVER   | <SYSONE> |          |
| 319 | DSMSR          | <SYSONE> |          |
| 320 | DSM_LOGGER     | <SYSONE> |          |
| 321 | SYSTEM_MANAGER | <SYSONE> |          |
| 323 | NAME_SERVER    | <SYSONE> |          |
| 325 | DSMASR         | <SYSONE> |          |

## LIST\_VAR

LIST\_VAR lists variables and their values from an active global variable file.

### Format

LIST\_VAR [*global-variable1* ... *global-variable-n*]

### Argument

*global-variable1* ... *global-variable-n* Specifies global variables to display. Use spaces to separate global variable names. Unrecognized variable names are ignored.

### Usage

If you specify no variable names, LIST\_VAR lists all global variables and their values, as in the following example:

```
OK, LIST_VAR
  .AWAY          BEECH>BRANCH2>TWIG4
  .HOME          BEECH>BRANCH5>TWIG3
  .ADAM          GARDEN>ADAM
OK,
```

If you specify one or more variable names, only those variables and their values are listed. Separate the variable names with a space or a comma, as in the following example:

```
OK, LIST_VAR .AWAY .HOME
  .AWAY          BEECH>BRANCH2>TWIG4
  .HOME          BEECH>BRANCH5>TWIG3
OK,
```

You can specify a wildcard name as a variable name, as follows:

```
OK, LIST_VAR .A@@
  .AWAY          BEECH>BRANCH2>TWIG4
  .ADAM          GARDEN>ADAM
OK,
```

For further information on global variables, see the *PRIMOS User's Guide*.

See also DEFINE\_GVAR; DELETE\_VAR; SET\_VAR.

.....

LL

LL                    *See* LIST\_LIMITS.

LLENT                *See* LIST\_LIBRARY\_ENTRIES.

LMC                  *See* LIST\_MINI\_COMMANDS.

LOAD

LOAD invokes LOAD, the Prime linking loader for R-mode binary files.

***Format***

LOAD

***Usage***

The LOAD utility converts R-mode binary (object) files into R-mode executable runfiles. The R-mode binary files are produced by the FTN, PMA, and RPG compilers. To load V-mode or I-mode segmented code, use the BIND or the SEG command. For a complete discussion of the loaders and an example of the use of LOAD, see the *SEG and LOAD Reference Guide*.

## LOGIN

LOGIN admits a user into a system. Use LOGIN at a terminal to begin a work session on a local or remote system. Only one login at a time can be active on a terminal. After you log in, you are attached to a directory established by the System Administrator as your Initial Attach Point. This directory is your origin directory.

### Format

LOGIN [*userid*] [*password*] [-ON *system*] [-PROJECT *project-id*]

### Arguments and Options

|                            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>userid</i>              | A unique string of characters identifying a user established by the System Administrator when a new account is set up. The first character of the ID must be a letter, and the rest may be any combination of letters, digits, periods (.), underscores (_), and dollar signs (\$) up to a maximum of 32 characters. Case does not matter because PRIMOS converts lowercase letters to uppercase. Your user ID identifies you to your local system and to other systems that recognize IDs defined on your system. You can specify only an existing user ID. The user ID you specify is not required to be the name of your origin directory. |
| <i>password</i>            | Specifies your current login password.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| -ON <i>system</i>          | Specifies the remote system on which you wish to log in. <i>system</i> is usually the PRIMENET node name of that system.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| -PROJECT <i>project-id</i> | Specifies the project ID with which you are associated after login. Your System Administrator determines if you are required to give a project ID at login. Project IDs conform to the same naming rules as user IDs. If you issue the LOGIN command without specifying any arguments, PRIMOS prompts you for your user ID and, depending on the configuration of your system, for your password and project ID. If your System Administrator has defined a default project for you, you are not prompted for a project ID.                                                                                                                   |

### ***Login Passwords***

A login password is a string of characters, known only to you. The maximum and minimum password lengths are configured by the System Administrator (the maximum cannot exceed 16 characters). The string may contain any ASCII characters, with the exception of the following PRIMOS reserved characters:

`() {} [] <> ~ ' ! @ % ^ + = ' " , ; : ? | \ space rubout/delete`

How you specify your password also depends on how your system is configured. On some systems, you can include the password on the command line. Other systems prompt you for the password after you issue the LOGIN command, in which case your terminal does not echo the password. Use of passwords is strongly recommended but not required.

### ***Implicit Login***

If your System Administrator has enabled Implicit Login you do not need to enter the LOGIN command when first logging in to a system. You may enter your user name only. The system then prompts you for your password. With Implicit Login enabled you can still use the LOGIN command.

### ***Remote Login***

Remote logins are accomplished with the `-ON` option. The remote system must be linked to your system by PRIMENET, and you must have the following information:

- The name of an existing ID on the remote system
- The PRIMENET node name of the remote system
- Any required passwords or validation codes

You are logged in to your local system if you either omit the `-ON` option and give a user ID that is valid on the local system, or use the `-ON` option with the node name of your local system and give a user ID that is valid on the local system. If you specify an incorrect node name, the PRIMENET connection is broken.

### ***Login-Over-Login***

You can log in to a system as another user without having to first log out your present user ID. Using this login-over-login method, you can either log in as a different user on the present (local) system, or you can log in remotely to another system. In order to use login-over-login, the configuration directive LOGLOG must be enabled on your system.

To log in as a different user on your current system, issue the LOGIN command with the desired user ID as you would normally. In this case, PRIMOS logs out your original user ID and then prompts you to supply the password of the new user ID.



Having done so, you are now logged in as a different user with the same user number. If you subsequently wish to log in again on the present (local) system employing your original user ID, you may do so without having to log out first.

To log in as a user on a remote system, issue the LOGIN command with the desired user ID and *-ON system* option as you would normally. PRIMENET first logs you out of the local system, then connects you to the remote system. The remote system asks you for your password. You may be asked for a remote password as an additional security measure (see your System Administrator for more details). If you subsequently wish to log in to yet another system, or if you wish to log back in to the local system, you must log out of the remote system first.

### ***Changing Passwords***

At Revision 22.0 or later, PRIMOS may require you to change your password periodically. Your System Administrator sets your password lifetime; that is, the frequency with which your password must be changed. Also, at Revision 22.0 or later, the System Administrator can require all passwords to be computer generated. In this case, you receive a computer-generated password when your password expires or system passwords change. Only you and the System Administrator know your password. See your System Administrator for more details.

### ***Failed Login***

If your attempt at login is not successful, PRIMOS often displays a message explaining why you could not log in. If your attempts repeatedly fail, make a note of the message and see your System Administrator. When you log in, you are notified of the number of failed login attempts under your user ID since your last successful login. For example, if there had been three unsuccessful login attempts under a users ID since the last successful login, the following message would appear after a successful login:

```
Warning!  There were 3 failed attempts to login under this ID since
the last successful login.
```

After PRIMOS displays a message like the one above, it resets to 0 the count of failed login attempts since your last successful login.

---

**Note**

If you are sure that the failed login attempts were not yours, contact your System Administrator. If there were no failed login attempts, PRIMOS does not give you a message when you login.

---

### **LOGIN Examples**

**Example 1:** A login procedure on a local system that requires a project ID:

```
LOGIN
User-id? CARROLL
Password? JABBERWOCK (not echoed)
Project-id? LOOKING_GLASS

CARROLL (user 149) logged in Wednesday, 04 Dec 91 08:15:32.
Welcome to PRIMOS version 23.3.0
Copyright (c) 1992, Prime Computer, Inc.
Last login Wednesday, 03 Dec 91 06:51:32.
OK,
```

**Example 2:** A login procedure on a local system with Implicit Login enabled:

```
login prompt: CARROLL
Password? JABBERWOCK (not echoed)

CARROLL (user 149) logged in Wednesday, 04 Dec 91 08:15:32.
Welcome to PRIMOS version 23.3.0
Copyright (c) 1992, Prime Computer, Inc.
Last login Wednesday, 03 Dec 91 06:51:32.
OK,
```

**Example 3:** A remote login by user HARRY onto a system with the node name HQA:

```
LOGIN HARRY -ON HQA
PRIMENET 23.2.0 HQA
Password? TUDOR (not echoed)

HARRY (user 7) logged in Wednesday, 04 Sep 91 08:28:49
Welcome to PRIMOS version 23.3.0
Copyright (c) 1992, Prime Computer, Inc.
Last login Monday, Tuesday, 27 Aug 91 10:50:48

Enter validation code: ARAGON (not echoed)

OK,
```



# LOGOUT

LOGOUT ends a user's work session on the system.

LOGOUT is the last command you issue when giving up access to a system. During the LOGOUT procedure, PRIMOS performs the following functions:

- Closes all user files.
- Releases all devices assigned to your terminal, including Network Terminal Service (NTS) lines.
- Detaches you from your origin directory.
- Returns to the supervisor all segments that you were using.
- Displays a logout message at both your terminal and, optionally, at the supervisor terminal. The logout message includes the date and time of the logout and the total hours and minutes of your connect time, CPU time, and I/O time. For more details about time accounting, see the TIME command.
- If the line is an NTS line, performs a disconnect sequence.

### Format

```
LOGOUT [ { username } ] [ -BATCH
                                     -CHILD
                                     -NO_VERIFY
                                     -PHANTOM
                                     -QUERY ]
```

### Options

- username* Specifies your user ID.
- username* The decimal number of the user you want to logout. You can only logout users owned by your user ID. (You can use the STATUS USERS command 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. Wildcards are supported.
- BATCH* Selects and logs out only matching Batch phantoms.
- CHILD* Selects and logs out only matching child processes.
- NO\_VERIFY* Disables querying when you use wildcarding.
- PHANTOM* Selects and logs out only matching phantoms.
- QUERY* Enables querying when you are not using wildcarding. You may use this option with the -*BATCH*, -*CHILD*, and -*PHANTOM* options.

### ***Logging Yourself Out***

To log yourself out, issue the LOGOUT command without an argument, as in the following example:

```
OK, LO
```

```
MARLOWE (user 35) logged out Wednesday, 28 Aug 91 11:47:52.  
Time used: 00h 01m connect, 00m 02s CPU, 00m 01s I/O.
```

```
OK,
```

### ***Logging Another User Out***

If another user (such as a phantom or a process on another terminal) has the same login name as yours, you can specify a user number to log out that user. (Use the STATUS USERS or LIST\_USERS commands to display the names and numbers of users.) When you specify a user number, do not leave a space between the hyphen and the user number, as in the following example:

```
OK, LOGOUT -68
```

To logout several user processes with similar numbers, you may use wildcarding. If process numbers 30 through 39 belong to your login name, you can use the following command to log them all out:

```
OK, LO -3@@
```

You can logout all of your phantom user processes with the command:

```
OK, LO -PH
```

Add the -QUERY option if you wish to verify non-wildcarded selections (you are always queried when you use a wildcard).

### ***Forced Logout***

You may be logged out by the system if there has been no activity at your terminal for longer than the maximum inactive time allowed. When this force logout occurs, the normal logout message is preceded by the following message:

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

If you are logged out by a command issued at the supervisor terminal, the message forced logout is displayed, followed by the normal logout message.

.....  
LON

LON

LON controls phantom logout notification.

### **Format**

LON [ { -ON }  
          { -OFF } ]

### **Options**

When you initiate a phantom process, the phantom is logged in under your user ID. When the phantom is finished, PRIMOS automatically logs it out and notifies you. The following is an example of a phantom logout message:

```
Phantom 99: Normal logout at 10:33  
Time used: 00h 02m connect, 00m 07s CPU, 00m 10s I/O
```

The LON command allows you to defer these messages or to display them immediately.

### **Deferring Phantom Logout Messages**

To defer the reception of phantom logout notification messages, specify the `-OFF` argument, as follows:

```
OK, LON -OFF
```

Instead of displaying the messages at your terminal immediately, PRIMOS stores the messages until you issue a `LON -ON` command.

You may want to defer logout messages when you are using a text editor or running a COMOUTPUT file.

### **Receiving Phantom Logout Messages**

To display logout notification messages that have been deferred by a `LON -OFF` command, use the `-ON` argument, as follows:

```
OK, LON -ON
```

The accumulated messages are displayed all at once.

The command also sets your terminal to receive subsequent phantom logout messages immediately.

The `-ON` argument is the default when you log in. Logout notification is sent to your terminal unless you specifically prevent it by using the `-OFF` argument.

### **LON Example**

The following example illustrates the use of the `-OFF` and `-ON` options. After issuing the `LON -OFF` command, you initiate three phantoms. The user numbers that are assigned to the phantoms are displayed, as follows:

```
PHANTOM is user 110
PHANTOM is user 115
PHANTOM is user 107
```

When these phantoms are logged out, the logout notification messages are stored by PRIMOS. To display these messages, issue the `LON -ON` command, as follows:

```
OK, LON -ON
```

```
Phantom 110: Normal logout at 11:40
Time used: 00h 00m connect, 00m 00s CPU, 00m 00s I/O.
```

```
Phantom 107: Abnormal logout at 11:40
Time used: 00h 00m connect, 00m 03s CPU, 00m 01s I/O.
```

```
Phantom 115: Normal logout at 11:40
Time used: 00h 00m connect, 00m 04s CPU, 00m 01s I/O.
```

Phantoms and phantom logout are discussed in more detail in the *PRIMOS User's Guide*.

- LPAC            See LIST\_PRIORITY\_ACCESS.
- LQ             See LIST\_QUOTA.
- LRE            See LIST\_REGISTERED\_EPF
- LRID           See LIST\_REMOTE\_ID.
- LS             See LIST\_SEGMENT.
- LSR            See LIST\_SEARCH\_RULES.

.....



*Replace this page with the tab page labeled*

**M-R**

**M**

See MESSAGE.

**MAGNET**

MAGNET reads, writes, and copies magnetic tapes that are not in Prime format to and from Prime disk files.

When you issue the MAGNET command, the MAGNET subsystem is invoked. Using MAGNET, you can read files from disk or tape, write files to disk or tape, and create output spool files.

**Format**

MAGNET [options]

**Options**

- |            |                                                                                                                          |
|------------|--------------------------------------------------------------------------------------------------------------------------|
| -OPERATOR  | Displays mount or dismount messages at the supervisor terminal. Do not use with the -USER option.                        |
| -OVERWRITE | Overwrites a BRMS-labeled tape. Without this option, the only BRMS tapes MAGNET can use are those labeled as free tapes. |
| -SILENT    | Displays only severity 2 and 3 error messages.                                                                           |
| -USER      | Displays mount or dismount messages at the user's terminal. Do not use with the -OPERATOR option.                        |

---

**Note**

MAGNET is not an archiving or system backup facility. For archiving and system backup, use MAGSAV/MAGRST.

---

**Usage**

For complete information on the commands and operation of the MAGNET subsystem, see the *MAGNET User's Guide*.

















## MESSAGE

MESSAGE sends a message to another user and controls the receive state of a user's terminal.

The MESSAGE command performs the following functions:

- Sends a message to the operator or to other users, either on your system or on remote systems
- Changes the receive state of your terminal
- Checks the receive state of your terminal or of another user's terminal
- Allows the operator to communicate with all or specific users, or with operators of other systems
- Supports optional carriage return/line (CRLF) feed mapping.

The command format of MESSAGE depends on which of these functions you are performing.

### Format

$$\text{MESSAGE} \left[ \left\{ \begin{array}{l} \textit{userid} \\ \textit{-usernumber} \\ \textit{-ACCEPT} \\ \textit{-DEFER} \\ \textit{-DISPLAY} \\ \textit{-REJECT} \\ \textit{-HELP} \end{array} \right\} \right] [-\textit{ON system}] \left\{ \begin{array}{l} \textit{-NOW} \\ \textit{-STATUS} [\{\textit{arguments}\}] \end{array} \right\}$$

### Arguments and Options

- userid*                      The name under which the recipient is logged in. If two or more users are logged in under *user-id*, all of them receive the message.
- usernumber*                The number of a specific terminal. You must supply the number with an initial hyphen (for example, -32). If you specify a user number, only the terminal with that number receives the message. (Use either the STATUS USERS or the MESSAGE -STATUS command to display a list of all user IDs and user numbers currently logged in.)

■ ■ ■ ■ ■ ■ ■ ■ ■ ■  
**MESSAGE**

- |                                     |                                                                                                                                                                                                                                                   |
|-------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-ACCEPT</b>                      | Allows reception of all messages. This is the default state when you log in.                                                                                                                                                                      |
| <b>-DEFER</b>                       | Allows reception only of deferred messages. Messages sent with the <b>-NOW</b> option are rejected; other messages are displayed when you return to command level. This option prevents others from initiating a TALK session with your terminal. |
| <b>-DISPLAY</b>                     | Displays the current configuration of the <b>MESSAGE</b> command. Reports the current maximum message length and whether or not carriage return/line feed (CRLF) mapping is enabled.                                                              |
| <b>-NOW</b>                         | Displays the message immediately on the recipient's screen. If you do not specify <b>-NOW</b> , the message is displayed when the recipient returns to <b>PRIMOS</b> command level.                                                               |
| <b>-ON <i>system</i></b>            | Directs <b>MESSAGE</b> to act on a remote system named <i>system</i> . The remote system must be linked to your own system through a network.                                                                                                     |
| <b>-REJECT</b>                      | Rejects all messages. This option prevents others from initiating a TALK session with your terminal.                                                                                                                                              |
| <b>-STATUS [{<i>arguments</i>}]</b> | Lists the receive state of all users on your system. See the <b>-STATUS</b> Option section below for an explanation of the arguments.                                                                                                             |
| <b>-HELP</b>                        | Displays the command syntax and options.                                                                                                                                                                                                          |

### ***Sending Messages***

Use **MESSAGE** to communicate with another user (either local or remote) or with the operator.

#### **Sending Messages to Other Users on Your System**

To send a one line message to another user on your system, use the following format:

**MESSAGE** { *userid*  
          -*usernumber* } [**-NOW**]

After you issue the **MESSAGE** command, the cursor moves to the beginning of the next line so that you can enter your message. The maximum number of characters in the message is 80 or 512 depending on how your System Administrator has configured the command.

When displayed on the recipient's terminal, the message is prefaced with your user ID and the time the message was sent. This format is

**\*\*\**user-id* (user *n* on *systemname*) at *hh:mm* *text-of-message***

*user-id* is your login name, *n* is your user number, *systemname* is the name of your system, and *hh:mm* is the time of day in hours and minutes.

In the following example, user DONNA on system SYS.B sends a message to user HOPE on the same system:

```
OK, MESSAGE HOPE -NOW
Marketing meeting has been changed to 3:00 PM tomorrow.
OK,
```

The message is displayed at HOPE's terminal as follows:

```
*** DONNA (user 15 on SYS.B) at 13:54
Marketing meeting has been changed to 3:00 PM tomorrow.
```

### **Sending a Message to the Operator**

To send a message to the supervisor terminal, use the following format:

**MESSAGE [-1] [-NOW]**

### **Sending Messages to Remote Users**

If your local system is attached to a network, use the following format to send a message to a user on a remote system:

**MESSAGE { *userid*  
-*usernumber* } -ON *system* [-NOW]**

*system* is the PRIMENET nodename of the remote system to which you are sending the message.

*userid* is the user ID or user number of the recipient.

The rest of the remote MESSAGE procedure is identical to the local procedure, as is the operation of the -NOW option. If used, -NOW must be the last word on the command line.

For example, to send a message to user QUEEN on a system with PRIMENET node name SYS.E, issue the following command:

```
OK, MESSAGE QUEEN -ON SYS.E -NOW
```

Then enter the message when the cursor moves to the beginning of the next line.

### Using the Carriage Return/Line Feed (CRLF) Feature

The CRLF feature enables you to send multiline messages. Special characters are used to create multiline messages, as follows:

- %/** 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.
- %%** 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 will appear just as you typed it (without the % signs).

When you want to include a % sign, use the combination %% , for example:

```
OK, M BOB -NOW
I am feeling 100%% better!
```

The recipient sees:

```
*** SALLY (user 45 on ARTEMIS ) at 12:38
I am feeling 100% better!
```

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.
```

---

**Note**

If you send a message from the local supervisor terminal to a local user who is logged in remotely to another system, you must specify the `-NOW` option. If the `-NOW` option is not specified, the message is not sent to the intended recipient, and an error message is returned to the supervisor terminal.

---

### ***-STATUS Option: Checking the Receive State of a Terminal***

Use the `-STATUS` option to check the receive state of a user's terminal (including your own). The `-STATUS` output lists the specified user's name, number, and receive state. Use the `-ON` option to determine the receive state of a user on a remote system.

The command format for checking the receive state of another terminal is

```
MESSAGE -STATUS [ { userid
                   { usernumber
                     ME
                   }
                 ]
```

#### ***-STATUS Arguments***

- |                   |                                                                                                                                 |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------|
| <i>userid</i>     | Lists the receive state of all users on your system with the login name of <i>userid</i> .                                      |
| <i>usernumber</i> | Lists the receive state of the terminal with the number <i>usernumber</i> . Do not use an initial hyphen before the user number |
| ME                | Lists the receive state of your terminal. You cannot use the <code>-ON</code> option with <code>MESSAGE -STATUS ME</code> .     |



## Changing the Receive State of a Terminal

The MESSAGE command also controls the flow of messages by setting the receive state of a terminal. To change the receive state, use the command format

```
MESSAGE { -ACCEPT
          -DEFER
          -REJECT }
```

Setting your terminal's receive state to defer or reject messages is necessary if you do not want messages to interrupt a terminal session. For example, you might change your receive state just before printing a file because a message received during printing would also be printed with the file's contents.

You cannot send deferred or immediate messages when your receive state has been changed by a MESSAGE -REJECT command. Nor can you send an immediate message (with the -NOW option) when your receive state has been changed by a MESSAGE -DEFER command. The reason for these restrictions is that a recipient could not respond to your message.

## Error Messages

Listed below are some error messages you may receive when using MESSAGE. Each error message is followed by an explanation.

Improper command usage or arguments. (MESSAG)

The command contains an invalid use of options, such as -ON and -STATUS ME.

Option "-option" not recognized by this command. (messag\_)

The option specified by -option is invalid.

\*\*\*Unknown addressee.

The command specifies the user ID or user number of a user who is not logged in.

\*\*\*User n not receiving now. (MESSAG)

This message means one of two things: if you sent a message with the -NOW option, the recipient's receive state is either DEFER or REJECT; if you sent a message without the -NOW option, the recipient's receive state is REJECT.

\*\*\*Unknown node (PRIMENET). (MSG\$)

You specified a system name unknown to your network.



• • • • •  
**MONITOR\_NET**

\*\*\*User n busy, please wait. (MSG\$)  
User n already has a deferred message waiting. Only one deferred message is allowed.

\*\*\*Requires -ACCEPT enabled.  
You must issue a MESSAGE -ACCEPT command before sending messages.

\*\*\*Requires -ACCEPT or -DEFER enable.  
You must issue a MESSAGE -ACCEPT or MESSAGE -DEFER command before sending messages.

See also TALK.

**MONITOR\_NET**

MONITOR\_NET invokes the PRIMENET Information Monitoring (PIM) utility. PIM monitors a system's network activity: ring, virtual circuits, sync lines, and LAN3xxs. The information includes performance, traffic, and status data.

**Format**

MONITOR\_NET [*options*]

**Usage**

For details on the operation and options of MONITOR\_NET, see the *Operator's Guide to Prime Networks*.



### **-REPORT Suboptions**

- report-pathname*** Specifies the file where output is saved. If *report-pathname* is omitted, the report is displayed at the terminal.
- DETAIL** Produces a detailed, entrypoint-by-entrypoint report that may be either system-wide or per-user.
- USER [*userid*]** Produces a report that shows dynamic-linking statistics for the whole system and for the user specified by *userid*.
- NEW\_FILE [*new-ENTRY\$-pathname*]** Produces a new, optimized ENTRY\$ search rules file that may be either system-wide or per-user. If *new-ENTRY\$-pathname* is omitted, a file named NEW.ENTRY\$.SR is written in the current directory.

### **The Per-user Report**

MONITOR\_SEARCH\_RULES can expand a system-wide report to show individual linkage counts for a specified user.

The per-user report can be useful to a programmer when tuning an application. The programmer watches the linkage counts as a test program runs. This facility lets the programmer create an ENTRY\$ search list tuned specifically for a program or a suite of programs. Besides improving performance, such tuning may save a user (or system) from running out of segments because unused libraries were mapped in and searched before the needed ones were found in the default, nonoptimal ENTRY\$ list search. Watching what libraries a program links to can also help in verifying that there are no hidden errors (such as linking to a subroutine in an earlier revision library or finding two subroutines of the same name in two separate libraries) or potential performance bottlenecks (such as making frequent attempts to find unknown routines via the CKDYN\$ routine or a LINKAGE\_FAULT\$ on unit, or linking to a large EPF library for a simple but important function that might be more efficiently used as in-line code).

Per-user reports are available on any user by specifying the appropriate user number as an argument to the -USER suboption. Per-user reports are sorted on the count for the specified user. Here is a sample per-user report:

OK, MONITOR\_SEARCH\_RULES -REPORT -USER 1

[MONITOR\_SEARCH\_RULES Rev. 23.3.0 Copyright (c) 1992, Prime Computer, Inc.]  
 [Serial #10b (Propulsions Engineering)]

Dynamic link monitor is installed as static library 1;  
 166 entries in use (out of 3275 max); 4 users being monitored individually.

| Library                                    | Total Count | User 1 Count |
|--------------------------------------------|-------------|--------------|
| <SYSTEM>LIBRARIES*>SYSTEM_LIB\$PRC.RUN     | 1893        | 169          |
| <SYSTEM>LIBRARIES*>SYSTEM_LIB\$PRG.RUN     | 260         | 90           |
| <SYSTEM>LIBRARIES*>SIT_LIBRARY.RUN         | 81          | 53           |
| Segment 2126                               | 98          | 27           |
| <SYSTEM>LIBRARIES*>APPLICATION_LIBRARY.RUN | 500         | 15           |
| Segment 2026                               | 12          | 8            |
| <SYSTEM>LIBRARIES*>SP\$LIB.RUN             | 5           | 5            |
| <SYSTEM>LIBRARIES*>TRANS_LIB\$PRC.RUN      | 125         | 2            |
| Segment 2342                               | 32          | 1            |
| <SYSTEM>LIBRARIES*>ECL\$LIB.RUN            | 80          | 0            |
| <SYSTEM>LIBRARIES*>FTN_LIBRARY.RUN         | 9           | 0            |
| Segment 4377                               | 109         | 0            |
| <SYSTEM>LIBRARIES*>DSMLIB.RUN              | 1           | 0            |
| <SYSTEM>LIBRARIES*>DSMLIB.RUN              | 1           | 0            |
| <br>                                       |             |              |
| Routines from Unknown Libraries            | Total Count | User 1 Count |
| U\$INVK                                    | 2           | 1            |

Erroneous search rules :

LIBRARIES\*>OLD\_LIB.RUN            Not found

OK,

### ***The Detailed Report***

MONITOR\_SEARCH\_RULES produces detailed reports showing complete subroutine-by-subroutine breakdowns of library usage, both for the whole system or for an individual user. The detailed report shows the name of a subroutine, its link address, and its linkage count. In cases where the subroutine linkage is to an Entry Control Block (by far the most common case), the segment containing the procedure code for the routine is also displayed; other linkages are either to short-called routines, which start at the link address itself, or occasionally to data other than a subroutine. The detailed report concludes with the usual library report, which is in fact a sum of the per-user routine reports, by library.

MONITOR\_SEARCH\_RULES

The -DETAIL option produces a detailed system-wide report. Use with the -USER suboption to produce a detailed per-user report. The following is an example of a detailed per-user report:

OK, MONITOR\_SEARCH\_RULES -REPORT -USER 1 -DETAIL

[MONITOR\_SEARCH\_RULES Rev. 23.3.0 Copyright (c) 1992, Prime Computer, Inc.]  
 [Serial #10b (Propulsions Engineering)]

Dynamic link monitor is installed as static library 1;  
 166 entries in use (out of 3275 max); 4 users being monitored individually.

| Routine Name                               | Address of Link | Proc Seg | Total Count | User 1 Count |
|--------------------------------------------|-----------------|----------|-------------|--------------|
| CLOSSA                                     | 4377/170040     | 4372     | 2           | 1            |
| P\$DATE                                    | 4407/7144       | 4404     | 8           | 0            |
| .                                          | .               | .        | .           | .            |
| SP\$ADDRESS                                | 4314/152126     | 4324     | 2           | 2            |
| LCKSF                                      | 6006/15634      | 2126     | 2           | 1            |
| P\$PCYC                                    | 4404/30762      |          | 5           | 2            |
| Library                                    |                 |          | Total Count | User 1 Count |
| <SYSTEM>LIBRARIES*>SYSTEM_LIB\$PRC.RUN     |                 |          | 1893        | 169          |
| <SYSTEM>LIBRARIES*>SYSTEM_LIB\$PRG.RUN     |                 |          | 260         | 90           |
| <SYSTEM>LIBRARIES*>SIT_LIBRARY.RUN         |                 |          | 81          | 53           |
| Segment 2126                               |                 |          | 98          | 27           |
| <SYSTEM>LIBRARIES*>APPLICATION_LIBRARY.RUN |                 |          | 500         | 5            |
| Segment 2342                               |                 |          | 32          | 1            |
| <SYSTEM>LIBRARIES*>ECL\$LIB.RUN            |                 |          | 80          | 0            |
| <SYSTEM>LIBRARIES*>FTN_LIBRARY.RUN         |                 |          | 9           | 0            |
| Segment 4377                               |                 |          | 109         | 0            |
| <SYSTEM>LIBRARIES*>DSMLIB.RUN              |                 |          | 1           | 0            |
| <SYSTEM>LIBRARIES*>DSMLIB.RUN              |                 |          | 1           | 0            |
| Routines from Unknown Libraries            |                 |          | Total Count | User 1 Count |
| U\$INVK                                    |                 |          | 2           | 1            |
| Erroneous search rules :                   |                 |          |             |              |
| LIBRARIES*>OLD_LIB.RUN                     |                 |          | Not found   |              |
| OK,                                        |                 |          |             |              |

## The New ENTRY\$ Search Rules File

Finally, the `-NEW_FILE [new_ENTRY$_pathname]` suboption produces a new version of the currently active ENTRY\$ search rules file, sorted into optimal order. The `-NEW_FILE` suboption must be used with the `-REPORT` option and can be used with any of the other `-REPORT` suboptions. If `new_ENTRY$_pathname` is omitted, the file `NEW.ENTRY$.SR` is written in the current directory. The `SET_SEARCH_RULES` command activates this file as a new ENTRY\$ search list.

The new, optimized ENTRY\$ search rules file has a header line stamped with the date and time. The header contains a message stating that the file was reorganized by `MONITOR_SEARCH_RULES`. After the header, any comment lines at the beginning of the old file are copied to the new file. Search rules follow the comment lines, sorted into optimal order: the most frequently used appear first. The rules are sorted for the whole system or, if the `-USER` suboption is used, for an individual user.

Lines describing EPF libraries that do not show up in a `MONITOR_SEARCH_RULES` report (that is, they effectively have a count of zero) are copied from the old file to the end of the new file in the same order as that of the old file. This behavior occurs even if an error (such as a nonexistent file or insufficient access rights) is detected, because the problem may be corrected later or may apply only to the user running `MONITOR_SEARCH_RULES` and not to all users.

`MONITOR_SEARCH_RULES` attempts to place the keyword `-SYSTEM` optimally by adding together the number of dynamic links to libraries named specifically in the file and subtracting that number from the total number of dynamic links. Rules that are definitely invalid – such as a misspelled keyword (for example, `-SYSYSTEM`) or a meaningless function (for example, `[RUBBISH]`) or a directory instead of an EPF (for example, `[HOME_DIR]` instead of `[HOME_DIR]>EPF.RUN`) – are not preserved in the new file.

Here is an example of using the `-NEW_FILE` suboption:

```
OK, MONITOR_SEARCH_RULES -REPORT MSR.REPORT -USER 59 -NEW_FILE  
[MONITOR_SEARCH_RULES Rev. 23.3.0 Copyright (c) 1992, Prime Computer, Inc.]
```

```
Creating new ENTRY$ search rule file ...  
New ENTRY$ search rule file NEW.ENTRY$.SR created.
```

For more information, see the *Advanced Programmer's Guide II: File System*

See also `EXPAND_SEARCH_RULES`, `LIST_SEARCH_RULES`, and `SET_SEARCH_RULES`.

■ ■ ■ ■ ■ ■ ■ ■ ■ ■

## MPACK

## MPACK

MPACK invokes an interactive MIDASPLUS utility for packing and restructuring MIDASPLUS data and index subfiles. The utility recovers space left by deleted data subfile entries, unlocks locked data records, and restructures the data subfile to correspond to the order of entries in the primary index.

### **Format**

MPACK

### **Usage**

See the *MIDASPLUS User's Guide*.

## MPLUSCLUP

MPLUSCLUP performs cleanup for MIDASPLUS after an abnormal program termination.

The cleanup procedure closes any open MIDASPLUS files and releases the internal resources held by the MIDASPLUS user. The resources include buffers, internal MIDASPLUS locks, and record locks.

### **Format**

MPLUSCLUP [*options*]

### **Usage**

See the *MIDASPLUS User's Guide*.

## MRGF

MRGF merges as many as five ASCII files into a single output file. *file1* is the original file (that is, the common ancestor of *file2* through *file5*). MRGF assumes that changes have been made in *file1* to produce the other files.

### Format

```
MRGF file1 file2 [file3 . . . file5] -OUTF outfile [options]
```

### Arguments

|                          |                                                                                                                                                                              |
|--------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>file1</i>             | The original file (that is, the common ancestor of <i>file2</i> through <i>file5</i> ). MRGF assumes that changes have been made in <i>file1</i> to produce the other files. |
| <i>file2 . . . file5</i> | Files resulting from edits to <i>file1</i> . MRGF assumes that changes have been made in <i>file1</i> to produce these files.                                                |

### -OUTF Argument and Suboptions

The -OUTF keyword and its *outfile* argument are mandatory.

|                         |                                                                                                                                                                                                                                                                                                                                             |
|-------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>outfile</i>          | Specifies the pathname of the output file.                                                                                                                                                                                                                                                                                                  |
| -BRIEF                  | Does not display the lines that differ between the files, only the file identification letters and the line numbers of the differing lines.                                                                                                                                                                                                 |
| -FORCE                  | Does not ask you to resolve conflicts between files, but automatically writes the changes in <i>file2</i> to <i>outfile</i> . Edit mode is not entered and manual selection is not allowed.                                                                                                                                                 |
| -MINL <i>number</i>     | Sets the minimum number of lines that must match (after a difference in the files is found) in order to resynchronize all file merging. (The default is 3.)                                                                                                                                                                                 |
| -REPORT <i>pathname</i> | Writes to a file (named <i>pathname</i> ) the lines of text that differ between files. Resolvable differences (that is, those resolved automatically by MRGF) are not displayed at the terminal. User-resolvable differences (that is, those requiring manual selection by you) are displayed as well as being written to <i>pathname</i> . |



## MRGF Operation

If you invoke MRGF without options, *file1* is compared line by line with each of the other files (*file2* through *file5*). Lines that match in all files are copied into *outfile*.

When MRGF finds differences between files, it attempts to resynchronize the files. Rematching is complete only when a minimum number of lines match in all files. The default of this minimum number is 3, but you can change it with the `-MINL` option.

After resynchronization is complete, the selection of the lines to be written to the output file takes place. If only one file differed from *file1*, the changes in that file are copied into *outfile*. If all files differed identically from *file1*, those changes are also copied. This type of selection is called **automatic selection** because you are not requested to verify the selections.

If, however, conflicting changes are found in several files (or if only one file is being merged with the original), MRGF prints the differing lines at the terminal. Each line is prefaced with an identification letter (letters A through E for *file1* through *file5*, respectively) and a line number.

MRGF then enters Edit mode to allow you to resolve the conflict. Select the lines you want written to *outfile* by entering one of the MRGF subcommands listed below. This type of selection is called **manual selection**. After selection (either automatic or manual) is completed, the line-by-line comparison continues.

### Edit Mode

As explained above, when MRGF encounters differences among the files that it cannot resolve, it displays the differing lines and then enters Edit mode so that you can manually select the line that is to be written to *outfile*. There is no prompt for Edit mode. Instead, MRGF prints the word "EDIT" and leaves the cursor at the beginning of the next line.

Use one of the MRGF subcommands listed below to make your selection. Issue the subcommands either in uppercase or lowercase letters.

| <i>Subcommand</i> | <i>Function</i>                                                                                                                                               |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|
| A                 | Inserts all of the differing lines from <i>file1</i> into <i>outfile</i> .                                                                                    |
| B                 | Inserts all of the differing lines from <i>file2</i> into <i>outfile</i> .                                                                                    |
| C                 | Inserts all of the differing lines from <i>file3</i> into <i>outfile</i> .                                                                                    |
| D                 | Inserts all of the differing lines from <i>file4</i> into <i>outfile</i> .                                                                                    |
| E                 | Inserts all of the differing lines from <i>file5</i> into <i>outfile</i> .                                                                                    |
| An                | Inserts line <i>n</i> of <i>file1</i> into <i>outfile</i> .                                                                                                   |
| Bn                | Inserts line <i>n</i> of <i>file2</i> into <i>outfile</i> . (Similarly for <i>file3</i> through <i>file5</i> ; for example, Dn to insert into <i>file4</i> ). |

|                        |                                                                                                                                                                         |
|------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>A<sub>m,n</sub></i> | Inserts lines <i>m</i> through <i>n</i> of <i>file1</i> into <i>outfile</i> . (Similarly for <i>file2</i> through <i>file5</i> ; for example, <i>B<sub>m,n</sub></i> .) |
| PA                     | Prints all of the differing lines from <i>file1</i> . (Similarly for <i>file2</i> through <i>file5</i> ; for example, PC.)                                              |
| PAn                    | Prints line <i>n</i> of <i>file1</i> . (Similarly for <i>file2</i> through <i>file5</i> ; for example, PD <i>n</i> .)                                                   |
| PAm, <i>n</i>          | Prints lines <i>m</i> through <i>n</i> of <i>file1</i> . (Similarly for <i>file2</i> through <i>file5</i> ; for example, PEm, <i>n</i> .)                               |
| OOPS                   | Undoes all previous editing for this discrepancy.                                                                                                                       |
| <i>blank line</i>      | Enters Input mode.                                                                                                                                                      |
| GO                     | Terminates editing and proceeds with the merge.                                                                                                                         |
| QUIT                   | Terminates editing, closes all files, and exits from MRGF.                                                                                                              |

### Input Mode

During Edit mode, you can insert new text into the output file by entering Input mode. To enter Input mode, enter a blank line (that is, press the RETURN key without typing anything). When MRGF enters Input mode, it prints the word "INPUT" and leaves the cursor at the beginning of the next line.

All text entered in Input mode is inserted into *outfile*. To return to Edit mode from Input mode, enter a blank line.

You cannot perform text editing on text entered in Input mode. Tab character expansion is not performed on lines entered in Input mode.

### Line Length

The MRGF command operates on compressed lines of any length and assumes that files of common ancestry contain lines compressed in identical fashion. It is possible, however, for a mismatch to occur between two lines that appear identical but were compressed differently.

### MRGF Example

This example illustrates the operation of MRGF by merging the following three files:

| FOX.1 | FOX.2    | FOX.3   |
|-------|----------|---------|
| The   | The      | The     |
| quick | quick    | quick   |
| brown | red      | brown   |
| fox   | fox      | fox     |
| jumps | jumps    | jumps   |
| over  | over     | over    |
| the   | the      | the     |
| lazy  | sleeping | snoring |
| dog   | dog      | dog     |

FOX.1 is the original file (*file1*). FOX.2 is *file2* and FOX.3 is *file3*. The three files will be merged to produce a fourth file named FOX.4. The merge process is illustrated below:

```
OK, MRGF FOX.1 FOX.2 FOX.3 -OUTF FOX.4  
[MRGF Rev. 23.3.0 Copyright (c) 1992, Prime Computer, Inc.]
```

```
A8      lazy  
CHANGED TO  
B8      sleeping  
BUT ALSO CHANGED TO  
C8      snoring  
EDIT.
```

```
MERGE FINISHED.  
1 MANUAL CHANGE.  
1 AUTOMATIC CHANGE AS FOLLOWS:  
  1 FROM FILE B
```

OK,

The merged output file (FOX.4) contains the following text:

```
The  
quick  
red  
fox  
jumps  
over  
the  
sleeping  
dog
```

The one manual change occurred when the user issued the command B, which specified that line B8, rather than lines A8 or C8, was to be inserted into FOX.4. MRGF made one automatic change, which was to insert the word *red* from FOX.2 into FOX.4. MRGF made this choice because FOX.2 differed from FOX.1, whereas FOX.3 was identical.

If you use the `-FORCE` option in this example, the same merged output would be produced. However, line B8 from FOX.2 would be inserted automatically because FOX.2, which is *file2*, is the preferred file. You would not be queried in this instance.

### **Purpose of MRGF**

The MRGF command, like the CMPF command, is invaluable for parallel software development. The MRGF command allows automated merging of program changes, and eliminates the need for editing of programs when two (or more) sets of changes made to a program are to be combined. However, you should check the resulting merged output file carefully before you use it.

MRGF is especially useful for combining changes to a program that have been made in parallel by several programmers. It can also be useful for distributing software changes to one or more sites, or to one or more persons.

See also CMPF.

## **NETLINK**

NETLINK connects you to another networked system.

The NETLINK command invokes the NETLINK utility which allows you to make as many as six remote login connections to other networked systems. The other systems must be connected to your system through PRIMENET or through a non-Prime network, such as a Packet Switched Data Network (PSDN). If your system has a PSDN link, NETLINK can access any system in the network, including non-Prime systems.

When you issue the NETLINK command without options, the utility enters Command mode and displays an at sign (@) as a prompt. You must then issue a NETLINK command to continue. Among the tasks NETLINK commands allow you to accomplish are the following:

- Connect to and use a maximum of six different remote systems at the same time
- Transfer files across networks
- Set data transmission characteristics
- Display the status of your connection
- Specify the various fields of the connect packet when data transmission characteristics of a foreign system differ from that of a Prime system

The following section describes the basic procedure for using NETLINK. For detailed information on the operation, options, and error messages of NETLINK, see the *User's Guide to Prime Network Services*.

## Format

NETLINK [*options*]

## Using NETLINK

The basic procedure for using NETLINK is described below:

1. Enter NETLINK's Command mode by issuing the NETLINK command without options. When Command mode is entered, the @ prompt appears.

Connect to the remote system by issuing one of the following two command formats:

C *address*  
NC *address*

*address* is either the host address assigned by the Packet Switched Data Network or a PRIMENET system name. Use C to connect to another system in a PRIMENET network or to reverse the charges over a PSDN, as in the following examples:

@ C SYS14

@ C 2080:12300011

Use NC when you do not want to reverse the charges. (NC is required for many international calls.)

When the connection is established, the following message is displayed:

*address* Connected

*address* is the address you specified with the C or the NC command.

2. Log in to the system as you would normally, entering any validation codes as required.
3. After you finish the terminal session, log out as you would normally. The message *address* Disconnected appears. (*address* is the address you specified in step 2.) NETLINK Command mode is reentered and the @ prompt is displayed.

4. You may now connect to another site or return to PRIMOS. To return to PRIMOS, enter the QUIT command. To connect to another site, repeat step 2 with the appropriate address.

### NETLINK Example

In the following example of a basic terminal session, user NOVAK connects to the remote system named SYS.D, which requires the validation code CURVE:

```
OK, NETLINK
[NETLINK Rev. 23.3.0 Copyright (c) 1992, Prime Computer, Inc.]
```

```
@ C SYS.D
```

```
SYS.D Connected
PRIMENET 23.2.0 SYS.D
LOGIN NOVAK
Password? TANAGER
```

```
NOVAK (user 48) logged in Wednesday, 28 Aug 91 15:14.52.
Welcome to PRIMOS version 23.3.0
Copyright (c) 1992, Prime Computer, Inc.
Last Login Tuesday, 27 Aug 91 18:33.21.
```

```
Enter validation code: CURVE
```

```
OK,
.
.      /* continue with normal terminal session
.
OK, LOGOUT
```

```
NOVAK (user 48) logged out Wednesday, 28 Aug 91 17:34.44.
Time used: 02h 20m connect, 00m 12s CPU, 00m 02s I/O.
```

```
Wait...
```

```
SYS.D Disconnected
@ QUIT
```

```
OK,
```



## NTS\_LIST\_ASSOCIATE

NTS\_LIST\_ASSOCIATE displays the current associations (if any) of assignable NTS PRIMOS line numbers with their corresponding LTS300 name or line numbers.

### Format

NTS\_LIST\_ASSOCIATE [ { -LINE *n*  
 -LTS\_NAME *name* [-LTS\_LINE *number*]  
 -HELP } ]

### Options

- LINE *n*                      Displays the association for a specific PRIMOS line number *n*.
- LTS\_NAME *name*              Displays the associations for an LTS300 named *name*.
- LTS\_LINE *number*            Displays the association for a specific line on the LTS300 named with the -LTS\_NAME option.
- HELP                          Displays command syntax and options.

NTS\_LIST\_ASSOCIATE, without options, displays all lines on all available LTS300s.

### Usage

NTS\_LIST\_ASSOCIATE is used with the Prime Network Terminal Service (NTS). For more information, see the *System Administrator's Guide, Volume II: Communication Lines and Controllers*.



■ ■ ■ ■ ■ ■ ■ ■ ■ ■

## NUMBER

## NUMBER

NUMBER numbers or renumbers statements in a BASIC program.

For NUMBER to work correctly, the BASIC commands must be in uppercase characters. The maximum line length that NUMBER can handle is 75 characters, plus 5 for the line number. Lines longer than 75 characters are truncated. If a statement is not numbered, NUMBER numbers it in order of its occurrence in the program.

### *Format*

NUMBER

### *Using NUMBER*

When NUMBER is invoked, it displays the following prompt with the cursor at the beginning of the next line:

INTREENAME, OUTREENAME, START, INCR,

The meanings of the four parameters are

| <i>Parameter</i> | <i>Meaning</i>                                                                                                                                                                                                                         |
|------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| INTREENAME       | The pathname of the input file. The input file is the file that contains the BASIC program with the statements to be renumbered.                                                                                                       |
| OUTREENAME       | The pathname of the output file. If no output filename is specified, the output is written to the input file.                                                                                                                          |
| START            | The number to be given to the first statement. The number can be from 1 to 9999, inclusive. If you omit START, the value 1 is assumed. You must specify a value for START if you are going to specify a value for INCR.                |
| INCR             | The number by which each subsequent statement is to be incremented. The incremental value can be from 1 to 9999, inclusive. If you omit INCR, the value 1 is assumed. To specify a value for INCR, you must specify a value for START. |

Only the first parameter (INTREENAME) is mandatory. If you enter one or more subsequent parameters, you must enter them all on the same line, separating each parameter with a comma or a space. Filenames can be entered either in uppercase or lowercase.

### Example of Using NUMBER

This example uses the following BASIC program, which is stored in a file named NUM.IN:

```
OK, SLIST NUM.IN

11 PRINT 'Enter the first number: '
12 INPUT A
23 PRINT 'Enter the second number: '
36 INPUT B
40 LET C = A * B
57 PRINT 'The answer is ',C
58 IF C <> 0 THEN 11
99 END
OK,
```

The sequence of commands shown below rennumbers the program. The first statement starts at 10, and the number of each subsequent statement is incremented by 5. The renumbered program is stored in a file named NUM.OUT:

```
OK, NUMBER
INTREENAME, OUTTREENAME, START, INCR,
NUM.IN NUM.OUT 10 5
OK,
```

The program is renumbered as follows:

```
OK, SLIST NUM.OUT
 10 PRINT 'First Number: '
 15 INPUT A
 20 PRINT 'Second Number: '
 25 INPUT B
 30 LET C = A * B
 35 PRINT 'Answer is ',C
 40 IF C <> 0 THEN 10
 45 END
OK,
```

### Error Messages

Some of the NUMBER error messages are shown below. Each error message is followed by an explanation.

```
DECODE FORMAT/DATA MISMATCH
```

A non-numerical value was specified for START or INCR.

• • • • •  
OAS

n DUP LINE NUMBER

There is more than one statement with the line number n.

Illegal name. BAD PARAMETERS

An illegal filename was specified.

INPUT FILE NULL

The specified input file is empty.

LINE NUMBER OVERFLOW

A new line number is greater than 9999. This error occurred because the START and/or the INCR parameters were too large.

MEMORY OVERFLO

There is not enough memory to contain a map of line numbers.

Not found. BAD PARAMETERS

The specified input file does not exist.

OAS

OAS starts up the Prime Office Automation System (OAS).

### ***Format***

OAS

### ***Usage***

The OAS command logs you in to the OAS Master Function Selection, which allows you access to all Word Processing and Management Communications and Support options.

For detailed information, see the following documents:

- *OAS Word Processing Guide (PT65)*
- *OAS Word Processing Guide (PST 100 and PT45)*
- *OAS Management Communications and Support Guide*
- *Using OAS on the PT25 Management Workstation*

# OPEN

OPEN opens a file on a specified file unit.

## Format

OPEN [*pathname*] *funit* *key*

## Arguments

- funit*      A number from 1g to 177g, inclusive, that specifies the file unit on which the file is opened. (Under special circumstances, PRIMOS may allow you to open file units larger than 177g.) To name the file, specify a *pathname* before *funit*.
- key*         An octal number that determines the status of the file when opened. The number is the logical OR of three octal numbers that specify the New File, Action, and Reference keys. The New File key represents the type of file. The Action and Reference keys specify the action to be taken with the file. The valid values of these keys are listed in the next section.

## Values for the *key* Argument

| <i>New File Key</i> | <i>Type of File Created</i>                            |
|---------------------|--------------------------------------------------------|
| 0000                | Sequential file (SAM)                                  |
| 2000                | Direct access file (DAM)                               |
| 4000                | SAM segment directory                                  |
| 6000                | DAM segment directory                                  |
| 10000               | Directory (avoid this; use the CREATE command instead) |

| <i>Action Key</i> | <i>Action Taken</i>                             |
|-------------------|-------------------------------------------------|
| 0001              | Open for reading                                |
| 0002              | Open for writing                                |
| 0003              | Open for reading and writing                    |
| 0004              | Close                                           |
| 0005              | Delete (avoid this; use DELETE command instead) |
| 0006              | Test for existence                              |



## ORIGIN

ORIGIN returns you to your origin directory from your current directory.

### **Format**

ORIGIN

### **Usage**

Your **origin directory** (also called your Initial Attach Point) is the directory to which you are first attached when you log in. Your **current directory** is the directory to which you are attached at any given moment. (Your origin directory is also your current directory when you are attached to it.) When you use the ATTACH command to attach to another directory, that directory becomes your current directory. Use the ORIGIN command to return to your origin directory from your current directory. ORIGIN does nothing if issued from your origin directory.

For more information on ORIGIN, see the *PRIMOS User's Guide*.

## OWLDSC

OWLDSC invokes the OWL interface program, which allows an OWL-1200 terminal to emulate an IBM 3277 Model 2 Display Station on systems where DPTX/DSC is running.

### **Format**

OWLDSC [-FAST] [-NOLOCK] [-REPORT]

### **Usage**

For details, see the *Distributed Processing Terminal Executive Guide*.

.....  
P

P See PM(Post Mortem).

## PASCAL

PASCAL loads the Prime PASCAL compiler and compiles an object program from an ASCII file containing PASCAL source code.

### **Format**

PASCAL *pathname* [*options*]

### **Usage**

*pathname* is an ASCII file containing PASCAL source code. It is recommended that you give *pathname* a .PASCAL suffix. Use the BIND or the SEG command to create a runfile from the object file. (BIND is recommended.)

For detailed information on the operation and options of PASCAL, see the *Pascal Reference Guide*.

## PASSWD

PASSWD changes the owner and nonowner passwords of the current password-protected directory.

---

### **Note**

Use the CHANGE\_PASSWORD command to change your login password.

---

The use of passworded directories has been superseded by Access Control Lists (ACLs).

### **Format**

PASSWD [*owner-password* [*nonowner-password*]]

### **Arguments**

*owner-password*

The new owner password that replaces the previous one. If you do not specify *owner-password*, the new owner password is blank.

*nonowner-password*

The new nonowner password that replaces the previous one. You cannot specify this argument unless you also specify *owner-password*. If you do not specify *nonowner-password*, the new nonowner password is null.

### **Guidelines for Using PASSWD**

The following are some guidelines to keep in mind when using the PASSWD command:

- To use PASSWD, you must be the owner of the directory whose passwords are to be changed and you must be attached to the directory. A nonowner cannot give this command.
- The PASSWD command applies only to password-protected directories. If you use PASSWD on a directory protected by an Access Control List (ACL), the password will be in effect only after you give the REVERT\_PASSWORD command.
- Passwords may be of any length, but they are matched by the first six characters only.
- To specify a lowercase password, enclose the password in single quotation marks.
- Do not confuse PASSWD with the CHANGE\_PASSWORD command, which changes login passwords.
- The MAGRST command cannot restore directories with passwords unknown to the MAGRST user.

### **PASSWD Example**

The following is an example of using PASSWD. The directory, named STUART and owned by the user, has an owner password of SATURN and a nonowner password of PLUTO. The user attaches to the directory and changes both passwords:

```
OK, ATTACH STUART SATURN
OK, PASSWD LION TIGER
OK,
```

STUART now has LION as its owner password and TIGER as its nonowner password.



.....  
PDEV

PDEV

PDEV converts to or from a pdev to its decoded values.

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

PDEV does the following depending on the specified options:

- Decodes a pdev into its four components.
- Displays physical device information about disks.
- Encodes four components into a corresponding pdev.

---

**Note**

PDEV will not check the validity of a pdev you encode. You should check the four values you use to generate a pdev against the values listed in the *Operator's Guide to File System Maintenance*.

---

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

**Format**

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

**Options**

-DECODE *pdev*

Decodes *pdev* and displays its four components:

- Disk controller address and 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 four decoded pdev components for disk *diskname*. When *diskname* includes a wildcard, PDEV displays information for all disks matching the wildcarded name. Do not use enclosing angle brackets (<>) with *diskname*.

- ENCODE** Generates a pdev from the four values passed as options. See the **-ENCODE** suboptions below. PDEV displays the input information and the resulting pdev.
- HELP** Displays a summary of the command's functions.

**-ENCODE Suboptions**

- CONTROLLER *aa*** Specifies a controller address *aa*, which is one of these octal numbers: 22, 23, 24, 25, 26, 27, 45, 46.
- UNIT *u*** Specifies a disk drive unit number *u*, which is an octal number from 0 through 7.
- STARTING\_HEAD *n*** Specifies a starting head (or surface) number *n*, which is an even number from 0 through 30.
- NUM\_HEADS *m*** Specifies the number of heads *m*, in the partition. *m* 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.

**Examples**

Display the device values of a disk with a pdev of 4060.

```
OK, PDEV -ENCODE 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 -ENCODE OSDSK3
```

| Disk   | LDEV | PDEV | Controller | Unit | Start | Heads |
|--------|------|------|------------|------|-------|-------|
| OSDSK3 | 2    | 7660 | '27 (5)    | 0    | 0     | 30    |

Using a wildcard to display a list of pdevs and their device values.

```
OK, PDEV -ENCODE *SK3
```

| 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,
```

Using PDEV to get the pdev of a disk device with the specified values.



### ***Running Command Input Files as Phantoms***

Command input files run as phantoms must have LOGOUT as the last command in the file, so that the phantom user can log out properly. Using COMINPUT -TTY or COMINPUT -END as the last line may cause an abnormal termination of the phantom. If LOGOUT is not the last line, the phantom reports an abnormal ending when it finishes processing the file.

You can control the file unit on which the command input file is opened by specifying an octal value for *funit*. The default value for *funit* is 6.

### ***Running CPL Programs as Phantoms***

In a CPL program run as a phantom, the &RETURN directive at the end of the program is interpreted as a LOGOUT command. *CPL-arguments* specify arguments to be passed to the CPL program. These arguments are used by the program's &ARGS directive. *funit* cannot be specified with CPL programs.

For further information on CPL programs, see the *CPL User's Guide*.

### ***Phantom Operation***

A phantom process is a job performed by a phantom user. A phantom user is similar to any other PRIMOS user, except that the phantom is not associated with a terminal. Therefore, during execution of a phantom process, all controlling input must be read from either a command input file or a CPL program instead of a user terminal. In addition, terminal output during a phantom process is suppressed unless a command output file has been opened with a COMOUTPUT command in the command input file or CPL program. Output is then written to the command output file.

---

#### **Note**

You must be attached to a directory on your local node in order to run a phantom process.

---

At any given moment, PRIMOS has only a fixed number of phantom processes available. The PHANTOM command searches for a free process into which the phantom user can log in. If no free processes are available, the following message is displayed at your terminal:

```
No phantoms are available.  FILENAME
```

If a process is available, the phantom user is logged in to your login directory. Then, the phantom feature of PRIMOS attaches to your current directory. If *pathname* is a command input file, it is opened on File Unit 6 (or on *funit*). If *pathname* is a CPL file, it is opened on an available unit. The phantom process then takes all further



If you log out and later log in to the same directory, you can use the STATUS and LOGOUT commands, as before, to control the phantom. Any phantom on your system can be logged out at the supervisor terminal.

A CPL program or command file may issue the PHANTOM command. Command files running in phantoms may also include PHANTOM commands (that is, you can chain phantom command files in a manner similar to command input command files).

See also JOB; LON.

## PL1

PL1 invokes the Prime full PL/I compiler and compiles an object program from an ASCII file containing PL1 source code.

### **Format**

PL1 *pathname* [*options*]

### **Usage**

*pathname* is an ASCII file containing PL1 source code. It is recommended that you give *pathname* a .PL1 suffix. Use the BIND or the SEG command to create a runfile from the object file. (BIND is recommended.)

For a full description of the PL/I compiler and its options, see the *PL/I Reference Guide*.

## PL1G

PL1G invokes the Prime PL/I Subset G compiler and compiles an object program from an ASCII file containing PL1G source code.

### **Format**

PL1G *pathname* [*options*]

### **Usage**

*pathname* is an ASCII file containing PL1G source code. It is recommended that you give *pathname* a .PL1G suffix. Use the BIND or the SEG command to create a runfile from the object file. (BIND is recommended.)

For a full description of the PL/I Subset G compiler and its options, see the *PL/I Subset G Reference Guide*.

.....  
PLOT

PLOT

PLOT plots a metafile or device-specific plot file.

**Format**

PLOT { *pathname* -AT *destination* [*options*]  
-CANCEL *plotid*  
-LIST }

**Usage**

For PLOT to work, your system must have the Prime Plotter Software installed. PLOT also lists the contents of the local plot queue and removes one or more plot files from the plot queue. For details on PLOT, see the *Prime Plotter Software Guide*.

PM

PM (Post Mortem) displays the contents of the RVEC user register vector, applicable only during the use of a static-mode program (not an EPF or a CPL program).

The RVEC parameters are described in Appendix A. PRIMOS first displays labels for the RVEC parameters, and then displays their values in the same order on the next line.

The PM command also displays the procedure base register (PB), the stack base register (SB), the link base register (LB), and the temporary base register (XB). These 32-bit registers are displayed at the user terminal on a text line separate from the other registers. Each of the Prime 350-class registers is displayed as two 16-bit octal numbers separated by a ring number and a slash (/) character.

**Format**

PM

**Usage**

The following example illustrates the PM command:

```
OK, P=
SA, EA, P, A, B, X, K=
1000 4543 1120 0 0 0 34000

PB, SB, LB, XB:
4000(3)/1120 4003(0)/4 4000(0)/5064 6(0)/14366
OK,
```

This example of PM under PRIMOS shows a PB of 4000(3)/1120, which indicates: ring 3, segment 4000<sub>8</sub>. The word number portion of PB (1120 in this example) is the same value as the P parameter in the first line of PM's output. This number specifies the location within the segment at which the next instruction executes upon a possible receipt of a START command.

---

#### Note

PM does not give an accurate picture of the machine state of a program if the program was halted by an on-unit that does not allow the static-mode overseer to update the PM data. This situation would occur if a user on-unit returned to command level by calling COMLV\$. The DUMP\_STACK command, however, always produces an accurate display of the state of the program.

---

## PMA

PMA loads the Prime Macro Assembler and starts assembly of a source file.

### Format

**PMA** *pathname* [*options*]

### Usage

*pathname* is the name of the source file. It is recommended that you give *pathname* a .PMA suffix. You can specify more than one option if the options do not conflict.

For a complete discussion of the assembler, including register settings, see the *Assembly Language Programmer's Guide*.

For a complete listing of the PMA instruction sets, see the *System Architecture Reference Guide*.

## PRERR

PRERR either displays the ERRVEC message set by the ERRSET subroutine and the first six octal locations in ERRVEC, or displays ERRVEC and the last error message.

The PRERR command is useful only for debugging obsolete programs. On encountering an error condition, PRIMOS sets up an internal vector called ERRVEC with several pieces of information. One of these pieces is an error message. Refer to the *Subroutines Reference III: Operating System* for a description of ERRVEC.





## PRIME/SNA

PRIME/SNA is not a PRIMOS command, but is a group of separately priced products that support the interconnection of Prime systems with IBM SNA (Systems Network Architecture) networks. The initial PRIME/SNA products are the following two subsystems that supply the services of IBM devices connected to an SNA network:

|                              |                                                                                                                                                                                                  |
|------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Server Subsystem</b>      | Provides the services of one or more SNA Physical Unit Type 2 (PU.T2) secondary SDLC stations.                                                                                                   |
| <b>Interactive Subsystem</b> | Provides emulation of devices in the IBM 3270 Information Display System family (3274 Control Unit, 3278 Display Station, and 3287 and 3289 Printers), in conjunction with the Server Subsystem. |

The following six PRIMOS commands control the PRIME/SNA products:

|                          |                                                                                                |
|--------------------------|------------------------------------------------------------------------------------------------|
| <b>SNADSC</b>            | User command for invoking IBM 3278 emulation and accessing IBM applications.                   |
| <b>SNA_3270</b>          | Operator command for controlling the various functions of the PRIME/SNA Interactive Subsystem. |
| <b>SNA_3270_CONFIG</b>   | Administrator command for creating, editing, or listing an Interactive configuration.          |
| <b>SNA_PRINT</b>         | Operator command for invoking the Interactive Printer Emulation facility.                      |
| <b>SNA_SERVER</b>        | Operator command for controlling the various functions of the PRIME/SNA Server Subsystem.      |
| <b>SNA_SERVER_CONFIG</b> | Administrator command for creating, editing, or listing the Server configuration.              |

These commands are briefly described in this chapter and are also described in detail in the following documents:

- *PRIME/SNA Installation and Configuration Guide*
- *PRIME/SNA Interactive Terminal User's Guide*
- *PRIME/SNA Operator's Guide*

.....  
PRIMON

## PRIMON

PRIMON monitors current system activity through screens of bar graphs, or stores this data in a file that can be analyzed with the PRIMAN tool. PRIMON is a separately priced product.

### Format

PRIMON

### Usage

For more information, see the *PRIMAN User's Guide*.

## PROP

PROP displays information about system printers.

For more information see the *Operator's Guide to Spooler Subsystems*.

### Format

PROP { *environment* -DISPLAY [-NO\_WAIT]  
-REPORT\_SIZE  
-STATUS [-ALL  
-NO\_WAIT]  
-HELP }

### Options

- |                    |                                                                                                                                                                                                                |
|--------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>environment</i> | The name of a printer environment file. A spooler environment is the set of specifications under which the printer operates. Environment specifications are kept in files of the form <i>environment.ENV</i> . |
| -DISPLAY           | Displays the contents of the environment file <i>environment.ENV</i> . An <i>environment.ENV</i> file lists the characteristics and attributes of a specific printer. You do not need to type the .ENV suffix. |
| -NO_WAIT           | Suppresses the —More— prompt and does not pause after every 23 lines of output. Output scrolls continuously.                                                                                                   |
| -REPORT_SIZE       | Report the current system spool job size limits.                                                                                                                                                               |

- STATUS** Lists the names of the system printers that have active despooler phantoms. See the **-STATUS Option** section below for more information.
- HELP** Displays the command syntax and all the available options. PROP options not documented here are reserved for System Administrators and Operators.

### **-STATUS Option**

**-STATUS** lists the names of the system printers that have active despooler phantoms. The **-STATUS** option format is

```
PROP -STATUS [--ALL] [--NO_WAIT]
```

### **-STATUS Suboptions**

- ALL** Displays the status of all environments for which environment files exist.
- NO\_WAIT** Same as above.

### **Examples**

The following example illustrates the **-STATUS** option.

```
OK, PROP -STATUS
[PROP Rev. 23.3.0 Copyright (c) 1992, Prime Computer, Inc.]
LASER           Idle
PLAIN           Idle
B1_LQP          Idle
OK,
```

Example of PROP **-STATUS** with the **-ALL** suboption.

```
OK, PROP -STATUS -ALL
[PROP Rev. 23.3.0 Copyright (c) 1992, Prime Computer, Inc.]
LASER           Idle
PLAIN           Idle
B1_LQP          Idle
MJD             Not Started
B1_LQP_MJD      Not Started
OK,
```

In the example, the despooler phantoms controlling printers LASER, PLAIN, and B1\_LQP are running, but those controlling MJD and B1\_LQP\_MJD are not. Though

.....  
PROP

a user could spool a file to all five printers, the file would not print on MJD and B1\_LQP\_MJD until the Operator started up their phantoms with PROP -START.

The following example illustrates the -DISPLAY output for the LASER printer.

```
OK, PROP LASER -DISPLAY
[PROP Rev. 23.3.0 Copyright (c) 1991, Prime Computer, Inc.]
LASER          Idle

/* Laser printer named LASER on SYS.A
/* Name of environment file is SPOOL*>LASER.ENV
/*
ASYNC -LINE 107 -PROTOCOL TTY -SPEED 9600
ATTRIBUTE LASER -MANDATORY /* LASER attr must be specified.
ATTRIBUTE WHITE
ATTRIBUTE SYS.A
FORMAT -WIDTH 80 -BOTTOM_MARGIN 1 -PAD_CHAR 0
/* Added -pad_char re: Rev 23 info file
LOG -ON
PRIORITY_SIZE 20 /* Print files <20 records first.
NODE -LOCAL /* Scan SYS.A's spool queue.
NODE ENR
POLL_RATE -DECAY 30 /* Poll at fast rate 30x before slowing
PLQ /* Give priority to jobs in local queue.
WARNING -ON
/*
/* End of Environment parameters
OK,
```

For a full description of printer environment directives, see the *Operator's Guide to the Spooler Subsystem*.

# PROTECT

PROTECT sets access rights for file system objects in password-protected directory.

## Format

**PROTECT** *pathname* [*owner-code* [*nonowner-code*]] [-REPORT]

## Arguments and Options

*pathname* Specifies the file, subdirectory, or segment directory to be protected. If you use a wildcard name with no file type selection options, the default selects all objects. To use PROTECT, you must have owner access to *pathname*.

*owner-code*  
*nonowner-code* Designates the type of access rights the owner and nonowners will have to the object. Specify the rights with the following codes:

| <i>Code</i> | <i>Description</i>                   |
|-------------|--------------------------------------|
| NIL         | No access of any kind (default)      |
| R           | Read access only                     |
| W           | Write access only                    |
| D           | Delete access only                   |
| RW          | Read and Write access only           |
| RD          | Read and Delete access only          |
| WD          | Write and Delete access only         |
| RWD         | All rights (Read, Write, and Delete) |

If you specify neither *owner-code* nor *nonowner-code*, both owner and nonowner receive NIL access rights. The owner must therefore modify the protection rights before the files can be accessed.

**-REPORT** Reports the results of each successful setting of protection rights. For example,

```
OK, PROTECT MAST>SAIL RWD R -REPORT
"MAST>SAIL" protected
OK,
```

.....  
PRTDSC

### **Using PROTECT in an ACL Directory**

When you use PROTECT with an object in an ACL directory, the object receives the designated rights but also retains its ACL rights. The ACL rights take precedence and the PROTECT rights are ignored when you access the object. If, however, you subsequently convert the directory to a password directory (with the REVERT\_PASSWORD command), the protection rights set by PROTECT come into effect.

To use PROTECT in an ACL directory, you must have Protect (P) access to the directory.

PRTDSC

PRTDSC invokes the printer emulation program on systems where DPTX/DSC is running. Printer output is spooled with form type equal to the first six characters of the station name.

#### **Format**

PRTDSC *station-1* [ ... *station-n* ]

#### **Usage**

For details, see the *Distributed Processing Terminal Executive Guide*.

PSD, PSD20

PSD loads and starts the R-mode version of the Prime Symbolic Debugger, an interactive debugging program.

PSD20 is a version of PSD for 16K PRIMOS II. It is identical to PSD, except in that it occupies locations '17760 to '26552.

#### **Formats**

PSD  
PSD20

#### **Usage**

When invoked, the PSD and PSD20 programs display a dollar-sign (\$) prompt and wait for you to enter a PSD command. To return to PRIMOS, enter QUIT. For details on PSD, see the sections on VPSD and IPSD in the *Assembly Language Programmer's Guide*.

See also DBG; HPSD; IPSD; VPSD.

## PT45DSC

PT45DSC invokes the PT45 interface program that allows a PT45 terminal to emulate an IBM 3277 Model 2 display station on a system that is running DPTX/DSC.

### **Format**

PT45DSC

### **Usage**

For details, see the *Distributed Processing Terminal Executive Guide*.

## PT46DSC

The PT46DSC command places a DPTX-configured PT46 terminal in DSC mode and allows it to emulate an IBM 3277 terminal. The PT46 terminal is an enhanced model of the PT45 that is better able to operate in a DPTX environment. This command is part of the DPTX product.

### **Format**

PT46DSC

### **Usage**

For details, see the *Distributed Processing Terminal Executive Guide*.



.....  
PTDSC

## PTDSC

The PTDSC command places a DPTX-configured PST 100 or PT200 terminal in DSC mode allowing it to emulate an IBM 3277 terminal. This command is part of the DPTX product.

### **Format**

PTDSC

### **Usage**

For details, see the *Distributed Processing Terminal Executive Guide*.

## PTELE

PTELE accesses the Office Automation System (OAS) Telephone Inquiry function.

### **Format**

PTELE

### **Usage**

For detailed information, see the *OAS Management Communications and Support Guide*.

## R

See RESUME.

## RDY

RDY sets or displays the system prompts.

### Format

RDY [*options*]

### Options

*text*, required for some options, is a character string of no more than 80 characters in length. If *text* contains special characters or blanks, the entire character string must be enclosed within single quotation marks.

- |                                 |                                                                                                                                                                                                                                                          |
|---------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-BRIEF</b>                   | Switches all prompts to the brief forms. (Default at login.)                                                                                                                                                                                             |
| <b>-ERROR_BRIEF <i>text</i></b> | Changes the brief form of the Error prompt to <i>text</i> . ER! is the default brief form.                                                                                                                                                               |
| <b>-ERROR_LONG <i>text</i></b>  | Changes the text portion of the long form of the Error prompt to <i>text</i> . By default, the text portion is a null string. Clock, CPU, and I/O times (plus the level number and the plus sign, if applicable) are still displayed after <i>text</i> . |
| <b>-EXPAND_PROMPT</b>           | Causes any prompt variables present in the prompt text to be expanded each time the prompt is displayed. Otherwise the prompt text is displayed without interpretation.                                                                                  |
| <b>-LONG</b>                    | Switches all prompts to the long forms.                                                                                                                                                                                                                  |
| <b>-NO_EXPAND_PROMPT</b>        | Causes any prompt variables present in the prompt text to be displayed as text strings rather than expanding them.                                                                                                                                       |
| <b>-OFF</b>                     | Suppresses the display of all prompt messages.                                                                                                                                                                                                           |
| <b>-ON</b>                      | Reenables the display of all prompt messages. Unless the <b>-LONG</b> or <b>-BRIEF</b> option is given with the <b>-ON</b> option, messages appear in the format last specified.                                                                         |
| <b>-READY_BRIEF <i>text</i></b> | Changes the brief form of the Ready prompt to <i>text</i> .                                                                                                                                                                                              |
| <b>-READY_LONG <i>text</i></b>  | Changes the text portion of the long form of the Ready prompt to <i>text</i> . Clock, CPU, and I/O times (plus the level number and the plus sign, if applicable) are still displayed after <i>text</i> .                                                |

- WARNING\_BRIEF *text* Changes the long form of the Warning prompt to *text*.
- WARNING\_LONG *text* Changes the text portion of the long form of the Warning prompt to *text*. Clock, CPU, and I/O times (plus the level number and the plus sign, if applicable) are still displayed after *text*.

### RDY Prompts

After PRIMOS executes a command, it displays a prompt message that indicates whether or not the command was executed successfully. There are three types of RDY prompts, each of which has a brief form and a long form. These types and their default messages at login are as follows (the underscore following the prompt indicates the position of the cursor):

| Type    | Brief Text | Long Text                                                                                                | Description                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|---------|------------|----------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Ready   | OK, _      | OK <i>clock-time CPU-time I/O-time levelno</i><br>-<br>Example:<br>OK 13:18:24 7.736 2.593 level 2+<br>- | Given after commands that executed successfully. The default long form is OK followed by the clock time, CPU time used since the last prompt, I/O time used since the last prompt, the level number if you are at a command level above 1 (that is, if a program was interrupted by a Ctrl-P or by an error-handling mechanism), and a plus sign (+) if the level is static mode. (This means that the last command executed was an external command.) |
| Error   | ER! _      | ER <i>clock-time CPU-time I/O-time levelno</i><br>-<br>Example:<br>ER 13:18:24 7.736 2.593 level 2+<br>- | Given after commands that ran into fatal errors. Except for the text string, the data in the long Error prompt is the same as the Ready prompt.                                                                                                                                                                                                                                                                                                        |
| Warning | OK, _      | OK <i>clock-time CPU-time I/O-time levelno</i><br>-<br>Example:<br>OK 13:18:24 7.736 2.593 level 2+<br>- | Given after commands that ran into error conditions but still executed to completion. The default brief and long forms are the same as the Ready prompts.                                                                                                                                                                                                                                                                                              |

The RDY command, when followed by one or more options, allows you to toggle between the long and brief forms of these prompts or to change any of the six default messages.

If issued without an option, RDY displays the current Ready long-form prompt, as shown:

```
OK, RDY
OK 15:12:42 7.674 2.343 level 2+
OK,
```

### ***RDY Prompt Variables***

The RDY command supports dynamic and expandable prompts. Variables can be included in the text for any RDY prompt. Variable prompt expansion works with both brief and long forms of all RDY prompts.

To instruct PRIMOS to expand RDY prompt variables, use the `-EXPAND_PROMPT` option. The current value of each variable in the prompt text is substituted each time the prompt is displayed. Expansion remains enabled until you disable it with the `-NO_EXPAND_PROMPT` option.

When the long form of RDY is enabled, the `-EXPAND_PROMPT` option causes the prompts to be entirely replaced with the prompt text. When expansion is not enabled (by omitting the `-EXPAND_PROMPT` option when creating the prompt, or by disabling it with the `-NO_EXPAND_PROMPT` option) only the text portion of the long prompt is replaced. For example,

```
OK, RDY -READY_LONG '%a : '           (Define a new long prompt)
OK, RDY -LONG                          (Toggle to long mode)
%a : 15:21:14 0.018 0.000              (An unexpanded long prompt)
RDY -EP                                 (Toggle expansion on)
<USERS>DRG :                            (The expanded long prompt)
<USERS>DRG : RDY -NEP -BRIEF           (Toggle expansion and long mode off)
OK,
```

Currently supported variables are:

|                  |                                           |
|------------------|-------------------------------------------|
| <code>%a</code>  | Current attach point                      |
| <code>%c</code>  | CPU time since last prompt                |
| <code>%dt</code> | Quick date. Defaults to MM/DD/YY          |
| <code>%da</code> | Abbreviated day name (e.g., Tue)          |
| <code>%dn</code> | Full day name (e.g., Tuesday)             |
| <code>%ds</code> | Day number with suppressed zero (e.g., 3) |
| <code>%dz</code> | Day number with leading zero (e.g., 03)   |

|                  |                                                                                                                                                                                                                                                                                                                                            |
|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>%en</code> | Entrypoint portion of current attach point pathname                                                                                                                                                                                                                                                                                        |
| <code>%ec</code> | Error code of last command (useful for Error and Warning prompts)                                                                                                                                                                                                                                                                          |
| <code>%f</code>  | AM or PM flag: if the time is 00:00:00 to 11:59:59 then the string am is returned, else pm is returned                                                                                                                                                                                                                                     |
| <code>%h1</code> | 24 hour value with leading zero (e.g., 08)                                                                                                                                                                                                                                                                                                 |
| <code>%h2</code> | 24 hour value with suppressed zero                                                                                                                                                                                                                                                                                                         |
| <code>%h3</code> | 12 hour value with leading zero                                                                                                                                                                                                                                                                                                            |
| <code>%h4</code> | 12 hour value with suppressed zero                                                                                                                                                                                                                                                                                                         |
| <code>%i</code>  | IO time since last prompt                                                                                                                                                                                                                                                                                                                  |
| <code>%l</code>  | Current command level. If the command level is 1 (the lowest command level) then a null string ("") is returned. If the command level is greater than 1 then the string "level <i>n</i> " is returned. If a static mode program was just executed and the command level is greater than 1, then the string "level <i>n</i> +" is returned. |
| <code>%mi</code> | Number of minutes with leading zero                                                                                                                                                                                                                                                                                                        |
| <code>%ma</code> | Abbreviated month name                                                                                                                                                                                                                                                                                                                     |
| <code>%mn</code> | Full month name                                                                                                                                                                                                                                                                                                                            |
| <code>%ms</code> | Numeric month number with suppressed zero                                                                                                                                                                                                                                                                                                  |
| <code>%mz</code> | Numeric month number with leading zero                                                                                                                                                                                                                                                                                                     |
| <code>%n</code>  | Number of users on the system                                                                                                                                                                                                                                                                                                              |
| <code>%p</code>  | ACL protection on the current directory                                                                                                                                                                                                                                                                                                    |
| <code>%rv</code> | PRIMOS revision                                                                                                                                                                                                                                                                                                                            |
| <code>%rn</code> | Remote name (the system name where the current attach point resides)                                                                                                                                                                                                                                                                       |
| <code>%se</code> | Seconds with leading zero                                                                                                                                                                                                                                                                                                                  |
| <code>%sn</code> | System name                                                                                                                                                                                                                                                                                                                                |
| <code>%t</code>  | Quick time (24 hour format HH:MM:SS)                                                                                                                                                                                                                                                                                                       |
| <code>%ui</code> | User ID                                                                                                                                                                                                                                                                                                                                    |
| <code>%un</code> | User number                                                                                                                                                                                                                                                                                                                                |
| <code>%y2</code> | Two-digit year number                                                                                                                                                                                                                                                                                                                      |
| <code>%y4</code> | Four-digit year number                                                                                                                                                                                                                                                                                                                     |
| <code>%l</code>  | CRLF (carriage return and linefeed)                                                                                                                                                                                                                                                                                                        |
| <code>%.</code>  | Global variable expansion. Syntax is <code>%.global-var%</code> (e.g., <code>%.terminal_type\$%</code> ) Returns the current value of a global variable.                                                                                                                                                                                   |

- %% The percent sign (%). This allows the user to override the meaning of a prompt variable.
- %- Reserved for future use.
- %[ ... %] PRIMOS active function call. The call must be in the format %[*command-function*%] where *command-function* is the EPF to be called as a function. PRIMOS runs the command function each time your prompt is displayed. For example, you could write a function that checks for mail, so that your prompt notifies you when mail arrives.
- Note that command function calls may be nested and may also contain prompt variables.
- For example, %[BEFORE %[AFTER %a '<'%] '>'%] would pass your current attach point (%a) to the AFTER function which would then pass the modified string to the BEFORE function.

### RDY Variable Error Messages

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

#### \$BUFFER\_TOO\_SMALL\$

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

#### \$CURRENT\_ATTACH\_POINT\_UNAVAILABLE\$

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

#### \$ENTRYNAME\_UNAVAILABLE\$

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

#### \$ACL\_UNAVAILABLE\$

The ACL on the current directory could not be obtained.

#### \$SYSTEM\_NAME\_UNAVAILABLE\$

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

#### \$USER\_ID\_UNAVAILABLE\$

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

**\$ERROR\_IN\_VARIABLE\_REFERENCES\$**

A global variable was referenced that does not exist.

**\$GLOBAL\_VARIABLE\_STORAGE\_UNDEFINED\$**

Global variable storage was not allocated for the current process (via DEFINE\_GVAR).

**\$NESTING\_TOO\_DEEP\$**

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

**\$UNBALANCED\_BRACKETS\$**

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

**\$IMPROPER\_USAGE\_OF\_BRACKETS\$**

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

**ECL Prompts and RDY Prompts**

ECL (EDIT\_COMMAND\_LINE) provides an option to define brief prompts. The brief prompts you define as an ECL option supersede the RDY brief prompts as long as ECL is enabled. When ECL is disabled with the ECL -OFF command, the RDY brief prompt is displayed. If you define a new RDY brief prompt when ECL is running, it is stored, but not used. Then, when ECL is disabled, the prompt you defined is used.

The following example demonstrates the relationship between ECL prompts and RDY brief prompts.

The example starts with ECL disabled and RDY in brief mode using default prompts.

First define new RDY brief prompts:

```
OK, RDY -RE 'RDY brief ok: ' -EB 'RDY brief er: '
RDY brief ok: BOGUS (A bad command)
Not found. BOGUS (std$cp)
RDY brief er: (You get the RDY error prompt)
```

Now define the ECL prompts:

```
RDY brief ok: ECL -RE 'ECL ok: ' -EB 'ECL er: '
RDY brief ok: ECL -ON (Still the RDY prompt. Now enable ECL)
ECL ok: (You get the ECL ready prompt)
ECL ok: BOGUS (A bad command)
Not found. BOGUS (std$cp)
ECL er: (You get the ECL error prompt)
```

Now redefine the RDY prompt while ECL is still enabled:

```
ECL ok: RDY -RB 'new RDY brief ok: '
ECL ok: ECL -OFF                (Still the ECL prompt. Now disable ECL)
new RDY brief ok:                (You get the revised RDY ready prompt)
```

When RDY is in long mode, and ECL is enabled, both the RDY prompt and the ECL prompt are displayed. For example, with long RDY prompts of RDY ok long and RDY er long and ECL prompts of ECL ok: and ECL er: , the prompts appear as follows:

```
RDY ok long 14:15:37  0.160  0.000
ECL ok: BOGUS                (A bad command)
Not found. BOGUS (std$cp)
RDY er long 14:16:23  0.060  0.000
ECL er:
```

## REENTER

REENTER reenters a subsystem following a QUIT or an error condition.

### Format

REENTER

### Usage

For REENTER to succeed, the subsystem being reentered must define an on-unit for the condition REENTER\$ that can go to the appropriate point within the subsystem. If no on-unit exists, the REENTER command fails and you return to PRIMOS command level.

For further information, see the *Advanced Programmer's Guide III: Command Environment*.

See also START, RESUME, and RELEASE\_LEVEL.



## RELEASE\_LEVEL

RELEASE\_LEVEL discards unwanted stack history. The stack history, which is automatically saved by PRIMOS, is a record of the calls and returns created by user commands.

### Format

RELEASE\_LEVEL {  
-ALL  
-TO *n*  
-LEVELS *n*}

### Options

Each of these options specifies how much of the stack to release.

- ALL Releases the entire stack down to listener level 1.
- TO *n* Releases stack levels down to level *n*. *n* must be a positive decimal integer less than the current level number.
- LEVELS *n* Releases *n* levels so that the new stack level will be the current level minus *n*. *n* must be a positive decimal integer such that the current level minus *n* is greater than or equal to 1. (Default of *n* is 1.)

If you do not specify an option, one of the following results:

- If your most recent command was an internal command, the current level of the stack is released.
- If your most recent command was an external command, the history of that command is released but the level number of the stack is not changed.

To determine your current stack level, issue the RDY command without options. If you are above level 1, the level number is displayed. If you are at level 1, no level number is displayed. In either case, the long form of the Ready prompt is also displayed.

If the stack grows large and unwieldy as a result of too many interruptions, PRIMOS warns you to release the stack with the message

Now at command level *n*. To release use RLS. (listen\_)

You cannot follow RELEASE\_LEVEL with a semi colon (;) on the same command line. All commands following RELEASE\_LEVEL on the command line are ignored.

For further information, see the *Advanced Programmer's Guide III: Command Environment*.

See also START, RESUME, and REENTER.

## REMOVE\_EPF

REMOVE\_EPF removes an Executable Program Format (EPF) from a user's address space.

The EPF to be removed must be mapped into your address space and must not be suspended. (Use the LIST\_EPF command to list the pathname and status of your mapped EPFs.) REMOVE\_EPF does not delete the EPF file itself. (To delete a mapped EPF, use REMOVE\_EPF to unmap the EPF, and then use the DELETE command to delete the file.)

### Format

REMOVE\_EPF [*pathname*] [*options*]

### Argument and Options

You can specify more than one option if they do not conflict.

#### *pathname*

The name of the EPF you want removed. You do not have to include the EPF suffixes .RUN or .RP*n* (where *n* is a single digit). Both wildcarding and iteration can be used, but not treewalking.

If the EPF indicated by *pathname* is not currently mapped into your address space or does not exist, REMOVE\_EPF displays the message

No EPFs removed (REMOVE\_EPF).

If you omit *pathname*, REMOVE\_EPF asks you to verify the removal of each EPF currently mapped into your address space. Enter Y or YES to remove the specific EPF, or enter N or NO to leave it mapped.

#### -ACTIVE

Removes only active (in-use) process-class library EPFs. Does not remove active suspended program EPFs.

#### -FORCE

Removes only active program-class library EPFs. Does not remove active program EPFs. Do not use the -FORCE option if an active program-class library EPF is to be called within the same program invocation by another EPF that has already established linkage to the active EPF.

■ ■ ■ ■ ■ ■ ■ ■ ■ ■  
**REMOVE\_EPF**

- NOT\_ACTIVE** Removes only non-active EPFs (that is, EPFs that are currently mapped to your address space, but are neither suspended program EPFs nor active process-class library EPFs).
- NO\_QUERY** Does not ask the user to verify that the EPF is currently in use within the user's address space.
- NO\_VERIFY** Disables verification checking if wildcards are used within *pathname*.
- QUERY** Asks the user to verify that the EPF is currently in use within the user's address space. (Default)
- VERIFY** Asks the user to verify the removal of an EPF. (The default is not to require verification unless wildcards are used within *pathname*.)
- HELP** Displays the command's correct syntax. The HELP screen is also displayed if PRIMOS detects an error while parsing the command.

***REMOVE\_EPF Example***

The following example illustrates the REMOVE\_EPF command without options:

```
OK, REMOVE_EPF
Ok to remove EPF file <OSGRP0>LIBRARIES*>FORTRAN_IO_LIBRARY.RUN? NO
Ok to remove EPF file <OSGRP0>LIBRARIES*>FORTRAN_LIBRARY.RUN? NO
Ok to remove EPF file <OSGRP0>LIBRARIES*>SYSTEM_LIBRARY.RUN? NO
Ok to remove EPF file <OSGRP0>CMDNCO>HELP.RUN? no
No EPFs removed (REMOVE_EPF).
OK,
```

For more information on REMOVE\_EPF and EPF's, see the *Programmer's Guide to BIND and EPFs*. For a detailed description of EPF's see the *Advanced Programmer's Guide, Volume 1: BIND and EPFs*.

## REMOVE\_REMOTE\_ID

Your remote-ID list contains the remote IDs that you can use to access files on remote systems. (Use the ADD\_REMOTE\_ID command to add a remote ID to the list.) The list can contain a maximum of 16 remote IDs, one ID per system. If your list has reached the 16 ID limit, you cannot add more remote IDs unless you first remove at least one remote ID with the REMOVE\_REMOTE\_ID command. To list the remote IDs on this list, use the LIST\_REMOTE\_ID command.

### Format

REMOVE\_REMOTE\_ID -ON *system*

### Option

-ON *system* Specifies the name of the remote system whose remote ID you want removed. If *system* is not on the remote-ID list, the message Not found is returned.

### Usage

The following example illustrates the removal of a remote ID from the system named T39. (LIST\_REMOTE\_ID is used to display your current remote IDs.):

```
OK, LIST_REMOTE_ID
System  User id  Project id
-----  -
T35     FRED_J
T54     FRED_REM POWER
T39     FRED2    WISHBONE

OK, REMOVE_REMOTE_ID -ON T39
OK, LIST_REMOTE_ID
System  User id  Project id
-----  -
T35     FRED_J
T54     FRED_REM POWER

OK,
```

REN

See REENTER.

■ ■ ■ ■ ■ ■ ■ ■ ■ ■  
*RESTOR*

## RESTOR

RESTOR restores an R-mode runfile from disk to memory, using the RVEC parameters saved with the file.

### **Format**

RESTOR *pathname*

### **Argument**

*pathname* Specifies the name of the runfile to be restored to memory.

### **Usage**

To restore a 64V segmented mode runfile, use the RESTOR subcommand of SEG instead of the RESTOR command.

## REST\_RBF

REST\_RBF activates (restores) an inactive ROAM file. RESTORE\_RBF is a synonym for REST\_RBF.

When you activate an inactive ROAM file, the recovery table (RCVTAB) is given the location of the ROAM master file and all its slaves. The file can then be accessed by runtime processing.

### **Format**

REST\_RBF *source-pathname* [*target-pathname*] [*options*]

### **Arguments and Options**

*source-pathname* The name of the file to be restored. If a currently active version of this file exists, REST\_RBF asks you if it should deactivate that file. If, however, you used the NO\_QUERY option, REST\_RBF deactivates the file without asking you.

|                        |                                                                                                                                                                                                                                                                               |
|------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>target-pathname</i> | The filename or pathname of the new location for the active file. If <i>target-pathname</i> is an existing file, it is overwritten only if it is a ROAM master segment directory. If you do not specify <i>target-pathname</i> , REST_RBF executes a restore in place.        |
| -CAM                   | Restores all the subfiles of the target segment directory as contiguous files. Cannot be used with the -DAM option. If neither -CAM nor -DAM is specified, the subfiles of the restored segment directory assume the same file type as those of the source segment directory. |
| -DAM                   | Restores all the subfiles of the target segment directory as DAM files. Cannot be used with the -CAM option. If neither -DAM nor -CAM is specified, the subfiles of the restored segment directory assume the same file type as those of the source segment directory.        |
| -NO_QUERY              | Suppresses system questions during the restore operation.                                                                                                                                                                                                                     |
| -PROTECT               | Gives the active file the same ACL protection as the archived file.                                                                                                                                                                                                           |
| -REPORT                | Reports each RBF subfile as it is restored. The default is not to report successful restores of individual subfiles. With or without the -REPORT option, REST_RBF reports the successful restore of the entire RBF.                                                           |

For details on REST\_RBF and on ROAM files, see the *ROAM Administrator's Guide*.

## RESUME

RESUME executes either a runfile or a CPL program.

The runfile can be a dynamic-mode Executable Program Format (EPF) file linked with the BIND command or an R-mode runfile loaded with the LOAD command, but not a static-mode runfile loaded with the SEG command.

RESUME searches for file suffixes in the following order:

1. .RUN
2. .SAVE
3. .CPL
4. No suffix

---

**Note**

Do not use RESUME to execute a file with a .SEG suffix. Use the SEG command instead.

---

**Format**

RESUME *pathname* [*arguments*]

**Arguments**

*pathname*                    The entryname or pathname of a runfile or CPL program (whose filename must have a .CPL suffix).

*arguments*                 The arguments (if any) to pass to the program you are invoking.

**Executing EPF Runfiles**

EPF runfiles are created by the BIND linking utility and have .RUN suffixes. These dynamic-mode runfiles are loaded into memory by PRIMOS, which allocates address space at execution time. Use the following format to execute an EPF runfile:

RESUME *pathname* [*program-arguments*]

You do not need to include the .RUN suffix on *pathname* because PRIMOS looks for it first. For example, to execute an EPF named PROG1.RUN which is in your current directory, issue the command

OK, RESUME PROG1

*program-arguments* are arguments passed to the program.

**Executing R-mode Runfiles**

R-mode runfiles are static-mode files with .SAVE suffixes or without suffixes. Files without suffixes are executed as .SAVE files. R-mode runfiles are created with the LOAD command.

R-mode runfiles are loaded into memory, using the saved RVEC values. You cannot change the SA and EA values but you can change one or more of the other five values by specifying them in *arguments* with the following format:

RESUME *pathname* [*pc*] [*a*] [*b*] [*x*] [*keys*] [*program-arguments*]

See Appendix A for an explanation of RVEC parameters and how you can specify new values. *program-arguments* are nonnumeric arguments passed to the program.

### Executing CPL Programs

CPL programs have .CPL suffixes. To execute a CPL program, use the format

**RESUME *pathname* [*CPL-arguments*]**

*CPL-arguments* are passed to the program as CPL arguments. You can also use the CPL, JOB, and PHANTOM commands to execute a CPL program. For more information on CPL programs, see the *CPL User's Guide*.

For detailed information about the command processor and command environment operations, see the *Advanced Programmer's Guide III: Command Environment*.

See also CPL, START, and REENTER.

## REVERT\_PASSWORD

REVERT\_PASSWORD converts the current directory from an ACL-protected directory to a password-protected directory.

Both of the following conditions must exist before you can convert an ACL directory:

- You must have Protect (P) access rights to the directory.
- The directory cannot have any access categories or subdirectories protected by ACLs.

If the directory was initially created as a password directory, its original password and protection keys are restored. If the directory was not previously a password directory, REVERT\_PASSWORD sets the owner password as blank and the nonowner password as null. To change these defaults, use the PASSWD command.

### Format

REVERT\_PASSWORD

### Usage

For further details, see the *PRIMOS User's Guide*.



.....

RJE

## RJE

RJE is not a PRIMOS command. Remote Job Entry (RJE) Phase II products are separately priced Prime software that enable multiuser Prime systems to emulate other vendor's RJE terminals over half-duplex, point-to-point, synchronous, and dialup or dedicated communications lines.

The two PRIMOS commands used with RJE are

- The RJQ command, which provides user interface to RJE
- The RJOP command, which provides operator control of RJE

The following terminals can be emulated with RJE:

- IBM 2780 and 3780
- HASP
- CDC 200UT
- Honeywell GRTS
- Univac 1004
- ICL 7020
- XBM (CO3)

For detailed information on the Prime Remote Job Entry emulators, see the *Remote Job Entry Phase II Guide*.

See also RJQ.

## RJQ

RJQ queues files for transmission to a remote site.

### Format

$$\text{RJQ } \left\{ \begin{array}{l} \textit{pathname} \text{ [-TO] } \textit{queuename} \text{ [queue-suboptions]} \\ \left\{ \begin{array}{l} \text{-CANCEL} \\ \text{-LIST} \\ \text{-RESET} \end{array} \right\} \left\{ \begin{array}{l} \textit{entry-number} \\ \text{ALL} \\ \text{OWN} \end{array} \right\} \left\{ \begin{array}{l} \text{-DEFER } \textit{time} \\ \text{-TO } \textit{queuename} \\ \text{-WITH } \textit{protocol} \end{array} \right\} \end{array} \right\}$$

### Arguments and Options

|                                                             |                                                                                                                                                                                                                                                                                                                                                                            |
|-------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>entry-number</i>                                         | Acts only on the queue entry specified by <i>entry-number</i> . You can specify <i>entry-number</i> either as a full queue entry number (RJ <i>nnnn</i> , where <i>nnnn</i> is a four-digit number) or as the four-digit number that follows the RJ prefix. You can specify more than one number if you separate each one with a comma.                                    |
| <i>pathname</i>                                             | The name of the file to be transmitted. You can specify a filename if the file is in your current directory. If you use the <code>-NO_COPY</code> option, you must specify the full pathname of the file.                                                                                                                                                                  |
| ALL                                                         | Acts on all entries in the general queue of submitted file transmission requests. (Default for <code>-LIST</code> .) Users cannot specify ALL for <code>-CANCEL</code> or <code>-RESET</code> . The operator (that is, a user logged in as SYSTEM) can use ALL with <code>-CANCEL</code> to cancel all entries in the referenced queue.                                    |
| <code>-CANCEL</code>                                        | Cancels some or all of your own queue entries.                                                                                                                                                                                                                                                                                                                             |
| <code>-DEFER time</code>                                    | Lists only entries deferred beyond the specified time. Can be used only with <code>-LIST</code> . <i>time</i> is given in the 24-hour hh:mm format. (The colon is optional.) The default for <i>time</i> is the current time.                                                                                                                                              |
| <code>-LIST</code>                                          | Lists some or all queue entries.                                                                                                                                                                                                                                                                                                                                           |
| <code>[-TO] queuename</code><br>[ <i>queue-suboptions</i> ] | The name of the queue in the remote system to which the file is being sent. The default name for <i>queuename</i> is the site name of the remote system. You do not have to specify the <code>-TO</code> keyword if <i>queuename</i> is the third item on the command line. See the section below called Queue Suboptions for the description of <i>queue-suboptions</i> . |

- OWN** Acts on all the specific queue entries for file transmission requests submitted under your login ID.
- RESET** Resets the state of one of your files that aborted during transmission, so that the file may be restarted and transmission retried.
- WITH *protocol*** Acts only on entries queued with the given protocol. The legal arguments for *protocol* are 2780, 3780, HASP, GRTS, 7020, XBM, 1004, and 200UT.

### Queue Suboptions

Use the following suboptions with the **-TO** option. You can specify more than one option if they do not conflict.

- AS *internal-name*** Defines the name that appears in the name field when you issue the RJQ **-LIST** command. (Default is the file's pathname, without passwords.)
- DEFER *time*** Delays transmission of the file until the specified time. *time* is given in the 24-hour hh:mm format. (The colon is optional.)
- DELETE** Deletes the user's source file after the file has been successfully entered in the queue.
- DEVICE *device-name*** Specifies the device to which or from which the file is being sent. *device-name* is one of the following: CP*n*, CR*n*, or LP*n*, where *n* is a device number from 1 to 7, inclusive. In slave (terminal) mode, files can be sent only as card input (CR). In master (host) mode, files can be sent to a remote printer (LP) or a remote punch (CP). In HASP and XBM only, more than one of each type of device may be on the system (for example, LP3).
- KEEP\_REQUEST** Prevents the emulator from deleting the queue entry after transmission, but marks the entry as "Sent" in the queue.
- NO\_COPY** Prevents a temporary copy of the user data from being made and queued for transmission. If you use this option, you must specify a full pathname for the file.
- NO\_TRANSLATE** Sends the file without translation.
- VFC [*argument*]** Defines the vertical forms translation codes to be sent with line printer files in master (host) mode. *argument* is either FTN or NONE (the default). FTN is supported only for 2780, 3780, or HASP emulators.

**-WITH *protocol***

Enforces the file-sending protocol specified in the remote site's site definition file. *protocol* is one of the following: 2780, 3780, HASP, GRTS, 7020, XBM, 1004, or 200UT.

### **Usage**

Remote Job Entry (RJE) users use the RJQ command to perform the following three functions:

- Queue a file for transmission to a remote site.
- Manage queued files. This function includes listing the RJE file transmission queue entries, canceling unwanted entries from the queue, and restarting aborted file transmissions.
- Build concatenated files.

After the file is entered in the queue, it is assigned a queue entry number in the following format:

**RJnnnn**

*nnnn* is a four-digit number. RJ0004 is an example of a queue entry number.

### **Queuing a File**

To queue a file for transmission to a remote site, use the following RJQ command format:

**RJQ *pathname* [-TO] *queuename* [*queue-options*]**

### **Building Concatenated Files**

RJQ has an interactive mode that allows you to build concatenated files. To enter this mode, issue the RJQ command without any arguments or options. A greater-than symbol (>) prompt preceded by *Concat .* is displayed, as follows:

```
OK, RJQ
Concat . >
```

You must enter a RJQ subcommand to continue. Enter **HELP** to invoke the RJQ Help facility.

For complete information on RJQ, see the *Remote Job Entry Phase II User's Guide*.



## RUNOFF

RUNOFF invokes Prime's text formatting program, RUNOFF.

### **Format**

**RUNOFF** [*pathname*]

### **Argument**

*pathname* specifies the ASCII source file that is to be formatted. If you do not specify *pathname*, RUNOFF prompts you for its name, as follows:

```
OK, RUNOFF
[RUNOFF Rev. T3.0-23.0 Copyright (c) 1987 Prime Computer, Inc.]
INPUT FILE:
```

### **Usage**

The input file must be in RUNOFF input form. It normally contains the necessary RUNOFF commands to control margins, indentation, line spacing, running heads, page numbering, and other textual matters.

After you have specified the input filename (either on the command line or at the INPUT FILE: prompt), RUNOFF displays a dollar-sign (\$) prompt so that you can enter any further RUNOFF commands. To finish entering RUNOFF commands, enter a blank line (a carriage return without text).

RUNOFF produces a formatted output to a designated disk file. For detailed and tutorial information, see the *New User's Guide to EDITOR and RUNOFF*.

.....

## RWLOCK

## RWLOCK

RWLOCK sets the read/write concurrency lock on a file or segment directory. The setting of a concurrency lock determines how many readers and/or writers can access the file or segment directory at one time.

### Format

RWLOCK *pathname lock* [-REPORT]

### Arguments and Option

|                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>pathname</i> | Identifies the object to be protected. You must have Owner access to the object if it is a password directory, or Protect (P) access if it is an ACL directory. Wildcards are supported.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <i>lock</i>     | Specifies the setting of the read/write concurrency lock. The value of <i>lock</i> must be only one of the following:<br><br>SYS      Sets protection to the value of the system read/write lock. (Default) The system lock is determined by the System Administrator.<br><br>EXCL     Sets for <i>n</i> readers <i>or</i> one writer (exclusive OR). An unlimited number of readers or one writer can access the object simultaneously.<br><br>UPDT     Sets for <i>n</i> readers <i>and</i> one writer. An unlimited number of readers and one writer can access the object simultaneously.<br><br>NONE     Sets for <i>n</i> readers <i>and</i> N writers. All users can access the object simultaneously. |
| -REPORT         | Reports the result of each successful lock setting.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |

### Usage

The following example illustrates the RWLOCK command:

```
OK, RWLOCK MAIL.LIST EXCL -REPORT
"MAIL.LIST" locked.
OK,
```

*Replace this page with the tab page labeled*

**S-Z**



S                    See START.

SAC                  See SET\_ACCESS.

SAVE

SAVE saves the contents of memory (segment 4000<sub>8</sub>) in a runfile.

Because SAVE writes the contents only of segment 4000<sub>8</sub>, use SAVE only to save static-mode programs residing entirely in segment 4000<sub>8</sub>. You cannot use SAVE on multisegment programs, which do not reside entirely in segment 4000<sub>8</sub> or on EPFs, which never reside in segment 4000<sub>8</sub>.

**Format**

SAVE *pathname* [*start-address*] [*end-address*] [*pc*] [*a*] [*b*] [*x*] [*keys*]

**Arguments and Options**

- |                                  |                                                                                                                                                                                                                                                                                                                                           |
|----------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>pathname</i>                  | Specifies the name of the static-mode runfile you wish to produce. If <i>pathname</i> is an entryname, the file is written to the current directory.                                                                                                                                                                                      |
| <i>start-address end-address</i> | Specifies the start and end addresses of the data to save from segment 4000 <sub>8</sub> .                                                                                                                                                                                                                                                |
| <i>pc a b x keys</i>             | RVEC parameters described in Appendix A of this guide. If you do not specify new values for these parameters, SAVE takes the existing values of RVEC from the current register set and stores them with the program. The RVEC parameters are used to initialize the processor registers and keys when the program is restored or resumed. |

**Usage**

The SAVE command produces only static-mode runfiles (.SAVE files). To produce 32I or 64V segmented runfiles (.SEG files), use the SEG command. To produce EPF runfiles (.RUN files), use the BIND command.

---

**Note**

All FORTRAN programs begin with ELM (Enter Load Mode). If macro assembler (PMA) users have ELM as the first instruction in the program, they do not have to use the SAVE command to set the keys after loading. The preferred way to save a memory image is to use the SAVE subcommand of LOAD, which automatically sets up the program keys as appropriate.

---

.....  
SAVE\_RBF

## SAVE\_RBF

SAVE\_RBF archives a ROAM file to disk.

Before you can use SAVE\_RBF, the after-image file must be on the system containing the ROAM file to be stored.

### Format

SAVE\_RBF *sourcename targetname* [-CAM] [-DAM] [-PROTECT] [-REPORT]

### Arguments and Options

|                   |                                                                                                                                                                                                                                                                               |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>sourcename</i> | Specifies the name of a ROAM master segment directory.                                                                                                                                                                                                                        |
| <i>targetname</i> | Specifies the name of the archived master file. It cannot be an existing filename.                                                                                                                                                                                            |
| -CAM              | Archives all the subfiles of the target segment directory as contiguous files. Cannot be used with the -DAM option. If neither -CAM nor -DAM is specified, the subfiles of the archived segment directory assume the same file type as those of the source segment directory. |
| -DAM              | Archives all the subfiles of the target segment directory as DAM files. Cannot be used with the -CAM option. If neither -DAM nor -CAM is specified, the subfiles of the archived segment directory assume the same file type as those of the source segment directory.        |
| -PROTECT          | Gives the archived file the same ACL protection as the source file.                                                                                                                                                                                                           |
| -REPORT           | Reports each RBF subfile as it is archived. The default is not to report successful saves of individual subfiles. With or without the -REPORT option, SAVE_RBF reports the successful save of the entire RBF.                                                                 |

### Usage

To archive an active or inactive ROAM file to magnetic tape, use the MAGSAV utilities. For details on SAVE\_RBF and on ROAM files, see the *ROAM Administrator's Guide*.

## SCHDEC

SCHDEC invokes the DBMS Schema Decompiler (SCHDEC).

### **Format**

SCHDEC [*schema-name*] [*output-pathname*]

### **Usage**

The schema decompiler translates a compiled schema back to an ASCII file (that is, into its DDL description). If you omit *schema-name* and *output-pathname*, SCHDEC prompts you for them. Access to the schema decompiler is limited to Database Administrators.

For detailed information, see the *DBMS Administrator's Guide*.

## SCHED

SCHED invokes the DBMS Schema Editor (SCHED). The schema editor is an interactive processor that allows a Database Administrator to alter the definition of a database. If you do not specify a schema name, SCHED prompts you for one.

### **Format**

SCHED [*schema-name*]

### **Usage**

For details, see the *DBMS Administrator's Guide*.

■ ■ ■ ■ ■ ■ ■ ■ ■ ■  
SCHEMA

## SCHEMA

SCHEMA invokes the DBMS schema Data Description Language (DDL) compiler.

### Format

SCHEMA *sourcename* [-OUTPUT *pathname*] [-LIST *pathname*] [-DAM]

### Argument and Options

- sourcename* Specifies the pathname of a schema definition file for the DDL compiler to translate.
- DAM Causes subfiles to be created as DAM files. Without the -DAM option, subfiles are created as contiguous (CAM) files.
- LIST *pathname* Specifies an alternative location or name for the output listing. The default name and location is *filename.LIST* in the same directory as the source file.
- OUTPUT *pathname* Specifies an alternate location for the new schema table. By default the schema table is placed in your current directory.

### Usage

For more information on SCHEMA, see the *DBMS User's Guide* or the *DBMS Data Description Language Reference Guide*. For more information on the -DAM option, see the *DBMS Administrator's Guide*.

## SEG

SEG invokes a utility for loading, modifying, running, and sharing segmented (V-mode and I-mode) programs.

### Format

```
SEG [ { pathname } ]
    [ -LOAD ]
```

### Argument and Option

|                 |                                                                                                                                                                                                  |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>pathname</i> | Specifies the name of the runfile.                                                                                                                                                               |
| -LOAD           | Invokes the SEG utility's LOAD subprocessor. The subprocessor displays a dollar-sign (\$) prompt and waits for a LOAD subcommand that loads an object file whose name usually has a .BIN suffix. |

### Executing Runfiles

Segment directories (also called SEG runfiles) are static V-mode and I-mode runfiles. To execute a SEG runfile, use the following command format:

```
SEG pathname
```

### Creating SEG Runfiles

To load (that is, create) a V-mode runfile from an object (binary) file, start the LOAD subprocessor with the following command line:

```
SEG -LOAD
```

For example,

```
OK, SEG -LOAD
[SEG Rev. T3.1-23.0 Copyright (c) 1991, Prime Computer, Inc.]
$
```

The runfile generated by the LOAD subprocessor uses the object file's basename and adds a .SEG suffix to produce the name of the output runfile.





- CATEGORY *acatname* Gives the target object the same access rights as an existing access category named *acatname*. It is not necessary to specify the .ACAT suffix.
- NO\_QUERY Suppresses verification of changes the command makes.

### **Access Control Lists**

An ACL is a list of users and the access rights granted to each user. The list may contain a maximum of 32 entries, but the list cannot exceed 160 characters, including blanks.

There are two types of ACLs:

- specific ACLs
- access categories

A **specific ACL** is an unnamed attribute of a file or directory. Specific ACLs do not exist as separate, named file system objects but are linked to the objects they protect. A specific ACL, therefore, does not appear when you issue the LD command.

An **access category** is a named file system object that has a separate existence and resides in a directory. The entryname of an access category has the suffix .ACAT (for example, GUARD.ACAT). The LD command lists access categories.

Each access category contains an ACL. The access category may protect any number of files, directories, and segment directories in the directory in which it resides. Thus, access categories provide an efficient way to group objects together for access control purposes.

The entries in an ACL define the rights that a user has when accessing a file or directory. Each entry is an ordered pair, in the following format:

*identifier:rights*

*identifier* is one of the following:

- A user ID that identifies a single user.
- A group name that identifies a set of users who are members of a particular access group. Your System or Project Administrator defines access groups for your system or project. The group name must begin with a period (.).
- The special identifier \$REST that identifies all users who are not listed by user ID or are not members of a group.



*rights* identify the privileges given to a user when using a particular file or directory. Specify *rights* by using one or more of the mnemonic codes listed below:

| <i>Code</i> | <i>Right</i> | <i>Applies to</i>     | <i>Allows the user to</i>                            |
|-------------|--------------|-----------------------|------------------------------------------------------|
| P           | Protect      | Directories           | Change accesses and attributes.                      |
| D           | Delete       | Directories           | Delete directory entries.                            |
| A           | Add          | Directories           | Add directory entries.                               |
| L           | List         | Directories           | Read directory contents.                             |
| U           | Use          | Directories           | Attach to directory.                                 |
| R           | Read         | Files                 | Read file contents.                                  |
| W           | Write        | Files                 | Change file contents.                                |
| X           | Execute      | Local EPFs            | Execute, but not read or copy a local EPF.           |
| O           | Owner        | Directories and files | Set read/write locks without P or ALL access rights. |
| ALL         | All          | Directories and files | All of the above.                                    |
| NONE        | None         | Directories and files | All access denied.                                   |

---

**Note**

Read (R) access includes Execute (X) access. Therefore, users who already have R access to a file do not need X access as well.

---

### Using SET\_ACCESS

To create an ACL for an object, use the format

**SET\_ACCESS** *target-pathname* [*acl*]

The result of the *acl* argument depends on the target's type and current protection according to the following rules:

- If *target-pathname* is a file or directory protected by neither a specific ACL nor an access category (protected by default by the parent directory's rights), SET\_ACCESS creates a specific ACL with the specified identifiers and rights (see Example 1).





### Copying Access Rights From Another File System Object

To copy the access rights of another file, directory, or access category, use the following command line:

```
SET_ACCESS target-pathname -LIKE reference-name
```

For example, user SAWYER gives the file RAFT the same protection as the file FENCE (from the previous example) in the following way:

```
OK, SET_ACCESS RAFT -LIKE FENCE
OK, LIST_ACCESS RAFT
```

```
ACL protecting "RAFT":
```

```
    SAWYER:  ALL
    $REST:   LUR
```

```
OK,
```

The files FENCE and RAFT now have specific ACLs with identical contents.

### Protecting a File System Object With an Existing Access Category

The -CATEGORY option adds a file or directory to the list of objects protected by an access category. The format is

```
SET_ACCESS target-pathname -CATEGORY categoryname
```

*categoryname* is an existing access category that is to provide protection for *target-pathname*.

For example, to protect the file RIVER with the existing access category named GUARD.ACAT, user SAWYER issues the following command:

```
OK, SET_ACCESS RIVER -CATEGORY GUARD
OK, LIST_ACCESS RIVER
```

```
ACL protecting "RIVER" (from access category "<TN>SAWYER>GUARD.ACAT")
:
```

```
    SAWYER:  ALL
    .GANG:   LUR
    $REST:   NONE
```

```
OK,
```

### Setting ACLs on the Current Directory

To use SET\_ACCESS on your current directory (that is, the directory to which you are currently attached), use the full pathname of the directory. For example, if the pathname of your current directory is <CITY>HOUSE>ROOM, use the command line

```
OK, SET_ACCESS HOUSE>ROOM MARY:ALL $REST:LUR
```



■ ■ ■ ■ ■ ■ ■ ■ ■ ■  
SET\_ASYNC

- TO *m*** Use following the **-LINE** option to configure a range of consecutively numbered lines with identical options. *m* specifies the last line number in a series beginning at the line number *n* specified with the **-LINE** option. *m* must be greater than *n*.
- HELP** Displays command syntax and a complete list of available options.

**-LINE Suboptions**

The options below are used to specify the characteristics of one or more than one asynchronous line. Enter these options after the **-LINE** option on the command line with the following syntax:

SET\_ASYNC **-LINE** *n* [**-TO** *m*] [*options*]

These options replace the octal parameters used by the obsolete AMLC command. To find out the SET\_ASYNC equivalent of a given AMLC command, use the interactive form of the CONVERT\_AMLC\_COMMANDS utility.

| <i>Suboption</i>             | <i>Function</i>                                                                                                                                                      |
|------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-DEFAULT</b>              | Sets all options to their default setting, as shown in Table 2-3. Desired exceptions to default settings may be specified after <b>-DEFAULT</b> on the command line. |
| <b>-LCHG</b>                 | Sets the line to full duplex (default).                                                                                                                              |
| <b>-NO_ECHO</b>              | Sets the line to half duplex.                                                                                                                                        |
| <b>-PROTOCOL <i>name</i></b> | Defines the format and relative timing of data. <i>name</i> may be any of the following. (TTY is the default.)                                                       |
|                              | ASD                    TTY8                                                                                                                                          |
|                              | TRAN                   TTY8HS                                                                                                                                        |
|                              | TT8BIT                TTYNOP                                                                                                                                         |
|                              | TTY                    TTYUPC                                                                                                                                        |

NTS lines support only TTY, TRAN, TTY8BIT, and TTYUPC; do not use ASD or TTYNOP on an NTS line. The obsolete protocols TTYHS, TRANHS, and TTYHUP are supported for older model 5054 AMLC controller boards.

---

**Note**

If you select the TRAN protocol, all characters normally pass transparently as data. However, if you select the SET\_ASYNC -XOFF option or if you set lword to bit 3 (recognize XOFF) with the AMLC command, then PRIMOS recognizes XON and XOFF as flow control characters, even if TRAN is in effect.

---

For descriptions of standard protocols, see the *System Administrator's Guide, Volume II: Communication Lines and Controllers*.

**-SPEED *value***

Sets the baud rate for the asynchronous line. *value* may be any of the following:

|       |          |      |            |
|-------|----------|------|------------|
| 50    | 200      | 2400 | 19200      |
| 75    | 300      | 3600 | CLOCK (2)  |
| 110   | 600      | 4800 | SPEEDA (3) |
| 134.5 | 1200 (1) | 7200 | SPEEDB (3) |
| 150   | 1800     | 9600 | SPEEDC (3) |

- (1) Supplied with the -DEFAULT option
- (2) Speed set with AMLCLK directive
- (3) Three jumper speeds set with the ASYNC JUMPER directive.

For further information on the ASYNC JUMPER and AMLCLK directives, see the *System Administrator's Guide, Volume II: Communication Lines and Controllers*.

**-STOP\_BITS *n***

Signals the receiving device to wait for the next character. *n* defines the number of stop bits to use, either 1 (default) or 2. All characters have 1 start bit, 1 parity bit, 7 information bits, and either 1 or 2 stop bits. The slower the transmission speed, the more stop bits are required to recognize the EOT. For this reason, an 11-bit character length and two stop bits are used for devices that operate at 110 baud.

**-PARITY *value***

Appends parity bits from the transmitting device to characters. -PARITY is used for error detection. *value* can be ODD, EVEN, or NONE (default). This option either sets the line parity to the desired setting or disables parity.

- CHAR\_LENGTH *n*** Sets the number of information and parity bits per character. *n* can be 5, 6, 7, or 8 (default). Character length can be adjusted for the nonstandard character sizes required by Baudot terminals, telex lines, or foreign devices. PRIMOS right-justifies the bits in a byte and sets the leftmost bits to zero. For more information, see the *System Administrator's Guide, Volume II: Communication Lines and Controllers*.
- LINE\_FEED** Echos a linefeed character for the RETURN key. This option is valid only when **-NO\_ECHO** (half duplex) is specified. **-NO\_LINE\_FEED** is the default.
- NO\_LINE\_FEED** Does not echo a linefeed character for RETURN. (Default)
- XOFF** Enables Ctrl-S and Ctrl-Q (**-XON** and **-XOFF**) to respectively stop and start the flow of data on the line from the host to the terminal. (Default)
- NO\_XOFF** Disables Ctrl-S and Ctrl-Q. This option is used for devices that cannot recognize these control key sequences.
- REVERSE\_XOFF** Enables Reverse Flow Control (RFC) for asynchronous lines. RFC sends XOFF characters to a device when the PRIMOS input ring buffer is 60% full. When the input ring buffer drops to 20% full, an XON character is sent to the device to indicate that transmission can resume.  
  
RFC also attempts to prevent DMQ input queue overruns for all ICS3 controllers and any ICS2 controllers that are not using BSC and ASYNC protocols. Choose this option only for lines connected to devices that can interpret XON and XOFF characters (such as PT45, PST 100, and PT200 terminals).
- NO\_REVERSE\_XOFF** Disables Reverse Flow Control for the line. (Default)
- DATA\_SENSE\_ENABLE** Enables the Data Set Sense (DSS) protocol, (reverse channel). This option is used for transmitting control information or for controlling the flow of data to devices that do not recognize XON/XOFF.
- NO\_DATA\_SENSE\_ENABLE** Disables the Data Set Sense (DSS) protocol (reverse channel). (Default)



- DATA\_SET\_SENSE**      Supports devices that toggle an RS-232-C pin (usually  
ready\_value            pin 8) to indicate when they are busy/ready instead of  
                             using XON/XOFF. The Data Set Sense (DSS) protocol  
                             sets ready\_value as either HIGH (pin signal raised) or  
                             LOW (pin signal lowered). The default is HIGH. Some  
                             devices use pins other than pin 8. If this is the case, ask  
                             your Customer Support Center to arrange your cables so  
                             that the Data set sense signal is wired into the pin used  
                             for carrier detect. If you use the -DSS option, you must  
                             also use -DSE and -NO\_DSE to enable and disable  
                             flow control. You must specify -NO\_XOFF as  
                             well. Data Set Sense is also referred to as buffered  
                             protocol or reverse channel protocol.
  
- DATA\_SET\_CONTROL**    Required for modems and port selectors to recognize  
                             when a block of information is transmitted. This option  
                             is ignored by terminals. Do not use -DSC on NTS lines.  
                             (Default)
  
- NO\_DATA\_SET\_CONTROL** Disables the -DSC option.
  
- ERROR\_DETECTION**      Used only for testing. When an input buffer overflows  
                             or when a parity error is detected, the incoming  
                             character is replaced with an ASCII NAK.
  
- NO\_ERROR\_DETECTION**   Prevents the line from sending an ASCII NAK character  
                             if an input parity or input buffer overflow error is  
                             sensed. (Default)
  
- SYSTEM**                Sets the line characteristics for the current line to system  
                             default settings that were in effect when you logged in.

---

**Note**

You can make changes only to your terminal line or to lines that you have assigned with the ASSIGN command. As a nonprivileged user, you cannot issue the SET\_ASYNC options -ASSIGNABLE, -SPEED\_DETECT, -NO\_SPEED\_DETECT, -LOOP\_LINE, -NO\_LOOP\_LINE, -DISLOG, -NO\_DISLOG, or USER\_NUMBER.

---

Table 2-2 lists the default settings given to a specified asynchronous line (or range of lines) when set using the -DEFAULT suboption to the -LINE option. The command line would have the following syntax:

SET\_ASYNC -LINE *n* [-TO *m*] -DEFAULT

Table 2-2. Default Settings for Asynchronous Lines

| <i>Line Option</i>           | <i>Default Setting</i>                                                                                                                                                                 |
|------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-ASSIGNABLE NO</b>        | Line is a login line.                                                                                                                                                                  |
| <b>-CHAR_LENGTH 8</b>        | Character length is 8 bits.                                                                                                                                                            |
| <b>-DATA_SET_CONTROL</b>     | Enables modems and port selectors to recognize when information is being transmitted.                                                                                                  |
| <b>-ECHO</b>                 | Full duplex line.                                                                                                                                                                      |
| <b>-LINE_FEED</b>            | Echo LINE FEED and carriage return for each RETURN.                                                                                                                                    |
| <b>-NO_DATA_SENSE_ENABLE</b> | Data Set Sense is disabled.                                                                                                                                                            |
| <b>-NO_DISLOG</b>            | Disables automatic disconnect.                                                                                                                                                         |
| <b>-NO_ERROR_DETECTION</b>   | NAK character is not placed in the input buffer when an input parity or input buffer overflow is detected.                                                                             |
| <b>-NO_LOOP_LINE</b>         | Line is not in loopback mode.                                                                                                                                                          |
| <b>-NO_REVERSE_XOFF</b>      | Reverse Flow Control is not enabled.                                                                                                                                                   |
| <b>-NO_SPEED_DETECT</b>      | Disables Auto Speed Detect.                                                                                                                                                            |
| <b>-PARITY NONE</b>          | Line parity is disabled.                                                                                                                                                               |
| <b>-PROTOCOL TTY</b>         | Line uses the terminal protocol.                                                                                                                                                       |
| <b>-SPEED 1200</b>           | Line speed is 1200 bits per second.                                                                                                                                                    |
| <b>-STOP_BITS 1</b>          | One stop bit.                                                                                                                                                                          |
| <b>-USER_NUMBER (n + 2)</b>  | Associates the buffers for USER_NUMBER (n + 2) with the physical line number n specified in the -LINE n option. For example, the default buffer associated with -LINE 16 is buffer 18. |
| <b>-XOFF</b>                 | Ctrl-S stops and Ctrl-Q resumes the flow of data to the terminal.                                                                                                                      |

### SET\_ASYNC Example

The following example illustrates the use of SET\_ASYNC to disable Ctrl-S and Ctrl-Q (-XON and -XOFF):

```
OK, SET_ASYNC -DISPLAY
[SET_ASYNC Rev 23.3.0 Copyright (c) 1992 Prime Computer, Inc.]
```

```
LINE = 0
  PARity           = NONE      ECHO
  PROtocol         = TTY       XOFF
  SPEED            = 9600      NO_LOOP
  Stop_Bits        = 1         Line_Feed
  Char_Length      = 8         NO_Data_Sense_Enable
  REVerse_XOFF     = OFF       NO_ERROR_DETECTION
  ASSiGNable       = NO        NO_Speed_Detect
  Data_Set_Sense   = LOW       DISLOG
  Owner Process    = 2         Data_Set_Control
```

```
OK, SET_ASYNC -LINE 0 -NOXOFF
[SET_ASYNC Rev 23.3.0 Copyright (c) 1992 Prime Computer, Inc.]
OK, SET_ASYNC -DISPLAY
[SET_ASYNC Rev 23.3.0 Copyright (c) 1992 Prime Computer, Inc.]
```

```
LINE = 0
  PARity           = NONE      ECHO
  PROtocol         = TTY       NO_XOFF
  SPEED            = 9600      NO_LOOP
  Stop_Bits        = 1         Line_Feed
  Char_Length      = 8         NO_Data_Sense_Enable
  REVerse_XOFF     = OFF       NO_ERROR_DETECTION
  ASSiGNable       = NO        NO_Speed_Detect
  Data_Set_Sense   = LOW       DISLOG
  Owner Process    = 2         Data_Set_Control
```

OK,

For more information on the SET\_ASYNC command, see the *System Administrator's Guide, Volume II: Communication Lines and Controllers*.

■ ■ ■ ■ ■ ■ ■ ■ ■ ■  
*SET\_DELETE*

## SET\_DELETE

SET\_DELETE sets the delete switch on a file system object.

When you create a file or a segment directory, its delete switch is set to allow you to delete it without your verification. The SET\_DELETE command changes this switch so that PRIMOS must first ask for your verification before it can delete the object. This protection, which is similar to the default protection on directories, assures that you cannot unintentionally delete the object.

SET\_DELETE works only for objects in ACL-protected directories. Before you can use this command, you must have Delete (D) access to the directory that contains the object.

### Format

```
SET_DELETE pathname [ { -PROTECT  
                        { -NO_PROTECT } } ]
```

### Argument and Options

|                 |                                                                                                                                                                                                                                                                                        |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>pathname</i> | Specifies the file, directory, or segment directory to receive delete-protection. Access categories cannot be protected. When you use this command on directories and segment directories, they receive double protection because they still retain their original default protection. |
| -PROTECT        | Sets the delete-protection on <i>pathname</i> .                                                                                                                                                                                                                                        |
| -NO_PROTECT     | Removes the delete-protection from <i>pathname</i> .                                                                                                                                                                                                                                   |

### Usage

For Example, to set delete protection on the file SPECS use the following command line:

```
OK, SET_DELETE SPECS -PROTECT
```

You can use the DELETE command to check if the file is protected (make sure to enter NO to the query if you wish to keep the file!):

```
OK, DELETE SPECS  
"SPECS" protected, ok to force delete? NO  
OK,
```

PRIMOS asks for your verification twice when you attempt to delete a directory that is delete-protected.:

```
OK, DELETE SUBDIR.3 -REPORT
Ok to delete directory "SUBDIR.3"? YES
"SUBDIR.3" protected, ok to force delete? YES
"SUBDIR.3" deleted.
OK,
```

If you answer N or NO (in uppercase or lowercase) at either query, the directory is not deleted.

To remove the delete protection on the file and directory in the above example enter:

```
OK, SET_DELETE (SPECS SUBDIR.3) -NO_PROTECT
OK,
```

## SET\_QUOTA

SET\_QUOTA sets the quota on a directory to a specific number of records.

A quota is the maximum number of records that a directory can contain. At most sites, only the System Administrator can set quotas on top-level directories. With SET\_QUOTA, users can set quotas on their own subdirectories, usually as a check on their own storage use.

To set a quota on a directory or subdirectory, you must have either Protect (P) access (for access control systems) or Owner access (for password systems) on the parent directory. If you use SET\_QUOTA without having these rights, the quota is not set and the following error message is returned:

```
Insufficient access rights. directory-name (set_quota)
```

### Format

```
SET_QUOTA directory-pathname [-MAX n]
```

### Argument and Option

*directory-pathname*

Specifies the directory or subdirectory on which you are setting a quota. If you are setting a quota on a subdirectory within your current directory, you need to specify only the subdirectory's entryname (the final element of its pathname).



To set the attributes of *pathname*, specify one or more of the options listed below. If you do not specify an option, SET\_RBF lists the file's existing attributes.

- AICLK**                      Specifies a runtime check of the save date/time stamp when after-imaging is enabled for the file. This option is meaningful only if the check was previously disabled.
- AIRC**                        Turns on after-image recovery and enables before-image recovery.
- BIRC**                        Enables the force-writing of before-images for before-image recovery.
- LOCK**                        Restricts access to the file. A locked file cannot be accessed by any run-units and can be unlocked only by the ROAM Administrator who issued the lock.
- NO\_AICLK**                    Disables the runtime check of the save date/time stamp.
- NO\_AIRC**                     Turns off after-image recovery.
- NO\_BIRC**                    Disables the force-writing of before-images for before-image recovery and turns off after-imaging.
- NO\_LOCK**                    Unlocks a file. Only the ROAM Administrator who locked a file can unlock it.
- NO\_TRANS\_ROLLBACK**      Disables before-image recovery and transactional concurrency control.
- TRANS\_ROLLBACK**        Enables before-image recovery and transactional concurrency control.
- WRITE\_ACCESS**            Permits runtime updates to an activated file that has not been rolled forward.

### **Usage**

For further information on SET\_RBF and ROAM files, see the *ROAM Administrator's Guide*.

## SET\_SEARCH\_RULES

SET\_SEARCH\_RULES modifies active search lists or adds new search lists to your current environment. Use the LIST\_SEARCH\_RULES (LSR) command to display the contents of your active search lists.

### Format

$$\text{SET\_SEARCH\_RULES} \left\{ \begin{array}{l} \text{template} \left[ \begin{array}{l} \text{--LIST\_NAME listname} \\ \text{--NO\_SYSTEM} \end{array} \right] \\ \text{--DEFAULT listname} \\ \text{--HELP} \end{array} \right\}$$

### Arguments and Options

|                                   |                                                                                                                                                                                                                                                                                                                 |
|-----------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>template</i>                   | Specifies the pathname of a file containing search rules. Without options <i>template</i> is used as the name of the list. To create a search list with a name different than <i>template</i> , use the <code>--LIST_NAME</code> option. It is recommended that you use the .SR suffix on templates you create. |
| <code>--DEFAULT listname</code>   | Resets the named search list by reloading the contents of the specified system default rules template file corresponding to <i>listname</i> .                                                                                                                                                                   |
| <code>--LIST_NAME listname</code> | Names the added search list <i>listname</i> rather than <i>template</i> . <i>listname</i> may be a maximum of 22 characters.                                                                                                                                                                                    |
| <code>--NO_SYSTEM</code>          | Contents of SEARCH_RULES*> <i>template</i> .SR is not prepended to the list. Without this option PRIMOS looks for a file called <i>template</i> .SR in the SEARCH_RULES* directory. If <i>template</i> .SR is not found an error message is displayed, and the list is not created.                             |
| <code>HELP</code>                 | Displays the command syntax. The Help facility is also displayed if PRIMOS detects an error while parsing the command.                                                                                                                                                                                          |

### Search Rules Overview

Each time your environment is initialized during login or reinitialized using the Initialize Command Environment (ICE) command PRIMOS automatically constructs a search list and places it in the system memory. PRIMOS builds a search list for each of the search rules files (called templates) it finds in the SEARCH\_RULES\*



directory. Every file in the SEARCH\_RULES\* directory with a .SR suffix is made into a search list with the same name as the file (without the .SR suffix).

When PRIMOS is installed the following five search rules template files containing default values are placed in the SEARCH\_RULES\* directory:

```
ATTACH$
BINARY$
COMMAND$
INCLUDE$
ENTRY$ (only ENTRY$ is required for system operation)
```

---

**Note**

If you have FS\_RECOVER installed on your system, the default search lists also contain the AUTOPSY and MAPS search lists. These search lists are only used by the system and do not need to be included in your search rules.

---

### **Search Rules Templates**

A search rules template file is an ASCII file created with a text editor, such as ED or EMACS (it is recommended that the name of this file have a .SR suffix, but it is not required). Each template file contains a list of directories or a list of disk partitions, one per line.

PRIMOS provides several variables and keywords that can be used in entries. The variables that may be used are:

|                   |                                                                                                                                         |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------|
| [HOME_DIR]        | Current directory                                                                                                                       |
| [ORIGIN_DIR]      | User's origin directory (initial attach point)                                                                                          |
| [REFERENCING_DIR] | Pathname specified as an option to a program that uses the search list, such as LIST_SEARCH_RULES. This variable is ignored if not set. |

The following general-purpose keywords can also be used in a search list (these keywords are not command options):

|                         |                                                                                                                                                                                                                                                                          |
|-------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -INSERT <i>template</i> | Inserts the contents of <i>template</i> at that point in the file.                                                                                                                                                                                                       |
| -SYSTEM                 | Contents of SEARCH_RULES*> <i>template</i> .SR are substituted in that place when the list is constructed. The -SYSTEM keyword has meaning only when the list being created has the same name as one of the .SR files in the SEARCH_RULES* directory, such as COMMAND\$. |

Additional special-purpose keywords are provided for specific search lists. Search rule keywords are further described in the *Advanced Programmer's Guide II: File System*.



---

**Note**

Reinitializing the login process by issuing an ICE command resets all the search lists to the system defaults (unless changes are made by SSR commands in the login CPL or other individualized login programs).

---

**SET\_SEARCH\_RULES Examples**

**Example 1**

Adding the directory MYPROGS to the COMMAND\$ search list. The template file called MYRULES.SR contains the following text:

```
/* search list template for my executables (optional comment)
[origin_dir]>myprogs
```

The command to activate the list is (using command and option abbreviations) :

```
OK, SSR MYRULES.SR -LNAM COMMAND$
```

You can display the active COMMAND\$ list using the LIST\_SEARCH\_RULES command:

```
OK, LIST_SEARCH_RULES COMMAND$
```

```
List: COMMAND$
```

```
Pathname of template: <DISKP>MYUFD>MYRULES
```

```
-public
CMDNCO
TOOLS
[ORIGIN_DIR]>myprogs
OK,
```

Note that, by default, the rules listed in the system's COMMAND\$.SR file is prepended to your list. If your search list contains all the directories you use, you may suppress the inclusion of the system COMMAND\$ rules with the command line:

```
OK, SSR MYRULES.SR -LIST_NAME COMMAND$ -NO_SYSTEM
```







- DISABLE\_GROUP**      Prevents users from watching your input and output stream via the .WATCH\$ ACL group privilege. This option can only be used by members of the .WATCH\$ ACL group. If a member user is watching you via .WATCH\$ access when you issue this option, that watch session is immediately terminated. Once disabled, ACL-group access to your terminal remains disabled until you log out.
- DISABLE\_USER**      Prevents any user from watching your input and output stream, unless that user has access via ACL-group access or System Administrator access. If a privileged user is watching when this option is used, that watch session is immediately terminated. Once disabled, user access to your terminal remains disabled until you issue a SHOW -ALL command, or until the end of the login session.
- LIST**                  Displays the System Administrator, ACL-group, and user access watch privileges.
- HELP**                  Displays command options.

**Usage**

SHOW grants permission to a user either for the duration of your login session or until the you disable access with one of the disable options. The system allows only one user to watch at a time. Issuing a subsequent SHOW commands replaces the access granted by the previous SHOW command.

See also WATCH.





## Output

The information returned by SIZE depends upon the type of file system object specified by the pathname, as follows:

|                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|--------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>File</b>              | Number of records allocated to the file (each record contains 1024 halfwords or 2048 bytes), type of file, name of file, and number of halfwords containing data. The logical and physical sizes of a CAM file are displayed. (The logical size is the number of records that contain data, and the physical size is the number of records allocated to the file.)<br><br>With the <code>-NORM</code> option, records contain 440 halfwords (880 bytes). |
| <b>Directory</b>         | Number of top-level entries (files, segment directories, sub-directories, and access categories), type of directory (" <code>pwd</code> " for password or " <code>acl</code> " for access control list), name of directory, and size in halfwords.                                                                                                                                                                                                       |
| <b>Segment Directory</b> | Number of top-level entries, type of directory file (sam SEGDIR or dam SEGDIR), name of directory, and total number of entries it can hold. To get the size of the directory in halfwords, multiply the last number by 2.                                                                                                                                                                                                                                |
| <b>Access Category</b>   | Number of entries and category name.                                                                                                                                                                                                                                                                                                                                                                                                                     |

---

### Note

Because size is logically an attribute of an object, reading an object's size with SIZE updates the parent directory's Date/Time Accessed, not the object's own Date/Time Accessed. This is also the case with an object's other attributes. However, using SIZE from a pre-Rev. 20.0 system on an object in a Rev. 20.0 or later directory updates the DTA of both the parent directory and the object.

---

When SIZE needs to print a number greater than 32767 before the word "entries" or "records", it allows 12 full character positions for the number instead of the normal 6 character positions.

The SIZE command cannot determine the size of an open file, but will display a message reporting the existence of an open file and its name similar to the following:

```
File in use.      "FILEA" (SIZE)
```

### SIZE Examples

The following examples illustrate the use and output of SIZE with non-wildcard names. See the following section for using SIZE with wildcard names.

**Example 1: Using SIZE on a file.**

```
OK, SIZE STAR.FILE
    51 records in sam file "STAR.FILE" (51550 halfwords)
OK,
```

```
OK, SIZE CTYP.FILE
    10 logical records in cam file "CTYP.FILE" (505 halfwords)
    11 physical records.
OK,
```

**Example 2: Using SIZE with the -NORM option on the same file as in Example 1.**

```
OK, SIZE STAR.FILE -NORM
    118 records in sam file "STAR.FILE" (51550 halfwords)
OK,
```

**Example 3: Using SIZE on a directory.**

```
OK, SIZE <SOLAR>PLANET
    128 entries in acl UFD "<SOLAR>PLANET" (2036 halfwords)
OK,
```

**Example 4: Using SIZE on a segment directory.**

```
OK, SIZE PROG3.SEG
    7 entries in sam SEGDIR "PROG3.SEG" (65 total)
OK,
```

**Example 5: Using SIZE on an access category.**

```
OK, SIZE GATE.ACAT
    3 entries in access cat. "GATE.ACAT"
OK,
```

## Using Wildcard Names With SIZE

SIZE uses the standard wildcard characters as described in Chapter 4. To obtain the size of all the objects in the current user's current directory, specify a double at sign (@@) as the wildcard name, as in the following example.

```
OK, SIZE @@
    1 record in sam file "LOGIN.CPL" (51 halfwords)
    1 record in sam file "LOGIN.ABBREV" (647 halfwords)
    3 entries in acl UFD "BOOK" (70 halfwords)
    6 records in dam file "BB.LIST" (5555 halfwords)
    5 entries in sam SEGDIR "AREA.SEG" (65 total)
    2 entries in access cat "GUARD.ACAT"
    0 entries in pwd UFD "PAS_DIR" (23 halfwords)
OK,
```

To obtain the size of all the objects in or below the current directory, use the wildcard -WALK\_FROM option, as in the following example where SIZE starts two levels down:

```
OK, SIZE *>@@>@@ -WALK_FROM 2
    1 record in sam file "**>BOOK>CHAP_1" (545 halfwords)
    1 record in dam file "**>BOOK>CHAP_2" (629 halfwords)
    1 record in sam file "**>BOOK>INDEX" (112 halfwords)
OK,
```

Because the PRIMOS command processor does not treewalk segment directories, the wildcard name and option in the previous example cannot produce the sizes of files in the segment directory AREA.SEG. To obtain the size of segment directory files, use the pathname format

**\*>segment-directory-name>@@**

as in the following example:

```
OK, SIZE *>AREA.SEG>@@
    1 record in sam file "**>AREA.SEG>0" (971 halfwords)
    1 record in dam file "**>AREA.SEG>1" (0 halfwords)
    2 records in sam file "**>AREA.SEG>2" (2048 halfwords)
    2 records in sam file "**>AREA.SEG>34" (2048 halfwords)
    2 records in sam file "**>AREA.SEG>35" (2048 halfwords)
OK,
```

For more information about PRIMOS file types and their characteristics see the *Advanced Programmers Guide II: File System*.

See also LD.

.....  
SLIST

## SLIST

SLIST displays the contents of a file at the user's terminal.

### Format

SLIST *pathname*

### Argument

*pathname* Specifies the name of the file. Use a simple filename if the file is in your current directory.

### Usage

The SLIST display scrolls continuously. If TERM -XOFF is enabled, use Ctrl-S to freeze the SLIST display. Use Ctrl-Q to continue the display or Ctrl-P to abort the command. On PST 100 terminals, you can also press PAUSE to halt the SLIST display and then press PAUSE again to continue the display. On a PT200/250 terminal, the Stop key will stop and start the display alternately.

## SNADSC

SNADSC invokes the PRIME/SNA Interactive Terminal Emulation program, which makes a supported Prime terminal emulate an IBM 3278 Display Station.

### Format

SNADSC [*station-name*] [-LOGON *logon-command*] [-TERMINAL\_TYPE *type*]

### Argument and Options

*station-name* Specifies a device name or device group name included in the PRIME/SNA Interactive configuration. The PRIME/SNA Administrator usually supplies this name to qualified users. If you invoke SNADSC without the station name, the SNADSC program prompts you for it.

-LOGON *logon-command* Specifies an IBM logon command for a particular application on the IBM host. If the IBM logon contains embedded spaces, it must be enclosed in single quotation marks.

**-TERMINAL\_TYPE** Specifies the type of terminal from which you are invoking SNADSC. Currently, the only supported terminal type is PT200/250, which is also the default.

**Usage**

For further information, see the *PRIME/SNA Interactive Terminal User's Guide*.

**SORT**

SORT sorts as many as 20 input files into a single output file. SORT performs two types of sorts: a default sort operation and an optional merge operation.

**Format**

$$\text{SORT} \left[ \begin{array}{l} \text{-MERGE} \\ \left\{ \begin{array}{l} \text{-TAG} \\ \text{-NONTAG} \end{array} \right\} \end{array} \right] \left[ \begin{array}{l} \text{-BRIEF} \\ \text{-SPACE} \end{array} \right]$$

**Options**

**-BRIEF** Does not display SORT prompts at the user's terminal. See the **-BRIEF** Option section below for further explanation.

**-MERGE** Specifies a merge operation, which merges a maximum of 11 presorted files into one sorted output file. SORT prompts you for the number of extra files to be merged and then prompts for their names, one per line. See the **-MERGE** Option section below for further explanation.

**-NONTAG** Specifies a NONTAG sort. Recommended for smaller or well-ordered files. A NONTAG sort stores each input record with its sort key in the work file, which eliminates the search for records after merging but requires more disk space.

**-SPACE** Deletes blank lines from the output file.

**-TAG** Specifies a TAG sort (default). Recommended for large or unordered files. A TAG sort stores input records separate from the key data. After all keys have been sorted and merged, the corresponding records are then located and output. This option cannot be specified if **-NONTAG** is used.

### **Input File Types**

SORT can process the four types of files listed below. The file types are defined by the records they contain. SORT has two default file types: compressed ASCII and variable length (also called binary). The type used depends on the type of key you select. SORT defaults to the variable length on integer or real keys; otherwise, the file type defaults to ASCII.

|                                |                                                                                                                                                                                                                                                                        |
|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| COMPRESSED<br>SOURCE (ASCII)   | Blank compressed record delimited by a NEWLINE character (^212). Source lines cannot contain data that may be interpreted as a blank compression indicator (^221) or a NEWLINE.                                                                                        |
| UNCOMPRESSED<br>SOURCE (ASCII) | Uncompressed record delimited by a NEWLINE character (^212). Source lines cannot contain data that may be interpreted as a NEWLINE. These files can also contain packed decimal data.                                                                                  |
| VARIABLE LENGTH                | Record stored with length (in words) stored in first word. (The first word is not included in the word count.)                                                                                                                                                         |
| FIXED LENGTH                   | Record containing data only, no length information. The length must be specified using the -INLENGTH or -OUTLENGTH keywords. If a NEWLINE character is appended to each record so that the file can be edited, that character must be included in the character count. |

### **Sort Operation**

The sort operation sorts a maximum of 20 unsorted input files into one sorted file. The input files can be sorted in ascending or descending (reverse) order using a maximum of 64 keys. SORT performs a stable sort; that is, it preserves the order of input for records with equal keys.

The sort operation works by comparing keys. A key is a field in a line of an input file. (A key can also be thought of as a portion of the line.) The data in the key is compared against the data in the same key of all the other lines in the input file. If more than one input file exists, the key is also compared against the same keys of all the input files.

For example, if you want to rearrange the lines in an ASCII file in alphabetical order according to the first word of the line, then the key for that sort might be the first 10 characters of each line. During the sort operation, the first 10 characters of each line would be compared to the first 10 characters of the other lines. At the end of the operation, SORT would write in alphabetical order each entire line (not just the key) into the output file.

### Sort Keys

Each key has a length and a type. The length begins at a certain column of the line and continues to a subsequent column of the same line. For example, the key might start at column 1 of a line and end at column 10. Or, if you wanted to sort on a middle part of a line, the key might start at column 40 and end at column 60.

The type of a key tells SORT what type of data (alphanumeric or numeric) is in the key and how the data is stored. ASCII files (both compressed and uncompressed) can be sorted on the following seven types of keys: A or AU for alphanumeric data, and LE, LS, TE, TS, or U for numeric data (default is A). Variable-length and fixed-length files can use any key type. Table 2-3 explains each key type in detail.

You can sort on as many as 64 keys for a line. The default number of keys is 1. If you are sorting on only one key that begins at column 1 and ends at the last possible column of a line (for example, 80), then the line is compared as a whole.

*Table 2-3. SORT Key Types*

| <i>Code</i> | <i>Key Type</i>                     | <i>Definition</i>                                                                                                                                       |
|-------------|-------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------|
| A           | ASCII (default)                     | Character strings, stored one character per byte. Their length is limited only by the length of the record. Sorted by ASCII collation, uppercase first. |
| AU          | ASCII, uppercase and lowercase sort | Storage is identical to regular ASCII. Lowercase characters are sorted as uppercase, but are put into the output file as lowercase.                     |
| D           | Double-precision real               | Length is 8 bytes; range is + ( 10**(-9902) through 10**(9825) ).                                                                                       |
| F           | Single-precision real               | Length is 4 bytes; range is + ( 10**(-38) through 10**(38) ).                                                                                           |
| I           | Single-precision integer (short)    | Length is 2 bytes; range is -32767 through +32767.                                                                                                      |
| J           | Double-precision integer (long)     | Length is 4 bytes; range is -(2**(31)) through +(2**(31)-1).                                                                                            |

| <i>Code</i>  | <i>Key Type</i>                       | <i>Definition</i>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |              |                 |                 |   |        |        |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|--------------|---------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------|-----------------|-----------------|---|--------|--------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| LE           | Numeric ASCII, leading embedded sign  | <p>One digit per byte. Alphabetic characters may represent digits, as shown in the table below. The first character represents both a digit and the sign of the field (for example, L579 represents -3579).</p> <table> <thead> <tr> <th><i>Digit</i></th> <th><i>Positive</i></th> <th><i>Negative</i></th> </tr> </thead> <tbody> <tr> <td>0</td> <td>+ or {</td> <td>} or -</td> </tr> <tr> <td>1</td> <td>A</td> <td>J</td> </tr> <tr> <td>2</td> <td>B</td> <td>K</td> </tr> <tr> <td>3</td> <td>C</td> <td>L</td> </tr> <tr> <td>4</td> <td>D</td> <td>M</td> </tr> <tr> <td>5</td> <td>E</td> <td>N</td> </tr> <tr> <td>6</td> <td>F</td> <td>O</td> </tr> <tr> <td>7</td> <td>G</td> <td>P</td> </tr> <tr> <td>8</td> <td>H</td> <td>Q</td> </tr> <tr> <td>9</td> <td>I</td> <td>R</td> </tr> </tbody> </table> | <i>Digit</i> | <i>Positive</i> | <i>Negative</i> | 0 | + or { | } or - | 1 | A | J | 2 | B | K | 3 | C | L | 4 | D | M | 5 | E | N | 6 | F | O | 7 | G | P | 8 | H | Q | 9 | I | R |
| <i>Digit</i> | <i>Positive</i>                       | <i>Negative</i>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |              |                 |                 |   |        |        |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 0            | + or {                                | } or -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |              |                 |                 |   |        |        |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 1            | A                                     | J                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |              |                 |                 |   |        |        |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 2            | B                                     | K                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |              |                 |                 |   |        |        |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 3            | C                                     | L                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |              |                 |                 |   |        |        |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 4            | D                                     | M                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |              |                 |                 |   |        |        |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 5            | E                                     | N                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |              |                 |                 |   |        |        |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 6            | F                                     | O                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |              |                 |                 |   |        |        |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 7            | G                                     | P                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |              |                 |                 |   |        |        |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 8            | H                                     | Q                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |              |                 |                 |   |        |        |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 9            | I                                     | R                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |              |                 |                 |   |        |        |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| LS           | Numeric ASCII, leading separate sign  | Numbers preceded by + or - indicate positive or negative value. (A blank space is treated as a positive sign.)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |              |                 |                 |   |        |        |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| PD           | Packed decimal                        | A four-bit nibble represents each digit; the number ends with a sign nibble. A negative sign is represented by hex D in the sign nibble; any other value in the sign nibble indicates a positive number. A packed field must have an odd number of digits plus the sign; since they are stored two nibbles (digit or sign) per byte, the result is a full number of bytes. Packed decimal keys may be no more than 63 digits plus the sign.                                                                                                                                                                                                                                                                                                                                                                             |              |                 |                 |   |        |        |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| TE           | Numeric ASCII, trailing embedded sign | Same as LE, except that the last digit carries the sign (for example, 357R represents -3579).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |              |                 |                 |   |        |        |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| TS           | Numeric ASCII, trailing separate sign | Same as LS, except that the + or - follows the number.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |              |                 |                 |   |        |        |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| ..           | Numeric ASCII, unsigned               | Like plain ASCII. These are stored one digit per byte, and are limited only by the length of the record.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |              |                 |                 |   |        |        |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| II           | Unsigned integer                      | Length is 2 bytes; range is 0 through 65535, inclusive.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |              |                 |                 |   |        |        |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |



## SORT Prompts

During a sort operation, SORT prompts you for information at two places. The first SORT prompt is

```
SORT PROGRAM PARAMETERS ARE:
  INPUT TREE NAME -- OUTPUT TREE NAME FOLLOWED BY
  NUMBER OF PAIRS OF STARTING AND ENDING COLUMNS.
```

At this prompt, enter the name of the input file (INPUT TREE NAME), the name of the output file (OUTPUT TREE NAME), and the number of keys for the sort (NUMBER OF PAIRS OF STARTING AND ENDING COLUMNS). If you are sorting on one key, you need not enter 1 because that is the default.

Enter the information on the line at which the cursor is positioned. If you are sorting only one input file, supply the two filenames as a simple parameter list. The following parameter list specifies two sort keys, with ALPHA.IN as the input file and BETA.OUT as the output file:

```
ALPHA? I B: - 1 1
```

Use file keywords to specify two or more input files and the number of keys. For example:

```
INPUTFILE ALPHA.BETA.GAMMA.DELTA.EPSILON SORTKEYS 2
```

For more on file keywords see the section Using Keywords below.

---

### **Caution**

Specifying identical names for input and output files is not recommended because the disk space used by the file becomes free (and hence vulnerable) during the sort. If the space is taken over by another user during this time, a "disk full" error — and loss of the file being sorted — may result.

---

■ ■ ■ ■ ■ ■ ■ ■ ■ ■  
SORT

The second SORT prompt is

INPUT PAIRS OF STARTING AND ENDING COLUMNS  
ONE PAIR PER LINE - - SEPARATED BY A SPACE.  
FOR REVERSE SORTING ENTER "R" AFTER DESIRED  
ENDING COLUMN - - SEPARATED BY A SPACE.  
FOR A SPECIFIC DATA TYPE ENTER THE PROPER CODE  
AT THE END OF THE LINE - - SEPARATED BY A SPACE.  
"A" - ASCII  
"I" - SINGLE PRECISION INTEGER  
"F" - SINGLE PRECISION REAL  
"D" - DOUBLE PRECISION REAL  
"J" - DOUBLE PRECISION INTEGER  
"U" - NUMERIC ASCII, UNSIGNED  
"LS" - NUMERIC ASCII, LEADING SEPARATE SIGN  
"TS" - NUMERIC ASCII, TRAILING SEPARATE SIGN  
"LE" - NUMERIC ASCII, LEADING EMBEDDED SIGN  
"TE" - NUMERIC ASCII, TRAILING EMBEDDED SIGN  
"PD" - PACKED DECIMAL  
"AU" - ASCII, UPPER & LOWER CASE SORT EQUAL  
"UI" - UNSIGNED INTEGER  
DEFAULT IS ASCII.

At this prompt, you must enter the starting and ending columns of each key.  
Optionally, you can specify a reverse sort on that key and the data type of the key.

Ascending sort is the default and need not be specified. The default for the data type is A for ASCII. Enter the information for each key on a separate line with either a parameter list or with the Key keywords listed in the section below called Using Keywords. For example, to sort on the first five characters, and then secondarily on the tenth through fifteenth characters (which are unsigned numeric ASCII characters) in descending order, the parameters you enter after the prompt would be:

```
1 5  
10 15 P D
```

After the sort is completed, SORT displays the number of passes needed for the sort and the number of items (lines) placed in the output file:

BEGINNING SORT

```
PASSES      2      ITEMS      520
```

```
[SORT-T3.0-23.0]  
OK,
```

### ***SORT Example Using Parameter Lists***

The following example of a sort operation uses ALPHA.IN as the input file and BETA.OUT as the output file. The file is sorted on two keys. The first key, which contains alphanumeric data, is columns 1 to 20 of each line. The second key, which is columns 30 to 40 of the line, contains numeric data and will be sorted in reverse order.

```
OK, SORT
SORT PROGRAM PARAMETERS ARE:
  INPUT TREE NAME - - OUTPUT TREE NAME FOLLOWED BY
  NUMBER OF PAIRS OF STARTING AND ENDING COLUMNS.
ALPHA.IN BETA.OUT 2
  INPUT PAIRS OF STARTING AND ENDING COLUMNS
  ONE PAIR PER LINE - - SEPARATED BY A SPACE.
  FOR REVERSE SORTING ENTER "R" AFTER DESIRED
  ENDING COLUMN - - SEPARATED BY A SPACE.
  FOR A SPECIFIC DATA TYPE ENTER THE PROPER CODE
  AT THE END OF THE LINE - - SEPARATED BY A SPACE.
  "A" - ASCII
  "I" - SINGLE PRECISION INTEGER
  "F" - SINGLE PRECISION REAL
  "D" - DOUBLE PRECISION REAL
  "J" - DOUBLE PRECISION INTEGER
  "U" - NUMERIC ASCII, UNSIGNED
  "LS" - NUMERIC ASCII, LEADING SEPARATE SIGN
  "TS" - NUMERIC ASCII, TRAILING SEPARATE SIGN
  "LE" - NUMERIC ASCII, LEADING EMBEDDED SIGN
  "TE" - NUMERIC ASCII, TRAILING EMBEDDED SIGN
  "PD" - PACKED DECIMAL
  "AU" - ASCII, UPPER & LOWER CASE SORT EQUAL
  "UI" - UNSIGNED INTEGER
  DEFAULT IS ASCII.
1 20
30 40 R TS
BEGINNING SORT
```

```
PASSES      2      ITEMS      520
```

```
[SORT-T3.0-23.0]
OK,
```

After the sort is completed, BETA.OUT contains the same lines as ALPHA.IN, except that they are in alphabetical order. If two or more lines have the same data in the first key, they are entered in reverse order according to the data in the second key.

The following example shows the same sort operation using the **-BRIEF** option:

```
OK, SORT -BRIEF
ALPHA IN BETA.OUT 2
1 20
30 40 R TS

BEGINNING SORT

      PASSES      2      ITEMS      520

[SORT-T3.0-23.0]
OK,
```

### Using Keywords

For more complicated sort operations, use keywords instead of simple parameter lists to enter file and key information.

There are two types of keywords: file keywords, used at the first SORT prompt; and key keywords used at the second SORT prompt.

### File Keywords

The keywords listed below specify multiple input filenames and types or provide SORT with information about record length. The keywords must be on a single line, but can be in any order.

#### *File Keyword*

**-INPUTFILE** *name*

**-OUTPUTFILE** *name*

**-KEYS** *n*

**-INTYPE** { COMPRESSED  
UNCOMPRESSED  
FIXED  
VARIABLE }

#### *Function*

Specifies a file to be sorted. The name can be a pathname of no more than 80 characters. Repeat this keyword for each input file.

Creates a file to hold the sorted output. Only one output file per sort is allowed.

Specifies the number of keys for the sort. *n* is from 1 to 64, inclusive.

Specifies the type of file to be sorted. All input files must be of the same type. If you do not give this keyword, a default file type is taken from the key type.

- OUTTYPE** *type*


Specifies the file type for the output file. *type* is one of the four types of the **-INTYPE** keyword. If you do not specify **-OUTTYPE**, the output file is the same type as the input file(s).
- INLENGTH** *n*


Specifies the maximum length of the input records (in bytes). The maximum value of *n* is 32760, which is also the default. You must give this keyword for fixed-length records.
- OUTLENGTH** *n*


Specifies the maximum length for records in the output file. Default is the length of input records. If you specify a fixed-length record output file, you must also specify its record length.

### Key Keywords

At the second SORT prompt, use the keywords listed below to specify information about the keys. The keywords must be on a single line, but can be in any order.

| <i>Key Keyword</i>       | <i>Function</i>                                                                                                                    |
|--------------------------|------------------------------------------------------------------------------------------------------------------------------------|
| <b>-START</b> <i>n</i>   | Uses <i>n</i> to specify the first column of the key.                                                                              |
| <b>-END</b> <i>n</i>     | Uses <i>n</i> to specify the last column of the key.                                                                               |
| <b>-DESCENDING</b>       | Requests a sort in descending (reverse) order.                                                                                     |
| <b>-TYPE</b> <i>code</i> | Specifies a key type for the sort. <i>code</i> is a key code from Table 2-3.                                                       |
| <b>-EBCDIC</b>           | Requests that the EBCDIC collating sequence, rather than the ASCII sequence, be used for sorting. Use only with A or AU key types. |

### Example of Using Keywords

The following example uses file and key keywords to specify information for a sort operation (the `-BRIEF` option suppresses the prompts):

```
OK, SORT -BRIEF
-INPUT A.ONE -INPUT B.TWO -OUTPUT C.OUT -KEYS 2
-START 1 -END 15 -DESCENDING
-START 16 -END 25 -TYPE LE
```

BEGINNING SORT

```
PASSES      2      ITEMS      80
```

[SORT-T3.0-23.0]

OK,

### **-MERGE Option**

A merge operation merges as many as 10 presorted files into one output file. To specify a merge operation, use the `-MERGE` option, as follows:

```
SORT -MERGE [options]
```

*options* can be one or more of the SORT options.

### Merge Prompts

The merge operation requires the number and names of the files to merge with those named at the first SORT prompt. Two additional prompts are displayed when the information is required.

After you have entered the sort keys, you see this prompt:

```
INPUT THE NUMBER OF ADDITIONAL FILES TO BE MERGED. (MAX= 10):
```

The maximum number of files SORT will merge is 11. The value of MAX in the prompt tells you how many files you may name in addition to those you specified at the first prompt. If you enter a zero, the merge operation begins. If you enter a number equal to or less than MAX you are prompted for their names as follows:

```
INPUT FILES TO BE MERGED, ONLY ONE PER LINE.
```

Enter the name of each file on a separate line.

After the merge is completed, SORT displays the number of passes needed for the merge and the number of items (lines) placed in the output file.

## SORT -MERGE Example

The following example illustrates a merge of three input files (ALPHA, BRAVO, and CHARLIE) into one sorted output file named DELTA:

```
OK, SORT -MERGE
SORT PROGRAM PARAMETERS ARE:
  INPUT TREE NAME - - OUTPUT TREE NAME FOLLOWED BY
  NUMBER OF PAIRS OF STARTING AND ENDING COLUMNS.
ALPHA DELTA 2
  INPUT PAIRS OF STARTING AND ENDING COLUMNS
  ONE PAIR PER LINE - - SEPARATED BY A SPACE.
  FOR REVERSE SORTING ENTER "R" AFTER DESIRED
  ENDING COLUMN - - SEPARATED BY A SPACE.
  FOR A SPECIFIC DATA TYPE ENTER THE PROPER CODE
  AT THE END OF THE LINE - - SEPARATED BY A SPACE.
  "A" - ASCII
  "I" - SINGLE PRECISION INTEGER
  "F" - SINGLE PRECISION REAL
  "D" - DOUBLE PRECISION REAL
  "J" - DOUBLE PRECISION INTEGER
  "U" - NUMERIC ASCII, UNSIGNED
  "LS" - NUMERIC ASCII, LEADING SEPARATE SIGN
  "TS" - NUMERIC ASCII, TRAILING SEPARATE SIGN
  "LE" - NUMERIC ASCII, LEADING EMBEDDED SIGN
  "TE" - NUMERIC ASCII, TRAILING EMBEDDED SIGN
  "PD" - PACKED DECIMAL
  "AU" - ASCII, UPPER & LOWER CASE SORT EQUAL
  "UI" - UNSIGNED INTEGER
  DEFAULT IS ASCII.
8 10 R
1 5
INPUT THE NUMBER OF ADDITIONAL FILES TO BE MERGED. (MAX= 10): 2
  INPUT FILES TO BE MERGED, ONLY ONE PER LINE.
BRAVO
CHARLIE

BEGINNING MERGE

  PASSES          2          ITEMS          950

[ SORT-T3.0-23.0 ]
OK,
```

Use the -BRIEF option to suppress all merge prompts.

## SPOOL

SPOOL prints files and manages files on the spool queue.

The SPOOL command allows you to perform the following functions:

- Print a file on a system printer or plotter. SPOOL print options allow you to specify how and where the file is to be printed.
- List all files or specific files on the spool queue.
- Modify the existing printing characteristics of a file on the spool queue.
- Cancel the printing of a file in the spool queue.

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

See also CONCAT, PROP.

### Format

```
SPOOL { pathname [printoptions]
        -CANCEL request [suboptions]
        -LIST request [suboptions]
        -MODIFY request [suboptions]
        -HELP }
```

### Options

- |                                                   |                                                                                                                                                                                                     |
|---------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>pathname</i> [ <i>printoptions</i> ]           | Specifies the name of the file you want printed or plotted. Use a simple filename if the file is in your current directory. <i>pathname</i> must be the first argument in the spool command syntax. |
|                                                   | <i>printoptions</i> , which are listed in the next section called Printing Options, specify other instructions about how the file is to be printed.                                                 |
| -CANCEL<br><i>request</i> [ <i>suboptions</i> ]   | Cancels all or specific files in a spool queue. See the section on the -CANCEL option below for the explanation of usage and suboptions.                                                            |
| -LIST [ <i>request</i> ]<br>[ <i>suboptions</i> ] | Lists all or specific files in a spool queue. See the section on the -LIST option below for the explanation of usage and suboptions.                                                                |
| -MODIFY <i>request</i><br>[ <i>suboptions</i> ]   | Modifies the printing characteristics of a file in the spool queue. See the section on the -MODIFY option below for the explanation of usage and suboptions.                                        |
| -HELP                                             | Displays the SPOOL syntax and options.                                                                                                                                                              |



## Printing Options

The following options follow the name of a file to be printed. Options may be specified in any order. You can use only one formatting option (`-COBOL`, `-FORTRAN`, `-PLOT`, `-NPH` or `-NO_FORMAT`) at a time.

- `-ALIAS alias_name` Replaces the user name in the file header and in SPOOL listings with *alias\_name*. *alias\_name* must be 16 characters or less.
- `-ALIAS` cannot be used on a pre-Rev. 21.0 system. Also, `-ALIAS` cannot be used with the `-DISK` option because `-DISK` places a spool request to a specified disk on a pre-Rev. 21 system.
- `-AS alias` Replaces *pathname* in the file header and in SPOOL listings with *alias*. *alias* must be 16 characters or less.
- `-ATTRIBUTE name1 [... name-n]` Specifies one or more device attributes to be used in spooling the file. Attributes are named properties of certain characteristics of the printer, such as paper type and destination. Your System Administrator supplies the valid attribute names for a printer in its printer environment file (use the `PROP` command to display environment files). The option can take one or more names and each can be specified more than once for multiple attributes. These options are not position-dependent. See the sections below called `Printing to a PostScript Printer` and `Printing to an HP Printer` for a description of PostScript and HP printer specific attributes.
- `-COBOL` Prints COBOL-format files. This option may be used when submitting a file to a pre-Rev. 21.0 system.
- `-COPIES n` Prints *n* copies of the file. (*n* cannot exceed 99.) When this option is used, the file's length is considered to be its actual length multiplied by the number of copies.
- `-DEFER [time]` Defers printing of the file. The file remains in the spool queue but is not printed until after the time you specify. Specify *time* in either 24-hour format (00:00 = midnight) or in 12-hour format with AM or PM (12:00 AM = midnight). The colon is optional. You must not leave a space between the digits and the AM or PM. If you issue the `-DEFER` option on the SPOOL command line without specifying a time, midnight (00:00) is assumed to be the print time. If the network crosses time zones, the deferred time used is that of the system holding the queue.

SPOOL

**-DISK *name***

Enters the file on the spool queue of the specified remote disk on a pre-Rev. 21.0 system. *name* is the name of the disk (partition). The remote disk must contain a SPOOLQ directory. (Use the STATUS DISKS command to list the disks connected to your system.) At Rev. 21.0, SPOOL ignores -DISK if it references a local disk. You cannot use -DISK with the options -ALIAS and -NO\_COPY.

**-FROM *m* -TO *n***

Specifies a range of pages to print. *m* and *n* are the page numbers of the first and last pages you want to print. *m* and *n* are printed inclusively; when used with FORTRAN or COBOL format they print logical pages. The -FROM value must not be greater than the -TO value. Both -FROM and -TO work with multiple copies of files using the -COPIES *n* option. Note that embedded specific escape sequences from pages prior to *m* may not take effect, because the sequences are never sent to the printer.

**-FTN**

Prints the file using the FORTRAN output conventions in the file. Cannot be used in some other formatting mode (-COBOL, -PLOT, -NO\_FORMAT, -NPH) is specified. The characters in column one of each line in the file have the following meanings:

| <i>Character</i> | <i>Meaning</i>                        |
|------------------|---------------------------------------|
| <b>1</b>         | Eject to top of page before printing. |
| <b>-</b>         | Skip 3 lines.                         |
| <b>0</b>         | Skip 2 lines.                         |
| <b>space</b>     | Skip 1 line.                          |
| <b>+</b>         | Overprint last line.                  |

**-HEADER [*text*]**

Replaces the default page header. *text* is the header you want to appear. If *text* is not supplied, the first line of the file is used. Any format commands embedded in the file that change the page header still function. If *text* contains spaces, you must enclose it in single quotation marks. Use the keyword FILE in place of *text* in order to use the name of the file as the page header.

**-LNUMBERS**

Each line of the printed file is prefixed with a number enclosed in parentheses and followed by a space. A line that uses more than one line of printed output is numbered only on the first printed line. Lines that overprint the previous line are not numbered.

- NO\_COPY** Prevents the SPOOL command from performing its normal operation of making a copy of the file to be printed. Instead, the contents of the file are printed from that file's present location in the file system. Therefore, any changes made to the file between the time it is spooled and the time it is printed appear in the printed file. If you specify **-NO\_COPY**, the file to be printed must be on the same node as the spool queue that is handling the spool request. You cannot use the **-NO\_COPY** option with password directories unless the nonowner password is null. You cannot use **-NO\_COPY** for requests submitted to a pre-Rev. 21.0 spooler. The despooler must have access rights to read the file if **-NO\_COPY** is used.
- NO\_EJECT** Disables the form feed, which normally occurs after the file has completed printing. Because a form feed is never performed before a banner page, use this option only when the banner page has been inhibited with the **-NOHEAD** option.
- NO\_FORMAT** Disables normal spooler format control (pagination and header generation). Used for files containing EVFU "skip-to-channel" commands. Cannot be used with other formatting options.
- NO\_HEADER** Does not print header pages.
- NO\_OVERSTRIKE** Inhibits overstriking by the printer. This option ensures that SPOOL properly handles the FORTRAN or COBOL format plus sign (+) control character, and trailing carriage return characters.
- NO\_NOTIFY** Notifies you when the file has completed printing.
- NO\_NPH** Enables file pagination without producing page headers and page numbers. **-NPH** is not allowed with other formatting modes.
- TO system** Adds a spool request to a Rev. 21.0 or later spool queue that resides on a different node. *system* is the name of any system that is networked to the current system. If you wish to add a spool request to a pre-Rev. 21.0 system on the network, use the **-DISK** option.

**-OPEN** [*pathname*]

Opens a data file in the spool queue directory. Data is generated by a program and directed to the data file. The file is opened on File Unit 2, unless you specify another file unit with a **-TUNIT** option. The file remains open for writing until you close it with the **CLOSE** command. After the file is closed, it is printed and then deleted from the spool queue. This option is useful for quickly printing listings from compilations.

**-PLOT** [*n*]

Denotes a plot file to be printed on the system plotter. *n* is the decimal number of words to be read and output per raster scan. (The default is 128 for a 200 raster/inch plotter.) A file spooled with this option cannot be printed on the line printer. **-PLOT** cannot be used with other formatting modes.

**-PROC** *name*

Specifies the PostScript procedure that allows you to print a file on a PostScript laser printer. You must download the PostScript procedure each time the printer is powered up. See the section below called Printing to a PostScript Printer for information about the **SPOOL** command's PostScript support.

**-SET\_FONT** *fontname*

Specifies a particular typeface for the file. *fontname* has a maximum length of 32 characters. If *fontname* is not supported, the file is printed using the default font name.

**-SET\_LANDSCAPE**

Specifies that the file is printed lengthwise along the paper. The options for *n*-up (thumbnail) printing on PostScript and HP printers are described in the sections below called Printing to a PostScript Printer and Printing to an HP Printer.

**-SET\_PAPER\_BIN**  
 $\left. \begin{array}{l} n \\ \text{MANUAL} \end{array} \right\}$

Specifies the paper bin to print the file. *n* is a number between 1 and 9, which selects the paper bin. **MANUAL** selects a manual paper feed.

**-SET\_PORTRAIT**

Specifies that the file is printed widthwise along the paper. The options for *n*-up (thumbnail) printing on PostScript and HP printers are described in the sections below called Printing to a PostScript Printer and Printing to an HP Printer.

**-SFI**

Suppresses the printing of both the *pathname* and the date/time modified notifications on the banner page and the trailer page of the printed file. (The *pathname* still appears in the **SPOOL -LIST -DETAIL** display, however, even if you use the **-SFI** option.)

- SPOOL\_WHILE\_OPEN** Allows a file to print while it is open for writing to disk. If used with **-OPEN**, then the file is held in the spool queue directory. If an existing file is spooled with **-SWO**, then **-NO\_COPY** is forced. **-SWO** does not guarantee immediate printing of a file. Printing starts when a printer with the correct environment is available.
- TRUNCATE** Truncates all lines longer than the printer's width as defined in the environment file. (Use the **PROP -DISPLAY** command to list the value of this parameter.) If this option is not specified, lines longer than the printer width wrap around two or more lines so that no printed text is lost.
- UNIT *n*** Specifies the file unit on which *pathname* is opened with the **-OPEN** option. *n* is a decimal number ranging from 1 through 125. (Default of *n* is 2.)
- XLATE *mapping*** Selects an alternate character set for the printer. *mapping* is a character string with a maximum of 32 characters that you use to select the mapping. *mapping* must begin with an alphabetic character; the remaining characters may be alphabetic, numeric, or any of the following three characters: . \$ \_

### Printing a File With SPOOL

To print a file, use the following command format:

**SPOOL *pathname [options]***

You select a suitable printer to print your file by specifying attributes that the printer must possess. Attributes are named properties of important features of the printer. For example, an attribute can denote the type of paper that is required or the location of the printer. You can specify all printer attributes simultaneously with one option, **-ATTRIBUTE**, described below. The names for these attributes are established by the System Administrator in special files called **printer environment files**. (Use the **PROP -DISPLAY** command to list the attributes of a particular printer.)

You should be aware of two requirements when spooling a file:

- You must specify the printer attributes. Only a printer with all the named attributes is used, even if that printer has additional attributes. SPOOL does not inform you if the attributes you specify are not valid.
- Some of the printer attributes may be specified as *mandatory* by the System Administrator. You must include all the mandatory attributes of the device so that the spool request is processed.

.....  
*SPOOL*

When you issue the SPOOL command, the file is copied onto the spool queue (unless the `-NO_COPY` option is used) and is printed according to its priority on the queue. Entering a file onto the spool queue is called spooling a file. There is only one queue for each system.

When a file is entered on the spool queue, it is identified by a request number that is unique across any cold start of the Spooler subsystem. The request number is reset to one (1) when the Spooler subsystem is cold started and the queue is empty. (This request number replaces the pre-Revision 21.0 `PRTnnn` identifier for the print request.) SPOOL then displays a message listing the spooled file's request number, number of records, and the pathname of the spool file, as in the following example:

```
OK, SPOOL BALANCE.SHEET  
[SPOOL Rev. 23.3.0 Copyright (c) 1992, Prime Computer, Inc.]  
Request 26 added to queue, 2 records :  
<ACCTNG>FORMS>BALANCE.SHEET  
OK,
```

The request number is necessary for canceling or modifying the spooled file. You can also use the SPOOL `-LIST` command to obtain the request number of a spooled file.

Although each system has only one queue, the spoolers copies of the files to be printed may be distributed across multiple partitions. You do not have to know the names of the partitions holding the data areas; the SPOOL command automatically distributes data files if necessary.

### **Example of Printing a File using SPOOL**

The following example of the SPOOL command prints a file on a printer controlled by your local system:

```
OK, SPOOL CMD.2 -ATT LINEP3 -ALIAS COLERIDGE -COPIES 3 -DEFER 1800  
[SPOOL Rev. 23.3.0 Copyright (c) 1992, Prime Computer, Inc.]  
Request 7 added to queue, 12 records : <SYSONE>BYRON>CMD.2
```

The command requests that three copies of the file `CMD.2` (in the current directory) be printed with the user name `COLERIDGE` at 6 p.m. (18:00) at the printer with the attribute named `LINEP3`.

---

**Note**

Your System Administrator may set the attributes file so that SPOOL issues an error message if you specify an invalid attribute. If so, spooling a file with the invalid attribute BAD\_ATTRIBUTE, for example, produces the following message:

```
OK, SPOOL LOGIN CPL ATT BAD_ATTRIBUTE
[SPOOL Rev. 23.3.0 Copyright (c) 1992, Prime Computer, Inc.]
Error from Spooler (Spool-107):
Unknown device attribute : BAD_ATTRIBUTE
```

Check with your System Administrator for further details.

---

**-CANCEL Option: Canceling a Spooled File**

The -CANCEL option cancels the specified print requests by deleting them from the spool queue.

If the file is in the spool queue of your local system, you can cancel the file while it is being printed. Files that are being printed on remote systems cannot be canceled after printing has begun.

To cancel the printing of files in a spool queue, use the following command format:

```
SPOOL -CANCEL {request [, request, ...]} [ [-ON system] ] [ [-DISK name] ] [-LIST]
```

**-CANCEL Suboptions**

- request* [, *request*, ... ]      Cancels request number(s) *request*. You can specify more than one spool request number. Request numbers must be separated by commas or blank spaces.
- ALL      Cancels all of your spool requests. Members of the .SPOOL\_ADMINISTRATOR\$ group and the user of the supervisor terminal can cancel any queued print request and any request on the local system that has started printing.
- DISK *name*      Specifies the pre-Rev. 21.0 disk with the spool queue that holds the print request you want canceled. *name* is the name of the disk or the logical device number of a disk on a pre-Rev. 21.0 system. Once a file spooled with the -DISK suboption has begun printing, it cannot be canceled.

SPOOL

**-ON *system*** Specifies the remote system with the spool queue that holds the print request you want canceled. *system* is the name of the remote system. Once a file spooled with the -ON suboption has begun printing, it cannot be canceled.

**Example of Cancelling a File**

The following example cancels request number 87.

```
OK, SPOOL -CANCEL 87
[SPOOL Rev. 23.3.0 Copyright (c) 1992, Prime Computer, Inc.]
Request 87 cancelled
OK, SPOOL -LIST
[SPOOL Rev. 23.3.0 Copyright (c) 1992, Prime Computer, Inc.]
System SYSONE
Request  Time      User           File           No Size  State
-----  -
26 August 91
86      20:34    FESTER         UNCLE.COMO     1  1
27 August 91
88      13:18    COLERIDGE     LOGIN.CPL      3  6    Defer
89      13:22    FESTER         LOGIN.CPL      1  2    Print
OK,
```

**-LIST Option: Listing Spooled Files**

To check the status of a spool queue, use the following command format:

SPOOL -LIST [*request*] [*suboptions*]

**-LIST Argument and Suboptions**

- request*** Lists only request number *request*.
- ALL** Lists all queues that the System Administrator has placed in the SPOOL\*>QUEUES file.
- ATTRIBUTE** Lists only entries with the specified attributes, and allows one or more than one attribute to be specified.
- BRIEF** Produces an abbreviated report with a single line per request. Use this option if you do not need a detailed listing. See section on -BRIEF below for further explanation.



- DETAIL**                    Produces a more detailed report, including the full pathname of the file being spooled, the printer attributes that are in effect, and the options being used (if any). See section on **-DETAIL** below for further explanation.
- DISK**                     Gives a queue report for a pre-Rev. 21.0 queue on a named partition.
- FULL**                    Adds a line to the **-DETAIL** report, which identifies the partition holding the spooler copy of the file.
- NO\_WAIT**                Suppresses the **—More—** prompt and does not pause after every 23 lines of output. Output scrolls continuously.
- ON**                      Gives a queue report for a named node.
- USER [name]**            Restricts the queue report to *name* and defaults to the current user if no name is given.

**-LIST Output**

If you specify only the **-LIST** keyword, the command lists only your files in your system's spool queue, as in the following example:

```
OK, SPOOL LIST
[SPOOL Rev. 23.3.0 Copyright (c) 1992, Prime Computer, Inc.]
System SYSONE
Request  Time      User           File           No Size  State
-----  -
26 August 91
86      20:34    FESTER        UNCLE.COMO     1  1
27 August 91
87      13:17    FESTER        LOGIN.CPL      1  2
88      13:18    COLERIDGE     LOGIN.CPL      3  6    Defer
89      13:22    FESTER        LOGIN.CPL      1  2    Defer
OK,
```

When used with no other options, **-LIST** displays the same information as **-LIST -BRIEF**.

The meanings of the header labels are as follows:

| <i>Label</i>   | <i>Meaning</i>                                                                                                                             |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------|
| <b>System</b>  | The name of the system on which the spool queue is located.                                                                                |
| <b>Request</b> | The request number of the spooled file and the date on which the file was spooled.                                                         |
| <b>Time</b>    | The time at which the file was spooled.                                                                                                    |
| <b>User</b>    | The name (user ID) of the user who spooled the file. If the <b>-ALIAS</b> option was used, lists the name specified by <i>alias_name</i> . |

.....  
SPOOL

**File** The filename of the spooled file. If the `--AS` option was used, lists the name specified by *alias*.

**No** The number of copies to be spooled.

**Size** The number of records in the spooled file. One record contains 1024 words and one word equals 2 bytes.

**State** The status of the spooled file. If this field is empty, the file is not printing. If the word `Print` appears, the file is printing. If the word `Defer` appears, the file's printing has been deferred.

### **--DETAIL Suboption**

The `--DETAIL` option displays detailed spool queue information. The `--DETAIL` report always shows which system the spool queue is on, the request number, the submittal date and time of the request, the user submitting the request, the number of copies requested, its size in records, and the complete pathname of the file. In addition, `--DETAIL` lists the following items, if specified, one per line:

| <i>Label</i>      | <i>Meaning</i>                                                                                                                        |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------|
| <b>Attributes</b> | The attributes given both explicitly by the user and supplied as defaults. If no attributes are specified, this line does not appear. |
| <b>Options</b>    | The options supplied by the user on the command line. If no options are specified, this line does not appear.                         |
| <b>Deferred</b>   | The earliest time at which the file will be printed.                                                                                  |

Here is an example of the `SPOOL --LIST --DETAIL` output:

```
OK, SPOOL --LIST --DETAIL
[SPOOL Rev. 23.3.0 Copyright (c) 1992, Prime Computer, Inc.]
System SYSONE
Request   Time           User                Copies  Size  State
-----
26 August 91
86        20:34:04  FESTER              1       1
   File name <USER1>FESTER>UNCLE.COMO
   Attributes DISK, PRINT_ROOM
27 August 91
87        13:17:20  FESTER              1       2
   File name <USER1>FESTER>LOGIN.CPL
   Attributes DISK, PRINT_ROOM
88        13:18:32  FESTER              3       6
   File name <USR1>FESTER>LOGIN.CPL
   Attributes DISK, PRINT_ROOM
   Deferred until 18:00:00 on 27 August 91
   Options   -ALIAS COLERIDGE
OK,
```

### Using -LIST When Spooling a File

You can use -LIST when spooling a file, as shown in the following example:

```
OK, SPOOL LOGIN.CPL -DEFER 1800 -LIST
[SPOOL Rev. 23.3.0 Copyright (c) 1992, Prime Computer, Inc.]
Request 89 added to queue, 1 records : <SYSONE>FESTER>LOGIN.CPL
System SYSONE
Request Time      User           File           No Size  State
-----
26 August 91
86      20:34    FESTER        UNCLE.COMO      1  1
27 August 91
87      13:17    FESTER        LOGIN.CPL        1  2
88      13:18    COLERIDGE     LOGIN.CPL        3  6    Defer
89      13:22    FESTER        LOGIN.CPL        1  2    Defer
OK,
```

### -MODIFY Option: Modifying a Spooled File

The -MODIFY option modifies the printing characteristics of a file in the spool queue.

If you change the attributes of a spool request, you must respecify all the attributes the changed request is to have.

Use the following command format to change the printing characteristics of a file in the spool queue:

```
SPOOL -MODIFY request [suboptions]
```

### -MODIFY Argument and Suboptions

*request* Request number of the spooled file.

In addition to the following, suboptions to -MODIFY may be any of the SPOOL printing options listed above under Printing Options, except -NO\_COPY, -OPEN, -SPOOL\_WHILE\_OPEN and -TUNIT.

- NO\_SWO Cancels the -SPOOL\_WHILE\_OPEN process.
- NO\_XLATE Removes the -XLATE option from a spooler request. The default mapping is used.
- NO\_DEFER Removes the -DEFER attribute.
- NO\_RUSH Removes the -RUSH attribute. (Default)

**-RUSH**

Gives priority to a file so that it is printed before other spooled files. If more than one file in the spool queue has this **-RUSH** attribute, the files are printed in the order they were spooled. Only users in the **.SPOOL\_ADMINISTRATOR\$** group or a user at the supervisor terminal can specify this option.

**Results of Using -MODIFY**

The new attributes specified with **-MODIFY** are compared to the existing attributes of the spooled file, with the following results:

- If a new attribute conflicts with a previous attribute, the file is given the new attribute and the old one is removed.
- If a new attribute does not conflict with previously specified attributes, it is added to the file.
- If an existing attribute is not modified by a new attribute, the existing attribute is retained.
- If an attribute is modified with the **-ATT** option, all attributes that were previously specified with this option are overridden. For example, if you specified **-ATT CARBON FLOOR3** when you spooled the file and subsequently use **-MODIFY -ATT WHITE** in an attempt to change **CARBON** to **WHITE**, the result is that both **CARBON** and **FLOOR3** attributes are overridden by the new **WHITE** attributes.

**Example of Modifying a Spooled File**

The following example illustrates the use of the **-MODIFY** option:

```
OK, SPOOL -MODIFY 89 -NO_DEFER -LIST
[SPOOL Rev. 23.3.0 Copyright (c) 1992, Prime Computer, Inc.]
System SYSONE
Request  Time  User          File          No Size  State
-----  -
26 August 91
86      20:34 FESTER        UNCLE.COMO    1  1
27 August 91
87      13:17 FESTER        LOGIN.CPL     1  2
88      13:18 COLERIDGE  LOGIN.CPL     3  6      Defer
89      13:22 FESTER        LOGIN.CPL     1  2      Print
OK,
```

In this example, the spooled file **LOGIN.CPL** has the **-DEFER** option removed from the file after the file has been submitted on the spool queue.

## Printing to a PostScript Printer

This section describes formatting options, fonts, and attributes that are available for PostScript printer environments only.

### PostScript Printer Font Options

You can use the `-SET_FONT` option to select a specific font and its point size, overriding any fonts selected by environment directives, or the PostScript printer default of Courier 11.5 pt. The following fonts are available for most PostScript printers (font names without point sizes are 11.5 pt). Refer to your printer documentation for more information on implementing fonts for your particular printer.

|                     |                                          |
|---------------------|------------------------------------------|
| AvantGarde          | Sans serif, proportional spaced, 11.5 pt |
| Bookman             | Serif, proportional spaced, 11.5 pt      |
| Courier             | Serif, monospaced, 11.5 pt               |
| Courier-10          | Serif, monospaced, 10 pt                 |
| Courier-12          | Serif, monospaced, 12 pt                 |
| Courier-Bold        | Serif, monospaced, 11.5 pt               |
| Courier-Bold-10     | Serif, monospaced, 10 pt                 |
| Courier-Bold-12     | Serif, monospaced, 12 pt                 |
| Courier-BoldOblique | Serif, monospaced, 11.5 pt               |
| Courier-Oblique     | Serif, monospaced, 11.5 pt               |
| Helvetica           | Sans serif, proportional spaced, 11.5 pt |
| Helvetica-Narrow    | Sans serif, proportional spaced, 11.5 pt |
| NewCentury          | Serif, proportional spaced, 11.5 pt      |
| Palatino            | Serif, proportional spaced, 11.5 pt      |
| Symbol              | Symbols                                  |
| Times               | Serif, proportional spaced, 11.5 pt      |
| Times-Bold          | Serif, proportional spaced, 11.5 pt      |
| ZapfChancery        | Serif, proportional spaced, 11.5 pt      |
| ZapfDingbats        | Symbols                                  |

### `-SET_PORTRAIT` Printing Option

`-SET_PORTRAIT` prints text across the shortest dimension of the paper. The default portrait format is 66 lines of 80 characters.

Use the following command format to print in portrait mode on a PostScript printer:

```
SPOOL -SET_PORTRAIT [-ATTRIBUTE n-UP]
```

The following is the PostScript attribute for the `-SET_PORTRAIT` option:

`-ATTRIBUTE n-UP` Enables printing of multiple formatted pages on a single sheet of paper, or *n*-up printing. When using portrait formats, *n* can be 2, 2X, 4, 4X, 8, 8X, 16, or 16X. Figure 2-2 shows the page layouts for each value of *n*. This method of printing is sometimes referred to as galley-proof printing or thumbnail printing. See the section below called Thumbnail Printing on a PostScript Printer.

### `-SET_LANDSCAPE` Printing Option

`-SET_LANDSCAPE` prints text across the longest dimension of the paper. The default landscape format is 66 lines of 138 characters.

Use the following command format to print in landscape mode on a PostScript printer:

SPOOL `-SET_LANDSCAPE` [*PostScript-attributes*]

The following are the PostScript attributes for the `-SET_LANDSCAPE` option:

`-ATTRIBUTE n-UP` Enables printing of multiple formatted pages on a single sheet of paper, or *n*-up printing. When using landscape formats, *n* can be 2, 2X, 4, 4X, 8, 8X, 16, or 16X. Figure 2-2 shows the page layouts for each value of *n*. This method of printing is sometimes referred to as galley-proof printing or thumbnail printing. See the section below called Thumbnail Printing on a PostScript Printer.

You can use most of these options with the SPOOL `-PROC` command to specify PostScript laser printer procedures. Refer to the `-PROC` command for more information. (You cannot use the `-PROC` command to specify 8-up, 16-up, and NOSCALE pages.) Point sizes and other size characteristics may differ slightly from the corresponding `-SET_PORTRAIT` or `-SET_LANDSCAPE` command.

`-ATTRIBUTE DRAFT/` Superimposes the word DRAFT in large outline letters diagonally across the entire page. The slash character (/) in the option name indicates that the word *DRAFT* is printed diagonally across the page. You can use this option for both PostScript files and text files. You cannot use this option when spooling files with the `-PROC` option.

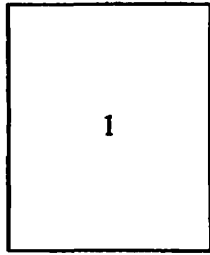
**-ATTRIBUTE NOSCALE** Prints landscape format, 50 lines of 105 characters per line, across the longest width of the paper. This is approximately the same number of characters per page as the standard portrait-format page.

### ***Thumbnail Page Printing on a PostScript Printer***

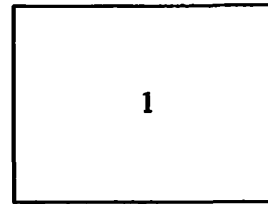
Thumbnail page printing, or *n*-up printing allows you to print multiple document pages on a single sheet of paper. Printing labels is one example of using *n*-up printing. Each sheet of paper is divided into 2 to 16 equally sized sections, each of which contains the contents of one formatted page. You can use thumbnail page printing with either portrait or landscape page formats.

Thumbnail printing options enable you to control the sequence in which the pages are printed on a sheet of paper. *n*-UP, which is the default, orders the pages in a top-down sequence, beginning at the top left-side of the paper, and down, in columns. *nX*-UP prints the pages across the paper from right to left in rows, starting at the top left corner.

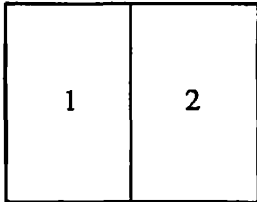
Figure 2-2 illustrates thumbnail page printing sequences.



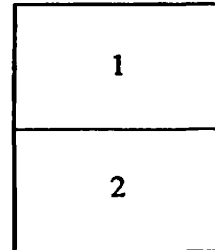
*Portrait*



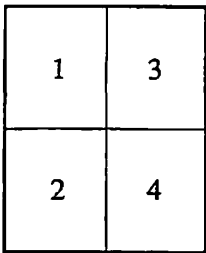
*Landscape*



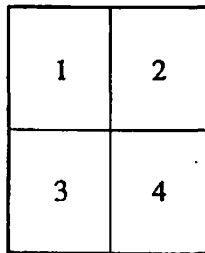
*2-UP Portrait*



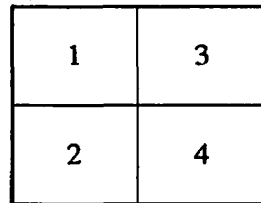
*2-UP Landscape*



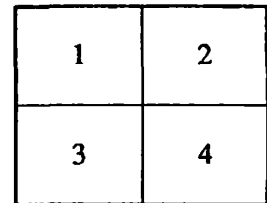
*4-UP Portrait*



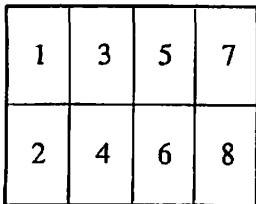
*4X-UP Portrait*



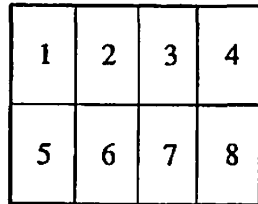
*4-UP Landscape*



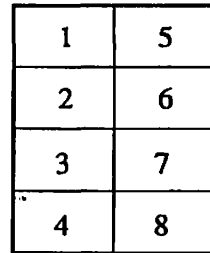
*4X-UP Landscape*



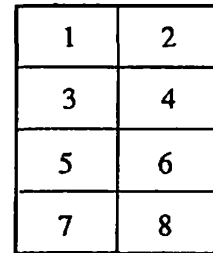
*8-UP Portrait*



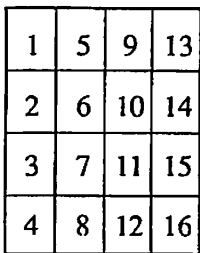
*8X-UP Portrait*



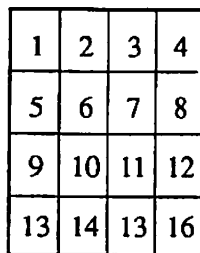
*8-UP Landscape*



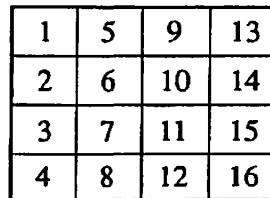
*8X-UP Landscape*



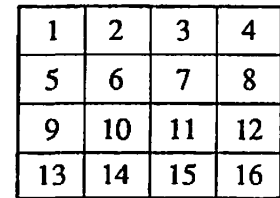
*16-UP Portrait*



*16X-UP Portrait*



*16-UP Landscape*



*16X-UP Landscape*

*Figure 2-2. PostScript n-UP Page Layouts for 8½ x 11 Inch Sheets*



The following tables describe the output from the thumbnail page printing options for both portrait and landscape formats:

| <i>Portrait Thumbnail Page Printing Options (PostScript)</i> |                    |                               |                  |                            |
|--------------------------------------------------------------|--------------------|-------------------------------|------------------|----------------------------|
| <i>-SET_PORTRAIT</i>                                         | <i>Sheet Size</i>  |                               | <i>Page Size</i> |                            |
|                                                              | <i>Orientation</i> | <i>Lines x chars per line</i> | <i>Font size</i> | <i>Page width x height</i> |
| Default (1 page per sheet)                                   | Portrait           | 66 x 80                       | 11.5 pts         | 8.0" x 10.5"               |
| -AT 2-UP                                                     | Portrait           | 66 x 80                       | 6.7 pts          | 4.7" x 6.1"                |
| -AT 4-UP or 4X-UP                                            | Portrait           | 66 x 80                       | 5.1 pts          | 3.6" x 4.7"                |
| -AT 8-UP or 8X-UP                                            | Portrait           | 66 x 80                       | 3.4 pts          | 2.3" x 3.1"                |
| -AT 16-UP or 16X-UP                                          | Portrait           | 66 x 80                       | 2.6 pts          | 1.8" x 2.3"                |

| <i>Landscape Thumbnail Page Printing Options (PostScript)</i> |                    |                               |                  |                            |
|---------------------------------------------------------------|--------------------|-------------------------------|------------------|----------------------------|
| <i>-SET_LANDSCAPE</i>                                         | <i>Sheet Size</i>  |                               | <i>Page Size</i> |                            |
|                                                               | <i>Orientation</i> | <i>Lines x chars per line</i> | <i>Font size</i> | <i>Page width x height</i> |
| Default (1 page per sheet)                                    | Landscape          | 66 x 138                      | 8.8 pts          | 10.5" x 8.0"               |
| -AT 2-UP                                                      | Landscape          | 66 x 138                      | 5.1 pts          | 6.1" x 4.7"                |
| -AT 4-UP or 4X-UP                                             | Landscape          | 66 x 138                      | 3.9 pts          | 4.7" x 3.6"                |
| -AT 8-UP or 8X-UP                                             | Landscape          | 66 x 138                      | 2.6 pts          | 3.1" x 2.3"                |
| -AT 16-UP or 16X-UP                                           | Landscape          | 66 x 138                      | 1.9 pts          | 2.3" x 1.8"                |

| <i>Landscape Thumbnail Page Printing Options (PostScript)</i> |                    |                               |                  |                            |
|---------------------------------------------------------------|--------------------|-------------------------------|------------------|----------------------------|
| <i>-SET_LANDSCAPE</i>                                         | <i>Sheet Size</i>  |                               | <i>Page Size</i> |                            |
|                                                               | <i>Orientation</i> | <i>Lines x chars per line</i> | <i>Font size</i> | <i>Page width x height</i> |
| <i>-AT NOSCALE</i>                                            |                    |                               |                  |                            |
| Default (1 page per sheet)                                    | Landscape          | 50 x 105                      | 11.5 pts         | 10.5" x 8.0"               |
| -AT 2-UP                                                      | Landscape          | 50 x 105                      | 6.7 pts          | 6.1" x 4.7"                |
| -AT 4-UP or 4X-UP                                             | Landscape          | 50 x 105                      | 5.1 pts          | 4.7" x 3.6"                |
| -AT 8-UP or 8X-UP                                             | Landscape          | 50 x 105                      | 3.4 pts          | 3.1" x 2.3"                |
| -AT 16-UP or 16X-UP                                           | Landscape          | 50 x 105                      | 2.6 pts          | 2.3" x 1.8"                |

### Special Options for HP Laser Jet Printers

The following Spooler features are available for use with the HP Laser Jet family of printers:

- Landscape page formatting
- Portrait page formatting
- Font size of 10, 12, and 16.66 (although not all printers support all three font sizes)

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*].
- -DELAY is optional, and is useful to prevent buffer overflows at the printer. The default setting for -DELAY is 4 seconds, which can be set to a lesser or greater number of seconds, depending on the particular printer.
- SET\_LANDSCAPE.
- SET\_PORTRAIT.

You must include both of these entries if you wish to use landscape and portrait page formatting.

Attributes for defining the paper size and number of characters per inch (cpi) or centimeter (cpcm) are shown in the following table:

| <i>Attributes for Defining HP Laser Jet Printer Environments</i> |                   |            |                                            |                   |             |
|------------------------------------------------------------------|-------------------|------------|--------------------------------------------|-------------------|-------------|
| <i>Attributes for U.S. Paper Sizes</i>                           |                   |            | <i>Attributes for European Paper Sizes</i> |                   |             |
| <i>Attribute</i>                                                 | <i>Paper Size</i> | <i>cpi</i> | <i>Attribute</i>                           | <i>Paper Size</i> | <i>cpcm</i> |
| LETTER                                                           | 8.5" x 11"        | 10         | A4                                         | 210x297mm         | 3.9         |
| LETTER_MED                                                       | 8.5" x 11"        | 12         | A4_MED                                     | 210x297mm         | 4.68        |
| LETTER_SMALL                                                     | 8.5" x 11"        | 16.66      | A4_SMALL                                   | 210x297mm         | 6.5         |
| LEGAL                                                            | 11" x 14"         | 10         | B5                                         | 182x257mm         | 3.9         |
| LEGAL_MED                                                        | 11" x 14"         | 12         | B5_MED                                     | 182x257mm         | 4.68        |
| LEGAL_SMALL                                                      | 11" x 14"         | 16.66      | B5_SMALL                                   | 182x257mm         | 6.5         |

**SPY**

SPY displays the information that MIDASPLUS uses.

***Format***

SPY

***Usage***

SPY is an offline menu-driven utility for use with MIDASPLUS files. You can use SPY whether or not you are using MIDASPLUS. The SPY output displays all or part of the information that MIDASPLUS uses and updates during runtime. The information includes the following:

- A table of data record locks taken
- A display of internal values at user-specified time intervals during runtime
- Systemwide and user-configurable parameters

When you invoke the SPY program, it displays a Main menu of options and asks you what type of information you want to see. Enter the appropriate response to continue.

For more information, see the *MIDASPLUS User's Guide*.

**SQ**

See SET\_QUOTA.

**SSR**

See SET\_SEARCH\_RULES.

START

## START

START restarts a program in memory.

Use START to restart a program that was interrupted before it finished execution. The program was previously loaded into memory with a RESTOR or RESUME command. The program can be an EPF, a static-mode program, a CPL program, or one of certain internal PRIMOS commands.

The interruption to the program was caused by a Ctrl-P, a BREAK, a FORTRAN PAUSE statement, a CALL EXIT statement (static-mode programs only), or a runtime error.

### Format

START [*pc*] [*a*] [*b*] [*x*] [*keys*]

### Arguments

[*pc*] [*a*] [*b*] [*x*] [*keys*]      Specify values for the RVEC vector. See the section called Static-mode Programs below.

### CPL Programs, EPF Runfiles, and Internal Commands

You can have more than one suspended CPL program, EPF program, and/or internal PRIMOS command in memory. The START command restarts the last program or internal command that was suspended. The System Administrator sets the number of EPFs that you can interrupt and restart.

Use START without arguments to restart a suspended CPL program, EPF program, or internal PRIMOS command, as follows:

OK,      .

The program restarts from the point of interruption.

The following example illustrates the START command with an EPF program that was interrupted with a Ctrl-P:

```
OK,      .      /* begin execution of program CIRCLE
Enter the radius of the circle:      /* user presses Ctrl-P

QUIT.
OK,      /* user issues START to restart CIRCLE
         /* program was waiting for input; user enters 8
The area of the circle is 50.3349.
OK,      /* program ends and returns control to PRIMOS
```

To restart a suspended EPF from its beginning, use the RESUME command.

## Static-mode Programs

Unlike CPL and EPF programs, you can have only one suspended static-mode program in memory at a time. If you interrupt a static-mode program and then execute another static-mode program, the second program overwrites the first because they execute in the same place in memory.

For static-mode programs, START initializes the processor's registers and keys from the command line (or from RVEC, for any values not specified in the command line).

To restart a suspended static-mode program, use the following command format:

```
START [pc] [a] [b] [x] [keys]
```

The arguments for START specify new values for the RVEC vector. You must specify the new values in octal. *pc* (program counter) is the starting address of the program. *a* is the A register, *b* is the B register, and *x* is the X register. *keys* stands for the status keys associated with INK and OTK instructions.

If you issue START without arguments, one of the following two actions results:

- If no static-mode program has been invoked or restored at this command level (as is the case after a Ctrl-P, for example), the interrupted program (whether it is static mode or dynamic mode) continues from the point of interruption.
- Otherwise, the static-mode program defined by the current RVEC is invoked.

If you issue START with arguments, the register settings specified by the arguments are applied to the RVEC contents (thus defining a new static-mode machine state), and that static-mode program is invoked.

Supplying new RVEC values is generally not recommended. The exception is specifying 1000 (octal) as the value for *pc*. START 1000 restarts the most recently invoked static-mode program. (This command does not, however, reinitialize variables in the program.) For further information on specifying RVEC values, see Appendix A of this guide.

If a static-mode program is interrupted and another static-mode program is subsequently invoked, an attempt to start the first program is refused with the following error message:

```
Attempt to proceed to overwritten program image. (listen_)
ER!
```

This error occurs because the machine state for the first static-mode program has been overwritten in the RVEC, and hence the system does not have the information needed to restart the program correctly. In addition, the second program may have overwritten the first program.

For further information, see the *Advanced Programmer's Guide III: Command Environment*.

See also REENTER, RESUME, and RELEASE\_LEVEL.

.....

## STATUS

## STATUS

STATUS displays the current status of various aspects of the system where the command is invoked.

As an operator command, STATUS monitors system usage. For more information on STATUS as an operator command, see the *Operator's Guide to System Commands*.

### **Format**

STATUS [*argument*]

### **Arguments**

Specify only one argument at a time.

|        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|--------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ALL    | Displays information accessible through all other STATUS arguments except for PROJECTS.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| COMM   | Displays information about the system's communication controllers (except for the Prime Node Controller). The information for each controller includes the name (ICS1, ICS2, ICS3, AMLC, MDLC, HSSMLC, SMLC), type, device address (in octal), the number of operable asynchronous lines, and the number of operable synchronous lines. The controller types are as follows: DMT (50xx old style) or DMQ (51xx new style) for AMLC controllers, a 4-digit PROM set ID number for MDLC controllers, and no information for other controllers. If the system is part of the Local Area Network (LAN), STATUS COMM displays LAN300 information as well. |
| DEVICE | Displays the physical and logical device numbers of any assigned magnetic tape drives, as well as the user ID and user number of the user to whom the device is assigned.                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |

**DISKS**

Displays information about disks running on your system, including remote disks if your system is on a network and is not running the Name Server. (When the Name Server is running, use LIST\_MOUNTS to get information about all available disks.) The information includes the diskname, octal logical device number (ldev), physical device number (pdev) for local partitions, and system node name. At PRIMOS Revision 22.0, DISKS gives information about local robust partitions and local or remote mirrored partitions.

Some commands use a decimal logical device number. Use STATUS DISKS to get the octal logical device number. To convert the octal logical device number to decimal, use the command function [OCTAL ldev].

For example, to get the decimal equivalent of octal 70, enter

```
type [OCTAL 70]
```

PRIMOS returns

56

56 is the decimal equivalent of 70<sub>8</sub>.

**ME**

Displays your user ID, your user number, your terminal's line number (in octal and decimal), and the devices you are using. If any other users (including phantoms) are using your user ID, the same information is also displayed about those users.

**NETWORK**

Displays the status of the full-duplex, ring, and public data networks. The information includes each system's node name and whether the system is connected ("Up" if the system is available, "Down" if not available). STATUS NETWORK also displays the nodes on the Local Area Network (LAN).

**NTS**

Displays information about the Network Terminal Service (NTS), including its status (started, not started) and, if it is started, the pathname of the NTS configuration file.

**PROJECTS**

Displays the user ID, project ID, and user number of each user on your system.





## Usage

The following example illustrates the use of STATUS without an argument:

OK, STATUS

System B52 is currently running PRIMOS rev. 23.3.0  
 Copyright (c) Prime Computer, Inc. 1992

User MARTHA

B52

| File Unit | File Position | Open Mode | File Type | RWlock | Treename                               |
|-----------|---------------|-----------|-----------|--------|----------------------------------------|
| 31        | 000000000     | VMr       | DAM       | NR-1W  | <USLAB>DSM*>SIT_TEXT_DBS>DSM_USA.TODAY |

| Disk   | Ldev | Pdev   | System | Robust | Mirror  |           |       |
|--------|------|--------|--------|--------|---------|-----------|-------|
|        |      |        |        |        | Primary | Secondary | State |
| USLAB  | 0    | 3460   |        |        |         |           |       |
| USUSR3 | 1    | 70460  |        |        |         |           |       |
| PAGER  | 2    | 100461 |        |        |         |           |       |
| SYSUSA | 3    | B29    |        |        |         |           |       |
| USA1   | 4    | B29    |        |        |         |           |       |
| USA2   | 5    | B29    |        |        |         |           |       |
| USA3   | 6    | B29    |        |        |         |           |       |
| USA4   | 7    | B29    |        |        |         |           |       |
| USA5   | 10   | B29    |        |        |         |           |       |

Note: The Name Server is started on your system. Therefore, the STATUS DISKS command may not display all the disks to which you have access. To see the complete list, use the LIST\_MOUNTS command.

| Sem. | Value  | Users |
|------|--------|-------|
| 65   | 177777 | 1     |

Ring network

| Node | State |
|------|-------|
| B52  | ****  |
| B29  | Up    |
| B17  | Down  |

NTS is not currently started

| User   | User No (In Decimal) | Line No | Devices (AL in Decimal)     |
|--------|----------------------|---------|-----------------------------|
| MARTHA | 6                    | rem     | <USUSR3> <USLAB> (from B52) |

OK,

.....  
SVCSW

## SVCSW

SVCSW sets the SVC switch.

### **Format**

SVCSW  $\left\{ \begin{array}{l} 0 \\ 1 \end{array} \right\}$

### **Options**

The SVC switch controls the handling of SVC instructions in a virtual memory environment. Setting the SVC switch to 0 (the default setting) causes all SVC instructions to be trapped and processed by the system supervisor. The SVC switch is initialized to 0 by the LOGIN command.

If the SVC SWITCH is on (SVC 1), almost all SVC instructions cause a virtual trap, and SVC instructions are handled through the user's location 65g. The class of SVC instructions always processed by the PRIMOS operating system, regardless of the SVCSW command, are those determined by FUNCTION code 5XX. Currently, the SVC instructions are RREC (for reading from disk), WREC (for writing to disk), TIMDAT (for obtaining the time and date from PRIMOS), and RECYCL (for passing control to the next user).

### **Usage**

For detailed information on SVCSW and on SVC instructions, see the *Assembly Language Programmer's Guide*.

## SYNCSORT

SYNCSORT invokes the SyncSort/PRIME utility to sort, merge, or copy one or more files. This is a separately priced product that offers functionality not available with the PRIMOS Sort utility.

### **Format**

SYNCSORT

### **Usage**

For more information, see the *SyncSort/PRIME Reference Manual*.

T            See TIME.

# TALK

TALK invokes the TALK facility, an interactive message system that allows two users to conduct a real-time conversation at their terminals.

The TALK command performs the following functions:

- Begins a TALK session with another user.
- Responds to a request by another user to participate in a TALK session.

## Format

$$\text{TALK} \left\{ \begin{array}{l} \text{-TO } \left\{ \begin{array}{l} \text{username} \\ \text{usernumber} \end{array} \right\} \text{ [-ON system]} \\ \text{-RESPOND} \\ \text{-HELP} \end{array} \right\} \left[ \begin{array}{l} \text{-TERMINAL\_TYPE } \textit{terminal-type} \\ \text{-STATISTICS} \end{array} \right]$$

## Options

- |                                                                                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -ON <i>system</i>                                                                        | Specifies the remote system of the person you are calling. <i>system</i> is the name of the remote system as displayed by the STATUS NET command.                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| -RESPOND                                                                                 | Invokes TALK in response to a request from another user.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| -TERMINAL_TYPE<br><i>terminal-type</i>                                                   | Specifies your terminal type (PST 100 or PT200, for example). The <i>terminal-type</i> argument specifies a valid terminal type. Use this option if you do not have the .TERMINAL_TYPE\$ global variable set to the proper terminal type. If you do not have .TERMINAL_TYPE\$ defined and you do not use -TTP, TALK prompts you for your terminal type.                                                                                                                                                                                                                                       |
| -TO $\left\{ \begin{array}{l} \text{username} \\ \text{usernumber} \end{array} \right\}$ | <p>Transmits a request to open a TALK session with the specified user.</p> <p><i>username</i> specifies the user ID of the person you wish to call. Since more than one person may be logged in as <i>username</i>, the first to respond to your TALK request becomes the other participant in the dialogue. Others logged in as <i>username</i> cannot participate.</p> <p><i>usernumber</i> specifies the user number (as defined by the STATUS USERS command) of the person you wish to call. If you send the TALK request to <i>usernumber</i>, only that user is allowed to respond.</p> |

**-STATISTICS**

Displays certain statistics at the end of the TALK session. The TALK server keeps these statistics in a logfile in the TALK\* directory; the most recent logfile and next most recent logfile are preserved in TALK\*. These statistics include

- The number of characters sent and received
- The number of InterServer Communication (ISC) messages sent and received
- The number of ISC delivery failures and retries

These statistics are useful in debugging problems caused by ISC and X.25 message delivery failures. For more information on ISC messages, see *Subroutines Reference V: Event Synchronization*.

**-HELP**

Displays a list of the command line options and their abbreviations. You cannot use this option with any other options on the command line.

***Initiating a TALK Session***

To initiate a TALK session, use the following format:

```
TALK -TO { username
           username } [-ON system] [ -TERMINAL_TYPE terminal-type
                                     -STAT ]
```

If the user you are calling can receive calls, a notification message is displayed on that user's terminal, and the TALK screen appears on your terminal. If the person does not respond, or if you wish to terminate the session before the person responds, type Ctrl-D. This sends control of your terminal back to PRIMOS and sends a message to the user you are calling that you have terminated the session.

Your TALK request will fail if any of the following three conditions are true:

- The user you are calling is not logged in.
- The user you are calling is not accepting messages (has used either the MESSAGE -DEFER command or the MESSAGE -REJECT command to put the receiving terminal in a DEFER or REJECT state).
- The user you are calling is already participating in another TALK session.

## Responding to a TALK Request

When another user calls you to participate in a TALK session, you will see the following message on your terminal screen:

```
***TALK_SERVER (user nnn on sys_name) at hh:mm
user_name on sys_name is calling you. Use TALK -RESPOND to answer.
```

If you do not wish to participate in the TALK session, ignore the request. If you choose to participate in the TALK session, issue the TALK command in the following format:

```
TALK -RESPOND [-TERMINAL_TYPE term_type] [-STATISTICS]
```

After you have issued the TALK -RESPOND command, your terminal screen also divides into a top and bottom window. Your user ID appears in the upper left portion of the top window, and the user ID of the person to whom you are responding appears in the upper left portion of the bottom window. A reminder that the Ctrl-D key sequence terminates the session appears on the lower right portion of the bottom window.

---

### Note

You cannot initiate a TALK session at the supervisor terminal (User 1), nor can you respond to a TALK request at the supervisor terminal.

---

## Using the TALK screen

If the user you call is logged in and receives the notification that you wish to open a TALK session, your terminal screen divides into a top and bottom window. Your user ID appears in the upper left portion of the top window, and the user ID of the person you are calling (when that person answers) appears in the upper left portion of the bottom window. A reminder that the Ctrl-D key sequence terminates the session appears on the lower right portion of the bottom window. At this point, you cannot type any text until the called user responds to your request.

When the user you call types TALK -RESPOND, the session starts. Either person may enter text first. In the example shown below, HOLMES initiates the session. When WATSON accepts the connection, HOLMES begins typing. The characters appear in the top window of his screen (as shown in the example below). WATSON watches the characters appear as they are typed, including backspaces and corrections, if any, in the bottom window of his screen. WATSON then types his response as HOLMES watches the characters WATSON types appear in his bottom window.



## Terminating the TALK Session

When you wish to end the TALK session, type Ctrl-D. This refreshes your terminal screen, and also refreshes the screen of the other participant. Both of you will then see the following message:

```
Session terminated.
```

If you used the `-STAT` option when you issued the TALK command, you will also see the statistics from the TALK session. For example:

```
Session terminated.
```

```
Characters sent =          176.
Characters received =       238.
Messages sent =             10.
Messages received =        103.
Delivery failures =          0.
Total retries =             0.
```

---

### Note

Do not use Ctrl-P to terminate a TALK session, as this may suspend your terminal process; use Ctrl-D instead.

---

## TCF

TCF accesses a remote IBM host from a DPTX-configured 3277 terminal.

### Format

```
TCF { -AUTOPORT n } -TERMINAL terminal-name [quit-signal]
```

### Arguments and Options

The TCF command invokes DPTX/TCF (Transparent Connect Facility) on a system where DPTX/TSF and DPTX/DSC are running. `-HOST` specifies a valid Virtual Buffer Emulator (VBE) station name (*hostname*) for the remote IBM host computer to which you want to connect. `-AUTOPORT` specifies the logical line number (*n*) of the synchronous line which is searched for an available terminal address (station address). *n* is a number ranging from 0 through 7, inclusive. `-TERMINAL` specifies the PRIMOS name (*terminalname*) of your 3277-type-terminal. Use an asterisk (\*) for *terminalname* to specify your current terminal.

*quit-signal* specifies the signal that you enter at your terminal when you want to break the connection with the host computer and return to PRIMOS command level. The quit signal, which can be a character string or a key, is one of the following:

| <i>Quit Signal</i>         | <i>Function</i>                                                                                                                                                                                                              |
|----------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-QUIT</b> <i>string</i> | Specifies a string of eight or fewer characters as the quit signal. When you enter the quit string at your terminal, you must enter it exactly as you specified it on the TCF command line.                                  |
| <b>-PA</b> <i>n</i>        | Specifies that a Program Attention (PA) key is the quit signal. <i>n</i> is 1 or 3.                                                                                                                                          |
| <b>-PF</b> <i>n</i>        | Specifies that a Program Function (PF) key is the quit signal. <i>n</i> is a number ranging from 1 through 12, inclusive. The PF key specified with this option cannot perform any other function in an application program. |
| <b>-TR</b>                 | Specifies that TEST REQUEST key is the quit signal.                                                                                                                                                                          |

When you enter the quit string at the terminal, the program breaks the connection with the host, prints out a **TCF HALT** message, and returns to PRIMOS. To the host, it seems that the terminal was powered off. Some applications programs may halt abnormally. In this case consult your System Administrator.

### **Usage**

For a detailed description of TCF, see the *Distributed Processing Terminal Executive Guide*.



## TERM

TERM sets the characteristics of a terminal line.

**Format**

TERM *options*

**Options**

You can specify more than one option if they do not conflict.

- |                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|--------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -BREAK { OFF<br>ON }     | Enables or disables the break character. The break character (which is Ctrl-P or the BREAK key) can interrupt a running program or command. The default is BREAK ON.                                                                                                                                                                                                                                                                                                                                                                                                                       |
| -DISPLAY                 | Lists the current settings of the erase and kill characters, the duplex mode, the XON/XOFF feature, and break.                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| -ERASE <i>char</i>       | Sets <i>char</i> as the erase character. <i>char</i> can be any character except certain PRIMOS reserved characters, such as the percent symbol (%), the single quotation mark ('), parentheses (), the semicolon (;), and the left bracket ([). In general, use of reserved characters as erase characters is not recommended, but if you do use a reserved character, enclose that character (except delete/rubout) in single-quotation marks ('). The default erase character is the double-quotation mark ("), although your System Administrator may have selected another character. |
| -FULL                    | Sets the terminal line to full-duplex mode, which is the normal PRIMOS duplex mode. With full-duplex operation, all characters are echoed except a LINEFEED input character, which is ignored. A carriage-return input character is passed into the system as a LINEFEED and echoed to the user as carriage return followed by LINEFEED.                                                                                                                                                                                                                                                   |
| -HALF [ { LF<br>NOLF } ] | Sets the terminal line to half-duplex. With LF, no characters are echoed except a carriage return, which echoes a NEWLINE; input is sent to the LINEFEED system in the same manner as with TERM -FULL. NOLF functions identically except that the carriage return does not echo as LINEFEED. LF is the default value for -HALF.                                                                                                                                                                                                                                                            |

■ ■ ■ ■ ■ ■ ■ ■ ■ ■  
**TERM**

- KILL *char*** Sets *char* as the kill character. *char* can be any character except certain PRIMOS reserved characters, such as the percent symbol (%), the single quotation mark ('), parentheses (), the semicolon (;), and the left bracket ([). In general, use of reserved characters as kill characters is not recommended, but if you do use a reserved character, enclose that character (except delete/rubout) in single-quotation marks ('). The default kill character is the question mark (?), although your System Administrator may have selected another character.
- NOXOFF** Disables the XOFF/XON feature for the terminal line; this is the default setting. If -NOXOFF is in effect, you cannot suspend output from the computer to the terminal by using Ctrl-S.
- XOFF** Enables the XOFF/XON feature for the terminal line; -NOXOFF is the default setting. With XOFF/XON enabled, output from the computer to the terminal is suspended by using Ctrl-S (XOFF); output is resumed by using Ctrl-Q (XON).

If you issue **TERM** without an option, it displays a list of the **TERM** options available.

---

**Note**

At Rev. 21.0, the **TERM** command options **-BREAK**, **-DISPLAY**, **-ERASE**, **-FULL**, **-HALF**, **-KILL**, **-NOXOFF**, and **-XOFF** support Network Terminal Service (NTS) lines. For more information on NTS, see the *NTS User's Guide*.

---

**Usage**

Use **TERM** to change the characteristics of the asynchronous line of your terminal from their previous settings. The new settings remain in effect until you reset them or until you log out. The login procedure resets the line's settings (except for the duplex mode) to their default values. To reset the duplex mode to full duplex at login, use the **-FULL** option.

The following example illustrates the TERM command:

**Example 1: Listing current settings:**

```
OK, TERM -DISPLAY

Erase = ^ = '242 octal
Kill = ? = '277 octal
Full Duplex
Xon/Xoff disabled
Break on
OK,
```

**Example 2: Changing the erase character to \$, the kill character to \* and turning on Xon/Xoff:**

```
OK, TERM -ERASE $ -KILL * -XOFF -BREAK ON
OK, TERM -DISPLAY

Erase = $ = '244 octal
Kill = * = '252 octal
Full Duplex
Xon/Xoff enabled
Break on
OK,
```

## THEMIS

THEMIS invokes the THEMIS logic simulation program.

### **Format**

```
THEMIS
```

### **Usage**

THEMIS, a separately priced product, is Prime's logic simulation program that allows you to interactively examine digital circuit models. For detailed information, see the *THEMIS User's Guide* or the *THEMIS Reference Guide*.

TIME

TIME

TIME displays the amount of computer time used during a terminal session.

### Format

TIME

### Usage

TIME lists the current values stored in the PRIMOS time accounting registers. The three values displayed correspond to the three values in the logout message, which are as follows:

| <i>Value</i>         | <i>Units</i>     | <i>Meaning</i>                                                                                                                   |
|----------------------|------------------|----------------------------------------------------------------------------------------------------------------------------------|
| <b>Connect time</b>  | Hours, minutes   | Time since login.                                                                                                                |
| <b>Compute time</b>  | Minutes, seconds | Time accumulated executing commands or using programs (does not include disk I/O time).                                          |
| <b>Disk I/O time</b> | Minutes, seconds | Time accumulated for disk input/output. Includes user-requested I/O to files and paging I/O time generated on the user's behalf. |

All times include PRIMOS supervisor services, such as the time spent executing supervisor subroutines on the user's behalf. Some supervisor service associated with the PRIMOS scheduler is charged to the supervisor (at the supervisor's terminal) and not the user. When the system is idle, CPU time is charged to the supervisor. Computer time does not include I/O time for diskettes or for disks connected to a type 4000 controller.

The following example illustrates the TIME command:

```
OK, 12s  
Time used: 00h 52m connect, 00m 27s CPU, 00m 26s I/O.  
OK,
```

TRAMLC

TRAMLC transmits or receives a file over an assigned asynchronous line between two Prime computer systems using transparent protocol.

**Format**

TRAMLC

**Usage**

TRAMLC is intended for transferring files when PRIMENET is not available. Both computers must use the same baud rate and both must issue the TRAMLC command, one to transmit and the other to receive. Either computer can issue the command first. Before issuing TRAMLC, assign the appropriate asynchronous line with the following command format:

ASSIGN ASYNC LINE *n* [-TO *m*]

*n* and *m* are the line numbers in decimal.

When you issue TRAMLC, it displays the following prompt:

FNAME

Reply by entering the filename or pathname of the file you wish to transmit or receive. TRAMLC then prompts

T/R LINE# BLOCK

You must enter three parameters on the same line for this prompt. The meaning and parameters of the prompt are as follows:

| <i>Parameter</i> | <i>Input Required</i>                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| T/R              | Transmit or Receive? Enter T to send a file or R to receive one.                                                                                                                                                                                                                                                                                                                                                                                                      |
| LINE#            | Enter the octal number of your asynchronous line.                                                                                                                                                                                                                                                                                                                                                                                                                     |
| BLOCK            | When TRAMLC transfers a file, it divides it into 60-halfword blocks and transmits one block at a time. If you want to monitor the progress of your transmission, TRAMLC will display a Block <i>n</i> transmitted message at your terminal. The number of messages you receive depends on the response you give to this prompt. If you enter 0, TRAMLC sends no messages. If you enter a positive integer <i>n</i> , TRAMLC sends a message at every <i>n</i> blocks. |



## UNASSIGN

UNASSIGN releases exclusive control of a previously assigned line or device.

The UNASSIGN command can be entered at the supervisor terminal or at the user terminal to which the device is currently assigned. When issued from a user terminal, UNASSIGN unassigns only the device that was previously assigned to the user. On selected devices, this command turns off the device and clears the associated I/O buffers.

All devices assigned by a user are released when that user logs out. If a user has assigned a device and has left without logging out, this command releases that device from the supervisor terminal.

**Format**

```
UNASSIGN {
  device [-UNLOAD]
  -ALIAS MTldn [-UNLOAD]
  ASYNC -LINE n [-TO m]
  DISK pdev
  MTpdn [-UNLOAD]
}
```

**Arguments and Options**

|                                      |                                                                                                                                                                                                                                                                                                                                     |
|--------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>device</i>                        | Specifies the device that was assigned with the ASSIGN command. The device codes are the same as those listed in the description of the ASSIGN command.                                                                                                                                                                             |
| -ALIAS MT <i>ldn</i>                 | Unassigns the tape drive with logical device number <i>ldn</i> . The values of <i>ldn</i> numbers range from 0 to 7, inclusive. Do not place a blank between MT and <i>ldn</i> . For example the tape drive with logical device number 0 is MT0.                                                                                    |
| ASYNC -LINE <i>n</i> [-TO <i>m</i> ] | Releases control of an asynchronous line or a range of consecutive lines. <i>n</i> is a decimal line number. If you are unassigning more than one line, use the -TO option. <i>n</i> is the first value in a series of line numbers and <i>m</i> (following -TO) is the last. The value of <i>m</i> must be greater than <i>n</i> . |
| DISK <i>pdev</i>                     | Unassigns the disk partition with physical device number <i>pdev</i> . <i>pdev</i> is the physical device number as listed by the STATUS DISKS command.                                                                                                                                                                             |
| MT <i>pdn</i>                        | Unassigns the tape drive with physical device number <i>pdn</i> . The values of <i>pdn</i> range from 0 to 7, inclusive. Do not place a blank between MT and <i>pdn</i> . For example the tape drive with physical device number 0 is MT0.                                                                                          |
| -UNLOAD                              | Rewinds and unloads the tape.                                                                                                                                                                                                                                                                                                       |

## Unassigning Asynchronous Lines

To unassign an asynchronous line, use the following format:

```
UNASSIGN ASYNC -LINE n [-TO m]
```

---

### Note

At Rev. 21.0, the UNASSIGN AMLC command is obsolete. Although its use is still supported, it is not recommended.

---

To unassign Network Terminal Service (NTS) lines, you must specify the PRIMOS line number in decimal notation. Also, the Network Terminal Service must be running in order for you to unassign NTS lines with the UNASSIGN command. For more information on NTS, see the *NTS User's Guide*.

## Unassigning Tape Drives

Magnetic tape drives are unassigned by specifying a physical or logical device number. To unassign a tape drive using the physical device number (pdn) use the following format:

```
UNASSIGN MT pdn [-UNLOAD]
```

To unassign by specifying a logical device number use the following format:

```
UNASSIGN -ALIAS MT ldn [-UNLOAD]
```

The logical device number must have been previously assigned to that particular drive and must be specified with the -ALIAS option as shown.

If an operator unassigns a user-dedicated tape drive, no message is displayed at that user's terminal. Should the user subsequently attempt to unassign the same device, an error message is displayed.

For further details on unassigning magnetic tape drives, see the *Data Backup and Recovery Guide*.

## Unassigning Disks

For a disk to be assigned to a user as an I/O device, it must neither be assigned to another user nor be in the PRIMOS file system. The disk must also be specified by an entry in the Assignable Disks Table. (Refer to the description of the ASSIGN command.) A disk that has been assigned by a user cannot be entered as an argument in the ADDISK command.

If the disk is assigned to PRIMOS, it must be released by the SHUTDN command at the supervisor terminal. The operator can also use UNASSIGN to unassign disk partitions as part of system maintenance.



To unassign a disk partition that is assigned to you, use the format:

**UNASSIGN DISK *pdev***

For information on UNASSIGN as an operator command, see the *Operator's Guide to System Commands*.

## UPCASE

UPCASE creates an uppercase-only file from a file containing both lowercase and uppercase characters.

UPCASE scans an input file and then creates an output file that contains the same data, except that all lowercase alphabetic characters are replaced with their uppercase equivalents. The contents of the input file are not changed in any way.

### Format

**UPCASE *in-pathname out-pathname***

### Argument

*in-pathname* is the input file and *out-pathname* is the output file.

### Usage

In the following example of UPCASE, LOWERCASE.FILE is the name of the input file and UPPERCASE.FILE is the output file. The SLIST command displays the contents of both files.

```
OK, SLIST LOWERCASE.FILE
This file contains both UPPERCASE LETTERS and
lowercase letters.
OK, UPCASE LOWERCASE.FILE UPPERCASE.FILE
OK, SLIST UPPERCASE.FILE
THIS FILE CONTAINS BOTH UPPERCASE LETTERS AND LOWERCASE
LETTERS.
OK,
```

UPCASE is especially useful for preparing files for output to a device (such as a printer) that handles only uppercase alphabetic characters.







writes to the locate buffer. Discounting a very small number of assigned disk operations, the number of I/O operations (IO/S) is calculated as follows:

$$\text{IO/S} \approx \text{PIO/S} + \text{LM/S} + \text{LocateWrites/S}$$

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

$$\text{LocateWrites/S} \approx \text{IO/S} - (\text{PIO/S} + \text{LM/S})$$

The following example illustrates a command line that specifies automatic sampling:

```
OK, USAGE -FREQ 1800 -TIMES 10
```

The command causes PRIMOS to monitor the system 10 times with an interval of 1800 seconds (30 minutes) between each sampling for a period of 5 hours (1800 seconds x 10 times / 3600 seconds-per-hour).

For detailed information on USAGE, including user, system, and disk I/O metering display definitions, see the *Operator's Guide to System Monitoring*.

## USERS

USERS displays the number of users currently logged in to the system.

### **Format**

```
USERS
```

### **Usage**

USERS does not take arguments or options. The supervisor terminal is not included in the count nor are users who are logged in remotely to other systems (unless they are using NETLINK). The following example illustrates this command:

```
OK, USERS
Users = 54
OK,
```

.....  
UX\_TAPE

## UX\_TAPE

UX\_TAPE either saves files to tape in a format that the UNIX CPIO and TAR utilities can read or it restores files from a tape created by either CPIO or TAR.

### Format

$$\text{UX\_TAPE } \textit{source-pathname} \text{ -MT } n \left[ \begin{array}{l} \text{-LIST} \\ \left\{ \begin{array}{l} \text{-RESTORE } [\textit{suboptions}] \\ \text{-SAVE } [\textit{suboptions}] \end{array} \right\} \end{array} \right]$$

### Argument and Options

|                                       |                                                                                                                         |
|---------------------------------------|-------------------------------------------------------------------------------------------------------------------------|
| <i>source-pathname</i>                | Specifies the objects you want to save or restore. You can use wildcards and iteration.                                 |
| <b>-MT</b> <i>n</i>                   | Specifies the unit number of the tape drive on which the reel is mounted. The drive must be online and assigned to you. |
| <b>-LIST</b>                          | Displays the names of the files on a tape or those being copied to or from a tape.                                      |
| <b>-RESTORE</b> [ <i>suboptions</i> ] | Restores files from tape. See the <b>-RESTORE</b> Option section below for an explanation of the available suboptions.  |
| <b>-SAVE</b> [ <i>suboptions</i> ]    | Writes files to tape. See the <b>-SAVE</b> Option section below for an explanation of the available suboptions.         |

#### **-LIST** Option

The **-LIST** option used alone lists the files on a tape written by a UNIX system. When used with either the **-SAVE** or **-RESTORE** options, **-LIST** displays the names of the files successfully copied. Use the following format for UX\_TAPE with the **-LIST** option:

$$\text{UX\_TAPE } \textit{source-pathname} \text{ -MT } n \text{ -LIST } \left[ \left\{ \begin{array}{l} \text{-RESTORE } [\textit{suboptions}] \\ \text{-SAVE } [\textit{suboptions}] \end{array} \right\} \right]$$

#### **-SAVE** Option

The **-SAVE** option causes UX\_TAPE to save files to tape. Use the following format for UX\_TAPE with the **-SAVE** option:

$$\text{UX\_TAPE } \textit{source-pathname} \text{ -MT } n \text{ -SAVE } [\textit{suboptions}]$$

The **-SAVE** suboptions are:

- APPEND** Appends the files to the tape. The files must be in the same format (CPIO-readable or TAR-readable) as the files already on the tape.
- BLOCK\_FACTOR *n*** Specifies the blocking factor *n*, to use when writing. *n* is in the range 1-24, with the default value of 20. This option is an alternative to the **-BLOCK\_SIZE** option, and is for users who are familiar with the UNIX concept of blocking factors. A blocking factor of 1 is equivalent to a block size of 512 bytes. **-BLOCK\_SIZE** and **-BLOCK\_FACTOR** are mutually exclusive options; you cannot use both on the same command line.
- BLOCK\_SIZE *n*** Specifies the block size *n*, in bytes, to use when writing. The default value for *n* is 10K bytes. This option is an alternative to the **-BLOCK\_FACTOR** option, and is for users familiar with the PRIMOS concept of block size. A block size of 512 bytes is equivalent to a blocking factor of 1. **-BLOCK\_SIZE** and **-BLOCK\_FACTOR** are mutually exclusive options; you cannot use both on the same command line.
- CPIO** Writes the tape in a format that the UNIX CPIO utility can read.
- LCASE** (Default) All PRIMOS pathnames convert to lowercase UNIX pathnames. For information on processing nonalphanumeric characters, see the *Data Backup and Recovery Guide*.
- MAX\_FILENAME\_LEN** Allow you to use filenames up to 32 characters long.
- NO\_REWIND** Suppresses tape rewind after the save or restore ends.
- NO\_TRANSLATE** Suppresses the text translation function. **-NO\_TRANSLATE** is used when the data is in binary format.
- POS *n*** Positions the tape *n* filemarks from the beginning of the tape before the save or restore begins. This option is not valid for a drive with a 60MB cartridge tape.
- SWAP** Reverses the order of the bytes in each word. This option provides compatibility with systems on which data words have the least-significant byte first.
- TAR** (Default) Writes the tape in a format that the UNIX TAR utility can read.

**-UPCASE** All PRIMOS pathnames convert to uppercase UNIX pathnames. For information on processing nonalphanumeric characters, see the *Data Backup and Recovery Guide*.

**-RESTORE Option**

The **-RESTORE** option causes UX\_TAPE to restore files from tape. Follow this syntax line when using UX\_TAPE with the **-RESTORE** option:

`UX_TAPE source-pathname -MT n -RESTORE [suboptions]`

The **-RESTORE** suboptions are:

- CPIO** Reads a tape written in CPIO format.
- LCASE** (Default) All lowercase UNIX pathnames convert to uppercase PRIMOS pathnames. Uppercase UNIX pathnames remain in uppercase with each uppercase letter preceded by a slash (/). Filenames that begin with numerals, the dash (-), or the underscore (\_) characters are preceded by a slash and an ampersand (/&). For information on processing nonalphanumeric characters, see the *Data Backup and Recovery Guide*.
- NO\_QUERY** Suppresses queries when a restored file has the same name as a file already in the current directory.
- NO\_REWIND** Suppresses tape rewind after the save or restore ends.
- NO\_TRANSLATE** Suppresses the text translation function. **-NO\_TRANSLATE** is used when the data is in binary format.
- POS n** Positions the tape *n* filemarks from the beginning of the tape before the restore begins. This option is not valid for a drive with a 60MB cartridge tape.
- SWAB** Reverses the order of the bytes in each word. This option provides compatibility with systems on which data words have the least-significant byte first.
- TAR** (Default) Reads a tape in TAR format.
- UPCASE** All UNIX pathnames convert to uppercase PRIMOS pathnames. For information on processing nonalphanumeric characters, see the *Data Backup and Recovery Guide*.



---

**Caution**

Do not use this option unless all filenames are unique regardless of case. For example, using `-UPCASE`, the UNIX files `README` and `ReadMe` both translate to the PRIMOS file `README`. One file overwrites the other in the translation process because only one file called `README` can exist in the PRIMOS directory.

---

For more information, see the *Data Backup and Recovery Guide*.

V

See `VRTSSW`.

## VPSD, VPSD16

`VPSD` loads and starts the V-mode version of the Prime Symbolic Debugger (PSD).

`VPSD16` loads the V-mode version of the Prime Symbolic Debugger (PSD) at `160000g` and starts it. The commands and operation of `VPSD16` are the same as those for `VPSD`.

### Format

`VPSD`  
`VPSD16`

### Usage

When the `VPSD` program starts, it displays a dollar-sign prompt (\$) and waits for a debugging command. For detailed information on both commands, see the discussion of `VPSD` in the *Assembly Language Programmer's Guide*.

See also `DBG`; `HPSD`; `IPSD`; `PSD`.

## VRPG

`VRPG` loads the Prime RPG II V-mode compiler and compiles an object program from an ASCII source file named *pathname*.

### Format

`VRPG` *pathname* [*options*]

### Argument

*pathname* may have either a `.VRPG` or a `.RPG` suffix, because the `VRPG` compiler checks for a suffix in that order. Use the `BIND` or the `SEG` command to create a runfile from the object file. (`BIND` is recommended.)



## WATCH

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

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

To use the WATCH command, access privilege must be granted either by the user to be watched or by the System Administrator. The monitored user grants access by issuing a SHOW command. The second way to get WATCH access is by being a member of the .WATCH\$ ACL group. Users in this ACL group can watch any user on the system. This type of access is helpful when assistance is needed and a SHOW command can not be executed (when a user needs help while an application is running). The System Administrator can add or remove a user from this ACL group by using the CONFIG\_USERS command.

### Format

```
WATCH { username
        -OFF
        -RAW
        -STATUS
        -USER_NUMBER n
        -HELP
      }
```

### Options

- |                 |                                                                                                                                                                                                                                        |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>username</i> | Starts to watch the input and output of <i>username</i> . <i>username</i> must be the only user logged in with this user ID. If there are multiple users logged in with the same user ID, the WATCH command displays an error message. |
| -OFF            | Stops the current watch session of the user issuing the command.                                                                                                                                                                       |
| -RAW            | Sends non-printable characters, such as control characters and escape sequences, to the monitoring terminal. Without the -RAW option, non-printable characters are converted to at signs (@) before they are sent to your terminal.    |

.....  
**WATCH**

- STATUS** Displays the user name and user number you are currently watching (if any) and the user name and user number of the user (if any) that is watching you.
- USER\_NUMBER *n*** Starts to watch input and output of user number *n*. The user must be currently logged in. Use the STATUS USER command to obtain the correct user number.
- HELP** Displays command options.

**Usage**

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

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

On the watched user's terminal the following message appears:

```
*** watching-user (user nn on SYSTEM) at hh:mm  
Watch session started.
```

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

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

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

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

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

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

## WORD

WORD invokes the PRIMEWORD Word Processing system. If PRIMEWORD is not installed on your system, this command is not recognized by PRIMOS.

**Format**

WORD [*document\_name*] [*options*]

**Usage**

If you do not specify a document name or any options, the PRIMEWORD Main Menu is displayed. If you specify *document\_name* without any options, the document is displayed and you enter the PRIMEWORD Edit function directly, bypassing the usual menus and prompts.

For further information, see the *PRIMEWORD Reference Guide*.

A command line that begins with an asterisk (\*) signifies a PRIMOS null command, that is, a command that performs no action. *text* is any character string. However, unlike a command line beginning with a slash and an asterisk (\*), this command line is evaluated by the command processor. Thus, it is possible for this type of line to produce error messages that will halt a program, even though the line was not intended to execute.

**Format**

[*text*]

**Usage**

The following series of command lines illustrate the effect of an initial asterisk:

```
OK, ...
OK, ...
Not found. THIS (std$cp)
Error in variable or command function reference. (std$cp)
ER!
Function call contains too many left brackets. (EVAL_AF)
Error in variable or command function reference. (std$cp)
ER!
```



*Replace this page with the tab page labeled*

**3 PRIMOS Command  
Functions**

# Dictionary of Command Functions

## ***Introduction***

A command function performs a specific operation and returns a value which replaces the command function expression on the command line.

Command functions are intended primarily for use with Prime's Command Procedure Language (CPL). However, they may be invoked on the command line as well. On the command line, these functions are most commonly used following the TYPE command, which displays the function's returned value.

Functions and their arguments are always enclosed in square brackets. For example,

```
TYPE [ 4826 ]
```

When the command line is processed, the function call is evaluated. The value returned by the function call then replaces the function call itself in the command line. For example, the command line shown above would evaluate to "TYPE 4826". Users do not see this intermediate stage, but see only the following at their terminals:

```
OK, [ 4826 ]
4826
OK,
```

Several of the command functions either require or allow the use of variables. If these functions are invoked at command level, all variables must be global. Chapter 4 describes the use of global variables at command level.

If a variable is required as an argument within a function call, the variable name need not be bounded by % symbols. The % symbols are required, however, if a variable value is substituted for an integer or character string argument in a function call. For example, .VAR1 requires % symbols in the following function call, which can accept either variables or integers:



■ ■ ■ ■ ■ ■ ■ ■ ■ ■

However, the variable `.VAR1` does not require `%` symbols in the following function call, which accepts only variables as its arguments:

```
[GET_VAR .VAR1]
```

Command functions fall into four categories: arithmetic functions, file system functions, string-handling functions, and miscellaneous functions. The functions are listed by category in Chapter 1.

For detailed information on user-written command functions and on calling command functions from programs, see the *Advanced Programmer's Guide, Volume III: Command Environment*.

## Command Functions

### [ABBREV]

ABBREV expands the value of an abbreviation or returns the status of an abbreviation file.

#### Format

```
[ABBREV { -EXPAND text
          -STATUS } ]
```

#### Usage

The ABBREV -EXPAND format retrieves the value of *text* from the currently active abbreviation file, and returns the expanded value as its result. If *text* is not an abbreviation, *text* itself is returned.

If the abbreviation AM stands for ATTACH DIR>MYDIR, then

```
[ABBREV -EXPAND AM]
```

returns

```
ATTACH DIR>MYDIR
```

The ABBREV -EXPAND format treats *text* as a command line and expands it. Only command abbreviations (that is, abbreviations created by the commands ABBREV -ADD and ABBREV -ADD\_COMMAND) are expanded. To expand argument abbreviations, put a null string (") in the command position, as follows:

```
[ABBREV -EXPAND '' text]
```

The ABBREV -STATUS format returns the absolute pathname of your currently active abbreviation file. For example, if the name of your abbreviation file is LOGIN.ABBREV and is in the directory <FRUIT>PEACH, then

```
[ABBREV -STATUS]
```

returns

```
<FRUIT>PEACH>LOGIN.ABBREV
```

.....  
[AFTER]

If you turn off your abbreviation file with the ABBREV -OFF command, the ABBREV -STATUS function displays the pathname of the file followed by the word -OFF, as in the following example:

```
OK, ABBREV -OFF
OK, TYPE [ABBREV -STATUS]
<FRUIT>PEACH>LOGIN.ABBREV -OFF
OK,
```

If you have not activated your abbreviation file since you logged in, ABBREV -STATUS returns only -OFF.

Invocation of the ABBREV function fails if abbreviations are not enabled at your site.

[AFTER]

AFTER prints all text or characters that appear after the specified text or characters.

### Format

```
[AFTER string find-string]
```

### Usage

*string* is the text or characters to be searched and *find-string* is the text or characters to be located.

```
[AFTER HELLO E]
returns
LLO
```

If *find-string* is not in *string* or is at the end of *string*, AFTER returns the null string.

## [ATTACH\_POINT]

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

### Format

[ATTACH\_POINT [-ORIGIN]]

### Usage

For example, if your origin directory is <USERS>MYHOME and you have attached to <ACCOUNTING>REPORTS>OCTOBER,

[ATTACH\_POINT]  
returns  
<ACCOUNTING>REPORTS>OCTOBER

[ATTACH\_POINT -ORIGIN]  
returns  
<USERS>MYHOME

## [ATTRIB]

ATTRIB returns information about a specified file system object.

### Format

[ATTRIB *pathname* {*options*} [-BRIEF]]

### Argument and Options

|                 |                                                                         |
|-----------------|-------------------------------------------------------------------------|
| <i>pathname</i> | The name of the file, directory, segment directory, or access category. |
|-----------------|-------------------------------------------------------------------------|



[BEFORE]

BEFORE prints all text or characters that appear before the specified text or characters.

**Format**

[BEFORE *string find-string*]

**Usage**

*string* is the text or characters to be searched and *find-string* is the text or characters to be located. For example,

```
[BEFORE HELLO L]
returns
HE
```

If *find-string* is not in *string*, BEFORE returns *string*. If *find-string* is at the beginning of *string*, BEFORE returns the null string.

[CALC]

CALC evaluates any arithmetic and logical expressions found in *expression*.

**Format**

[CALC *expression*]

**Usage**

CALC 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:

.....  
[CALC]

Highest: ^ unary + unary -  
/ \*  
+ -  
= ^= < > <= >=  
&  
Lowest: |

---

**Note**

All operators evaluated by CALC 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.

---

Logical and relational operators return the Boolean values TRUE and FALSE. For example,

```
[CALC 9 > 3]  
returns  
TRUE
```

Relational operators accept either numeric or nonnumeric operands. If a relational operator is given a nonnumeric operand, an ASCII-based string comparison is done. If both operands are numeric or Boolean, an arithmetic comparison is done. Boolean TRUE is interpreted as 1 and FALSE as 0.

Arithmetic operators must have, as operands, integers or variables that convert to integers. The values of the operands, as well as the value returned by the function call, must be in the range -2147483648...+2147483647. (This is the range of signed values that can be represented by a 32-bit integer.)

Arithmetic operators return a character string representation of the numeric result. Arithmetic operators apply only to integer values; no floating-point arithmetic is supported at command level or in CPL.

All the arithmetic operators have the usual definition, except for / which returns only the truncated integer part of any noninteger result. For example,

```
[CALC 99 / 25]  
returns  
3
```

## [CND\_INFO]

CND\_INFO allows a CPL condition handler to examine the condition information of the most recent condition on the stack.

### **Format**

[CND\_INFO *option*]

### **Usage**

Use only one option at a time.

- CONTINUE\_SWITCH    Boolean value of the continue-to-signal switch. If no condition frame is on the stack, FALSE is returned.
- NAME                Name of the condition. If no condition frame is on the stack, \$NONE\$ is returned.
- RETURN\_PERMIT     Boolean value of the returned permitted switch. If no condition frame is on the stack, FALSE is returned.

For further information on CND\_INFO, see the *CPL User's Guide*. For further information on the Prime Condition Mechanism, see the *Subroutines Reference III: Operating System*.

## [COMO\_INFO]

The COMO\_INFO function returns the current state of your COMOUTPUT file in a format that can be used as an argument to a subsequent COMO command.

### **Format**

[COMO\_INFO]

### **Usage**

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

If there is no open como file, COMO\_INFO returns -TTY or -NTTY (depending on the current output state).



.....  
[DATE]

A typical use of this function in a CPL program would be to activate an alternate COMO file temporarily:

```
&SET_VAR COMO_STATE := [COMO_INFO]
                    COMO NEW_COMO_FILE
                    .
                    .
                    .
                    COMO [UNQUOTE %COMO_STATE%]
```

In the example, the current pathname and state of the active COMO file is saved in the variable `COMO_STATE`. The `COMO` command opens up a new COMO file named `NEW_COMO_FILE`. The second `COMO` command reactivates the original COMO file based on the information saved in `COMO_STATE`. The `UNQUOTE` command removes the quotes around the saved string so that the spaces will be interpreted as delimiters between the arguments.

[DATE]

DATE returns the current date and/or time in a variety of formats.

### Format

[DATE [option]]

### Options

Only one option can be specified. If you omit *option*, the date is returned in the format: `yy-mm-dd`. For example, `91-09-04` for September 4, 1991. If you use one of the options, the date and/or time is returned as follows (the format is shown by the example date of September 4, 1991 at 10:44:00 AM):

- AMPM        10:44 AM
- CAL         September 4, 1991
- DAY         4
- DOW         Wednesday
- FTAG        910904.104400
- FULL        91-09-04.10:44:00.Wed
- MONTH       September
- TAG         910904

```
-TIME      10:44:00
-USA       09/04/91
-UFULL     09/04/91.10:44:00.Wed
-VFULL     04 Sep 91 10:44:00 Wednesday
-VIS       04 September 91
-YEAR      1991
```

[DIR]

DIR returns the directory portion of a pathname, that is, all of a pathname except its final component (entryname). The function does not check for the existence of the file system object named by pathname.

**Format**

[DIR *pathname* [-BRIEF]]

**Usage**

For example,

```
[DIR JOHN>DATA>EXP1]
returns
JOHN>DATA
```

\* (asterisk) is returned if *pathname* is a simple filename.

If you specify -BRIEF, most error messages produced by the function are suppressed. Only error messages indicating improper invocation of the function or its arguments are printed.

.....  
[ENTRYNAME]

[ENTRYNAME]

ENTRYNAME returns the entryname portion of a pathname, that is, its final component. The function does not check for the existence of the pathname.

### Format

[ENTRYNAME *pathname*]

### Usage

For example,

[ENTRYNAME JOHN>DATA>EXP1]

returns

EXP1

*pathname* itself is returned if it is a simple filename. For example,

[ENTRYNAME HELLO]

returns

HELLO

[EXISTS]

EXISTS is a Boolean function that determines

- Whether a file system object exists
- Whether the file system object is of a specified type (file, directory, segment directory, or access category)

### Format

EXISTS *pathname* [*type*] [--MKIEF]

### Usage

*pathname* is the name or pathname of a file or directory.

*type* is a file type specifier. It may be omitted (defaulting to `-ANY`) or may be one of the following:

- `-ANY`
- `-ACCESS_CATEGORY`
- `-FILE`
- `-DIRECTORY`
- `-SEGMENT_DIRECTORY`

The value `TRUE` is returned if *pathname* exists and matches the file type specified. The value `FALSE` is returned if *pathname* cannot be found or does not match the file type specified. If *type* is `-ANY` or is omitted, only the existence of *pathname* is checked.

For example, if the current directory contains a file called `HELLO`. The function calls

```
[EXISTS HELLO -ANY]
[EXISTS HELLO]
[EXISTS HELLO -FILE]
```

all return  
`TRUE`

because `HELLO` exists and is a file. However, the function call

```
[EXISTS HELLO -DIRECTORY]
```

returns  
`FALSE`

because `HELLO` is not a directory.

If you specify `-BRIEF`, most error messages produced by the function are suppressed. Only error messages indicating improper invocation of the function or its arguments are printed.

.....  
[EXPAND\_SEARCH\_RULES]

## [EXPAND\_SEARCH\_RULES]

EXPAND\_SEARCH\_RULES returns the fully-qualified pathname of an entryname. The entryname can be any file system object: file, directory, ACAT, or segment directory.

### Format

```
[EXPAND_SEARCH_RULES entryname [ -LIST_NAME listname  
-REFERENCING_DIR pathname  
-SUFFIX .sfx [, .sfx, ... ]  
filetype-options ]
```

### Options

-LIST\_NAME *listname*

Specifies the search list to use to locate the file system object. *listname* is the name of a search list. Only one search list can be specified at a time.

-REFERENCING\_DIR *pathname*

Specifies a search rule for PRIMOS to substitute for the [REFERENCING\_DIR] entries in the search list. EXPAND\_SEARCH\_RULES then uses the search list to expand the name of the file system object. *pathname* is the fully-qualified pathname of a directory.

-SUFFIX *.sfx* [, *.sfx*, ... ]

Specifies the suffix(es) to append to entryname before beginning the search. Each *sfx* must begin with a period, for example .RUN. Suffixes are appended in turn, in the order listed. If no match is found with any of the suffixes, PRIMOS searches for the entryname without a suffix.

### File Type Options

-ACCESS\_CATEGORY

Returns the fully-qualified pathname only if *entryname* is an access category.

-DIRECTORY

Returns the fully-qualified pathname only if *entryname* is a directory.

-FILE

Returns the fully-qualified pathname only if *entryname* is a file.

**-SEGMENT\_DIRECTORY**

Returns the fully-qualified pathname only if *entryname* is a segment directory.

## Usage

This function uses the PRIMOS search rules facility to determine the fully qualified path name of a file system object. It searches all of the locations specified to locate the desired file system object. If the *entryname* cannot be found, EXPAND\_SEARCH\_RULES returns the value \$ERROR\$.

When EXPAND\_SEARCH\_RULES is used without the -LIST\_NAME option PRIMOS checks *entryname* for one of the suffixes .RUN, .SAVE, or .CPL. If one is present, the COMMAND\$ search list is used. For all other *entrynames*, the ATTACH\$ search list is used. Typically, the ATTACH\$ search list directs a search for top-level directories.

Note that sub-directories are not searched unless listed in the specified (or default) search list used.

EXPAND\_SEARCH\_RULES might be used in a CPL program to set a variable to the fully-qualified pathname of a top-level directory. For example, if USERS is a top-level directory in <SYSTEM1, when you pass USERS as an argument to the following CPL program,

```
&ARGS FILENAME  
&SET_VAR PATHNAME := [EXPAND_SEARCH_RULES %FILENAME%]
```

the value of the variable PATHNAME will be set to <SYSTEM1>USERS.

You can create a search list in your directory (see SET\_SEARCH\_RULES in chapter 2) and direct EXPAND\_SEARCH\_RULES to use it with the -LIST\_NAME option. If you create a search list called MYRULES, and the file MYFILE is in one of the directories listed in MYRULES, you set the variable PATHNAME to the fully-qualified pathname of MYFILE as follows:

```
&S PATHNAME := [EXPAND_SEARCH_RULES MYFILE -LIST_NAME [LIST MYRULES]]
```

For further details concerning the search rules facility, refer to the *Advanced Programmer's Guide II*.

EXPAND\_SEARCH\_RULES can also be invoked on the command line. See the EXPAND\_SEARCH\_RULES entry in Chapter 2.

.....  
[GET\_VAR]

[GET\_VAR]

GET\_VAR returns the value of the variable name defined by *expr*.

### **Format**

[GET\_VAR *expr*]

### **Usage**

The string \$UNDEFINED\$ is returned if the variable named by *expr* has not been defined, or if no global variable file is active.

GET\_VAR is useful for testing whether a variable has been set. If the function is used at command level, only global variables may be accessed. For example, if .VAR1 has been set to 32, but .VAR2 has not been set to any value, then

```
[GET_VAR .VAR1]
```

returns

32

but

```
[GET_VAR .VAR2]
```

returns

\$UNDEFINED\$

## [GROUP\_LIST]

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

### Format

```
[GROUP_LIST [wildcard-pattern].
```

### Usage

GROUP\_LIST is functionally identical to the LIST\_GROUPS command, except that it returns the information as a quoted character string. A quoted string is handled as a single value, even if it consists of several words. For example,

```
[GROUP_LIST]
returns the string
.ADMINISTRATION .TOOLS .PROJECTS
```

The three words, for example, cannot be assigned to three variables until you unquote the string using the UNQUOTE function. See the CPL User's Guide for more information about quoted strings.

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

```
[GROUP_LIST *]
returns
.ADMINISTRATION
```

```
[GROUP_LIST *.PROJECTS]
returns
.ADMINISTRATION .PROJECTS
```



.....  
[GVPATH]

[GVPATH]

GVPATH returns the pathname of your active global variable file.

**Format**

[GVPATH]

**Usage**

If, for example, your active global variable file is <TOP>DIR>VARFILE, then

[GVPATH]  
returns  
<TOP>DIR>VARFILE

If no global variable file is defined or active, the function returns

-OFF

[HEX]

HEX converts a nonnegative hexadecimal number to its decimal equivalent.

**Format**

[HEX *number*]

**Usage**

*number* is the hexadecimal number or letter to be converted. For example,

[HEX C]  
returns  
12

[IF]

IF provides conditional branching.

### Format

```
[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 left and right parenthesis must be bounded by blanks.

---

For more information about expressions see CALC in this chapter.

The following are examples of valid expressions:

```
1 < 2
FLORIDA <= CALIFORNIA
'FLORIDA' ^= 'CALIFORNIA'
( MAINE ^= DELAWARE ) & ( CAROL = JOAN )
%terminal_type% = PST100
^ [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)
```

.....  
[IF]

*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 tilde (~) when variables are to be part of the abbrev.)

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

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

Using abbreviation created above:

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

The abbreviation EXIST prints the argument you entered (LOGIN.CPL), 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, ~ AB -AC EDIT [IF [USER_INFC -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 matches, then the function call becomes a null string. Notice 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 .      (the new prompt you created)
A REMOTE_UFD                  (attach to a remote directory)
PLATO->SATRE:REMOTE_UFD      (the prompt displays the remote name too)
```

## [INDEX]

INDEX locates and prints the starting position number of a substring in a string.

### **Format**

```
[INDEX string find-string]
```

### **Usage**

*string* is the text to be searched and *find-string* is the text or characters to be located in *string*. For example,

```
{INDEX ABCDEFGHIJ GH}
```

returns

```
7
```

.....  
[KLMD]

[KLMD]

KLMD returns an ASCII string containing information about Prime software.

### Format

[KLMD *pathname* { -ALL  
-DST  
-STD } ;

### Argument and Options

|                         |                                                                                                                                                                                                                                                                                                                                                                                                                     |               |                |                 |               |                   |               |              |                |                         |               |             |               |
|-------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------|----------------|-----------------|---------------|-------------------|---------------|--------------|----------------|-------------------------|---------------|-------------|---------------|
| <i>pathname</i>         | The pathname of a system code object file (for example EMACS.RUN). If <i>pathname</i> is not fully-qualified, KLMD uses the COMMAND\$ search rules to locate it. If KLMD cannot find the file, or if the file is not of the appropriate type, it issues an error.                                                                                                                                                   |               |                |                 |               |                   |               |              |                |                         |               |             |               |
| -ALL                    | Returns both standard and distribution data. The string consists of the following fields: <table><tr><td>standard data</td><td>148 characters</td></tr><tr><td>[undefined]</td><td>88 characters</td></tr><tr><td>distribution data</td><td>62 characters</td></tr><tr><td>[undefined]</td><td>130 characters</td></tr></table>                                                                                     | standard data | 148 characters | [undefined]     | 88 characters | distribution data | 62 characters | [undefined]  | 130 characters |                         |               |             |               |
| standard data           | 148 characters                                                                                                                                                                                                                                                                                                                                                                                                      |               |                |                 |               |                   |               |              |                |                         |               |             |               |
| [undefined]             | 88 characters                                                                                                                                                                                                                                                                                                                                                                                                       |               |                |                 |               |                   |               |              |                |                         |               |             |               |
| distribution data       | 62 characters                                                                                                                                                                                                                                                                                                                                                                                                       |               |                |                 |               |                   |               |              |                |                         |               |             |               |
| [undefined]             | 130 characters                                                                                                                                                                                                                                                                                                                                                                                                      |               |                |                 |               |                   |               |              |                |                         |               |             |               |
| -DST                    | Returns distribution data. The string consists of the following fields: <table><tr><td>organization</td><td>20 characters</td></tr><tr><td>individual</td><td>6 characters</td></tr><tr><td>issue date</td><td>18 characters</td></tr><tr><td>order number</td><td>8 characters</td></tr><tr><td>customer service number</td><td>10 characters</td></tr></table>                                                    | organization  | 20 characters  | individual      | 6 characters  | issue date        | 18 characters | order number | 8 characters   | customer service number | 10 characters |             |               |
| organization            | 20 characters                                                                                                                                                                                                                                                                                                                                                                                                       |               |                |                 |               |                   |               |              |                |                         |               |             |               |
| individual              | 6 characters                                                                                                                                                                                                                                                                                                                                                                                                        |               |                |                 |               |                   |               |              |                |                         |               |             |               |
| issue date              | 18 characters                                                                                                                                                                                                                                                                                                                                                                                                       |               |                |                 |               |                   |               |              |                |                         |               |             |               |
| order number            | 8 characters                                                                                                                                                                                                                                                                                                                                                                                                        |               |                |                 |               |                   |               |              |                |                         |               |             |               |
| customer service number | 10 characters                                                                                                                                                                                                                                                                                                                                                                                                       |               |                |                 |               |                   |               |              |                |                         |               |             |               |
| -STD                    | Returns standard data (default). The string consists of the following fields: <table><tr><td>product name</td><td>20 characters</td></tr><tr><td>revision number</td><td>20 characters</td></tr><tr><td>serial number</td><td>20 characters</td></tr><tr><td>licensee</td><td>40 characters</td></tr><tr><td>expiry date</td><td>18 characters</td></tr><tr><td>[undefined]</td><td>30 characters</td></tr></table> | product name  | 20 characters  | revision number | 20 characters | serial number     | 20 characters | licensee     | 40 characters  | expiry date             | 18 characters | [undefined] | 30 characters |
| product name            | 20 characters                                                                                                                                                                                                                                                                                                                                                                                                       |               |                |                 |               |                   |               |              |                |                         |               |             |               |
| revision number         | 20 characters                                                                                                                                                                                                                                                                                                                                                                                                       |               |                |                 |               |                   |               |              |                |                         |               |             |               |
| serial number           | 20 characters                                                                                                                                                                                                                                                                                                                                                                                                       |               |                |                 |               |                   |               |              |                |                         |               |             |               |
| licensee                | 40 characters                                                                                                                                                                                                                                                                                                                                                                                                       |               |                |                 |               |                   |               |              |                |                         |               |             |               |
| expiry date             | 18 characters                                                                                                                                                                                                                                                                                                                                                                                                       |               |                |                 |               |                   |               |              |                |                         |               |             |               |
| [undefined]             | 30 characters                                                                                                                                                                                                                                                                                                                                                                                                       |               |                |                 |               |                   |               |              |                |                         |               |             |               |

If no options are used, standard data is returned.

[KLMF]

KLMF returns an ASCII string containing an item of information about Prime software.

### **Format**

[KLMF *pathname* {*option*}]

### **Argument and Options**

*pathname*                      The pathname of a system code object file (for example EMACS.RUN). If *pathname* is not fully-qualified, KLMF uses the COMMAND\$ search rules to locate it. If KLMF cannot find the file, or if the file is not of the appropriate type, it issues an error.

Use only one of the following options at a time.

- CSM\_NUMBER                      Customer service maintenance number.
- EXPIRY\_DATE                      Date the software license expires.
- INDIVIDUAL                      Name of your Prime distribution contact.
- ISSUE\_DATE                      Date the copy of the product was issued.
- LICENSEE                      Name of the software licensee.
- ORDER\_NUMBER                      Order number used by Prime.
- ORGANIZATION                      Prime software distribution organization.
- PRODUCT                      Product name.
- REVISION                      Revision number.
- SERIAL\_NUMBER                      Serial number of your copy of the software.

.....  
[KLMT]

[KLMT]

KLMT compares a value you specify with the corresponding value of a Prime software attribute. KLMT returns TRUE if the value you specify matches the corresponding software attribute and FALSE if it does not match.

### Format

[KLMT *pathname* [*option* [*value*]] [*option* [*value*]] . . . [--PART]]

### Argument and Options

- pathname* The pathname of a system code object file (for example EMACS.RUN). If *pathname* is not fully-qualified, KLMT uses the COMMAND\$ search rules to locate it. If KLMT cannot find the file, or if the file is not of the appropriate type, it issues an error.
- PART Instructs KLMT to compare the characters in *value* with the same number of characters at the beginning of the attribute data. For example, if you specify --REVISION 21 --PART, KLMT compares 21 with the first two digits of the revision number of the Prime software. If the --PART option is present, all option/value pairs are affected.

Each *option/value* pair specifies an attribute of Prime software and a string to match against the corresponding attribute data. Values are converted to uppercase before they are compared. If *value* is omitted, it is assumed to be the null string. Select one or more attributes from the following:

- |                                 |                                                                                     |
|---------------------------------|-------------------------------------------------------------------------------------|
| --CSM_NUMBER [ <i>value</i> ]   | Compares <i>value</i> against the customer service maintenance number.              |
| --EXPIRY_DATE [ <i>value</i> ]  | Compares <i>value</i> against the date the software license expires.                |
| --INDIVIDUAL [ <i>value</i> ]   | Compares <i>value</i> against the name of your Prime distribution contact.          |
| --ISSUE_DATE [ <i>value</i> ]   | Compares <i>value</i> against the date the copy of the product was issued.          |
| --LICENSEE [ <i>value</i> ]     | Compares <i>value</i> against the name of the software licensee.                    |
| --ORDER_NUMBER [ <i>value</i> ] | Compares <i>value</i> against the order number used by Prime.                       |
| --ORGANIZATION [ <i>value</i> ] | Compares <i>value</i> against the name of Prime software distribution organization. |

- PRODUCT *[value]*                      Compares *value* against the product name.
- REVISION *[value]*                      Compares *value* against the revision number.
- SERIAL\_NUMBER *[value]*                Compares *value* against the serial number of your copy of the software.

**Usage**

KLMT returns TRUE if the value you specify matches the specified attribute's value stored in the Prime software. For example

```
[KLMT PROG.RUN -REVISION 23.3.00]
```

returns TRUE if the software revision number is 23.3.00, or FALSE if the software revision number is any other value.

When more than one option/value pair is given, KLMT returns TRUE if all of the values match, or FALSE if any do not match.

[LENGTH]

LENGTH prints the number of characters in a given character string.

**Format**

```
[LENGTH string]
```

**Usage**

For example,

```
[LENGTH HOW LONG IS THIS?]
```

returns

17

Leading, trailing, and multiple blanks within *string* are not counted by the function unless *string* is quoted.



.....  
[MOD]

[MOD]

MOD divides one number by another and returns the remainder (modulus).

### **Format**

[MOD *number1 number2*]

### **Usage**

*number1* is the dividend and *number2* is the divisor. For example,

```
[MOD 299 100]  
returns  
99
```

[NULL]

NULL tests a string for the occurrence of any text or characters and returns TRUE if no text or characters exist and FALSE otherwise.

### **Format**

[NULL *string*]

### **Usage**

*string* is the text or characters to be tested. For example, if the variable .VAR3 has been set to a null string, then

```
[NULL % .VAR3%]  
returns  
TRUE
```

[OCTAL]

OCTAL converts a nonnegative octal number to its decimal equivalent.

**Format**

[OCTAL *number*]

**Usage**

*number* is the octal number to be converted. For example,

```
[OCTAL 12]
returns
10
```

[OPEN\_FILE]

OPEN\_FILE opens a file for reading or writing. It returns the unit number, a decimal integer indicating the file unit on which it opened the file.

**Format**

[OPEN\_FILE *pathname status-var* -MODE *access-mode*]

**Arguments**

|                   |                                                                                                                                                                                                                                                                                                               |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>pathname</i>   | The name or pathname of the file to be opened.                                                                                                                                                                                                                                                                |
| <i>status-var</i> | The name of a global or local variable that is automatically set to 0 if the operation is successful and nonzero otherwise. If OPEN_FILE is used at command level, <i>status-var</i> must be a global variable. <i>status-var</i> may be either global or local if OPEN_FILE is invoked within a CPL program. |

*access-mode* specifies the mode of file access as follows:

|          |                |
|----------|----------------|
| R or r   | Read only      |
| W or w   | Write only     |
| WR or wr | Read and write |

.....  
[PATHNAME]

### Usage

It is often convenient to set a variable to the file unit number by using the SET\_VAR command in conjunction with the OPEN\_FILE function, as follows:

```
OK, SET_VAR .UNIT [OPEN_FILE YOYO .STAT -MODE R]  
OK,
```

%UNIT% may then be supplied as the file unit number when the READ\_FILE or WRITE\_FILE function is invoked. The variable must be global if it is set or used at command level. It may be either local or global if it is used within a CPL program.

[PATHNAME]

PATHNAME returns the full pathname (including the root entryname) of *path*.

### Format

```
PATHNAME path [-BRIEF]
```

### Usage

For PATHNAME to work, the entire directory portion of *path* must exist and the user's process must be able to attach to that directory. However, the final element in the pathname need not exist; if you supply PATHNAME with a nonexistent filename, it returns the same value that it would return if the filename exists.

If *path* is either an entryname or an asterisk (\*), PATHNAME returns the full pathname of the current directory with either the entryname or \* appended to it. If, for example, the current directory is ROOM, a subdirectory of HOUSE on the disk TOWN, then

```
PATHNAME *)  
returns  
<TOWN>HOUSE>ROOM>*
```

If *path* is a relative pathname, then its elements are appended to the pathname of the current directory. Thus, continuing the example above,

```
PATHNAME *)  
returns  
<TOWN>HOUSE>ROOM>KEY
```

If *path* is a pathname that includes one or more right-angle brackets (>) and does not begin with an asterisk (\*), PATHNAME assumes that *path* is a full pathname, that is, PATHNAME assumes that the specified pathname begins with a top-level directory. If the directory <FARM>BARN exists and is accessible, then the function call

```
[PATHNAME <FARM>BARN]
```

returns

```
<FARM>BARN>STALL
```

If <FARM>BARN does not exist or is not accessible, an error message is generated.

The -BRIEF option suppresses most error messages produced by the function. Only error messages indicating improper invocation of the function or its arguments are displayed.

[PDEV]

PDEV returns the pdev (physical device number), given the disk partition name or the four device specifications: controller number, disk unit number, the starting head number, and the total number of heads.

### Format

```
PDEV { -DISK diskname
      -ENCODE [-CONTROLLER cno -UNIT diskno -START headno -HEADS nheads ] }
```

### Usage

*diskname* is the name of a disk without enclosing angle brackets.

Suboptions for the -ENCODE option are:

|                              |                                                                      |
|------------------------------|----------------------------------------------------------------------|
| -CONTROLLER <i>cno</i>       | <i>cno</i> is the controller number.                                 |
| -UNIT <i>diskno</i>          | <i>diskno</i> is the unit number of the disk.                        |
| -STARTING_HEAD <i>headno</i> | <i>headno</i> is the number of the start head of the disk partition. |
| -NUM_HEADS <i>nheads</i>     | <i>nheads</i> is the number of heads for the partition.              |



The `-TTY` option forces the `QUERY` function to take input from the terminal. If you omit this option, the function takes its response from the command input stream. That is, if the function is invoked interactively, it goes to the terminal for its response. If the function is invoked from a command input file, or from a `&DATA` group within a CPL program, it goes to the command input file or CPL program for its input. A CPL program containing the function `QUERY` with the `-TTY` option cannot be executed as a batch job or phantom.

[QUOTE]

`QUOTE` places an outer pair of quotes around the text specified in *string* and doubles the quotes already inside the given string.

**Format**

```
[QUOTE string1 string2 ... stringn]
```

**Usage**

The `QUOTE` function is used to keep the meaning of special symbols from being interpreted during calls to subsystems. For example,

```
[QUOTE XY' 'Z]  
returns  
'XY' 'Z'
```

Note that the `TYPE` command removes a level of quotes from its argument, so that the following results are obtained when `TYPE` and `QUOTE` are used together:

```
OK, TYPE [QUOTE XY' 'Z]  
XY' 'Z'  
OK,
```

.....  
[READ\_FILE]

## [READ\_FILE]

READ\_FILE reads the next line from a previously opened ASCII file and returns the line as its value. The line is quoted if it contains special characters. The true null string (a string of zero length, containing no characters) is returned if end-of-file is encountered.

### Format

[READ\_FILE *unit status-var* [-BRIEF]]

### Usage

*unit* is the file unit number of the file to be read. It may be the decimal integer returned by the OPEN\_FILE function or a variable whose value was set using the SET\_VAR command with the OPEN\_FILE function. If you use a variable at command level, it must be a global variable.

*status\_var* is a variable that is automatically set to 0 if the operation is successful and to a nonzero integer otherwise. When end-of-file is reached, *status-var* is set to 1. If you invoke the function READ\_FILE at command level, *status-var* must be a global variable.

For example, if the file YOYO was opened on unit 35 and the next line in YOYO is "This is yoyo.", then the command

```
READ_FILE 35 .STAT
```

returns

```
This is yoyo.
```

and sets .STAT to 0. Subsequent calls to READ\_FILE return successive lines from the file. After the last line of YOYO is read, the next call to READ\_FILE returns the null line and sets .STAT to 1.

If you specify -BRIEF, most error messages produced by the function are suppressed. Only error messages indicating improper invocation of the function or its arguments are printed.

[RESCAN]

RESCAN removes one level of quotes from *string* and evaluates any function calls or variable references no longer appearing in quotes.

**Format**

[RESCAN *string*]

**Usage**

For example,

[RESCAN ' [DATE -DOW] ']

returns the day of the week.

The function may be used to force evaluation of quoted variables.

[RESPONSE]

RESPONSE displays the specified prompt text followed by a colon and a space and waits for you to enter a string. The string you enter is returned as the value of the function. If your response is only a carriage return, a default string is returned, if specified in the function call, otherwise RESPONSE returns a null string. The response string can also be provided by a command input file or CPL program when the function is used by batch or phantom processes.

**Format**

[RESPONSE *text* [*default*] [-TTY];

**Usage**

*text* is the prompt text to display. If *text* is null, no prompt is displayed, but your response is required. *text* and *default* must be entered in single quotes if they contain special characters or embedded blanks. One level of quotes is stripped before printing.



.....  
[RESPONSE]

Examples:

```
OK, TYPE [RESPONSE HELLO]
HELLO: HI THERE
HI THERE
OK,
```

With embedded blanks in the prompt text and the default used, it would be:

```
OK, TYPE [RESPONSE 'HELLO THERE' 'NO ANSWER' ]
HELLO THERE: <CR>
NO ANSWER
OK,
```

RESPONSE is typically used to make a CPL program interactive. The following lines exit the program if you enter N or <CR> , or continues if you enter Y or any other character:

```
.
.
.
&S ANSWER := [RESPONSE 'DO YOU WISH TO CONTINUE? (Y or N)' N]
&IF %ANSWER% = N &THEN &RETURN
.
.
.
```

The `-TTY` option forces the RESPONSE function to take input from the terminal. If you omit this option, the function takes its response from the command input stream. That is, if the function is invoked interactively, it goes to the terminal for its response. If the function is invoked from a command input file, or from an &DATA group within a CPL program, it goes to the command input file or CPL program for its input. A CPL program containing the RESPONSE function with the `-TTY` option cannot be executed as a batch job or phantom.

[REVERSE]

REVERSE returns an ASCII string with the characters of the argument string in reverse order.

**Format**

[REVERSE *string*]

**Usage**

*string* is the string to reverse.

For example,

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

[SEARCH]

SEARCH returns the index of the first character in *string1* that appears in *string2*.

**Format**

[SEARCH *string1 string2*]

**Usage**

For example,

```
[SEARCH ABCDEFG XYEF]
returns
5
```

because E is the fifth character in the string ABCDEFG. If no character in *string1* appears in *string2*, SEARCH returns 0.

.....  
[SUBST]

[SUBST]

SUBST (substitute) substitutes *string3* for *string2* wherever *string2* occurs within *string1*, and returns the altered *string1*.

### Format

[SUBST *string1 string2 string3*]

### Usage

For example,

```
[SUBST ABCDEFG DE QZ]
```

returns

ABCQZFG

[SUBSTR]

SUBSTR (substring) returns a substring of *string* that begins at position *start-position* and extends for length *num-chars*.

### Format

[SUBSTR *string start-position [num\_chars]*]

### Usage

If you omit *num-chars*, SUBSTR prints all characters from *start-position* to the end of *string*. *start-position* and *num-chars* (if given) must be positive integers.

For example,

```
[SUBSTR ABCDEFGHIJ 4 3]
```

returns

DEF

[SYSTEM\_INFO]

The SYSTEM\_INFO function retrieves system information similar to that provided by the USAGE, PRIMAN, or STATUS commands.

**Format**

[SYSTEM\_INFO *option*]

**Options**

- MEMORY Returns the amount of memory configured for the system. If the memory is less than 1 megabyte, it will be returned in kilobytes (e.g., 512KB). If it is at least 1 megabyte, it is returned in megabytes (e.g., 32MB).
- NAME Returns the current system name, as specified when it was booted. This is the default option.
- NUM\_CONTROLLERS Returns the number of disk controllers in use on the system.
- NUM\_DISKS Returns the number of disks in use on the system, from all controllers.
- PROCESSOR Returns the Prime standard name of the system's CPU (9955-II, 6355, 4150, etc.) A \$UNKNOWN\_CPU\$ result indicates that the processor type is outside of the function's knowledge (this only happens with very old processors, or if older revisions of this command function are run on very new systems).
- REVISION Returns the current PRIMOS revision.
- USERS Returns the number of users currently logged in.

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

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

.....  
[SYSTEM\_USAGE]

## [SYSTEM\_USAGE]

The SYSTEM\_USAGE function returns information about the system normally available only through USAGE or PRIMAN.

### **Format**

[SYSTEM\_USAGE *option* ]

### **Options**

|                     |                                                                                                            |
|---------------------|------------------------------------------------------------------------------------------------------------|
| <b>-CPTIME</b>      | Returns the number of seconds of CPU time used since boot.                                                 |
| <b>-IO_TIME</b>     | Returns the number of seconds of I/O time used since boot.                                                 |
| <b>-MEMORY</b>      | Returns the amount of memory in use, in pages.                                                             |
| <b>-SEGMENTS</b>    | Returns the number of segments currently being used.                                                       |
| <b>-TIME</b>        | Returns the total time of operation since boot, in HMS format (e.g., 4h7m24s). This is the default option. |
| <b>-UP_SINCE</b>    | Returns the boot time of the system in ISO format (e.g., 91-3-31.17:31:44.Sun).                            |
| <b>-WIRED_PAGES</b> | Returns the number of pages that are wired (cannot be flushed or re-allocated).                            |

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

[TO\_HEX]

TO\_HEX converts a decimal integer to its hexadecimal equivalent. Negative numbers are supported by this function.

**Format**

[TO\_HEX *number*]

**Usage**

*number* is the decimal number to be converted. For example,

[TO\_HEX 12]  
returns  
C

[TO\_OCTAL]

TO\_OCTAL converts a decimal integer to its octal equivalent. Negative numbers are supported by this function.

**Format**

[TO\_OCTAL *number*]

**Usage**

*number* is the decimal number to be converted. For example,

[TO\_OCTAL -8]  
returns  
-10

.....  
[TRANSLATE]

## [TRANSLATE]

TRANSLATE replaces characters in one string with characters from another.

### Format

```
[TRANSLATE string1 string2 string3]
```

### Usage

TRANSLATE looks for *string3* characters in *string1*, replaces them with corresponding characters from *string2*, then returns the altered *string1*.

For example,

```
[TRANSLATE BRAVE 12345 ABCDEF]
```

returns

```
2R1V5
```

If *string2* and *string3* are omitted, TRANSLATE converts all *string1* characters to uppercase, then returns *string1*.

## [TRIM]

TRIM removes a specified character from the left, the right, or both sides of a given string.

### Format

```
[TRIM string { { -LEFT  
-RIGHT  
-BOTH } } [char ] ]
```

### Usage

*string* is the string of characters to be modified, and *char* is the character to be removed. For example,

```
[TRIM XWHELLOXX -BOTH X]
```

returns

```
HELLO
```

If you omit *char*, leading and/or trailing blanks are removed. If you do not specify a side (-LEFT, -RIGHT, or -BOTH) when blanks are removed, -BOTH is assumed.

## [UNQUOTE]

UNQUOTE removes the outer pair of quotes around the text specified in *string* and changes all other remaining pairs of quotes within *string* to single quotes.

### Format

[UNQUOTE *string*]

### Usage

For example,

```
[UNQUOTE '''XX''''YY''']
```

returns

```
'XX' 'YY'
```

Note that the TYPE command also removes a level of quotes from its argument, so that the following results are obtained when TYPE and UNQUOTE are used together:

```
TYPE [UNQUOTE '''XX''''YY''']
```

returns

```
XX' YY
```

## [USER\_INFO]

The USER\_INFO function returns information about a user's process. This function normally defaults to the current user's process but may be used to examine other processes as well.

### Format

[USER\_\_INFO [*user-number*] [*option*]]

### Options

Use only one option at a time.

-NAME Returns the user's login ID.

-NUMBER Returns the user's number. This is the default option.



.....  
[USER\_INFO]

|                    |                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|--------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-PRIORITY</b>   | Returns the user's current task priority, either a number from 4 (highest) to 0 (lowest), IDLE (process on idle queue) or SUSPEND (process is suspended).                                                                                                                                                                                                                                                                                   |
| <b>-PROCESS_ID</b> | Returns the unique process ID of the user. This option cannot be specified when requesting information about other users — it is restricted to information on one's own process only.                                                                                                                                                                                                                                                       |
| <b>-PROJECT</b>    | Returns the user's project ID.                                                                                                                                                                                                                                                                                                                                                                                                              |
| <b>-TYPE</b>       | Returns the user's type, which is assumed to be one of the following: <b>TERMINAL</b> , <b>REMOTE</b> , <b>CONSOLE</b> , <b>PHANTOM</b> , <b>FAM</b> , <b>SLAVE</b> , <b>SERVER</b> , <b>PROCESS</b> , <b>CHILD</b> , <b>NETMAN</b> , <b>OTHER</b> . <b>PROCESS</b> is a label for the various system processes, such as the logout server or kernel processes. <b>SERVER</b> is used for such tasks as the name server, login server, etc. |

### Usage

If no options are specified, the default is to return the user's number (**-NUMBER**). If a user number is specified, but no command option is given, the default is to return the user's name (**-NAME**).

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

```
[USER_INFO -NAME]
```

```
returns  
YOURID
```

```
[USER_INFO 150 -TYPE]
```

```
returns  
TERMINAL
```

```
[USER_INFO 300 -TYPE] (when user number 300 is unassigned)
```

```
returns  
$UNKNOWN$
```

## [USER\_USAGE]

USER\_USAGE returns usage information for your own (or a specified) process.

### Format

[USER\_USAGE [*user-number*] [*option*]]

### Options

Use only one option at a time.

|                     |                                                                                                       |
|---------------------|-------------------------------------------------------------------------------------------------------|
| <b>-CONNECTED</b>   | Returns the time logged in for the process in hms format (e.g., 5h12m4s). This is the default option. |
| <b>-CPTIME</b>      | Returns the amount of CPU time (in seconds) used by the process since login.                          |
| <b>-IO_TIME</b>     | Returns the amount of I/O time (in seconds) used by the process since login.                          |
| <b>-LOGIN</b>       | Returns the date and time the process logged in, in ISO format (e.g., 91-03-31.17:31:44.Sun).         |
| <b>-MEMORY</b>      | Returns the number of non-shared pages currently in use by the process.                               |
| <b>-SEGMENTS</b>    | Returns the number of segments in use by the process.                                                 |
| <b>-WIRED_PAGES</b> | Returns the number of pages wired by the process.                                                     |

### Usage

If no option is specified, the default is to return the process's connect time (-CONNECTED).

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

Examples:

```
{USER_USAGE -WIRED_PAGES}
```

returns

340

```
{USER_USAGE 300 -SEGMENTS}    (when user number 300 is unassigned)
```

returns

\$UNKNOWN\$

[USER\_USAGE 1 -CONNECTED] is equivalent to [SYSTEM\_USAGE -TIME],  
[USER\_USAGE 1 -LOGIN] is equivalent to [SYSTEM\_USAGE -UP\_SINCE].

.....  
[VALIDATE]

[VALIDATE]

The VALIDATE function allows the user to validate input from a CPL RESPONSE function.

### **Format**

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

### **Options**

- |                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|---------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-DATE</b>        | Validates that the argument is a valid date. Date can be in USA form (MM/DD/YY), ISO form (YY-MM-DD), or Visual form (DD Mmm YY); a Visual form date must be a quoted string. Checks for valid character types and numeric values, including leap year check. Validates complete date strings as supported by CL\$PIX, including time of day and day of week; day of week is only validated to be a legal value, not necessarily the correct value for that date. The year field can be omitted or given as one, two, or four digits. Leading zeros are not required. Wildcards not accepted. |
| <b>-DECIMAL</b>     | Validates that the argument is a valid decimal integer. Wildcards not accepted.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| <b>-ENTRYNAME</b>   | Validates that the argument is a valid entryname.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| <b>-GROUP_ID</b>    | Validates that the argument is a valid ACL group ID.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| <b>-HEXADECIMAL</b> | Validates that the argument is a valid hexadecimal integer. Letters can be uppercase or lowercase. Wildcards not accepted.                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| <b>-NO_WILDCARD</b> | Disallows wildcard characters in any arguments.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| <b>-OCTAL</b>       | Validates that the argument is a valid octal integer. Wildcards not accepted.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| <b>-PASSWORD</b>    | Validates that the argument is a valid password. Wildcards not accepted.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <b>-PROJECT</b>     | Validates that the argument is a valid project ID.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| <b>-TREENAME</b>    | Validates that the argument is a valid PRIMOS treename.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| <b>-USER</b>        | Validates that the argument is a valid user ID.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |

### Usage

The VALIDATE function verifies that the argument string contains valid characters and is the proper length for the specified option, it does not verify the existence of an object.

For example, [VALIDATE -USER FOOBARHEAD] returns TRUE, indicating that FOOBARHEAD is a valid user ID. It does NOT verify that user FOOBARHEAD actually has a user profile on the system. For further details on acceptable values, refer to the CL\$PIX appendix to the *Subroutines Reference II: File System*.

The -NO\_WILDCARD option may be used to disallow wildcard characters in arguments. By default wildcards are accepted, except where indicated.

For example, [VALIDATE -TREE LOGIN.@] returns TRUE, whereas [VALIDATE -TREE LOGIN.@ -NO\_WILDCARD] returns FALSE.

### [VERIFY]

VERIFY returns an integer that represents the position (beginning with 1) of the first character in *string1* that does not appear in *string2*.

### Format

[VERIFY *string1 string2*]

### Usage

For example,

```
[VERIFY 123X456Y 654321]
```

returns

4

because X, the fourth character of 123X456Y, is the first character that does not appear in 654321.

If all characters of *string1* appear in *string2*, VERIFY returns 0.

■ ■ ■ ■ ■ ■ ■ ■ ■ ■  
[WILD]

[WILD]

WILD produces a list of all names within a directory that match one or more wildcard names.

The WILD function has two forms, which are discussed below. The first form returns all matching names at once, in a single list. Names within the list are separated by blanks. The list may not exceed 1024 characters in length. The second form, which uses the `-SINGLE` option, returns one matching name per invocation until the list of names is exhausted.

WILD returns the true null string if no entries are matched, or, when in `-SINGLE` mode, the end of the directory is reached.

### Format

[WILD *wildname1* [...] *wildname-n*] [*options*] [`-SINGLE` *unit-var*] [`-BRIEF`]

### Arguments and Options

*wildname1* [...] *wildname-n*

Specify wildcard names that the WILD function matches. If *wildname1* is a pathname, all the wildnames are looked for in the directory that *wildname1* specifies. Otherwise, all names are searched for in the current directory. *wildname2* through *wildname-n* may not be pathnames.

`-SINGLE` *unit-var*

Causes the WILD function to return names one at a time, rather than as a list. Use `-SINGLE` whenever you think the WILD list might overrun its limit of 1024 characters, or whenever it is more convenient to work with filenames one at a time.

Set *unit-var* to 0 before using the WILD function with the `-SINGLE` option.

(*unit-var* must be a global variable if the function is invoked at command level.)

WILD uses *unit-var* to store the number of the file unit on which it opens the directory for reading. The directory remains open until all matching names have been returned.

**-BRIEF** Most error messages produced by the function are suppressed. Only error messages indicating improper invocation of the function or its arguments are printed.

### Selection Options

More than one selection option may be used. For example, if you use both the **-CAM** and **-AFTER** options, the names of all CAM files and the names of all files modified after the date specified will be returned.

- ACCESSED\_AFTER *date*** Matches objects in Rev. 20.0 or later directories last accessed on or after *date*. (One format of *date* is mo/dd/yy.hh:mm:ss. For complete information on how to specify *date*, see the section Wildcard Options in Chapter 4.)
- ACCESSED\_BEFORE *date*** Matches objects in Rev. 20.0 or later directories last accessed before *date*.
- ACCESS\_CATEGORY** Matches access categories.
- AFTER *date*** Matches objects last modified on or after *date*.
- BACKEDUP\_AFTER *date*** Matches objects backed up with the BACKUP utility on or after *date*.
- BACKEDUP\_BEFORE *date*** Matches objects backed up with the BACKUP utility before *date*.
- BEFORE *date*** Matches objects last modified before *date*.
- CAM** Matches CAM files. (This option returns the names of all file system objects when used on a disk of a pre-Rev. 23.3 system. Use the [ATTRIB] function to perform selection.)
- CREATED\_AFTER *date*** Matches objects in Rev. 20.0 or later directories created on or after *date*.
- CREATED\_BEFORE *date*** Matches objects in Rev. 20.0 or later directories created before *date*.
- DAM** Matches DAM files. (This option returns the names of all file system objects when used on a disk of a pre-Rev. 23.3 system. Use the [ATTRIB] function to perform selection.)

.....  
[WILD]

|                                     |                                                                                                                                                                                                                 |
|-------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-DIRECTORY</b>                   | Matches directories.                                                                                                                                                                                            |
| <b>-FILE</b>                        | Matches files.                                                                                                                                                                                                  |
| <b>-MODIFIED_AFTER <i>date</i></b>  | Same as <b>-AFTER <i>date</i></b> .                                                                                                                                                                             |
| <b>-MODIFIED_BEFORE <i>date</i></b> | Same as <b>-BEFORE <i>date</i></b> .                                                                                                                                                                            |
| <b>-RBF</b>                         | Matches ROAM files.                                                                                                                                                                                             |
| <b>-SAM</b>                         | Matches SAM files. (This option returns the names of all file system objects when used on a disk of a pre-Rev. 23.3 system. Use the [ATTRIB] function to perform selection.)                                    |
| <b>-SEGMENT_DIRECTORY</b>           | Matches segment directories.                                                                                                                                                                                    |
| <b>-TRUNCATED</b>                   | Matches objects that have been truncated by FIX_DISK. (This option returns the names of all file system objects when used on a disk of a pre-Rev. 23.3 system. Use the [ATTRIB] function to perform selection.) |

---

**Note**

Objects with undefined date/time values are matched by both members of each pair of date-selection options. Thus, an object with an undefined DTA value is matched by both **-ACCESSED\_AFTER** and **-ACCESSED\_BEFORE**. An object with an undefined DTC value is matched by both **-CREATED\_AFTER** and **-CREATED\_BEFORE**. Each pair in the **-BEFORE** and **-AFTER** families of options behaves in this way.

---

WILD options may be specified with any PRIMOS command that accepts wildcards, such as LD, COPY, DELETE, and SIZE.

**Example1:** If the directory MYDIR contains the files LETTER1 through LETTER5 and the directories SUB1 and SUB2, then

```
[WILD MYDIR>LETTER( SUB(]
```

returns

```
LETTER1 LETTER2 LETTER3 LETTER4 LETTER5 SUB1 SUB2
```

The entries are not necessarily returned in alphabetical order.

## [WRITE\_FILE]

WRITE\_FILE writes the contents of *text* into a previously opened ASCII file.

### Format

```
[WRITE_FILE unit text]
```

### Usage

WRITE\_FILE returns 0 if the operation is successful and returns a positive integer otherwise. *text* must be entered in single quotes if it contains special characters. One level of quoting is stripped prior to writing.

*unit* is the decimal integer identifying the file to be written. It may be either the number returned by the OPEN\_FILE function or a variable whose value was set to that number. (At command level, only global variables may be used. Both local and global variables are allowed within CPL programs.)

*text* is the information to be written. It is written as a single line in the file; that is, the newline is automatically added. If, for example, the file STUFF was opened on unit 37, then

```
[WRITE_FILE 37 'ADD THIS STUFF.']
```

writes the text to the file and returns a value of 0. If the variable .FILE\_NUM was previously set to the file unit number, then

```
[WRITE_FILE &.FILE_NUM 'ADD THIS STUFF.']
```

does the same thing.

If WRITE\_FILE is used to write to a file that already contains *text*, the old text is overwritten by the new text. To add new text to a file without writing over the old text, use READ\_FILE to move down to the end of the old text before using WRITE\_FILE.



*Replace this page with the tab page labeled*

**4 Command-line  
Features**

# Command-line Features

## Introduction

The PRIMOS command processor supports many command-line features that you can use to control and facilitate command processing. Among these features are the following:

- Global variables to supply information to PRIMOS commands and to user programs
- Multiple command entry to give several commands at once
- Iteration lists to repeat a command, substituting a new file system object each time
- Wildcard symbols to execute a command on a group of objects without listing their names individually
- Treewalking to search through a directory tree to execute a specified command on the appropriate file system objects
- Name generation to avoid repeating long entrynames by having PRIMOS substitute the full names for generation symbols
- Combinations of command-line features
- Syntax suppression to ignore special characters and command functions on the command line

You can also use the `ABBREV` command to create abbreviations for commands and command lines that you often use. For details on using `ABBREV`, see Chapter 2 of this book or the *PRIMOS User's Guide*.

The command processor recognizes the features listed above by looking for special characters on the command line. These special characters, in the order they are



### **Supplying Long Pathnames or Arguments**

Global variables, like abbreviations created with the `ABBREV` command, can be used to supply long pathnames in a shortened form. When used as an argument to a command, a global variable must be specified with a percent sign (%) on each side.

For example, if you store source programs in the subdirectory `BRANCH1` and their corresponding binary files in subdirectory `BRANCH2`, you could create a global variable for the complete pathname of each subdirectory:

```
OK, SET_VAR .SRC <FOREST>MAPLE>BRANCH1
OK, SET_VAR .BNY <FOREST>MAPLE>BRANCH2
```

To compile `GRADES.FTN` (a FORTRAN source file in `BRANCH1`) and store the resulting binary file in `BRANCH2`, you would give the command

```
OK, FTN % .SRC%>GRADES.FTN -B % .BNY%>GRADES.BIN
```

### **Supplying Variables to Command Functions**

Certain command functions, such as `OPEN_FILE` and `READ_FILE`, require either a local or a global variable as an argument. You can supply such an argument with a predefined global variable. Do not use the percent sign (%) with command functions if the argument requires a variable. For further information, see the *CPL User's Guide* or Chapter 3 of this guide.

## **Multiple Commands**

You can issue several commands on one command line if each command is separated by the command separator character, the semicolon (;). Each command is executed separately, as if it were on its own command line. For example, the following command line attaches you to the directory `MARKET` and then lists its contents:

```
ATTACH MARKET; LD
```

If one of the commands contains an error, `PRIMOS` still attempts to execute the remaining commands on the line. For instance, if `PRIMOS` is unable to attach to `MARKET` (in the previous example), it still lists the contents of whatever directory you are attached to.

The `ABBREV` command disables the effect of semicolons as command separators and interprets them literally, as part of the abbreviation rather than as a separator between `ABBREV` and another command. Therefore, if you give an `ABBREV` command followed by another command on the same line, the second command will

become part of the preceding abbreviation. This feature allows you to create an abbreviation that executes more than one command. For example, the command

```
ABBREV -ADD_COMMAND CA CLOSE -ALL; RLS -ALL
```

defines an abbreviation that, when executed, closes all your files and then releases the entire stack.

The effect of semicolons is also disabled if the command line begins with a tilde (~), the syntax suppression character.

---

**Note**

Do not use the COMINPUT command on a command line that contains multiple commands.

Some commands cannot be followed by other commands on the command line because the execution of the command resets the user's command environment, making execution of any following command impossible. These commands must appear as the last (or only) command on a command line: CHANGE\_PROJECT, ICE, LOGIN, LOGOUT, RELEASE\_LEVEL, and START.

---

## Iteration

There are occasions when you must repeat the same command several times, specifying a new file (or other file system object) each time. For example, to compile three FORTRAN files, you must give the following three commands:

```
FTN A -54V -XREF  
FTN B -64V -XREF  
FTN C -54V -XREF
```

You can do this more easily by giving one command that contains the three filenames enclosed in parentheses. Thus, the previous three commands can be given as one:

```
FTN (A B C) -64V -XREF
```

The list in parentheses is called an iteration list. Each member of the iteration list must be separated by a blank or by a comma. The previous example can also be given as

```
FTN (A,B,C) -64V -XREF
```

PRIMOS executes the command once for each member of the list, as if you had typed each command separately.

---

**Note**

To separate items within an iteration list, use spaces, commas, or a combination of a comma followed by a space. You cannot use the combination of a space followed by a comma.

---

**Multiple Iteration Lists**

The previous example replaced one argument with one iteration list. You can, however, replace as many arguments as you like with iteration lists. You must separate each list with a blank or a comma. For example, the command

```
COPY (A B C) (D E F)
```

copies files A, B, and C, renaming them D, E, and F, respectively. The iteration list (A B C) is the first argument and the list (D E F) is the second.

When an iteration list runs out of arguments, a null string is substituted. For example, the command

```
CMPF (L M N) (O P) (Q R S)
```

produces the following three commands:

```
CMPF L O Q
CMPF M P R
CMPF N S
```

**Iteration Lists as Parts of Arguments**

The previous examples used iteration lists as complete arguments. An iteration list, however, can be used as part of a single argument if the list is not separated by a blank or a comma from the rest of the argument. For example, the following command uses an iteration list as part of each of the two arguments:

```
COPY ALLEN>(A B C) JUNE>(D E F)
```

When executed, the command copies the files A, B, and C from the directory ALLEN into the directory JUNE, naming them D, E, and F, respectively.

No more than two iteration lists can be used in a single argument.

### **Cross-product Iteration Lists**

When you use two (but no more than two) iteration lists within a single argument, you create a cross-product iteration list. In such a list, each member of the first iteration list is paired with each member of the second iteration list, and the command is executed once for each object that results. For example, the command

```
DELETE (YOUR,MY,HIS) .(MEMO,DRAFT)
```

uses one argument consisting of two iteration lists and produces a cross-product iteration list that deletes the following files: YOUR.MEMO, YOUR.DRAFT, MY.MEMO, MY.DRAFT, HIS.MEMO, and HIS.DRAFT.

Similarly, the command

```
DELETE (A B C) (D E F)
```

deletes the following nine files: AD, AE, AF, BD, BE, BF, CD, CE, and CF.

## **Wildcarding**

Wildcarding allows you to specify groups of file system objects on which a command can act. For example, the following command compiles all files in the current directory whose names have .FTN as a suffix:

```
FTN @@.FTN
```

The following command lists all file system objects in the directory BEECH whose names begin with the letter A:

```
LD BEECH>A@@
```

Wildcarding is specified by using a wildcard name as an argument to a command. A **wildcard name** is a pathname in which the final element (or the only element, if the name is an entryname) contains one or more of the wildcard characters shown in Table 4-1.

*Table 4-1. Wildcard Characters*

| <i>Character</i> | <i>Function</i>                                                                                                                                                                   |
|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| @@               | Replaces any number of characters in any number of components within a file or directory name.                                                                                    |
| @                | Replaces any number of characters within one component of a filename or directory name. Stops matching at the period (.) that separates a name and its suffix.                    |
| +                | Replaces a single character, except a period(.).                                                                                                                                  |
| ^                | Inverted match character. Must be the first character in the wildcard name. A wildcard name that begins with ^ matches all names that do not match the rest of the wildcard name. |

Examples of each of these characters are given in subsequent sections.

In most cases, only one argument per command can contain wildcard names. That argument can contain either a single wildcard name (as in the examples above) or a single iteration list containing any number of wildcard names, as in the following examples:

```
DELETE (A@@ B@@)
```

```
DELETE @(.BIN .LIST)
```

### **Wildcard Matching**

When a command containing a wildcard name is given, the command processor searches the specified directory for all entrynames that match the given wildcard name. An entryname matches a wildcard name when it contains both of the following:

- The same number of components as the wildcard name
- The same literal characters as the wildcard name contains, in the same relative positions

The three wildcard symbols +, @, and @@ differ only in the number of characters they can match.

- + matches only one character (except a period).
- @ matches any number of characters in a single component of a name, stopping at the period (.) that separates components.
- @@ matches any number of characters in any number of components.



A wildcard name consisting only of the symbol @ matches all single-component names in the directory, whereas a wildcard name consisting only of the symbol @@ matches all names in the directory, regardless of how many components they contain.

The selection order for wildcard matching cannot be predicted.

**Examples of Wildcard Names**

Suppose that the current directory contains the following file system objects:

|                 |             |                  |             |
|-----------------|-------------|------------------|-------------|
| BARR1 . COBOL   | BARR1 . SEG | BARR2 . COBOL    | BARR2 . SEG |
| CLR . CPL       | EDD . COMO  | EDD . COMO . OLD | EDD . CPL   |
| EDD . CPL . OLD | FILL        | SCROLL           | SKILL       |

The wildcard name @ matches all single-component names:

|      |        |       |
|------|--------|-------|
| FILL | SCROLL | SKILL |
|------|--------|-------|

The wildcard name S@ matches all single-component names that begin with S:

|        |       |
|--------|-------|
| SCROLL | SKILL |
|--------|-------|

The wildcard name @@ matches all two-component names:

|               |             |               |             |
|---------------|-------------|---------------|-------------|
| BARR1 . COBOL | BARR1 . SEG | BARR2 . COBOL | BARR2 . SEG |
| CLR . CPL     | EDD . COMO  | EDD . CPL     |             |

The wildcard name @.SEG matches all two-component names that end in .SEG:

|             |             |
|-------------|-------------|
| BARR1 . SEG | BARR2 . SEG |
|-------------|-------------|

The wildcard name @.@ matches all three-component names:

|                  |                 |
|------------------|-----------------|
| EDD . COMO . OLD | EDD . CPL . OLD |
|------------------|-----------------|

The wildcard name BARR+.COBOL matches

|               |               |
|---------------|---------------|
| BARR1 . COBOL | BARR2 . COBOL |
|---------------|---------------|

The wildcard name BARR+.@ matches

|               |             |               |             |
|---------------|-------------|---------------|-------------|
| BARR1 . COBOL | BARR1 . SEG | BARR2 . COBOL | BARR2 . SEG |
|---------------|-------------|---------------|-------------|

The wildcard name `@@L` matches all names, of any number of components, that end in L:

|             |             |         |      |
|-------------|-------------|---------|------|
| BARR1.COBOL | BARR2.COBOL | CLR.CPL | FILL |
| EDD.CPL     | SCROLL      | SKILL   |      |

The wildcard name `@@` matches every name in the directory.

### ***Inverted Matching***

The caret (^) functions as an inverted match character. Any wildcard name beginning with a caret matches every object whose name does not match the rest of the wildcard name. Using the directory above, the wildcard name `^@@L` matches all names that do not end in L:

|             |           |          |              |
|-------------|-----------|----------|--------------|
| BARR1.SEG   | BARR2.SEG | EDD.COMO | EDD.COMO.OLD |
| EDD.CPL.OLD |           |          |              |

### ***Wildcard Options***

You can modify the effect of wildcard names by specifying one or more of the wildcard options described in Table 4-2. **Wildcard options** can be given on the command line anywhere after the command.

Wildcard options are used to

- Select only file system objects of a particular type or types
- Select only file system objects that were last modified before or after a particular date
- Enable or disable verification

Table 4-2. Wildcard Options

| Option                                          | Matches                                                                                  |
|-------------------------------------------------|------------------------------------------------------------------------------------------|
| <code>--ACCESSED_AFTER <i>date.time</i></code>  | Objects in Rev. 20.0 or later directories last accessed on or after <i>date.time</i> .   |
| <code>--ACCESSED_BEFORE <i>date.time</i></code> | Objects in Rev. 20.0 or later directories last accessed before <i>date.time</i> .        |
| <code>--ACCESS_CATEGORY</code>                  | Access categories only.                                                                  |
| <code>--AFTER <i>date.time</i></code>           | Objects last modified on or after <i>date.time</i> .                                     |
| <code>--BACKEDUP_AFTER <i>date.time</i></code>  | Objects backed up with the BACKUP utility on or after <i>date.time</i> .                 |
| <code>--BACKEDUP_BEFORE <i>date.time</i></code> | Objects backed up with the BACKUP utility before <i>date.time</i> .                      |
| <code>--BEFORE <i>date.time</i></code>          | Objects last modified before <i>date.time</i> .                                          |
| <code>--CREATED_AFTER <i>date.time</i></code>   | Objects in Rev. 20.0 or later directories created on or after <i>date.time</i> .         |
| <code>--CREATED_BEFORE <i>date.time</i></code>  | Objects in Rev. 20.0 or later directories created before <i>date.time</i> .              |
| <code>--DIRECTORY</code>                        | Directories only.                                                                        |
| <code>--FILE</code>                             | Files only.                                                                              |
| <code>--MODIFIED_AFTER <i>date.time</i></code>  | Same as <code>--AFTER <i>date.time</i></code> .                                          |
| <code>--MODIFIED_BEFORE <i>date.time</i></code> | Same as <code>--BEFORE <i>date.time</i></code> .                                         |
| <code>--NO_VERIFY</code>                        | Suppresses verification requests, even for a command that usually requires it. (Default) |
| <code>--RBF</code>                              | ROAM files only.                                                                         |
| <code>--SEGMENT_DIRECTORY</code>                | Segment directories only.                                                                |
| <code>--VERIFY</code>                           | Causes PRIMOS to request verification before executing the command on any object.        |

The options are discussed in the following three sections. The formats for specifying *date.time* are discussed in the second section.

## Type-designation Options

The five options that select particular types of file system objects are the following:

```
-ACCESS_CATEGORY
-DIRECTORY
-FILE
-RBF
-SEGMENT_DIRECTORY
```

When you specify one or more of these options in a command line containing a wildcard name, the command executes only on objects of that type. For example, the following command lists all files beginning with A in the current directory:

```
LD A@@ -FILE
```

The following command lists all files and access categories beginning with A in the current directory:

```
LD A@@ -FILE -ACCESS_CATEGORY
```

If you specify none of the type-specification options, all file object types (files, directories, segment directories, and access categories) are matched.

## Date-selection Options

The following options select an object according to the date on which it was created, last modified, last accessed, or backed up:

```
-ACCESSED_AFTER date.time
-ACCESSED_BEFORE date.time
-AFTER date.time
-BEFORE date.time
-CREATED_AFTER date.time
-CREATED_BEFORE date.time
-MODIFIED_AFTER date.time
-MODIFIED_BEFORE date.time
-BACKEDUP_AFTER date.time
-BACKEDUP_BEFORE date.time
```

When you specify these options, the command processor matches only objects that were created, last modified, last accessed, or backed up before, on, or after a given date and time.

The four full formats with which you specify *date.time* are

```
mo/dd/yy.hh:mm:ss
yy-mo-dd.hh:mm:ss
'dd mon yy.hh:mm:ss'
'dd mon yy hh:mm:ss'
```







## Treewalking

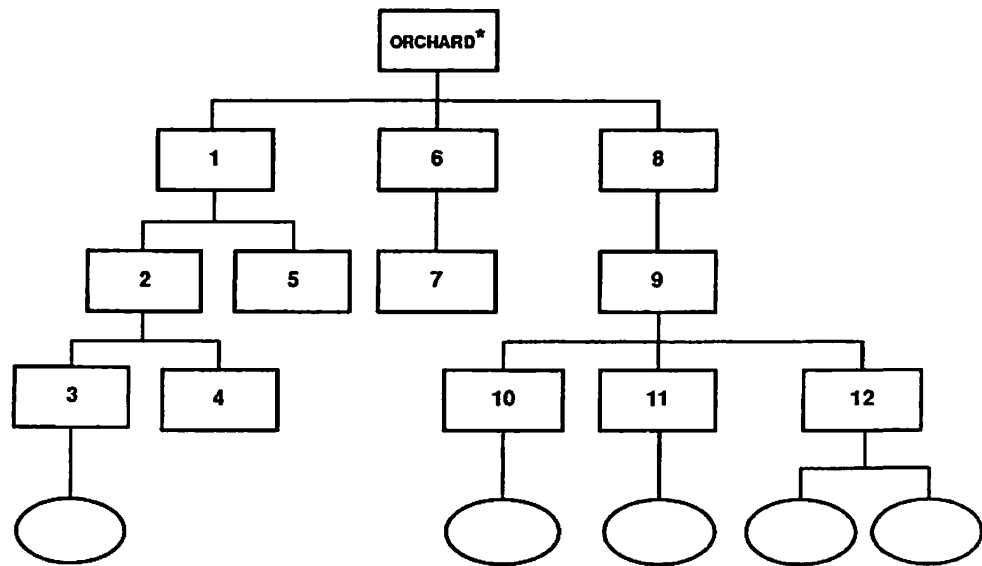
Wildcarding allows PRIMOS commands to act on a group of file system objects located within a single directory. Treewalking takes this convenience one step further, and allows a command to act on designated objects within a directory tree (that is, a directory, its subdirectories, their subdirectories, and so forth).

You specify a treewalking pattern by using wildcard characters in an intermediate position within a pathname and/or in the final position of the pathname. The wildcard characters cannot be in the first position of the pathname.

When you give a command that contains a treewalk name, the command processor searches all directories subordinate to the specified starting directory for file system objects that match the given treewalk pathname. (The starting directory itself is not searched unless you specify the `-WALK_FROM 1` option, explained below.)

Figure 4-1 illustrates how PRIMOS proceeds vertically through directories in a standard treewalk. The horizontal order of visitation within a single directory cannot be predicted. Hence, the horizontal order of the top three subdirectories (numbers 1, 6, and 8) in Figure 4-1 might differ in a subsequent treewalk. The command for Figure 4-1 is

```
LD ORCHARD>@@>@@
```



\* = attach point

104.01.D3108.8LA

Figure 4-1. Order of Visiting Directories in Sample Standard Treewalk



### Examples of Treewalking

Figure 4-2, representing a sample directory tree stemming from the directory ORCHARD, is used for the following examples.

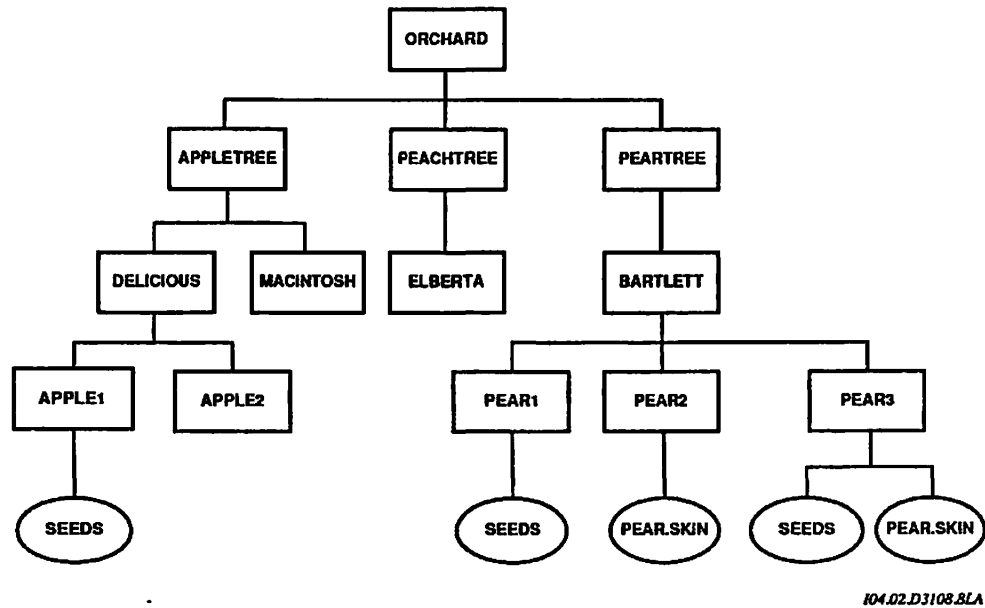


Figure 4-2. Sample Directory Tree

If you are attached to directory ORCHARD, the command

```
LD @@ -DIR
```

produces the following list of all the directories contained in ORCHARD:

```
OK, LD @@ -DIR
```

```
<FOREST>ORCHARD (ALL access)
```

```
1 record in this directory, 18 total records out of quota of 0.
```

```
3 Directories.
```

```
APPLETREE          PEACHTREE          PEARTREE
```

(The command could also be given as LD -DIR because @@ is assumed when no pathname follows the LD command.)

To list all subdirectories subordinate to ORCHARD, issue the command

```
LD ORCHARD:GC:GC -DIR
```

This displays information on all subdirectories from APPLETREE to PEAR3, as follows:

OK, LD ORCHARD>@@>@@ -DIR

<FOREST>ORCHARD>APPLETREE (ALL access)  
1 record in this directory, 6 total records out of quota of 0.  
2 Directories.

DELICIOUS           MACINTOSH

<FOREST>ORCHARD>APPLETREE>DELICIOUS (ALL access)  
1 record in this directory, 4 total records out of quota of 0.  
2 Directories.

APPLE1              APPLE2

<FOREST>ORCHARD>APPLETREE>DELICIOUS>APPLE1 (ALL access)  
1 record in this directory, 1 total record out of quota of 0.  
No entries selected.

<FOREST>ORCHARD>APPLETREE>DELICIOUS>APPLE2 (ALL access)  
1 record in this directory, 1 total record out of quota of 0.  
No entries selected.

<FOREST>ORCHARD>APPLETREE>MACINTOSH (ALL access)  
1 record in this directory, 1 total record out of quota of 0.  
No entries selected.

<FOREST>ORCHARD>PEACHTREE (ALL access)  
1 record in this directory, 2 total records out of quota of 0.  
1 Directory.

ELBERTA

<FOREST>ORCHARD>PEACHTREE>ELBERTA (ALL access)  
1 record in this directory, 1 total record out of quota of 0.  
No entries selected.

<FOREST>ORCHARD>PEARTREE (ALL access)  
1 record in this directory, 5 total records out of quota of 0.  
1 Directory.

BARTLETT

<FOREST>ORCHARD>PEARTREE>BARTLETT (ALL access)  
1 record in this directory, 4 total records out of quota of 0.

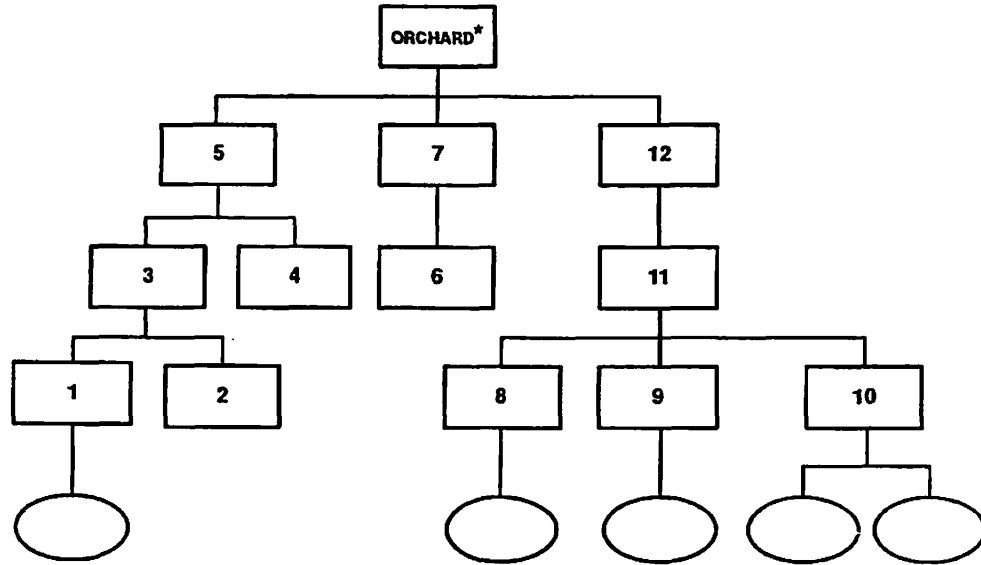


| <i>Option</i>             | <i>Function</i>                                                                                                                                                                                                                                                                           |
|---------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>-WALK_FROM n</code> | Executes the command in directories at levels greater than or equal to <i>n</i> . The default is <code>-WALK_FROM 2</code> , which executes the command at the first directory under the starting directory. For execution in the starting directory, specify <code>-WALK_FROM 1</code> . |
| <code>-WALK_TO n</code>   | Executes the command in directories at levels less than or equal to <i>n</i> .                                                                                                                                                                                                            |
| <code>-BOTTOM_UP</code>   | Executes the command in specified directories starting at the deepest level (that is, starting at the largest level number and going to the smallest level number). The default is to start at the highest level (that is, the smallest level number) and work down.                      |

### Examples of Treewalking Using Options

Figure 4-3 illustrates the order of visiting directories in a bottom-up treewalk in the sample tree ORCHARD. The command is

```
LD ORCHARD>@@>@@ -BOTUP
```



\* = attach point

104.03.D3108.BLA

Figure 4-3. Order of Visiting Directories in Sample Bottom-up Treewalk



## **Requirements for Name Generation**

Generated names are created from the following sources:

- One **source pathname**, from which to create new names. The number of components in the generated name is usually less than or equal to the number of components in the source pathname.
- **Generation patterns**, contained in the entryname portions (that is, the final positions) of one or more subsequent pathnames.

### **Source Pathname**

Except for RESUME and SEG commands, the source pathname is the first pathname in the command line. That is, it forms the first argument to the command. For example, in the command line

```
CMPE DIR>NAMES.OLD DIR>NAMES.NEW
```

the source pathname is DIR>NAMES.OLD.

For RESUME and SEG commands, the source pathname is the second argument in the command line. For example, in the command line

```
RESUME MYPROGRAM DIR>ARGUMENT1 DIR>ARGUMENT2
```

the source pathname is DIR>ARGUMENT1.

### **Generation Patterns**

Name generation patterns are composed of two items:

- Name generation symbols, usually the equal (=) and double-equal (==) signs. The equal sign (=) copies a single component of the source name; the double-equal sign (==) copies as many components as can be copied without adding components to the name. Only one double-equal sign can appear in a name generation pattern.
- Literal strings of characters, each of which replaces a component in the source name.

A summary of name generation symbols and their effects is shown in Table 4-3.

Table 4-3. Name Generation Symbols and Their Effects

| <i>Character</i>       | <i>Function</i>                                                                                                                                                                                   |
|------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| =                      | Copies a single component from the source name to the generated name. (Parallels the wild character @ for copying.)                                                                               |
| ==                     | Copies one or more components from the source name to the generated name. (Parallels the wild character @@ for copying, except for such substitutions, additions, or deletions as are specified.) |
| ^=                     | Skips over a single component from the source name without copying it to the generated name.                                                                                                      |
| ^==                    | Skips over one or more components of the source name without copying them to the generated name.                                                                                                  |
| <i>literal-string</i>  | Replaces a component from the source name with the component given by <i>literal-string</i> .                                                                                                     |
| <i>+literal-string</i> | Adds to the source name the component shown after the plus sign (+).                                                                                                                              |

### Examples of Name Generation

| <i>Source Name</i> | <i>Generated Pattern</i> | <i>Generated Name</i> | <i>Description</i>                                                                                             |
|--------------------|--------------------------|-----------------------|----------------------------------------------------------------------------------------------------------------|
| A.B.C.D            | = .X                     | A.B.C.X               | Three components are copied to create a four-component name.                                                   |
| A.B.C.D            | = .X.Y                   | A.B.X.Y               | Only the first two components are copied.                                                                      |
| A.B.C.D            | X.= .X                   | X.B.C.X               | Only the middle two components are copied.                                                                     |
| A.B.C.D            | X.=                      | X.B                   | A single equal sign copies only one component.                                                                 |
| A.B.C.D            | = .X.Y.Z.=               |                       | An error message results because the pattern specifies five components and the source name contains only four. |

#### Note

A name generation pattern can be used only in the entryname portion (that is, the final position) of the pathname. The following command is therefore illegal:

```
COPY A>LONGNAME>B X>=>Y
```

### ***Adding Components***

To add a component to a generated name, precede the new characters or name generation symbols with a plus sign (+), as in the following examples:

| <i>Source Name</i> | <i>Generated Pattern</i> | <i>Generated Name</i> |
|--------------------|--------------------------|-----------------------|
| A.B                | = = .+C                  | A.B.C                 |
| A.B                | = .+C.=                  | A.C.B                 |
| A.B                | +C.= =                   | C.A.B                 |

The exception to this rule occurs when you are adding a component to the end of a name and have specified each preceding component explicitly. In this case, you specify the new component literally without using the plus sign, as in the following example:

| <i>Source Name</i> | <i>Generated Pattern</i> | <i>Generated Name</i> |
|--------------------|--------------------------|-----------------------|
| A.B                | = . = .C                 | A.B.C                 |

### ***Deleting Components***

To delete one or more components, precede the equal or double-equal sign with a caret (^), which acts as a negation symbol. The examples below show the creation of generated names with deleted components:

| <i>Source Name</i> | <i>Generated Pattern</i> | <i>Generated Name</i> |
|--------------------|--------------------------|-----------------------|
| A.B.C              | = . ^ = =                | A                     |
| A.B.C              | ^ = . = =                | B.C                   |
| A.B.C              | = . ^ = . =              | A.C                   |
| A.B.C              | ^ = = . =                | C                     |

### ***Combining Additions, Substitutions, and Deletions***

You can specify additions, substitutions, and deletions in a single name generation pattern, as in the following example:

| <i>Source Name</i> | <i>Generated Pattern</i> | <i>Generated Name</i> |
|--------------------|--------------------------|-----------------------|
| A.B.C.D.E.F        | = .X.+Y.= .^=. =         | A.X.Y.C.E             |

However, only one double-equal sign, with or without a caret (negation) sign, can appear in the pattern.



## Combining Command-line Features

Some command-line features can be used in combination with others. The use of wild characters in treewalking is such an example. The next two sections discuss other combinations.

### *Combining Iteration With Other Features*

Wildcards, treewalk patterns, and name generation patterns, as well as abbreviations, variables, and function calls, can all be used within iteration lists. For example, the following are all legal commands:

```
DELETE (A& B& C&);
COPY (G LIST (BIN, ARCHIV>|= OLDLIST =.OLDBIN)
SPOOL (Y YESTERDAY& %.TODAY& A.&)
```

### *Wild Characters and Name Generation*

If a source pathname includes a wildcard name, the generated names match whatever names are produced by the wildcarding process. For example, the command

```
COPY ALPHA>*& LIST ARCHIV>== OLDLIST
```

copies all listing files in the directory ALPHA into the directory ARCHIV and changes the suffix on each file from LIST to OLDLIST. The following three commands are among those that might be generated:

```
COPY ALPHA>A.LIST ARCHIV>A.OLDLIST
COPY ALPHA>B.LIST ARCHIV>B.OLDLIST
COPY ALPHA>C.LIST ARCHIV>C.OLDLIST
```

## Syntax Suppression

The tilde (~) is the PRIMOS syntax suppression character. When you begin a command line with a tilde, you force PRIMOS to interpret the rest of the line literally and ignore special characters, such as the wild, iteration, and name generation characters. Because these special characters initiate the command-line features described in this chapter, the features are therefore suppressed. Command functions (which use brackets as special characters) are also suppressed.

For example, the following command line executes both the TYPE and LD commands, thus printing the answer 200 and then listing the contents of your current directory:

```
TYPE [CALC 10 * 20];LD
```

If, however, you use a tilde as the first character of the command line, PRIMOS suppresses the special characters (that is, brackets and semicolon) and prints everything literally after TYPE, as follows:

```
OK, ~ TYPE [CALC 10 * 20];LD
[CALC 10 * 20];LD
OK,
```

Syntax suppression is particularly useful when you are creating an abbreviation that includes a command function or command-line feature. For example, in the following command, the DATE function is executed and the abbreviation LOG would stand for COMO LOG.MAY (if the current month were May):

```
ABBREV -&C LOG COMO LOG [DATE -MONTH]
```

If, however, you want the DATE function inserted into the abbreviation without being evaluated, then you give the command

```
~ ABBREV -&C LOG COMO LOG.[DATE -MONTH]
```

The abbreviation LOG now stands for COMO LOG.[DATE -MONTH]. Thus, each time you issue the command LOG, the DATE function is evaluated and a command output file is created with a name beginning with LOG and having the current month as its suffix.

*Replace this page with the tab page labeled*

**5 Command-line  
Processing**

# Command-line Processing

## Introduction

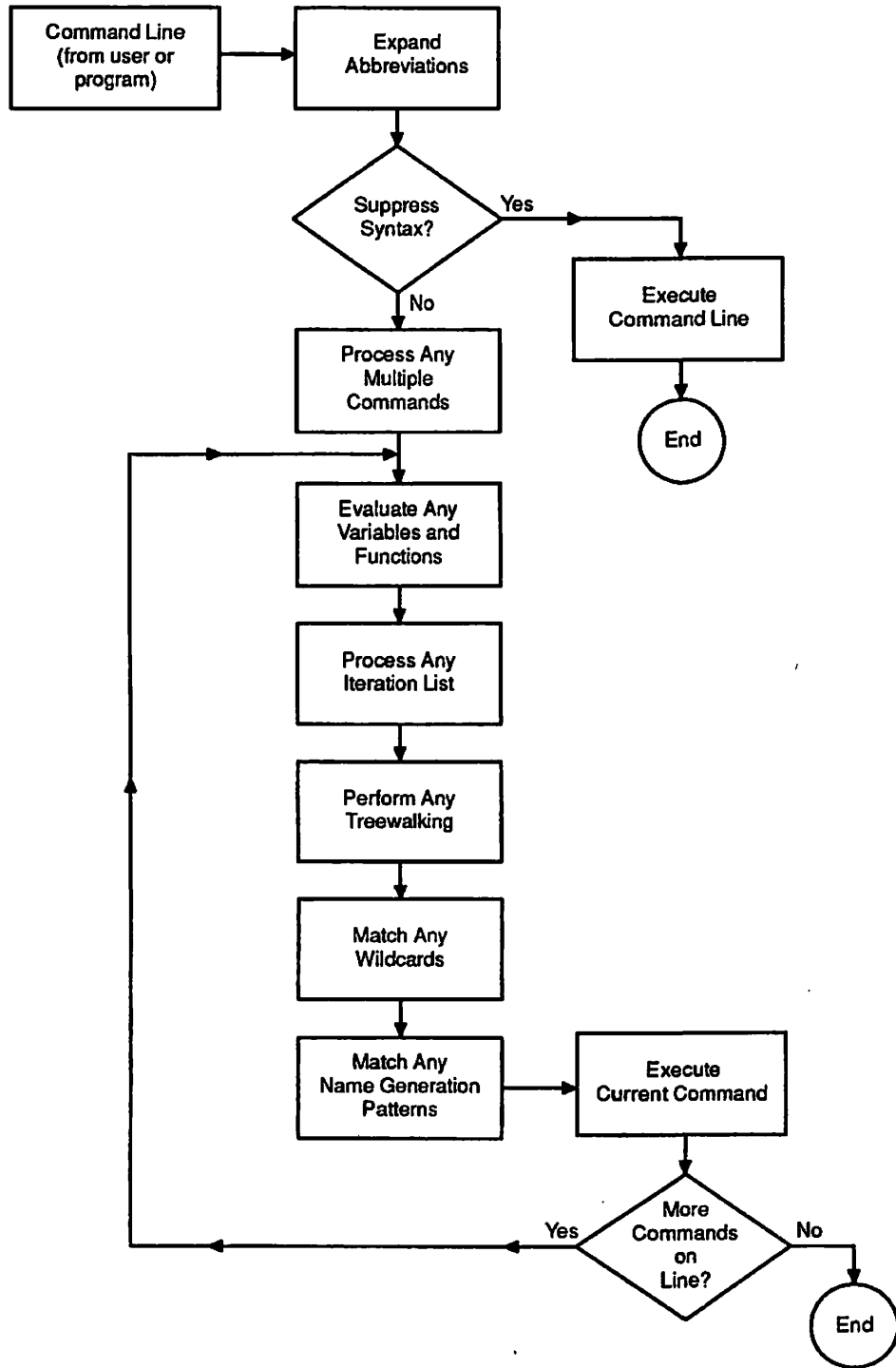
If you frequently use the many features of the PRIMOS command processor, you may find it useful to know the order in which the features are processed and the interactions between them.

## Command-line Processing Order

Command-line processing proceeds in the following order:

1. Expanding abbreviations
2. Suppressing syntax
3. Processing multiple commands
4. Evaluating variables and functions
5. Identifying iteration lists
6. Implementing treewalking
7. Matching wildcard names
8. Matching name generation patterns
9. Executing the command

Figure 5-1 illustrates this processing order. Each step is discussed in the following sections.



105.01.D3108.BLA

Figure 5-1. Process Flow for Execution of a Command Line

## ***Expanding Abbreviations***

If the System Administrator has not disabled the abbreviation processor and if your abbreviations file is active, the first thing that happens in the command line is that abbreviations are expanded.

A command function or simple iteration list on the command line is treated as a single token by the abbreviation processor. Functions or iteration lists that require abbreviation parameters or that are used as arguments for an abbreviation parameter are treated as single tokens by the abbreviation processor. For example, if A is the abbreviation for `*>SUBDIR>%1%`, then

A (B C D)

expands to

`*>SUBDIR>` (B C D)

because the list (B C D) is treated as a single token and is assigned as the value of parameter 1 of abbreviation A.

Similarly, if B is the abbreviation for `*>BDIR>%1%` and C is the abbreviation for `*>CDIR>%1%` then

(A B C) D E F

expands to

`(*>SUBDIR>D *>BDIR>D *>CDIR>D)` E F

because the list (A B C) is treated as a single token having one parameter, which is assigned the value D.

Function calls are also treated as single tokens, but they are expanded as if they were a separate command line. Thus, the first token following a left bracket is considered to be in the command position, and no abbreviation parameters are taken from beyond the matching right bracket.

For example, if A is the abbreviation for `FOO %2%.TWO %1%.ONE`, then

[A B] C D

expands to

[FOO .TWO B.ONE] C D

Note that C does not become the second parameter of A because C lies outside the function call brackets.

## Suppressing Syntax

Following abbreviation expansion, the command processor checks to see if the first character on the command line is the syntax suppressor, the tilde (~). If so, processing of all subsequent command line features is suppressed. The command line processor removes the tilde and executes the remaining command line without further interpretation. See Syntax Suppression in Chapter 4 of this guide for examples of using syntax suppression.

## Processing Multiple Commands

The command processor next scans the command line for the command separator character, the semicolon (;). This character delimits multiple commands on the same command line. The command processor does not interpret the command separator character if the command line begins with the syntax suppressor character (~), or if a command separator follows the ABBREV command on the command line. In both cases, the semicolon is treated as a literal character. The latter exception is provided to allow definitions of abbreviations whose value contains the command separator character. For example,

```
ABBREV -AC ZOT CLOSE ALL; DELETE @@ -NO_VERIFY
```

is a single command that defines an abbreviation, ZOT, whose value is

```
CLOSE ALL; DELETE @@ -NO_VERIFY
```

If recognition of the command separator is not disabled, the features described hereafter are executed separately for each command on the command line. For example, the order of execution in the command line

*command1* [*function1 arguments*]; *command2* [*function2 arguments*]

is evaluate *function1*, execute *command1*, evaluate *function2*, and execute *command2*.

---

### Note

Because of its operation, the COMINPUT command must not be used in command lines containing multiple commands.

Some commands cannot be followed by other commands on the command line because the execution of the command resets the user's command environment, making execution of any following command impossible. These commands must appear as the last (or only) command on a command line: CHANGE\_PROJECT, ICE, LOGIN, LOGOUT, RELEASE\_LEVEL, and START.

---

## ***Evaluating Variables and Functions***

Once the current command has been identified, variable references are evaluated. Each reference of the form *%variable\_name%* is replaced by the value of *variable\_name*.

Text, command function references of the form *[function arguments]* are evaluated and replaced by their values. Evaluation proceeds from the inside out. For example, the command line

*command [function arguments]*

is replaced by

*command value*

where *value* is the value returned by *function*. (For details on functions, see the *CPL User's Guide* or Chapter 3 of this guide.)

The fact that variables and functions are evaluated after the command separator has been processed means that semicolons are not recognized as command separators when they form part of the value of a variable or function.

The fact that functions are evaluated after variables means that, if the value of a variable contains a function reference, the function will be evaluated. Conversely, if the value of a function contains a variable reference, the variable reference will not be evaluated because variable evaluation has already occurred.

If any error occurs during variable or function evaluation, such as a reference to an undefined variable or function, the command processor prints an error message and does not process that command.

## ***Identifying Iteration Lists***

All iteration sets (lists in parentheses) in the command line are identified. Conceptually, the command processor can be thought of as producing a series of command lines, one for each iteration specified by the simple iteration sets. For example,

*command (A B C) Y (D E F)*

can be thought of as the series of commands

*command A Y D*  
*command B Y E*  
*command C Y F*

In fact, the command processor does not generate the command strings at this time. Rather, it implements a kind of list structure at this level.



## ***Implementing Treewalking***

Next, the command line is examined for a **treewalk pathname**, that is, a pathname whose directory part contains a wildcard. A **directory part** is the part before the final name. Each command can contain only one treewalk pathname. (More than one treewalk pathname may appear on the command line, provided each iteration yields no more than one.)

The command processor opens the directory whose pathname appears before the directory wildcard in the treewalk pathname. It visits the subdirectories of the tree and substitutes the pathname of each directory visited for the part of the treewalk pathname to the left of and including the directory wildcard. (See Chapter 4 of this book for details on treewalking.)

For example,

```
command A>B>@>@ .LIST
```

could execute the following series of commands:

```
command A>B>C>@ .LIST  
command A>B>C>X>@ .LIST  
command A>B>C>Y>@ .LIST  
command A>B>D>@ .LIST
```

Commands of this series are passed on to the wildcarding step, below.

## ***Matching Wildcard Names***

Each command is scanned for a pathname whose last element (entryname) is a wildcard. Only one such pathname per command can be received at this step.

At the same time, the command is scanned for wildcard options (such as `-AFTER`) that specify selection criteria in addition to the wildcard name. These wildcard options are used only at this step in processing. Therefore, they are removed from the commands that are actually executed.

The command processor opens the directory given by the treename, and selects those entries in the directory that match the wildcard and the other selection criteria. If verify mode is enabled, the command processor asks the user to approve or disapprove each match.

The wildcard part of the pathname is then replaced with the actual name that is matched, and the command is passed to the next step.

For example,

```
command @.LIST -AFTER 12-1 -FILE
```

might execute as if it were the series

```
command A.LIST
command B.LIST
command C.LIST
.
.
.
```

### ***Matching Name Generation Patterns***

The final step is name generation. The command processor searches for any pathname in the command that contains an equal sign (=) in the entryname. Any number of such pathnames is permitted. Each generation pattern (that is, each name containing an equal sign) is replaced by the name it generates. The **source name** is usually the first object argument to the command, but individual commands may differ.

For example,

```
command ABC.LIST ==.+OLD
```

executes as

```
command ABC.LIST ABC.LIST.OLD
```

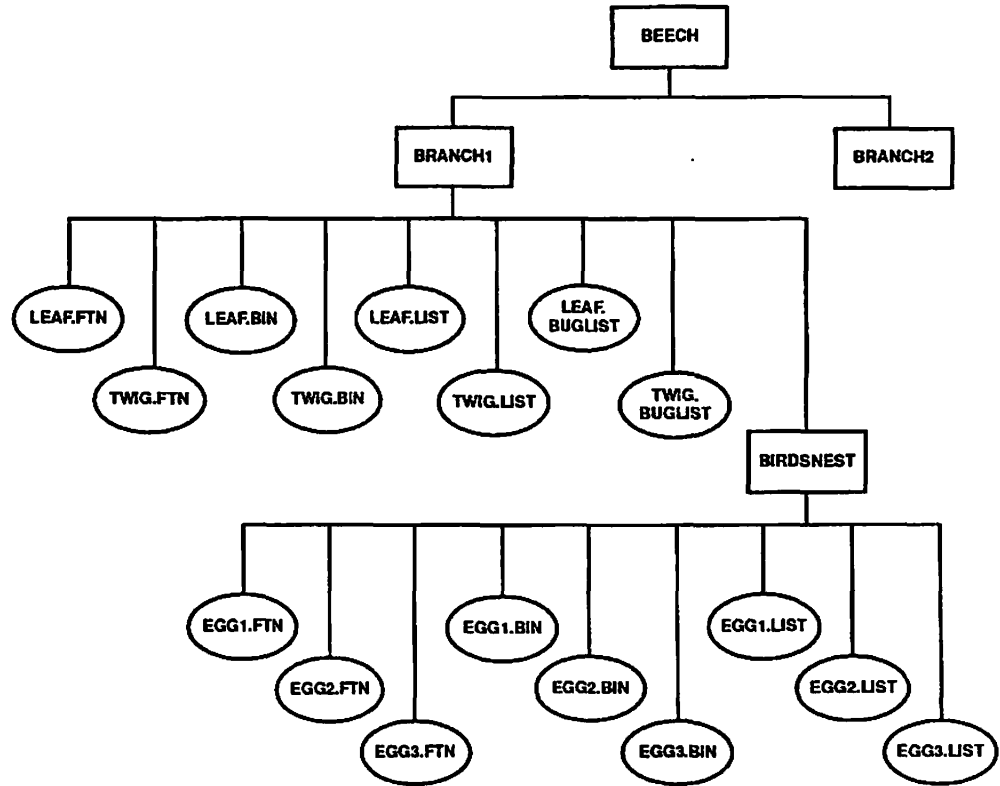
### ***Executing the Command***

The command that emerges from the name generation step is then executed. After execution (whether or not the command produced an error message), the next wildcard match, then the next treewalk step, and finally the next simple iteration step, is taken.

## **Example of Command-line Processing**

The example below follows a command line, step-by-step, through processing to illustrate the order in which command-line processing occurs. The command listed after "CURRENT:" is the command currently being processed. The commands listed at "WAITING:" show, in order, subsequent commands to be executed. (The stack of waiting command lines shown in this example would not be created by PRIMOS. The commands are shown in this form for easier reading.)

The directory used in this example is shown in Figure 5-2.



105.02.D3108.BLA

Figure 5-2. Sample Directory Tree

### Processing Steps

1. The user types the following command line:

```
CHECK % .PROJ1%
```

2. Abbreviation Expansion: The command processor expands the user's abbreviation CHECK, producing the command line

```
SPOOL % .PROJ1%> (@.LIST @.BUGLIST) ;DELETE % .PROJ1%> (@.BIN  
@.LIST)
```

3. Syntax Suppression: The command line does not begin with a tilde (~). No syntax suppression occurs and processing continues.
4. Multiple Commands: The presence of a semicolon breaks the command line into two commands. The processing status is

```
CURRENT:      SPOOL %.PROJ1%>(@.LIST @.BUGLIST)
WAITING:      DELETE %.PROJ1%>(@.BIN @.LIST)
```

5. Variable Evaluation: The global variable `%.PROJ1%` is removed from the current command and replaced with its value `BEECH>@@` from the global variable file previously activated. The processing status is

```
CURRENT:      SPOOL BEECH>@@>(@.LIST @.BUGLIST)
WAITING:      DELETE %.PROJ1%>(@.BIN @.LIST)
```

6. Command Function Evaluation: The command line contains no square brackets (`[ ]`), which denotes the absence of function calls. No change to the command occurs.

7. Iteration Evaluation: Next, the iteration lists are evaluated. The presence of an iteration list containing two items (`@.LIST` and `@.BUGLIST`); creates two command lines to replace the current command line. The first of the new command lines becomes the current line. The second is placed at the top of the waiting list. The processing status is

```
CURRENT:      SPOOL BEECH>@@>@.LIST
WAITING:      SPOOL BEECH>@@>@.BUGLIST
              DELETE %.PROJ1%>(@.BIN @.LIST)
```

8. Treewalking: Next, treewalking evaluation matches `@@` against all directories in `BEECH`'s subtree. Each match found creates a new command line. The first becomes current while the rest are placed at the top of the waiting list. The processing status is

```
CURRENT:      SPOOL BEECH>BRANCH1>@.LIST
WAITING:      SPOOL BEECH>BRANCH1>BIRDSNEST>@.LIST
              SPOOL BEECH>BRANCH2>@.LIST
              SPOOL BEECH>@@>@.BUGLIST
              DELETE %.PROJ1%>(@.BIN @.LIST)
```

9. Wildcard Matching: Matching of wildcards is done for the current command. The first match found replaces the wildcard in the current command line. Subsequent matches generate new commands for the waiting list. The processing status is

```
CURRENT:      SPOOL BEECH>BRANCH1>LEAF.LIST
WAITING:      SPOOL BEECH>BRANCH1>TWIG.LIST
              SPOOL BEECH>BRANCH1>BIRDSNEST>@.LIST
              SPOOL BEECH>BRANCH2>@.LIST
              SPOOL BEECH>@@>@.BUGLIST
              DELETE %.PROJ1%>(@.BIN @.LIST)
```

10. Name Generation: The current command is checked for name generation patterns. There are none, so no changes occur at this step.

11. Execution: The current command

SPOOL BEECH>BRANCH1>LEAF.LIST

is executed. The first command on the waiting list now becomes the current command. The processing status is

CURRENT: SPOOL BEECH>BRANCH1>TWIG.LIST  
 WAITING: SPOOL BEECH>BRANCH1>BIRDSNEST>@.LIST  
 SPOOL BEECH>BRANCH2>@.LIST  
 SPOOL BEECH>@@>@.BUGLIST  
 DELETE %.PROJ1%>(@.BIN @.LIST)

12. Name Generation: The current command is checked for name generation patterns and then executed. The next command then moves up to become the current command. The processing status is

CURRENT: SPOOL BEECH>BRANCH1>BIRDSNEST>@.LIST  
 WAITING: SPOOL BEECH>BRANCH2>@.LIST  
 SPOOL BEECH>@@>@.BUGLIST  
 DELETE %.PROJ1%>(@.BIN @.LIST)

13. Wildcard Matching: Matching on the current command produces

CURRENT: SPOOL BEECH>BRANCH1>BIRDSNEST>EGG1.LIST  
 WAITING: SPOOL BEECH>BRANCH1>BIRDSNEST>EGG2.LIST  
 SPOOL BEECH>BRANCH1>BIRDSNEST>EGG3.LIST  
 SPOOL BEECH>BRANCH2>@.LIST  
 SPOOL BEECH>@@>@.BUGLIST  
 DELETE %.PROJ1%>(@.BIN @.LIST)

14. Name Generation: The current command, and then the next two, are checked for name generation. The current command is then executed, which results in the following processing status:

CURRENT: SPOOL BEECH>BRANCH1>BIRDSNEST>EGG2.LIST  
 WAITING: SPOOL BEECH>BRANCH1>BIRDSNEST>EGG3.LIST  
 SPOOL BEECH>BRANCH2>@.LIST  
 SPOOL BEECH>@@>@.BUGLIST  
 DELETE %.PROJ1%>(@.BIN @.LIST)

15. Execution: The current command is executed, which produces the following status:

CURRENT: SPOOL BEECH>BRANCH1>BIRDSNEST>EGG3.LIST  
 WAITING: SPOOL BEECH>BRANCH2>@.LIST  
 SPOOL BEECH>@@>@.BUGLIST  
 DELETE %.PROJ1%>(@.BIN @.LIST)

16. Execution: The current command is executed, which produces the following status:

```
CURRENT:      SPOOL BEECH>BRANCH2>@.LIST
WAITING:      SPOOL BEECH>@@>@.BUGLIST
              DELETE %.PROJ1%>(@.BIN @.LIST)
```

17. Wildcard Matching: The current command requires wildcard matching. However, there are no wildcard matches for the directory BRANCH2. The current command line never executes, and the next command becomes current. The processing status is

```
CURRENT:      SPOOL BEECH>@@>@.BUGLIST
WAITING:      DELETE %.PROJ1%>(@.BIN @.LIST)
```

18. Treewalking: Treewalking evaluation produces the following status:

```
CURRENT:      SPOOL BEECH>BRANCH1>@.BUGLIST
WAITING:      SPOOL BEECH>BRANCH1>BIRDSNEST>@.BUGLIST
              SPOOL BEECH>BRANCH2>@.BUGLIST
              DELETE %.PROJ1%>(@.BIN @.LIST)
```

19. Wildcard Expansion: Wildcard expansion produces the following status:

```
CURRENT:      SPOOL BEECH>BRANCH1>LEAF.BUGLIST
WAITING:      SPOOL BEECH>BRANCH1>TWIG.BUGLIST
              SPOOL BEECH>BRANCH1>BIRDSNEST>@.BUGLIST
              SPOOL BEECH>BRANCH2>@.BUGLIST
              DELETE %.PROJ1%>(@.BIN @.LIST)
```

20. Name Generation: The top two commands are each checked for name generation. The first command is executed, resulting in the following processing status:

```
CURRENT:      SPOOL BEECH>BRANCH1>TWIG.BUGLIST
WAITING:      SPOOL BEECH>BRANCH1>BIRDSNEST>@.BUGLIST
              SPOOL BEECH>BRANCH2>@.BUGLIST
              DELETE %.PROJ1%>(@.BIN @.LIST)
```

21. Execution: The second command from step 20 is executed, producing the following status report:

```
CURRENT:      SPOOL BEECH>BRANCH1>BIRDSNEST>@.BUGLIST
WAITING:      SPOOL BEECH>BRANCH2>@.BUGLIST
              DELETE %.PROJ1%>(@.BIN @.LIST)
```

22. Wildcard Matching: Matching produces two more null matches. The top two commands do not execute, and the next command becomes current. Steps 23 to 43 result from this command:



```
CURRENT:  DELETE BEECH>BRANCH1>BIRDSNEST>@.BIN
WAITING:  DELETE BEECH>BRANCH2>@.BIN
          DELETE BEECH>@@>@.LIST
```

30. Wildcard Matching: Matching of wildcards replaces the wildcard name in the current command and generates new commands for the waiting list. The processing status is

```
CURRENT:  DELETE BEECH>BRANCH1>BIRDSNEST>EGG1.BIN
WAITING:  DELETE BEECH>BRANCH1>BIRDSNEST>EGG2.BIN
          DELETE BEECH>BRANCH1>BIRDSNEST>EGG3.BIN
          DELETE BEECH>BRANCH2>@.BIN
          DELETE BEECH>@@>@.LIST
```

31. Execution: The current command is executed, resulting in the following processing status:

```
CURRENT:  DELETE BEECH>BRANCH1>BIRDSNEST>EGG2.BIN
WAITING:  DELETE BEECH>BRANCH1>BIRDSNEST>EGG3.BIN
          DELETE BEECH>BRANCH2>@.BIN
          DELETE BEECH>@@>@.LIST
```

32. Execution: The current command is executed, resulting in the following processing status:

```
CURRENT:  DELETE BEECH>BRANCH1>BIRDSNEST>EGG3.BIN
WAITING:  DELETE BEECH>BRANCH2>@.BIN
          DELETE BEECH>@@>@.LIST
```

33. Execution: The current command is executed, resulting in the following processing status:

```
CURRENT:  DELETE BEECH>BRANCH2>@.BIN
WAITING:  DELETE BEECH>@@>@.LIST
```

34. Wildcard Matching: Nothing matches the wildcard, so the DELETE command does not execute. The processing status is

```
CURRENT:  DELETE BEECH>@@>@.LIST
```

35. Treewalking: The last treewalk generates new command lines. The processing status is

```
CURRENT:  DELETE BEECH>BRANCH1>@.LIST
WAITING:  DELETE BEECH>BRANCH1>BIRDSNEST>@.LIST
          DELETE BEECH>BRANCH2>@.LIST
```



36. Wildcard Matching: Matching of wildcard names replaces the wildcard name of the current command with an entryname. New commands are generated and placed on the waiting list. The processing status is

CURRENT:       DELETE BEECH>BRANCH1>LEAF.LIST  
WAITING:       DELETE BEECH>BRANCH1>TWIG.LIST  
                  DELETE BEECH>BRANCH1>BIRDSNEST>@.LIST  
                  DELETE BEECH>BRANCH2>@.LIST

37. Execution: The current command is executed, resulting in the following processing status:

CURRENT:       DELETE BEECH>BRANCH1>TWIG.LIST  
WAITING:       DELETE BEECH>BRANCH1>BIRDSNEST>@.LIST  
                  DELETE BEECH>BRANCH2>@.LIST

38. Wildcard Matching: A found match in the current command replaces the wildcard symbol @:

CURRENT:       DELETE BEECH>BRANCH1>BIRDSNEST>@.LIST  
WAITING:       DELETE BEECH>BRANCH2>@.LIST

39. Execution: The current command is executed, resulting in the following processing status:

CURRENT:       DELETE BEECH>BRANCH1>BIRDSNEST>EGG1.LIST  
WAITING:       DELETE BEECH>BRANCH1>BIRDSNEST>EGG2.LIST  
                  DELETE BEECH>BRANCH1>BIRDSNEST>EGG3.LIST  
                  DELETE BEECH>BRANCH2>@.LIST

40. Execution: The current command is executed, resulting in the following processing status:

CURRENT:       DELETE BEECH>BRANCH1>BIRDSNEST>EGG2.LIST  
WAITING:       DELETE BEECH>BRANCH1>BIRDSNEST>EGG3.LIST  
                  DELETE BEECH>BRANCH2>@.LIST

41. Execution: The current command is executed, resulting in the following processing status:

CURRENT:       DELETE BEECH>BRANCH1>BIRDSNEST>EGG3.LIST  
WAITING:       DELETE BEECH>BRANCH2>@.LIST

42. Wildcard Matching: The last wildcard matching is performed:

CURRENT:       DELETE BEECH>BRANCH2>@.LIST

43. Execution: Nothing matches the wildcard, so the DELETE command does not execute. The command line of step 1 has been completely processed and executed.

## Terminal Display

When the command in the example executes, you see the following response at the terminal.

```

OK, CHECK *.PROJ%
[SPool Rev. 23.2.0 Copyright (c) 1991, Prime Computer, Inc.]
Request 11 added to queue, 1 records : <SYSONE>BEECH>BRANCH1>LEAF.LIST
[SPool Rev. 23.2.0 Copyright (c) 1991, Prime Computer, Inc.]
Request 12 added to queue, 1 records : <SYSONE>BEECH>BRANCH1>TWIG.LIST
[SPool Rev. 23.2.0 Copyright (c) 1991, Prime Computer, Inc.]
Request 13 added to queue, 1 records:<SYSONE>BEECH>BRANCH1>BIRDSNEST>EGG1.LIST
[SPool Rev. 23.2.0 Copyright (c) 1991, Prime Computer, Inc.]
Request 14 added to queue, 1 records:<SYSONE>BEECH>BRANCH1>BIRDSNEST>EGG2.LIST
[SPool Rev. 23.2.0 Copyright (c) 1991, Prime Computer, Inc.]
Request 15 added to queue, 1 records:<SYSONE>BEECH>BRANCH1>BIRDSNEST>EGG3.LIST
[SPool Rev. 23.2.0 Copyright (c) 1991, Prime Computer, Inc.]
Request 16 added to queue, 1 records : <SYSONE>BEECH>BRANCH1>LEAF>BUGLIST
[SPool Rev. 23.2.0 Copyright (c) 1991, Prime Computer, Inc.]
Request 17 added to queue, 1 records : <SYSONE>BEECH>BRANCH1>TWIG>BUGLIST
(std$cp) Verify wildcard selections for "BEECH>BRANCH1>@.BIN :
"LEAF.BIN"? YES
"TWIG.BIN"? YES
(std$cp) Verify wildcard selections for "BEECH>BRANCH1>BIRDSNEST>@.BIN :
"EGG1.BIN"? YES
"EGG2.BIN"? YES
"EGG3.BIN"? YES
(std$cp) Verify wildcard selections for "BEECH>BRANCH1>@.LIST :
"LEAF.LIST"? YES
"TWIG.LIST"? YES
(std$cp) Verify wildcard selections for "BEECH>BRANCH1>BIRDSNEST>@.LIST :
"EGG1.LIST"? YES
"EGG2.LIST"? YES
"EGG3.LIST"? YES
OK,

```

*Replace this page with the tab page labeled*

**6 Command-line  
Editing (ECL)**

# Command-line Editing (ECL)

## ***Introduction***

EDIT\_CMD\_LINE (ECL) is a command-line editor that allows you to control command-line input to the terminal. On the simplest level, this means that you don't have to retype an entire command line if you've made a mistake. More importantly, ECL is a sophisticated way to manipulate command-line input. This chapter, divided into four sections, documents several important features of ECL.

- The section titled Introduction provides general information on ECL.
- The section titled Automatic Pathname Completion details how you can complete partially-entered pathnames, replace pathnames, and perform a treewalk wildcard expansion.
- The section titled ECL Key Bindings describes ECL's default key bindings, the key bindings file, key binding customization, and Programmable Function (PF) commands.
- The section titled ECL Commands Reference lists all of the ECL commands in alphabetical order and provides a description of each command.

For information about the ECL command and its command-line options, see Chapter 2. For additional information about ECL, including tutorial information, see the *PRIMOS User's Guide*.

## ***Control Characters***

Control characters entered into the command line are displayed by ECL as a two-character sequence: ^ (or ~ for the Prime Extended Character Set) followed by the corresponding ASCII-7 letter for the control character (for example, the ASCII mnemonic DEL is displayed as ^?, and the Prime ECS mnemonic SSA is displayed as ~F).



Ctrl-P. In order for this to work, the Ctrl-P to be quoted or echoed cannot be entered using the type-ahead feature of PRIMOS.

The suspension of the break character can lead to a confusing situation. For example, suppose you accidentally enter *do\_echo* or *do\_quote* and you don't know how to exit. Typing Ctrl-P does not allow you to break out because of the changed interpretation. In this case, type Ctrl-P twice (or more) in rapid succession in order to break out of the process.

### ***ECL as an EPF Command Function***

When invoked as an EPF command function, ECL returns as its value a string representing the package version number followed by the internal data structure version number, separated by a space (for example, 303 302). This is useful if, for example, the code in your CPL program references one or both of these numbers.

### ***ECL and Command Output (COMO) Files***

While a COMO file is open, ECL keeps the file clean by limiting output to the submitted command and its prompt. This aids the viewing, editing, and printing of such files. See also the `-CLEAN_COMO / -NO_CLEAN_COMO` options, documented in Chapter 2, for more information about controlling ECL terminal output to a COMO file.

### ***Automatic Pathname Completion***

ECL has a facility called automatic pathname completion. You type in part of the pathname, and ECL attempts to complete it for you. This is done with the *expand\_wild* command. *expand\_wild* attempts to complete the partially-typed pathname that the cursor is positioned within. The partially-typed pathname you enter must conform to the currently implemented PRIMOS pathname standards.

The *expand\_wild* command permits you to specify a pathname either above or below your current position in the directory tree. Typing `*<` references the directory above the current one, `*<<` the directory two levels above, and so on. You can then reference downward from such a point. Thus, to reference another branch in the directory tree, type `*<BRANCH2` (if you are one level below BRANCH2) or `*<>BRANCH2` (if you are in a different branch of the directory tree).

### ***How ECL Completes Pathnames***

The following procedure describes the method ECL uses to complete partial pathnames with the *expand\_wild* command.

1. The partially-typed pathname is extracted from the command line.



method instead of the menu, you can turn off the menu-list numbering feature with the `-NO_WILD_MENU` option.

### **Replacing Pathnames**

You can replace the pathname you entered with its equivalent full PRIMOS pathname within your command line. To do this, specify an *n* count of 16 for the `expand_wild` command (*multiplier multiplier expand\_wild*). This also causes abbreviation expansion of *all* pathname entrynames, including the last one (or the one that the cursor is within) when the `-WILD_ABBREV` option is in effect.

### **Treewalk Wildcard Expansion**

When you specify the `-WILD_TAIL` option, you can perform a *treewalk* wildcard expansion. Do this by positioning the cursor before the final entryname of the partially-completed pathname (that is, before the last `>`).

ECL attempts to locate the matching directories (and subdirectories, subsubdirectories, and so forth) to complete the pathname. When displaying the list of matches, full pathnames are shown and the list is *not* sorted. Be aware that PRIMOS treewalk pathnames implicitly bind the treewalk wildcard expression to the end of the pathname rather than to the beginning. That is, additional directories can be inserted to the left of the first wildcarded entryname position. For example, if the following four files exist,

```
<DISK>USER>DIR1>SOMEFILE
<DISK>USER>XXX1>XXX2>DIR3>SOMEFILE
<DISK>USER>DIR1>XXX2>SOMEFILE
<DISK>USER>XXX1>XXX2>XXX3>SOMEFILE
```

then the treewalk pathname `<DISK>USER>D@>SOMEFILE` finds only the first two, while `<DISK>USER>X@>SOMEFILE` finds only the last two.

### **Example Session**

The following is a sample dialog of a PRIMOS session. Commentary is provided between examples. The cursor position is shown using the underscore character. Completion requests are made with the `expand_wild` command.





```
1) FILE1.TXT   2) FILE2.TXT   3) PRODUCTS>   4) SUBDIR1>5)
SUBDIR2>      6) ZERO
```

The directories have the > character appended for easy identification. Now attach to the directory PRODUCTS by typing P and again requesting completion:

```
OK, ATTACH <DISK1>USER1>DIRECTORY1>P_
```

Here is the result:

```
OK, ATTACH <DISK1>USER1>DIRECTORY1>PRODUCTS>_
```

Note that the > has appeared because PRODUCTS is a directory. Alternatively, you can type Esc 3 Esc Ctrl-I, invoking the *expand\_wild\_menu* command with a numeric argument of 3 to directly choose PRODUCTS from the menu list display.

Because the ATTACH command does not take the > as the last token on the command line, you must delete it before submission. Having done this, attach to PRODUCTS. Now, try to reference upward from the current directory.

```
OK, LD *<_
```

There is no unambiguous expansion in the parent directory, so ECL lists its contents:

```
1) FILE1.TXT   2) FILE2.TXT   3) PRODUCTS>   4) SUBDIR1>
5) SUBDIR2>   6) ZERO
```

To list the file ZERO, you need only to type

```
OK, LD *<Z_
```

and request completion to produce

```
OK, LD <DISK1>USER1>DIRECTORY1>ZERO_
```

The \*< expression has been expanded automatically into a legal PRIMOS pathname. Finally, suppose that you want to create another directory in the directory above:

```
OK, CREATE *NEWDIR_
```

This produces an error if submitted because CREATE does not understand the new pathname syntax. Instead, request completion with a count of 16 (for example, by issuing the *multiplier multiplier* command), which replaces the pathname with its legal equivalent:

```
OK, CREATE <DISK1>USER1>DIRECTORY1>NEWDIR_
```



## ***ECL Key Bindings***

Each key on your terminal is *bound to* a specific ECL command. In other words, when you press a key you execute the command that is associated with, or bound to, that key. For example, within ECL Ctrl-A is bound to the *begin\_line* command, and issuing the command Ctrl-A positions the cursor at the beginning of the current line. Most of the keys on the main keypad are bound to the *self\_insert* command; when you press one of them, the character that appears on that key is echoed on the terminal screen.

You can change these default bindings to suit your needs by binding any character or sequence of characters to any ECL command. To define specific functions for specific keys, you must perform the following steps:

1. Create a **bindings file** with a standard text editor. A bindings file is a text file containing one binding per line. This file may have any legal PRIMOS filename.
2. Use the ECL `-BIND_TERM` option to compile and load the bindings.

To save time and increase efficiency, it is recommended that you also perform these additional steps:

3. Use the ECL `-SAVE_TERM` option to save the compiled bindings in a separate file.
4. Use the ECL `-RESTORE_TERM` option to quickly reload this file during a subsequent session.

The `-SAVE_TERM` and `-RESTORE_TERM` options are discussed later in the section titled Compiling, Saving, and Restoring the Bindings File.

## ***Syntax of the Bindings File***

The syntax and syntax descriptions of the bindings file are given below. Each binding must be on a separate line, with blank spaces or tab settings between the fields but not within any of them. (You can include comments in the file by entering the number character (#) in the first column of a line.)

*sequence command* [*program*] [*terminal\_list*]

### ***sequence Argument***

*sequence* is the character or character sequence that you want to bind to a particular ECL command. The *sequence* field can contain the character sequence sent by any key on your terminal. Two major categories of characters are printing characters and nonprinting characters.

|                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|-------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Printing characters</b>    | Most of the keys on the main keypad send printing characters because when you press one of them the character that appears on the keypad is echoed on the screen. You can specify printing characters as part of sequence simply by entering them in the bindings file.                                                                                                                                                                                                                                                                                                                                                                                   |
| <b>Nonprinting characters</b> | Several keys, or key combinations such as Ctrl-A, do not send a printing character to be echoed on the terminal screen. Instead, a nonprinting character or sequence of nonprinting characters is sent. Keys such as Backspace, Tab, and Esc send nonprinting characters. Function keys such as F1 through F10, PF1 through PF12, and PA1 through PA4 (depending on your terminal) send a sequence of nonprinting (and printing) characters. You cannot specify nonprinting characters by their keypad name, as you can with printing characters. Instead, you must use the methods discussed below in the section Other Nonprinting Character Sequences. |

### ***command* Argument**

*command* is the name of the ECL command to be invoked by the character or character sequence (for example, *begin\_line*). Refer to the section ECL Commands Reference for a complete list of command names.

### ***program* Argument**

*program* is the function to be programmed to an ECL programmable function command. If you use the *program* argument, *command* either must be one of ECL's programmable function commands (pfkey1 to pfkey512), or it must be *execute\_macro* to specify the initial keyboard macro program. Also, you must include the vertical bar character (|) between *command* and *program*. Spaces are not permitted.

### ***terminal\_list* Argument**

*terminal\_list* specifies the terminal type(s) that apply to this binding. If a particular binding has more than one terminal type, separate them by the vertical bar character (|) with no spaces in between. If you do not include this argument, ECL performs the binding regardless of what kind of terminal you are using. You also can enter the vertical bar character, either alone or at the beginning or end of the *terminal\_list*, to include ECL's default generic 24-line by 80-column terminal in the list of terminals for the binding.

**Note**

The number character (#) cannot appear in Column 1 in your bindings file because ECL interprets this character as a comment character in that position. To use the number character as a character other than a comment character, leave a space in Column 1.

**Key Binding Sequences**

You can specify character keys for the ECL binding sequences in two ways.

**Control Sequences With Printing Characters**

Use the caret (^) to indicate a control sequence. For example, the representation ^A in a bindings file specifies Ctrl-A.

**Other Nonprinting Character Sequences**

The backslash character (\) is the first character in the representation of any nonprinting character that is part of *sequence*. You can also use the backslash to specify printing characters, such as the caret (represented by \^ ) and the backslash character itself (represented by \\). Several backslash character combinations have special meanings, as shown in the following list.

| <i>Sequence</i> | <i>Meaning</i>                               |
|-----------------|----------------------------------------------|
| \b              | The BACKSPACE key (same as ^H)               |
| \E              | The ESC key (same as ^[)                     |
| \f              | A form-feed character (same as ^L)           |
| \l              | ASCII linefeed character                     |
| \n              | A linefeed character                         |
| \r              | ASCII carriage return character (same as ^M) |
| \t              | The TAB key (same as ^I)                     |
| \v              | The vertical tab character (same as ^K)      |

Use the backslash character to include the octal representation of a character in the sequence field. For instance, you can use \205 in the bindings file to denote ^E. Also, if you must include a space in the *sequence* or *program* argument, you must use the representation \240 (the octal representation of a space character).

Preceding any other printing character with the backslash results in binding only that character to the ECL command specified. For example, if you bind \x to *prev\_line*, ECL displays the previous command line when you press x.

As an example, suppose the bindings file for a PT45 contains these two lines:

```

\EC          forward_char          pt 45 | dm30
\ED          back_char             pt 45 | dm30

```

The first line shown above sets the sequence `\EC` (Esc C) as the command to move forward one character (*forward\_char*) on a PT45 terminal. (The right arrow key `->` is the same as Esc C on a PT45.)

Similarly, the next line shows that the sequence `\ED` (Esc D) is the key binding for the command to move backward one character (*back\_char*). (The left arrow key `<-` is the same as Esc D on a PT45.) Be aware that the sequence field in the bindings file is case-sensitive, but the command field and the terminal field are not.

Table 6-1 lists each character of the ASCII subset of the Prime ECS character set. This table contains character codes most often used in a bindings file. The left side of each column contains the ASCII mnemonic, and the right side of each column shows the corresponding character sequence. For example, the BS mnemonic is represented in the bindings file as `^H` (Ctrl-H). Refer to Appendix D for the complete Prime ECS character set.

Table 6-1. Quick-reference ASCII Character Chart

|     |    |     |    |    |      |   |   |   |   |   |   |   |   |     |    |
|-----|----|-----|----|----|------|---|---|---|---|---|---|---|---|-----|----|
| NUL | ^@ | DLE | ^P | SP | \240 | 0 | 0 | @ | @ | P | P | . | . | p   | p  |
| SOH | ^A | DC1 | ^Q | !  | !    | 1 | 1 | A | A | Q | Q | a | a | q   | q  |
| STX | ^B | DC2 | ^R | "  | "    | 2 | 2 | B | B | R | R | b | b | r   | r  |
| ETX | ^C | DC3 | ^S | #  | #    | 3 | 3 | C | C | S | S | c | c | s   | s  |
| EOT | ^D | DC4 | ^T | \$ | \$   | 4 | 4 | D | D | T | T | d | d | t   | t  |
| ENQ | ^E | NAK | ^U | %  | %    | 5 | 5 | E | E | U | U | e | e | u   | u  |
| ACK | ^F | SYN | ^V | &  | &    | 6 | 6 | F | F | V | V | f | f | v   | v  |
| BEL | ^G | ETB | ^W | ,  | ,    | 7 | 7 | G | G | W | W | g | g | w   | w  |
| BS  | ^H | CAN | ^X | (  | (    | 8 | 8 | H | H | X | X | h | h | x   | x  |
| HT  | ^I | EM  | ^Y | )  | )    | 9 | 9 | I | I | Y | Y | i | i | y   | y  |
| LF  | ^J | SUB | ^Z | +  | +    | : | : | J | J | Z | Z | j | j | z   | z  |
| VT  | ^K | ESC | ^E | *  | *    | ; | ; | K | K | [ | [ | k | k | {   | {  |
| FF  | ^L | FS  | ^  | ,  | ,    | < | < | L | L | \ | \ | l | l |     |    |
| CR  | ^M | GS  | ^] | -  | -    | = | = | M | M | ] | ] | m | m | }   | }  |
| SO  | ^N | RS  | ^^ | .  | .    | > | > | N | N | ^ | ^ | n | n | ~   | ~  |
| SI  | ^O | US  | ^_ | /  | /    | ? | ? | O | O | - | - | o | o | DEL | ^? |

### Commands Useful for ECL Key Bindings

This section introduces some fundamental ECL commands that are useful in setting your own bindings. These commands are discussed further in the sections that follow, and are also included in the complete list of ECL commands at the end of the chapter.

`do_echo` Esc @

Echoes to the terminal the characters that follow without interpreting them as

being part of a command. Echoing discontinues when *do\_echo* is repeated. For example, Esc @ PRT Esc @ sends the string "PRT" to the terminal.

***do\_quote*** Esc `

Self-inserts the characters that follow directly into the command line without further processing. Quoting discontinues when *do\_quote* is repeated. (This is useful for determining what a key on the terminal actually sends.)

***echo\_raw*** [*not bound*]

Echoes the bound character sequence at the terminal without including it in the command line. This is useful for keys such as Scroll Up/Down on a PST 100 so that you can scroll the screen locally without affecting the command line being entered.

***explain\_key*** Ctrl-\_\_

Prompts for the key sequence for which you need information and displays the command name bound to that sequence. This command is never collected within a keyboard macro. The numeric argument is ignored but saved for the next command entered.

***quote*** Ctrl-Q or Esc Q or Esc q

Quotes the next character that is typed for self-insertion. This permits the insertion of characters into the command line that would normally be bound to commands other than *self\_insert*.

***self\_insert*** *printable characters*

Inserts the last character of the bound sequence into the command line at the current position.

***submit*** Ctrl-J

Submits the entire command line for processing. The cursor can be anywhere on the line. *submit* finishes a keyboard macro if the macro was begun at the current level; this makes the macro self-terminating.

***unbound*** [*anything not otherwise bound*]

Disables any binding for this sequence. Everything not explicitly bound to another sequence is unbound. This is useful in nullifying a key whose effects are potentially dangerous.

### ***Programmable Function Commands***

Bindings can be made to Programmable Function commands (PF commands), and these bindings can be programmed in the program field in the bindings file. Binding ECL PF commands to the extra function keys on a terminal can be useful because they are not keys that you normally use to enter command line text. Also, you can reconfigure an undesired keyboard layout using PF command bindings.



You can use a maximum of 512 different PF commands in a bindings file. Each PF command value in program cannot exceed 200 characters, and the total number of characters you can use for programming all PF commands cannot exceed 8000 characters.

---

**Note**

PF commands are not necessarily synonymous to the special function keys on a terminal; PF commands can be any key or sequence of characters. In using PF commands, you must include in the command field one of the PF commands (pfkey1 through pfkey512) or *execute\_macro* to specify the initial keyboard macro program.

---

### **How the PF Commands Work**

In the bindings file, the program field (program) is the sequence of characters executed by the PF command. When a PF command is used, program is substituted for sequence and reinterpreted by ECL. For example, if the bindings file for a PT200 contains this line for F1:

```
\EO!          pfkey1          pt200
```

whenever you press the PT200's F1 key, its default program is inserted into the command line. Now suppose you want to bind the PRIMOS LD command to this key. Change this line in the bindings file, following the syntax rules described earlier, so that the line now looks like this:

```
\EO!          pfkey1|LD          pt200
```

Next, recompile the bindings file:

```
OK, LD  # The ECL command is now set to LD
Established PT200 terminal capabilities (9600 baud). (EDIT_CMD_LINE)
Established PT200 bindings from 'ECL_BINDINGS' (91% available). (EDIT_CMD_LINE)
OK,
```

For the duration of this terminal session, pressing F1 executes the LD command. To keep this change in compiled form for future sessions, use the `-SAVE_TERM` and `-RESTORE_TERM` options.

### **Multiple Command Sequences**

You can bind more than one sequence to the same PF command by programming the first binding to a particular PF command, and then making other normal (unprogrammed) bindings to the same PF command. If a normal binding is made to an unprogrammed PF command, this programs the key with a default of `FUNC#`, where # is the number (1 through 512) of the PF command.

Continuing the example in the previous section (when you programmed the LD command to F1 on a PT200 terminal), suppose you want F2 to execute the LD command just as F1 does. You know that the sequence \EO" is bound by default to the PF command pfkey2 (F2 on a PT200); here is its binding in the bindings file:

```
\EO"          pfkey2          pt200
```

To make F2 execute the LD command as F1 does, you don't have to put LD in the program field (pfkey2|LD) as you would expect. Instead, just change the PF command of F2 to match the PF command of F1, which executes the LD command. Thus, your bindings file looks like this:

```
\EO!          pfkey1|LD          pt200\EO"
pfkey1          pt200
```

Now, both F1 and F2 execute LD.

---

**Note**

Be aware that there is no default keyboard macro program for the *execute\_macro* command.

---

### Self-terminating Function Keys

Normally, when you press a terminal's function key, a key-specific character sequence plus a termination sequence is transmitted. In other words, the character sequence associated with that function key is present along with an instruction to *submit* that character sequence. You could equate this to executing the LD command at PRIMOS level; you type the letters L and D and then press a key, Return, that instructs PRIMOS to submit, or terminate, the LD command. Return is bound to the ASCII sequence Ctrl-J (represented in the bindings file as ^J) by default.

Function keys also have default termination sequences, but you can change this default if you so desire. For example, on the PT45, two switches at the back of the terminal allow you to terminate a function key with ^C, ^D, or ^J. On a PST 100 or a PT200, you can use software to change the default, and it can be set to nothing (which is the terminal default) or set to ^J. Since ^J is usually bound to the *submit* command, you can make function key bindings that self-terminate (submit themselves).

Suppose you have a PT45 terminal that terminates function keys with ^J. Now examine the contents of the following bindings file:

```
^B\Ep          pfkey1          pt45|dm30
^B\Eq          pfkey2|F2          pt45|dm30
^B\Er^J        pfkey3|\E@\ED\E@   pt45|dm30
^B\Es^J        pfkey4|\E'COPY\E'^J pt45|dm30
```

1. Binding #1 binds the terminal's F1 to the default programmed sequence FUNC1 because no *program* is supplied. This binding self-terminates because ^J has been left off the bound *sequence*.

2. Binding #2 binds F2 so that the string "F2" is inserted into the command line. (The characters "F" and "2" are bound to *self\_insert*.)

This function key also self-terminates.

3. Binding #3 makes F3 work this way: echo, Esc D, stop echo. (^E@ is bound to *do\_echo*.) This causes Esc D to be sent to the terminal. Notice that the ^J is added to the *sequence* field so that ^J is explicitly absorbed as part of the binding and *not* left to be interpreted as the *submit* command, since you don't want to submit Esc D. This key, therefore, does not self-terminate.

4. Binding #4 binds F4 to do the following: quote, "COPY", stop quote, terminate. (^E' is bound to *do\_quote*.) Notice that ^J is in the *sequence* field to be considered as part of the binding sequence. The ^J in the *program* field causes COPY to be inserted and then submitted explicitly (self-terminated).

You can append ^J to the end of the program field in order to explicitly establish a self-terminating PF command. This is useful for terminals that cannot be set up to terminate function keys with ^J themselves.

---

**Note**

To avoid programming a PF command to another PF command, use the sequence bound to *quote* or *do\_quote*. For example, suppose you don't like your keyboard layout and want to remap the caret (^) character and the tilde (~) character on a PT45 so that they self-insert one another. In this case, the following entries in the bindings file are illegal:

```
~          pfkey511 | \^          pt 45 | dm30
\^        pfkey512 | ~          pt 45 | dm30
```

Because each character is now a PF command binding and no longer bound to *self\_insert*, an endless loop results. The proper way to have these characters self-insert one another is to use the quote command (^Q), as follows:

```
~          pfkey511 | ^Q\^        pt 45 | dm30
\^        pfkey512 | ^Q~        pt 45 | dm30
```

---

### **Sample Bindings File**

The sample bindings file shows how to represent various terminal keys. The file includes comments, denoted by # in the first position, that explain the key binding directly below. This file is for a PST 100 terminal, and all command descriptions assume a count of 1.

```
#####
###
###
###
### PST100/PT200 bindings:
###
#####
###
#
# Sequence Command|Program Terminal
# -----
#
# pressing Esc n moves the cursor to the next line
# (\e = Escape key)
#
# \en next_line pst100
#
# pressing Esc p moves the cursor to the previous line
#
# \ep prev_line pst100
#
# pressing Ctrl a or Ctrl A moves the cursor one word to the right
# (^ = Control key)
#
# ^a forward_word pst100
#
# pressing the caret character and the letter a moves the cursor
# one character to the right
# (\^ = caret key)
#
# \^a forward_char pst100
#
# pressing Ctrl B (which is 202 octal) moves the cursor to the
# beginning of the line
#
# \202 begin_line pst100
#
#
```

### ***Bindings for Terminals***

Here is the standard bindings file for the PT45, PST 100, and PT200 terminals. In the listing shown here, the designation pt200 actually represents the four different kinds of PT200: the standard PT200, the PT200 in wide 132-column mode, the color PT200, and the color PT200 in wide 132-column mode.

The TERM\* directory also contains a file called EDIT\_CMD\_BINDSUIX which contains bindings for PT45, PST 100, PT200, and PT250 terminals for emulating EMACS SUIX mode.

```
#####
###
### PST100/PT200 bindings:
###
#####
#
# Sequence      Command          Terminal          <Key>-[Meta-]Unaugmented/Shift/Ctrl/Ctrl-Shift
# -----      -
#
# ^H            rubout_char      pst100            # <BACKSP>-ALL (PST100)
# ^H            rubout_char      pt200             # <BACKSP>-U/S (PT200)
# ^I            expand_wild      pst100            # <TAB>-ALL (PST100)
# ^I            expand_wild      pt200             # <TAB>-U/S (PT200)
# ^J            submit           pst100|pt200     # <RETURN>-ALL
# ^J            submit           pst100|pt200     # <ENTER>-U/S
# ^N            echo_raw         pst100            # <CHAR SET>-U/S (PST100)
# ^N            echo_raw         pt200             # <CHAR SET>-ALL (PT200)
# ^O            echo_raw         pst100            # <CHAR SET>-U/S (PST100)
# ^O            echo_raw         pt200             # <CHAR SET>-ALL (PT200)
# ^P            unbound         pst100            # <STOP>-U (PST100)
# ^P            unbound         pst100            # <CONFIG>-C (PST100)
# \E            (Prefix)        pst100|pt200     # <ESC>-ALL
# \E$A          echo_raw         pst100|pt200     # <HOME>-U          Cursor Relative Home
# \E$B          echo_raw         pst100|pt200     # <HOME>-S/C/CS     Cursor Absolute Home
# \E$G          echo_raw         pst100            # <RESET>-U (PST100) Reset
# \E$G          echo_raw         pt200             # <CLEAR>-U (PT200) Reset
# \E$a          echo_raw         pt200             # <PREV>-U (PT200) Page Up (aka <PG UP>)
# \E$b          echo_raw         pt200             # <NEXT>-U (PT200) Page Down (aka <PG DN>)
# \E?          echo_raw         pst100            # <RESET>-C (PST100) Clear Screen
# \E?          echo_raw         pt200             # <CLEAR>-C (PT200) Clear Screen
# \E{0i        echo_raw         pst100            # <AUX SND>-U (PST100) Media Copy Screen Transfer
# \E{0i        echo_raw         pt200             # <PRT SCN>-U (PT200) Media Copy Screen Print
# \E{4h        unbound         pst100|pt200     # <INSERT>-ALL     Insert Mode On
# \E{4l        unbound         pst100|pt200     # <INSERT>-ALL     Insert Mode Off
# \E{>0i       echo_raw         pst100|pt200     # <AUX SND>-S (PST100) Media Copy Area Transfer
# \E{>1i       echo_raw         pst100            # <AUX SND>-C (PST100) Media Copy Screen Transfer
# \E{>1i       echo_raw         pt200             # <PRT SCN>-C (PT200) Media Copy Screen Transfer
# \E{>2i       echo_raw         pt200             # <PTR SCN>-S (PT200) Media Copy Page Dump
# \E{>10h      echo_raw         pt200             # <NUM LOCK>-ALL (PT200) Function Keypad
# \E{>10l      echo_raw         pt200             # <NUM LOCK>-ALL (PT200) Numeric Keypad
# \E{>20h      unbound         pst100            # <CHAR SET>-C (PST100) DSC Mode Set
# \E{>20l      unbound         pst100            # <CHAR SET>-CS (PST100) DSC Mode Reset
# \E[A         prev_line        pst100            # <UP>-ALL (PST100) Cursor Up
# \E[A         prev_line        pt200             # <UP>-U (PT200) Cursor Up
# \E[B         next_line        pst100            # <DOWN>-ALL (PST100) Cursor Down
# \E[B         next_line        pt200             # <DOWN>-U (PT200) Cursor Down
# \E[C         forward_char     pst100            # <RIGHT>-ALL (PST100) Cursor Right
# \E[C         forward_char     pt200             # <RIGHT>-U (PT200) Cursor Right
# \E[D         back_char        pst100            # <LEFT>-ALL (PST100) Cursor Left
# \E[D         back_char        pt200             # <LEFT>-U (PT200) Cursor Left
# \E[J         kill_line        pst100            # <ERASE>-C/CS (PST100) Erase in Display to EOS
# \E[J         kill_line        pt200             # <ERASE>-CS (PT200) Erase in Display to EOS
# \E[K         kill_line        pst100|pt200     # <ERASE>-S Erase in Line to EOL
# \E[M         echo_raw         pst100|pt200     # <DELETE>-S Delete Line
# \E[O         echo_raw         pst100|pt200     # <ERASE>-C (PT200) Erase in Area to EOA
# \E[P         delete_char      pst100|pt200     # <DELETE>-U Delete Character
# \E[S         echo_raw         pst100            # <SCROLL>-S (PST100) Scroll Up
# \E[S         echo_raw         pt200             # <SCRL UP>-U (PT200) Scroll Up
# \E[T         echo_raw         pst100            # <SCROLL>-U (PST100) Scroll Down
# \E[T         echo_raw         pt200             # <SCRL DN>-U (PT200) Scroll Down
# \E[1U        echo_raw         pst100            # <SCROLL>-C (PST100) Next Page
# \E[U         echo_raw         pt200             # <END/BEGIN>-U (PT200) Next Page
# \E[1V        echo_raw         pst100            # <SCROLL>-CS (PST100) Preceding Page
# \E[V        echo_raw         pt200             # <END/BEGIN>-S (PT200) Preceding Page
```

PRIMOS Commands Reference Guide

|          |          |              |                           |                             |
|----------|----------|--------------|---------------------------|-----------------------------|
| \E[X     | echo_raw | pst100 pt200 | # <ERASE>-U               | Erase Character             |
| \E[Z     | unbound  | pst100       | # <B-TAB>-ALL (PST100)    | Cursor Backward Tabulation  |
| \E[2     | unbound  | pt200        | # <B-TAB>-U/S (PT200)     | Cursor Backward Tabulation  |
| \E[j     | echo_raw | pt200        | # <SCRL RIGHT>-U (PT200)  | Scroll Right                |
| \E[k     | echo_raw | pt200        | # <SCRL LEFT>-U (PT200)   | Scroll Left                 |
| \E]0\E\\ | echo_raw | pst100 pt200 | #                         | Operating System Command    |
| \E]1\E\\ | echo_raw | pst100 pt200 | #                         | Operating System Command    |
| \E]2\E\\ | echo_raw | pst100 pt200 | #                         | Operating System Command    |
| \E_0\E\\ | echo_raw | pst100 pt200 | # <MENU>-U                | Application Program Command |
| \E_1\E\\ | echo_raw | pst100 pt200 | # <HELP>-U                | Application Program Command |
| \E_2\E\\ | echo_raw | pst100 pt200 | # <HELP>-S                | Application Program Command |
| \E_3\E\\ | echo_raw | pst100       | # <HELP>-C/CS (PST100)    | Application Program Command |
| \E_3\E\\ | echo_raw | pt200        | # <HELP>-C (PT200)        | Application Program Command |
| \E_4\E\\ | echo_raw | pt200        | # <HELP>-CS (PT200)       | Application Program Command |
| \E_5\E\\ | echo_raw | pt200        | # <MENU>-S (PT200)        | Application Program Command |
| \E_6\E\\ | echo_raw | pt200        | # <MENU>-C (PT200)        | Application Program Command |
| \E_7\E\\ | echo_raw | pt200        | # <MENU>-CS (PT200)       | Application Program Command |
| \EN!     | unbound  | pt200        | # <PA1>-U (PT200)         |                             |
| \EN%     | unbound  | pt200        | # <PA1>-S (PT200)         |                             |
| \EN)     | unbound  | pt200        | # <PA1>-C (PT200)         |                             |
| \EN-     | unbound  | pt200        | # <PA1>-CS (PT200)        |                             |
| \EN"     | unbound  | pt200        | # <PA2>-U (PT200)         |                             |
| \EN&     | unbound  | pt200        | # <PA2>-S (PT200)         |                             |
| \EN*     | unbound  | pt200        | # <PA2>-C (PT200)         |                             |
| \EN.     | unbound  | pt200        | # <PA2>-CS (PT200)        |                             |
| \EN#     | unbound  | pt200        | # <PA3>-U (PT200)         |                             |
| \EN'     | unbound  | pt200        | # <PA3>-S (PT200)         |                             |
| \EN+     | unbound  | pt200        | # <PA3>-C (PT200)         |                             |
| \EN/     | unbound  | pt200        | # <PA3>-CS (PT200)        |                             |
| \EN\$    | unbound  | pt200        | # <PA4>-U (PT200)         |                             |
| \EN(     | unbound  | pt200        | # <PA4>-S (PT200)         |                             |
| \EN,     | unbound  | pt200        | # <PA4>-C (PT200)         |                             |
| \EN0     | unbound  | pt200        | # <PA4>-CS (PT200)        |                             |
| \EN1     | unbound  | pt200        | # <SCRL LOCK>-U (PT200)   |                             |
| \EN2     | unbound  | pt200        | # <SCRL LOCK>-S (PT200)   |                             |
| \EN3     | unbound  | pt200        | # <SCRL LOCK>-C (PT200)   |                             |
| \EN4     | unbound  | pt200        | # <SCRL LOCK>-CS (PT200)  |                             |
| \EN5     | unbound  | pt200        | # <SM UNLABEL>-U (PT200)  |                             |
| \EN6     | unbound  | pt200        | # <SM UNLABEL>-S (PT200)  |                             |
| \EN7     | unbound  | pt200        | # <SM UNLABEL>-C (PT200)  |                             |
| \EN8     | unbound  | pt200        | # <SM UNLABEL>-CS (PT200) |                             |
| \EN9     | unbound  | pt200        | # <LG UNLABEL>-U (PT200)  |                             |
| \EN:     | unbound  | pt200        | # <LG UNLABEL>-S (PT200)  |                             |
| \EN;     | unbound  | pt200        | # <LG UNLABEL>-C (PT200)  |                             |
| \EN<     | unbound  | pt200        | # <LG UNLABEL>-CS (PT200) |                             |
| \EN=     | unbound  | pt200        | # <SCRL UP>-S (PT200)     |                             |
| \EN>     | unbound  | pt200        | # <SCRL UP>-C (PT200)     |                             |
| \EN?     | unbound  | pt200        | # <SCRL UP>-CS (PT200)    |                             |
| \EN@     | unbound  | pt200        | # <SCRL DOWN>-S (PT200)   |                             |
| \ENA     | unbound  | pt200        | # <SCRL DOWN>-C (PT200)   |                             |
| \ENB     | unbound  | pt200        | # <SCRL DOWN>-CS (PT200)  |                             |
| \ENC     | unbound  | pt200        | # <SCRL LEFT>-S (PT200)   |                             |
| \END     | unbound  | pt200        | # <SCRL LEFT>-C (PT200)   |                             |
| \ENE     | unbound  | pt200        | # <SCRL LEFT>-CS (PT200)  |                             |
| \ENF     | unbound  | pt200        | # <SCRL RIGHT>-S (PT200)  |                             |
| \ENG     | unbound  | pt200        | # <SCRL RIGHT>-C (PT200)  |                             |
| \ENH     | unbound  | pt200        | # <SCRL RIGHT>-CS (PT200) |                             |
| \ENI     | unbound  | pt200        | # <UP>-S (PT200)          |                             |
| \ENJ     | unbound  | pt200        | # <UP>-C (PT200)          |                             |
| \ENK     | unbound  | pt200        | # <UP>-CS (PT200)         |                             |
| \ENL     | unbound  | pt200        | # <DOWN>-S (PT200)        |                             |
| \ENM     | unbound  | pt200        | # <DOWN>-C (PT200)        |                             |
| \ENN     | unbound  | pt200        | # <DOWN>-CS (PT200)       |                             |
| \ENO     | unbound  | pt200        | # <LEFT>-S (PT200)        |                             |

|       |         |              |                   |         |               |
|-------|---------|--------------|-------------------|---------|---------------|
| \ENP  | unbound | pt200        | # <LEFT>-C        | (PT200) |               |
| \ENQ  | unbound | pt200        | # <LEFT>-CS       | (PT200) |               |
| \ENR  | unbound | pt200        | # <RIGHT>-S       | (PT200) |               |
| \ENS  | unbound | pt200        | # <RIGHT>-C       | (PT200) |               |
| \ENT  | unbound | pt200        | # <RIGHT>-CS      | (PT200) |               |
| \ENU  | unbound | pt200        | # <CANCEL>-U      | (PT200) |               |
| \ENV  | unbound | pt200        | # <CANCEL>-S      | (PT200) |               |
| \ENW  | unbound | pt200        | # <CANCEL>-C      | (PT200) |               |
| \ENX  | unbound | pt200        | # <CANCEL>-CS     | (PT200) |               |
| \ENY  | unbound | pt200        | # <DELETE>-C      | (PT200) |               |
| \ENZ  | unbound | pt200        | # <DELETE>-CS     | (PT200) |               |
| \EN[  | unbound | pt200        | # <PREV>-S        | (PT200) | (aka <PG UP>) |
| \EN\[ | unbound | pt200        | # <PREV>-C        | (PT200) | (aka <PG UP>) |
| \EN]  | unbound | pt200        | # <PREV>-CS       | (PT200) | (aka <PG UP>) |
| \EN\^ | unbound | pt200        | # <NEXT>-S        | (PT200) | (aka <PG DN>) |
| \EN_  | unbound | pt200        | # <NEXT>-C        | (PT200) | (aka <PG DN>) |
| \EN`  | unbound | pt200        | # <NEXT>-CS       | (PT200) | (aka <PG DN>) |
| \ENa  | unbound | pt200        | # <END/BEGIN>-C   | (PT200) |               |
| \ENb  | unbound | pt200        | # <END/BEGIN>-CS  | (PT200) |               |
| \ENc  | submit  | pt200        | # <ENTER>-U [E2]  | (PT200) |               |
| \ENd  | submit  | pt200        | # <ENTER>-S [E2]  | (PT200) |               |
| \ENE  | submit  | pt200        | # <ENTER>-C [E2]  | (PT200) |               |
| \ENf  | submit  | pt200        | # <ENTER>-CS [E2] | (PT200) |               |
| \EO   | unbound | pt200        | # <TAB>-C/CS      | (PT200) |               |
| \EO{  | unbound | pt200        | # <B-TAB>-C/CS    | (PT200) |               |
| \EOy  | unbound | pt200        | # <STOP>-S        | (PT200) |               |
| \EOz  | unbound | pt200        | # <STOP>-C        | (PT200) |               |
| \EO!  | pfkey1  | pst100 pt200 | # <F1>-U          |         |               |
| \EO)  | pfkey1  | pst100 pt200 | # <F1>-S          |         |               |
| \EO1  | pfkey1  | pst100 pt200 | # <F1>-C          |         |               |
| \EO9  | pfkey1  | pst100 pt200 | # <F1>-CS         |         |               |
| \EO"  | pfkey2  | pst100 pt200 | # <F2>-U          |         |               |
| \EO*  | pfkey2  | pst100 pt200 | # <F2>-S          |         |               |
| \EO2  | pfkey2  | pst100 pt200 | # <F2>-C          |         |               |
| \EO:  | pfkey2  | pst100 pt200 | # <F2>-CS         |         |               |
| \EO#  | pfkey3  | pst100 pt200 | # <F3>-U          |         |               |
| \EO+  | pfkey3  | pst100 pt200 | # <F3>-S          |         |               |
| \EO3  | pfkey3  | pst100 pt200 | # <F3>-C          |         |               |
| \EO;  | pfkey3  | pst100 pt200 | # <F3>-CS         |         |               |
| \EO\$ | pfkey4  | pst100 pt200 | # <F4>-U          |         |               |
| \EO,  | pfkey4  | pst100 pt200 | # <F4>-S          |         |               |
| \EO4  | pfkey4  | pst100 pt200 | # <F4>-C          |         |               |
| \EO<  | pfkey4  | pst100 pt200 | # <F4>-CS         |         |               |
| \EO%  | pfkey5  | pst100 pt200 | # <F5>-U          |         |               |
| \EO-  | pfkey5  | pst100 pt200 | # <F5>-S          |         |               |
| \EO5  | pfkey5  | pst100 pt200 | # <F5>-C          |         |               |
| \EO=  | pfkey5  | pst100 pt200 | # <F5>-CS         |         |               |
| \EO&  | pfkey6  | pst100 pt200 | # <F6>-U          |         |               |
| \EO.  | pfkey6  | pst100 pt200 | # <F6>-S          |         |               |
| \EO6  | pfkey6  | pst100 pt200 | # <F6>-C          |         |               |
| \EO>  | pfkey6  | pst100 pt200 | # <F6>-CS         |         |               |
| \EO'  | pfkey7  | pst100 pt200 | # <F7>-U          |         |               |
| \EO/  | pfkey7  | pst100 pt200 | # <F7>-S          |         |               |
| \EO7  | pfkey7  | pst100 pt200 | # <F7>-C          |         |               |
| \EO?  | pfkey7  | pst100 pt200 | # <F7>-CS         |         |               |
| \EO(  | pfkey8  | pst100 pt200 | # <F8>-U          |         |               |
| \EO0  | pfkey8  | pst100 pt200 | # <F8>-S          |         |               |
| \EO8  | pfkey8  | pst100 pt200 | # <F8>-C          |         |               |
| \EO@  | pfkey8  | pst100 pt200 | # <F8>-CS         |         |               |
| \EOA  | pfkey11 | pst100 pt200 | # <PF1>-U         |         |               |
| \EOO  | pfkey11 | pst100 pt200 | # <PF1>-S         |         |               |
| \EO]  | pfkey11 | pst100 pt200 | # <PF1>-C         |         |               |
| \EOk  | pfkey11 | pst100 pt200 | # <PF1>-CS        |         |               |
| \EOB  | pfkey12 | pst100 pt200 | # <PF2>-U         |         |               |



|       |                |              |                 |          |
|-------|----------------|--------------|-----------------|----------|
| \BOp  | pFkey12        | pst100 pt200 | # <PF2>-S       |          |
| \EO\^ | pFkey12        | pst100 pt200 | # <PF2>-C       | (PST100) |
| \BO1  | pFkey12        | pst100 pt200 | # <PF2>-CS      | (PT200)  |
| \BOC  | pFkey13        | pst100 pt200 | # <PF3>-U       | (PST100) |
| \EOQ  | pFkey13        | pst100 pt200 | # <PF3>-S       | (PT200)  |
| \BO_  | pFkey13        | pst100 pt200 | # <PF3>-C       | (PST100) |
| \BOM  | pFkey13        | pst100 pt200 | # <PF3>-CS      | (PT200)  |
| \BOD  | pFkey14        | pst100 pt200 | # <PF4>-U       | (PST100) |
| \BOR  | pFkey14        | pst100 pt200 | # <PF4>-S       | (PT200)  |
| \EO`  | pFkey14        | pst100 pt200 | # <PF4>-C       | (PST100) |
| \EOh  | pFkey14        | pst100 pt200 | # <PF4>-CS      | (PT200)  |
| \BOE  | pFkey15        | pst100 pt200 | # <PF5>-U       | (PST100) |
| \BOS  | pFkey15        | pst100 pt200 | # <PF5>-S       | (PT200)  |
| \EOa  | pFkey15        | pst100 pt200 | # <PF5>-C       | (PST100) |
| \EOo  | pFkey15        | pst100 pt200 | # <PF5>-CS      | (PT200)  |
| \EOE  | pFkey16        | pst100 pt200 | # <PF6>-U       | (PST100) |
| \EOt  | pFkey16        | pst100 pt200 | # <PF6>-S       | (PT200)  |
| \EOp  | pFkey16        | pst100 pt200 | # <PF6>-C       | (PST100) |
| \EOG  | pFkey16        | pst100 pt200 | # <PF6>-CS      | (PT200)  |
| \EOU  | pFkey17        | pst100 pt200 | # <PF7>-U       | (PST100) |
| \EOc  | pFkey17        | pst100 pt200 | # <PF7>-S       | (PT200)  |
| \EOq  | pFkey17        | pst100 pt200 | # <PF7>-C       | (PST100) |
| \EOH  | pFkey18        | pst100 pt200 | # <PF8>-U       | (PST100) |
| \EOV  | pFkey18        | pst100 pt200 | # <PF8>-S       | (PT200)  |
| \EOD  | pFkey18        | pst100 pt200 | # <PF8>-C       | (PST100) |
| \EOR  | pFkey18        | pst100 pt200 | # <PF8>-CS      | (PT200)  |
| \EOI  | pFkey19        | pst100 pt200 | # <PF9>-U       | (PST100) |
| \EOW  | pFkey19        | pst100 pt200 | # <PF9>-S       | (PT200)  |
| \EOe  | pFkey19        | pst100 pt200 | # <PF9>-C       | (PST100) |
| \EOS  | pFkey19        | pst100 pt200 | # <PF9>-CS      | (PT200)  |
| \EOJ  | pFkey20        | pst100 pt200 | # <PF10>-U      | (PST100) |
| \EOX  | pFkey20        | pst100 pt200 | # <PF10>-S      | (PT200)  |
| \EOf  | pFkey20        | pst100 pt200 | # <PF10>-C      | (PST100) |
| \EOE  | pFkey20        | pst100 pt200 | # <PF10>-CS     | (PT200)  |
| \EOK  | pFkey21        | pst100 pt200 | # <PF11>-U      | (PST100) |
| \EOY  | pFkey21        | pst100 pt200 | # <PF11>-S      | (PT200)  |
| \EOg  | pFkey21        | pst100 pt200 | # <PF11>-C      | (PST100) |
| \EOu  | pFkey21        | pst100 pt200 | # <PF11>-CS     | (PT200)  |
| \EOL  | pFkey22        | pst100 pt200 | # <PF12>-U      | (PST100) |
| \EOZ  | pFkey22        | pst100 pt200 | # <PF12>-S      | (PT200)  |
| \EOh  | pFkey22        | pst100 pt200 | # <PF12>-C      | (PST100) |
| \EOv  | pFkey22        | pst100 pt200 | # <PF12>-CS     | (PT200)  |
| \EOM  | pFkey9         | pst100       | # <PF13>-U      | (PST100) |
| \EOM  | pFkey9         | pt200        | # <PF9>-U       | (PT200)  |
| \EOI  | pFkey9         | pst100       | # <PF13>-S      | (PST100) |
| \EOI  | pFkey9         | pt200        | # <PF9>-S       | (PT200)  |
| \BO1  | pFkey9         | pst100       | # <PF13>-C      | (PST100) |
| \BO1  | pFkey9         | pt200        | # <PF9>-C       | (PT200)  |
| \EOW  | pFkey9         | pst100       | # <PF13>-CS     | (PT200)  |
| \EOW  | pFkey9         | pt200        | # <PF9>-CS      | (PT200)  |
| \EOw  | pFkey9         | pt200        | # <PF14>-U      | (PST100) |
| \EON  | pFkey10        | pst100       | # <PF14>-S      | (PT200)  |
| \EON  | pFkey10        | pt200        | # <PF10>-U      | (PT200)  |
| \EO\  | pFkey10        | pst100       | # <PF14>-C      | (PST100) |
| \EO\  | pFkey10        | pt200        | # <PF10>-S      | (PT200)  |
| \EOj  | pFkey10        | pst100       | # <PF14>-CS     | (PT200)  |
| \EOj  | pFkey10        | pt200        | # <PF10>-C      | (PT200)  |
| \EOx  | pFkey10        | pst100       | # <PF14>-CS     | (PST100) |
| \EOx  | pFkey10        | pt200        | # <PF10>-CS     | (PT200)  |
| # ^?  | rubout_char    | pst100       | # <DEL>-ALL     | (PST100) |
| # ^?  | rubout_char    | pt200        | # <BACKSP>-C/CS | (PT200)  |
| # ^Q  | (flow-control) | pst100       | # <PAUSE>-ALL   | (PST100) |
| # ^Q  | (flow-control) | pt200        | # <STOP>-U      | (PT200)  |
| # ^S  | (flow-control) | pst100       | # <PAUSE>-ALL   | (PST100) |

```

# ^S      (flow-control)    pt200      # <STOP>-U      (PT200)
# (internal-to-term)       pst100     # <STOP>-S/C/CS (PST100)
# (internal-to-term)       pt200     # <STOP>-CS      (PT200)
# (internal-to-term)       pst100     # <DELETE>-C/CS (PST100)
# (internal-to-term)       pst100     # <AUX SND>-CS  (PT200)
# (internal-to-term)       pt200     # <PRT SCN>-CS  (PT200)
# (internal-to-term)       pst100     # <CONFIG>-U/S/CS (PST100)
# (internal-to-term)       pst100     # <MENU>-S/C/CS (PST100)
# (internal-to-term)       pst100     # <RESET>-S/CS  (PST100)
# (internal-to-term)       pt200     # <CLEAR>-S/CS  (PT200)
# (internal-to-term)       pst100|pt200 # <ENTER>-C/CS
# (internal-to-term)       pt200     # <SETUP>-ALL   (PT200)
# (internal-to-term)       pt200     # <CHNG MODE>-ALL (PT200)

```

# PRIMOS Commands Reference Guide

```

#####
###
### Pr45 bindings:
#####
# ~C submit pt45|dm30|b150 # <ENTER>-ALL (DIP switch selectable)
# ~D submit pt45|dm30|b150 # <ENTER>-ALL (DIP switch selectable)
# ~H rubout_char pt45|dm30|b150 # <BACKSP>-ALL
# ~I expand_wild pt45|dm30|b150 # <TAB>-ALL
# ~J submit pt45|dm30|b150 # <ENTER>-ALL
# ~W submt pt45|dm30|b150 # <RETURN>-ALL
# ~E (Prefix) pt45|dm30|b150 # <SEND>-ALL #
# ~E! echo_raw pt45|dm30|b150 # <PAGE>-ALL Page Mode
# ~E# echo_raw pt45|dm30|b150 # <PAGE>-ALL Line Mode
# ~E( echo_raw pt45|dm30|b150 # <FORMS>-C/CS Forms Mode On
# ~E* echo_raw pt45|dm30|b150 # <AUX ON>-U/S AUX port on
# ~E, echo_raw pt45|dm30|b150 # <AUX ON>-ALL AUX/Parallel port off
# ~E- echo_raw pt45|dm30|b150 # <CE>-ALL Monitor Mode Off
# ~E.0 echo_raw pt45|dm30|b150 # <AUX ON>-C/CS Clear Entry [FORMS]
# ~E.1 echo_raw pt45|dm30|b150 # <SCRL UP>-U/S Parallel port on
# ~E.2 echo_raw pt45|dm30|b150 # <SCRL UP>-C/CS Scroll Up
# ~E.3 echo_raw pt45|dm30|b150 # <SCRL DOWN>-U/S Next Page
# ~E.4 echo_raw pt45|dm30|b150 # <SCRL DOWN>-C/CS Scroll Down
# ~E.5 echo_raw pt45|dm30|b150 # <PRINT>-U/S Previous Page
# ~E.5 echo_raw pt45|dm30|b150 # <PRINT>-C/CS Parallel Print
# ~E.6 echo_raw pt45|dm30|b150 # <E-AUX>-C/CS Parallel Enable
# ~E0 echo_raw pt45|dm30|b150 # <HOME>-C/CS Parallel Enable
# ~E: echo_raw pt45|dm30|b150 # <ASEND>-ALL Absolute Home
# ~E: echo_raw pt45|dm30|b150 # Monitor Mode On
# ~E: unbound pt45|dm30|b150 # CPU Message Deposit
# ~E> unbound pt45|dm30|b150 # <B-TAB>-ALL Back Tab
# ~E@ unbound pt45|dm30|b150 # <I CHAR>-ALL Line Insert Mode Off
# ~EA prev_line pt45|dm30|b150 # Cursor Up
# ~EB next_line pt45|dm30|b150 # Cursor Down
# ~EC forward_char pt45|dm30|b150 # Cursor Right
# ~ED back_char pt45|dm30|b150 # Cursor Left
# ~EE echo_raw pt45|dm30|b150 # CLEAR EOP/EOF>-C/CS Clear Screen
# ~EH echo_raw pt45|dm30|b150 # <HOME>-U/S Home
# ~EJ kill_line pt45|dm30|b150 # <CLEAR EOP/EOF>-U/S Clear to End of Page
# ~EK kill_line pt45|dm30|b150 # <I LINE>-ALL Clear to End of Line
# ~EL echo_raw pt45|dm30|b150 # <I LINE>-ALL Insert Line
# ~EM echo_raw pt45|dm30|b150 # <D LINE>-ALL Delete Line
# ~EN echo_raw pt45|dm30|b150 # <LOCAL>-ALL Off Line
# ~EP delete_char pt45|dm30|b150 # Line Delete Character
# ~EO unbound pt45|dm30|b150 # <I CHAR>-U/S Line Insert Mode On
# ~EV unbound pt45|dm30|b150 # Reset
# ~EW unbound pt45|dm30|b150 # <FORMS>-U/S Forms Mode On
# ~EX unbound pt45|dm30|b150 # <FORMS>-ALL Forms Mode Off
# ~E' delete_char pt45|dm30|b150 # Cursor Display On/Off
# ~EA unbound pt45|dm30|b150 # <A-SET>-U/S Page Delete Character
# ~ED unbound pt45|dm30|b150 # <A-SET>-C/CS Page Insert Mode On
# ~EE unbound pt45|dm30|b150 # Attribute Set
# ~EG unbound pt45|dm30|b150 # <M-LOCK>-ALL Memory Lock Clear
# ~EH unbound pt45|dm30|b150 # <M-LOCK>-ALL Memory Lock On
# ~EJ echo_raw pt45|dm30|b150 # <E-AUX>-U/S Memory Lock Off
# ~EK echo_raw pt45|dm30|b150 # <E-AUX>-ALL AUX enable
# ~EL unbound pt45|dm30|b150 # AUX/Parallel disable
# ~EM unbound pt45|dm30|b150 # Start Blink
# ~EN echo_raw pt45|dm30|b150 # Normal Video
# ~EB back_char On Line
# ~EP pfkey1 f1

```

```

^B\Eq      pfkey2          pt45|dm30|b150    # <F2>-U/S          f2
^B\Er      pfkey3          pt45|dm30|b150    # <F3>-U/S          f3
^B\Es      pfkey4          pt45|dm30|b150    # <F4>-U/S          f4
^B\Et      pfkey5          pt45|dm30|b150    # <F5>-U/S          f5
^B\Eu      pfkey6          pt45|dm30|b150    # <F6>-U/S          f6
^B\Ev      pfkey7          pt45|dm30|b150    # <F7>-U/S          f7
^B\Ew      pfkey8          pt45|dm30|b150    # <F8>-U/S          f8
^B\Ex      pfkey9          pt45|dm30|b150    # <F9>-U/S          f9
^B\Ey      pfkey10         pt45|dm30|b150    # <F10>-U/S         f10
^B\Ez      pfkey11         pt45|dm30|b150    # <F11>-U/S         f11
^B\E{      pfkey12         pt45|dm30|b150    # <F12>-U/S         f12
^B\E|      pfkey13         pt45|dm30|b150    # <F13>-U/S         f13
^B\E}      pfkey14         pt45|dm30|b150    # <F14>-U/S         f14
^B\E~      pfkey15         pt45|dm30|b150    # <F15>-U/S         f15
^B\E^?     pfkey16         pt45|dm30|b150    # <F16>-U/S         f16
^B^Np      pfkey17         pt45|dm30|b150    # <F1>-C/CS         f1 control
^B^Nq      pfkey18         pt45|dm30|b150    # <F2>-C/CS         f2 control
^B^Nr      pfkey19         pt45|dm30|b150    # <F3>-C/CS         f3 control
^B^Ns      pfkey20         pt45|dm30|b150    # <F4>-C/CS         f4 control
^B^Nt      pfkey21         pt45|dm30|b150    # <F5>-C/CS         f5 control
^B^Nu      pfkey22         pt45|dm30|b150    # <F6>-C/CS         f6 control
^B^Nv      pfkey23         pt45|dm30|b150    # <F7>-C/CS         f7 control
^B^Nw      pfkey24         pt45|dm30|b150    # <F8>-C/CS         f8 control
^B^Nx      pfkey25         pt45|dm30|b150    # <F9>-C/CS         f9 control
^B^Ny      pfkey26         pt45|dm30|b150    # <F10>-C/CS        f10 control
^B^Nz      pfkey27         pt45|dm30|b150    # <F11>-C/CS        f11 control
^B^N{      pfkey28         pt45|dm30|b150    # <F12>-C/CS        f12 control
^B^N|      pfkey29         pt45|dm30|b150    # <F13>-C/CS        f13 control
^B^N}      pfkey30         pt45|dm30|b150    # <F14>-C/CS        f14 control
^B^N~      pfkey31         pt45|dm30|b150    # <F15>-C/CS        f15 control
^B^N^?     pfkey32         pt45|dm30|b150    # <F16>-C/CS        f16 control
# (ignored-by-Primos) pt45|dm30|b150    # <LF>-ALL
# (internal-to-term)  pt45|dm30|b150    # <ESC>-C/CS
# (internal-to-term)  pt45|dm30|b150    # <CLEAR>-U/S       No operation
# (internal-to-term)  pt45|dm30|b150    # <RESET>-ALL       Internal Reset

```

## Compiling, Saving, and Restoring the Bindings File

When you have finished setting up your bindings file, you must compile it to bind each sequence to the specified command. However, compiling the file makes those bindings effective for the current terminal session only. You have to recompile them the next time you want to use them. Instead, if you save the compiled version of the file, then you can restore the compiled version the next time you log in. This is faster than recompiling the bindings file again.

The compiled file is made up of two parts: the compiled bindings and the terminal capabilities. The bindings are the actual sequences bound to their corresponding commands. The capabilities are the characteristics of the particular type of terminal; baud rate, screen width, screen length, and so forth.

### Compiling the Bindings File

To compile the bindings file, use the `-BIND_TERM` option in the following format:

```
ECL -TERMINAL_TYPE terminal -BIND_TERM [bindings_file] [term_alias]
```



## Saving and Restoring Simultaneously

You can use both `-STERM` and `-RTERM` on the same command line. If you do so, the command line variable `compiled_file` is optional for `-SAVE_TERM`. `-STERM` and `-RTERM` together are used in conjunction with the `-BPS`, `-TERMINAL_TYPE`, and `-BIND_TERM` options, if you want to specify all these options simultaneously, as in the following example. (`-BPS` sets the terminal baud rate.)

```
OK, ECL -RTERM -BPS115200 -BIND_TERM /usr/lib/terminfo/sterm -SAVE_TERM /usr/lib/terminfo/sterm
Restored PT200 terminal capabilities and bindings (9600 baud). (EDIT_CMD_LINE)
OK,
```

ECL processes the above command line as follows:

1. If you specify `-RTERM`, ECL attempts to restore the compiled terminal characteristics and bindings in *compiled\_file*. *compiled\_file* contains the terminal type and baud rate with which it was compiled, and ECL checks that type and baud rate agree with those specified by `-TTP` and `-BPS` on the command line *before* completing the restore operation.
2. If the restore operation succeeds, ECL does not establish any more bindings by processing the `-BIND_TERM` option. In other words, the bindings in the existing compiled file are established, not those in the file specified by `-BIND_TERM` on the command line. Also, ECL ignores any `-STERM` option.
3. If the restore operation fails, either because *compiled\_file* does not exist or *compiled\_file*'s terminal type and baud rate do not match those specified on the command line, then terminal capabilities and bindings are compiled normally as if the restore operation were never requested.
4. If `-STERM` was also specified on the command line, the capabilities and bindings are saved in *compiled\_file*. This allows you, on one command line, to restore compiled bindings for your terminal and (if they haven't yet been compiled) to automatically compile them, so that the restore operation will succeed the next time.

---

### Note

If you change your terminal capability or bindings files, ECL notes that the compiled file is out-of-date and recompiles the out-of-date portion automatically. This requires that `-BTERM` be enabled (to verify the bindings portion of the compiled file) and that `-TTP` be enabled (to verify the capabilities portion of the compiled file) or `-BPS` (which implies `-TTP`).

---

## What Are the Current Bindings?

You can use several methods to find out what a particular key sends, that is, what action a particular key is bound to.



Table 6-2. PST100 and PT200 Function Key Sequences

| Key                                              | Unaugmented | Augmented with |      |            |
|--------------------------------------------------|-------------|----------------|------|------------|
|                                                  |             | Shift          | Ctrl | Ctrl-Shift |
| F1                                               | \EO!        | \EO)           | \EO1 | \EO9       |
| F2                                               | \EO"        | \EO*           | \EO2 | \EO:       |
| F3                                               | \EO#        | \EO+           | \EO3 | \EO;       |
| F4                                               | \EO\$       | \EO,           | \EO4 | \EO<       |
| F5                                               | \EO&        | \EO-           | \EO5 | \EO=       |
| F6                                               | \EO%        | \EO.           | \EO6 | \EO>       |
| F7                                               | \EO'        | \EO/           | \EO7 | \EO?       |
| F8                                               | \EO(        | \EO0           | \EO8 | \EO@       |
| PF1                                              | \EOA        | \EOO           | \EOJ | \EOk       |
| PF2                                              | \EOB        | \EOP           | \EOV | \EOl       |
| PF3                                              | \EOC        | \EOQ           | \EO_ | \EOm       |
| PF4                                              | \EOD        | \EOR           | \EO' | \EOn       |
| PF5                                              | \EOE        | \EOS           | \EOa | \EOo       |
| PF6                                              | \EOF        | \EOT           | \EOb | \EOp       |
| PF7                                              | \EOG        | \EOU           | \EOc | \EOq       |
| PF8                                              | \EOH        | \EOV           | \EOd | \EOr       |
| PF9                                              | \EOI        | \EOW           | \EOe | \EOs       |
| PF10                                             | \EOJ        | \EOX           | \EOf | \EOt       |
| PF11                                             | \EOK        | \EOY           | \EOg | \EOu       |
| PF12                                             | \EOL        | \EOZ           | \EOh | \EOv       |
| <i>The next two keys appear on PST100s only.</i> |             |                |      |            |
| PF13                                             | \EOM        | \EO[           | \EOi | \EOw       |
| PF14                                             | \EON        | \EO\           | \EOj | \EOx       |
| F9                                               | \EOM        | \EO[           | \EOi | \EOw       |
| F10                                              | \EON        | \EO\           | \EOj | \EOx       |
| PA1                                              | \EN!        | \EN%           | \EN) | \EN-       |
| PA2                                              | \EN"        | \EN&           | \EN* | \EN.       |
| PA3                                              | \EN#        | \EN'           | \EN+ | \EN/       |
| PA4                                              | \EN\$       | \EN(           | \EN, | \EN0       |



Table 6-3. PT45 Function Key Sequences

| Key | Unaugmented | Augmented with Ctrl |
|-----|-------------|---------------------|
| F1  | ^B\Ep       | ^B^Np               |
| F2  | ^B\Eq       | ^B^Nq               |
| F3  | ^B\Er       | ^B^Nr               |
| F4  | ^B\Es       | ^B^Ns               |
| F5  | ^B\Et       | ^B^Nt               |
| F6  | ^B\Eu       | ^B^Nu               |
| F7  | ^B\Ev       | ^B^Nv               |
| F8  | ^B\Ew       | ^B^Nw               |
| F9  | ^B\Ex       | ^B^Nx               |
| F10 | ^B\Ey       | ^B^Ny               |
| F11 | ^B\Ez       | ^B^Nz               |
| F12 | ^B\E{       | ^B^N{               |
| F13 | ^B\E        | ^B^N                |
| F14 | ^B\E}       | ^B^N}               |
| F15 | ^B\E~       | ^B^N~               |
| F16 | ^B\E^?      | ^B^N^?              |

### Display Manager

The ECL display manager, as the term suggests, controls the manner in which terminal output is displayed on the screen. The display manager efficiently maintains an accurate display of the command line, even if it wraps to more than one line on the screen. The display manager also optimizes cursor movement and does not update the screen until type ahead has been processed. Thus, inserting characters at the beginning of a long command line is relatively quick, even across a network connection.

The display manager is controlled by the ECL\$LIB subroutine, which replaces the PRIMOS CL\$GET routine when ECL is initialized. Also, in order that ECL be able to use different terminals, the display manager also references a database called TERMINFO (discussed below) to get information about the capabilities of a particular terminal.

### Terminal Requirements for the Display Manager

Here are the minimum requirements for your terminal in order for the display manager to properly operate:

- A character placed in the last column of the screen must wrap the cursor to the first column of the next line. (If not, then ECL -WIDTH 0 must be specified.)
- A linefeed (^J) character, or a screen wrap off the bottom line of the display, must cause a new blank line to scroll on the screen.

- A backspace (^H) character must move the cursor one position to the left without erasing anything.
- A backspace (^H) must wrap from the first column on a line to the last column of the previous line.

### **Other Requirements**

ECL assumes, for a generic terminal, a default terminal display of 24 lines by 80 columns (although you can change this with ECL's `-HEIGHT` and `-WIDTH` command line options, or with a `TERMINFO` entry). The display manager normally requires that ECL perform its own character echoing (as EMACS does), rather than PRIMOS.

### ***Specifying Terminal Type***

As mentioned above, the display manager references a database called `TERMINFO` to get information about the capabilities of a particular terminal. But first, you must tell ECL what type of terminal you are using. You can do this in two ways:

1. Using the `-TERMINAL_TYPE [type]` command-line option. This overrides any types defined in the global variable `.TERMINAL_TYPE$`.
2. Using the global variable `.TERMINAL_TYPE$` (the same variable used by EMACS). ECL checks this variable whenever `-TERMINAL_TYPE` is invoked with no argument.

If ECL cannot find an appropriate entry in `TERMINFO` for the specified terminal, or there is some other error, ECL defaults to its generic terminal capabilities. Once terminal capabilities and bindings are established for a particular terminal, you can then save and restore them in compiled form using ECL's `-SAVE_TERM` and `-RESTORE_TERM` options. (See the section `ECL Bindings` earlier in this chapter.)

### ***Prompt Handling***

In order for ECL's display manager to properly optimize its display operations, it must know about the prompt.

By default, ECL recognizes the standard PRIMOS `RDY` prompts, displaying their brief or long forms as appropriate. ECL also has its own internal set of brief-form prompts independent of those specified by the `RDY` command; these default to `OK`, `ER!`, and `OK`, for the ready, error, and warning brief prompts, respectively. You specify these prompts with the ECL command-line options `-RB`, `-EB`, and `-WB`. Using these options causes ECL to use its own set in place of the PRIMOS set; long-form prompts are unaffected.

The advantage of using ECL's own set of prompts is that the number sign (`#`) character, with an ECL prompt, acts as a placeholder. `#` instructs ECL to substitute the current command history entry number in place of the `#` character when



---

**Note**

If the number you specify does not accurately reflect the number of characters displayed by the prompt, the prompt may appear to work correctly on some terminals. However, problems may occur when moving through the history stack, or when editing long command lines that wrap to more than one line in the screen.

---

**The TERMINFO Database**

The TERMINFO database is a text file that contains descriptions called entries. Each of these entries lists a set of capabilities for a terminal. Entries are compiled to produce individual files each describing a single terminal. The format and capabilities of Prime TERMINFO are documented fully in the AT&T™ System V (and PRIMIX Version 3.0) edition of the *Unix™ Programmer's Manual*.

TERMINFO itself resides in the top-level directory TERM\*. Your System Administrator determines if you are using the TERMINFO database or an alternative database.

---

**Note**

This TERMINFO database is in the public domain and is made available to the user by courtesy of Prime Computer, Inc. Prime makes no representations or warranties whatsoever regarding this database, or the ability of any Prime software, when combined with this database, to operate on any terminals other than Prime terminals. Prime also disclaims any obligations to maintain or support this database or any similar database now or in the future.

---

Table 6-4 shows how the ECL display manager uses the following terminal capabilities.

Table 6-4. ECL TERMINFO Capabilities

| <i>Name</i>  | <i>Type</i> | <i>Full Name</i>             | <i>Function</i>                           |
|--------------|-------------|------------------------------|-------------------------------------------|
| <b>am</b>    | bool        | <i>auto_right_margin</i>     | Wraps at right margin                     |
| <b>bel</b>   | str         | <i>bell</i>                  | Rings bell                                |
| <b>bw</b>    | bool        | <i>auto_left_margin</i>      | Wraps at left margin                      |
| <b>cols</b>  | \$          | <i>columns</i>               | Number of columns in a line               |
| <b>cr</b>    | str         | <i>carriage_return</i>       | Carriage return                           |
| <b>cub1</b>  | str         | <i>cursor_left</i>           | Moves cursor left one space               |
| <b>cubw*</b> | str         | <i>cursor_left_wrap</i>      | Wraps cursor to the left                  |
| <b>cud1</b>  | str         | <i>cursor_down</i>           | Moves cursor down one line                |
| <b>cuf1</b>  | str         | <i>cursor_right</i>          | Moves cursor right one space              |
| <b>cuu1</b>  | str         | <i>cursor_up</i>             | Moves cursor up one line                  |
| <b>dcd1*</b> | str         | <i>delete_char_display</i>   | Deletes one character to end of display   |
| <b>dch1</b>  | str         | <i>delete_char</i>           | Deletes one character                     |
| <b>ed</b>    | str         | <i>clr_eos</i>               | Clears to end of display                  |
| <b>el</b>    | str         | <i>clr_eol</i>               | Clears to end of line                     |
| <b>ht</b>    | str         | <i>tab</i>                   | Tabs to next column                       |
| <b>idd*</b>  | str         | <i>insert_delete_display</i> | Insert and delete in display mode         |
| <b>idde*</b> | str         | <i>insert_delete_line</i>    | Insert and delete in line mode            |
| <b>in</b>    | bool        | <i>insert_null_glitch</i>    | Distinguishes nulls in insert mode        |
| <b>ind</b>   | str         | <i>scroll_forward</i>        | Scrolls forward                           |
| <b>ip</b>    | str         | <i>insert_padding</i>        | Insert delay after an inserted character  |
| <b>ipd*</b>  | str         | <i>insert_eos_padding</i>    | Insert delay for insert to end of display |
| <b>it</b>    | #           | <i>init_tabs</i>             | Tabs exist every # spaces                 |
| <b>lines</b> | #           | <i>lines</i>                 | Number of lines on a screen or page       |
| <b>lis1*</b> | str         | <i>line_oriented_init1</i>   | Initialization string for ECL             |
| <b>mir</b>   | bool        | <i>move_insert_mode</i>      | Allows movement in insert mode            |
| <b>pad</b>   | str         | <i>pad_char</i>              | Pad delay character (rather than null)    |
| <b>rlom*</b> | str         | <i>line_oriented_end</i>     | Exits line-oriented (ECL) mode            |
| <b>rmir</b>  | str         | <i>exit_insert_mode</i>      | Exits insert mode                         |
| <b>slo*</b>  | str         | <i>line_oriented_start</i>   | Enters line-oriented (ECL) mode           |
| <b>smid*</b> | str         | <i>enter_insert_display</i>  | Enters insert-in-display mode             |
| <b>smir</b>  | str         | <i>enter_insert_mode</i>     | Enters insert or insert-in-line mode      |
| <b>xenl</b>  | bool        | <i>eat_newline_glitch</i>    | Ignores newline after 80 columns          |
| <b>xt</b>    | bool        | <i>telera_y_glitch</i>       | Has Teleray tabs                          |

\*Non-standard TERMINFO extensions for Prime ECL.

## **Using TERMINFO**

The display manager makes certain assumptions about several terminal capabilities. It assumes that

- **am** is specified (true)
- **bel** is an ASCII BEL (CtrlG)
- **cr** is an ASCII CR (CtrlM)
- **cub1** is an ASCII BS (CtrlH)
- **cud1** is an ASCII LF (CtrlJ)
- **ht** is an ASCII HT (CtrlI)
- **in** is not specified (false)
- **ind** is an ASCII LF (CtrlJ)
- **xt** is not specified (false)

If these assumptions are contradicted by the capabilities specified, ECL produces the following warning message:

```
Note: xxx terminal may not be properly supported. (EDIT_CMD_LINE)
```

The **lines** and **cols** capabilities may be overridden by ECL's **-HEIGHT** and **-WIDTH** options, respectively.

## **Backspace and Wrapping**

If the terminal has **bw**, then **cub1** is used as a backspace that wraps backward from the first column to the last column of the previous line; otherwise, specify **cubw**. The display manager uses **cuu1** to move up one line if neither **bw** nor **cubw** have been specified.

TERMINFO guidelines state that **cuf1** may not necessarily wrap from the end of a line to the beginning of the next with **am** specified; however, ECL assumes this is possible.

## **Using the Clearing Capabilities**

If your terminal supports both **ed** and **el**, it is best to specify both, because **ed** is usually slower and the display handler uses the faster of the two. The TERMINFO guidelines state that **ed** is only defined from the first column of a line, but ECL assumes it is valid from any column.



version command). In the latter case, the display manager restores **slo**m when resumed, just before redisplaying the command line.

### **Disabling a Linefeed**

The `xenl` capability indicates that the terminal ignores a newline (linefeed) after wrapping to a new line. Certain terminals exhibit this behavior.

## ***ECL Commands Reference***

The commands documented in this section, also called fundamental bindings, are ordered alphabetically by command name. Each command name is accompanied on the same line by the appropriate control or escape sequence. In the discussion of each command, any reference to numeric argument *n* refers to the `esc_digit` command used immediately before the listed command.

### ***abort\_cmd* Ctrl-G**

Aborts your last input command. `abort_cmd` sets the return error code to -1 and returns the null string to the caller; this is useful when, for example, you wish to abort from a forward search. When used at command level (at the ECL prompt), `abort_cmd` always changes your position in the stack history to the end; that is, one position after the last command you issued. The numeric argument is not applicable and is ignored.

### ***back\_char* Ctrl-B**

Moves the cursor back *n* characters in the command line.

### ***back\_word* Esc B**

Moves the cursor back *n* words, leaving the cursor positioned on the first character of that word. A word consists only of alphanumeric and underscore characters. The `-ENTRY` option to ECL modifies this definition to include all characters possible in a PRIMOS entryname, including wildcards. For more information about the `-ENTRY` option, see the discussion on the ECL command in Chapter 2.

### ***begin\_line* Ctrl-A**

Moves the cursor to the beginning of the command line (the first character position after the prompt). The numeric argument is ignored.

### ***collect\_macro* Esc (**

Starts collecting keystrokes into a macro program that, once completed with the `finish_macro` (or `submit`) command, can be reexecuted with the `execute_macro` command. Keyboard macros have the ability to collect responses to prompts like those generated by the search commands. Only one keyboard macro, limited to 200 keystrokes, can be programmed at a time. The bell rings if a keyboard macro





***end\_line* Ctrl-E**

Moves the cursor to the end of the command line. The numerical argument is ignored.

***esc\_digit* Esc *n***

Specifies how many times the next ECL command is to be executed. The numeric argument *n* defaults to one (1) for each command. *esc\_digit* is a two-part sequence. First, press the Esc key. Second, press the decimal digit or digits desired. If you enter more than one decimal digit, they continue the count specification; for example, Esc 24 specifies a count of 24. (In this case, the next ECL command you entered would be executed 24 times.) This command is similar to the *multiplier* command, discussed later in this section.

***exchange\_mark* Ctrl-X Ctrl-X**

Exchanges the cursor position with the mark position (see the *mark* command). The actual text region delimited by these two positions remains unchanged.

***execute\_macro* Esc E**

Executes the most recent keyboard macro you defined by using the *collect\_macro* and *finish\_macro* commands. You can execute the previous keyboard macro while collecting a new one in order to include it within the new macro program. The bell rings if the keyboard macro program is null. To view the contents of the keyboard macro program, use the *explain\_key* command for a sequence bound to *execute\_macro*. The numeric argument, *n*, is applied only to the first command within the keyboard macro program.

***expand\_abbrev* Esc A**

Replaces a token with its abbreviation expansion. (The cursor must be within the token.) Any PRIMOS global variable references are also expanded; to prevent this, use a numeric argument (*n*) of 0. If *n* is 4, the entire command line (not beginning with ~) is expanded for abbreviations. If *n* is 16, the entire command line is expanded for global variables and abbreviations. In all of the above cases, the mark position is set at the beginning of the expansion and the cursor is placed at the end. (See the *mark* command later in this section.) Be aware that the token or command line replaced by the expansion is not implicitly saved in a kill buffer. Tokens are delimited by any characters in the set > , ; [ ] ( ).

***expand\_wild* Ctrl-I**

Performs an automatic pathname completion on a partially completed pathname. (The cursor must be within the partially completed pathname.) The completion occurs as if you had used wildcards at the end of the partial pathname; however, ECL's -WILD\_TAIL option makes wildcarding occur at the current cursor position within the pathname. If *n* is 1, the partially completed pathname is expanded as far as possible without the pathname becoming ambiguous. If this is already the case, or if *n* is 4, then the menu list of possible completions is displayed without affecting the partially completed pathname in the command line. You can make ECL process abbreviation expansion before pathname

completion by using the option `-WILD_ABBREV`. The mark position is always set to the beginning of the completed pathname (see the *mark* command). If the `-WILD_TAIL` option is in effect, you can restrict the display to only the directories that match the partial pathname by doing the following:

1. Add a `>` to the end of the partial pathname
2. Use `Ctrl-B` to position the cursor on the `>`
3. Press `Ctrl-I` to request completion.

See the section Automatic Pathname Completion in this chapter for more information on pathname completion.

***expand\_wild\_menu*** `Esc Ctrl-I`

Selects the *n*th entry corresponding to the number in the displayed menu list of the possible pathname completions generated by the *expand\_wild* command. *expand\_wild\_menu* substitutes the entry in the numbered list for the partial pathname in the command line. This command is typically used immediately after the *expand\_wild* command.

***explain\_key*** `Ctrl-_`

Prompts for a key sequence and displays the command name to which that sequence is bound. This is useful for determining what terminal function keys are bound to, and for debugging binding-related problems. This command is never collected within a keyboard macro (see *collect\_macro*). The numeric argument, while ignored, is retained for the next command entered.

***extend\_command*** `Esc X`

Executes an ECL command at the `Command:prompt`. The command can be any command listed in this section. The use of the numeric argument depends on the specified command. Executing *extend\_command* twice reexecutes the most recent command rather than reprompting.

***finish\_macro*** `Esc )`

Completes a keyboard macro being collected with the *collect\_macro* command. The macro replaces the previous keyboard macro, if it exists, and can be executed with the *execute\_macro* command. The bell is rung if no keyboard macro was being collected when this command is issued. The numeric argument is ignored.

***forward\_char*** `Ctrl-F`

Moves the cursor forward *n* characters toward the end of the command line.

***forward\_search*** `Ctrl-S` or `Esc S`

Prompts for a string, and searches *n* times in the forward direction (toward the end of the command history) for that string in each command of the command history. If the string is found, the corresponding command becomes the current command with the cursor positioned on the first occurrence of the matching

string; if not, the bell rings. If the current command line is null, the search starts with the hidden command in the history; otherwise, the search ends with this hidden command. The search string may begin, end, or both begin and end with a quoted newline character ('^J'), which instructs ECL to search for lines beginning (and/or ending) with the search string. A newline character is quoted using the *quote* command followed by pressing Return on your terminal. The search facility maintains its own ring of search strings. This is analogous to the command history, but the search ring is accessible only from within a search prompt. Entering a null string in response to the search prompt instructs ECL to use the most recent search string. The most recent string is also used automatically when a search command is reexecuted with the *reexecute* command. By default, searching is not case-sensitive, but this can be changed by using the `-CASE_SEARCH` option.

***forward\_word* Esc F**

Moves the cursor forward one word (or *n* words if *esc\_digit* is used), leaving the cursor positioned after the last character of that word. See *back\_word* for the definition of a word.

***goto\_line* Esc G**

Moves the cursor directly to the entry in the command history specified by the numeric argument *n*. For example, Esc 5 Esc G moves the cursor to command history entry number 5. That entry then becomes the current command line being edited. If *n* is associated with the current command, the hidden command from the command history is recalled. An *n* value of 0 repositions to an empty command at the end of the command history (the top of the history stack). Use the *refresh* command to determine the entry number associated with commands in the history list. You can also specify (with arguments to the ECL `-RB/-EB/-WB` options) a command-level prompt displaying the current history entry number for each command. See the discussion about ECL in Chapter 2 for more details about setting up such a prompt.

***kill\_line* Ctrl-K**

Kills the remainder of the line from the cursor to the end of the command line. A numeric argument of 0 or greater than 1 instructs ECL to kill from the beginning of the line to the cursor. The deleted text is placed into a kill buffer for possible recall with the *yank* or *yank\_replace* command. If the cursor is positioned at the end of a command line when the command is issued, the next line in the command history is moved up and placed at the end of the current command line. This moves the subsequent entries in the command history up one line. This command allows you to delete commands from the command history.

***kill\_region* Ctrl-W**

Kills the contents of the region delimited by the cursor and mark positions (see the *mark* command). A numeric argument of 4 instructs *kill\_region* to kill the entire command line instead. The deleted text is placed into a kill buffer.



***pfkey1 – pfkey512*** [not bound]

Executes the program bound to the corresponding Programmable Function command (PFkey) sequence. The numeric argument, *n*, is effective only on the first command within the programmed sequence. If no sequence has been programmed, the terminal bell rings, indicating the PFkey is unprogrammed. Programming PFkeys is discussed in the section ECL Key Bindings.

***prev\_line*** Ctrl-Z

Moves the cursor one entry (or *n* entries if *esc\_digit* is used) toward the beginning of the command history. The command line saved there becomes the current command line being edited. Executing *prev\_line* with no numeric argument retrieves your last command. An argument of zero recalls the command at the current place within the command history hidden by the current command. Relative movement within the command history is affected by the ECL options *-STACK / -NO\_STACK* and *-STICK / -NO\_STICK*, which are discussed in Chapter 2.

***quote*** Ctrl-Q or Esc Q

Quotes the next character typed in (including Ctrl-P) for literal insertion one time (or *n* times if *esc\_digit* is used). This permits the insertion of characters into the command line that are normally bound to nonprintable commands. If there are many characters to quote, use the *do\_quote* command, discussed earlier in this section.

***reexecute*** Ctrl-C

Reexecutes the previous command one time (or *n* times if *esc\_digit* is used). If the previous command has a numeric argument associated with it, the two counts are multiplied together. *reexecute* itself is never considered the previous command; that is, reexecuting *reexecute* acts upon the command prior to the first reexecute.

***refresh*** Ctrl-L

Reinitializes the terminal and redisplay the current command line. Use *refresh* to clean up the terminal display or to reset your terminal. This command has a default numeric value of 1; if *n* is greater than 1, the previous *n-1* commands in the history are displayed before the current command line is redisplayed. A numeric value of zero reveals the current history event number and any history command hidden by the current command line.

***reverse\_search*** Ctrl-R or Esc R

Prompts for a string, and searches one time (or *n* times if *esc\_digit* is used) in the reverse direction (toward the beginning) for that string in the command history. If found, the corresponding command becomes the current command with the cursor positioned on the first match found, otherwise the bell rings. *reverse\_search* always starts with the previous command in the history and includes the hidden command last. See the *forward\_search* command for more information on searching.



***upcase\_word*** Esc U

Moves the cursor forward *n* words while converting all lowercase letters to their uppercase equivalents. This is the converse of the *downcase\_word* command. See *back\_word* earlier in this section for the definition of a word.

***version*** Ctrl-V

Displays the current version number for ECL\$LIB.

***yank*** Ctrl-Y

Yanks back the text saved in the previous kill buffer *n*, inserting it into the command line at the current position. If *n* is 1, the current kill buffer, containing the text most recently deleted, is yanked. The mark position is always set to the beginning of the yanked text (see the *mark* command earlier in this section).

***yank\_replace*** Esc Y

Functions like the *yank* command except that most recently yanked text is replaced by the text of the indicated kill buffer. This permits you to search the kill ring for the deleted text of interest. The mark position is always set to the beginning of the yanked text (see the *mark* command earlier in this section).

## ECL Commands by Function

This section divides the ECL commands into functional groups. An example illustrates the use of each command. The series of examples for each functional group forms a terminal session, with each command acting upon the result of the previous command line. The cursor position is denoted by an underscore ( ). At the beginning of the session, you are examining the contents of the BAR subdirectory. However, you have mistyped the string BAR on the command line. At this point, the command line looks like this:

```
LD *>ABR>@@_
```

Now go on to the following sections for a running example of the uses of ECL.

### ***Cursor Movement***

This section illustrates the use of the cursor movement commands.

| <i>Command</i>    | <i>Keystroke</i> | <i>Result</i>         | <i>Comment</i>     |
|-------------------|------------------|-----------------------|--------------------|
| <i>begin_line</i> | Ctrl-A           | <u>LD</u> *>ABR>@@    |                    |
| <i>end_line</i>   | Ctrl-E           | LD *>ABR>@@ <u> </u>  |                    |
| <i>back_char</i>  | Esc-7-Ctrl-B     | LD *>ABR> <u> </u> @@ | (Executed 7 times) |



|                     |        |                      |
|---------------------|--------|----------------------|
| <i>forward_char</i> | Ctrl-F | LD *> <u>ABR</u> >@@ |
| <i>forward_word</i> | Esc-F  | LD *> <u>ABR</u> >@@ |
| <i>back_word</i>    | Esc-B  | LD *> <u>ABR</u> >@@ |

### Insertion

ECL causes all printable characters to insert themselves into the command line. Non-printable characters can also be inserted.

| <i>Command</i>     | <i>Syntax</i> | <i>Keystroke</i> | <i>Result</i>              |
|--------------------|---------------|------------------|----------------------------|
| <i>self_insert</i> |               | AB               | LD *> <u>ABABR</u> >@@     |
| <i>quote</i>       | Esc-Q         | Esc-QCtrl-C      | LD *> <u>AB^CABR</u> >@@   |
| <i>do_quote</i>    | Esc-'Esc-'    | Esc-'Ctrl-D      | LD *> <u>AB^C^DABR</u> >@@ |

### Deletion

You can delete by characters, words, lines, or regions.

| <i>Command</i>     | <i>Keystroke</i> | <i>Result</i>             | <i>Comment</i> |
|--------------------|------------------|---------------------------|----------------|
| <i>delete_char</i> | Ctrl-D           | LD *> <u>AB^C^DBR</u> >@@ |                |
| <i>rubout_char</i> | Ctrl-H           | LD *> <u>AB^CBR</u> >@@   | (PRIMOS erase) |
| <i>delete_word</i> | Esc-D            | LD *> <u>AB^C</u> >@@     |                |
| <i>rubout_word</i> | Esc-Ctrl-H       | LD *>>@@                  |                |
| <i>kill_line</i>   | Ctrl-K           | LD *>_                    |                |
| <i>kill_region</i> | Ctrl-UCtrl-W     | _                         | (Kills line)   |

### Deletion Recall

All deletion commands except *delete\_char* and *rubout\_char* save the deleted text in the kill ring. Most consecutive deletions are kept together in the same kill ring entry.

| <i>Command</i>      | <i>Keystroke</i> | <i>Result</i>            | <i>Comment</i>                                                              |
|---------------------|------------------|--------------------------|-----------------------------------------------------------------------------|
| <i>yank</i>         | Ctrl-Y           | LD *>_                   | (From <i>kill_region</i> above)                                             |
|                     | Ctrl-Y           | LD *> LD *>_             |                                                                             |
| <i>yank_replace</i> | Esc-Y            | LD *> <u>AB^CBR</u> >@@_ | (From <i>delete_word</i> , <i>rubout_word</i> , and <i>kill_line</i> above) |

## Regions

This section introduces the ECL commands that work specifically with regions. A *region* is a user-defined portion of text. The beginning of the region is called the *mark*, which is invisible on the terminal screen; the end of the region is called the *point*. (Often, the point is where the cursor rests.) In the examples used in this section, the mark is denoted by the red underscore.

| <i>Command</i>       | <i>Keystroke</i> | <i>Result</i>   | <i>Comment</i>             |
|----------------------|------------------|-----------------|----------------------------|
| <i>copy_region</i>   | Esc-W            | LD *>AB^CBR>@@_ |                            |
| <i>exchange_mark</i> | Ctrl-X-Ctrl-X    | LD *>AB^CBR>@@_ |                            |
| <i>kill_region</i>   | Ctrl-W           | LD *>_          |                            |
| <i>yank</i>          | Ctrl-Y           | LD *>AB^CBR>@@_ | (From <i>kill_region</i> ) |
| <i>yank_replace</i>  | Esc-Y            | LD *>AB^CBR>@@_ | (From <i>copy_region</i> ) |
| <i>mark</i>          | Ctrl-@           | LD *>AB^CBR>@@_ |                            |

## Repetition

This section introduces the ECL commands that allow you to reexecute commands or to execute commands any number of times.

| <i>Command</i>    | <i>Syntax</i>                                   | <i>Keystroke</i> | <i>Result</i>    |
|-------------------|-------------------------------------------------|------------------|------------------|
| <i>multiplier</i> | Ctrl-U<br>[ <i>digit(s)</i> ]<br><i>command</i> | Ctrl-UCtrl-B     | LD *>AB^CBR>@@   |
|                   |                                                 | Ctrl-U2X         | LD *>AB^CBXXR>@@ |
| <i>esc_digit</i>  | Esc-[ <i>digit(s)</i> ]<br><i>command</i>       | Esc-2Ctrl-H      | LD *>AB^CBR>@@   |
| <i>reexecute</i>  |                                                 | Ctrl-C           | LD *>ABR>@@      |

## Twiddle

This section illustrates the character position-switching ECL command.

| <i>Command</i> | <i>Keystroke</i> | <i>Result</i> |
|----------------|------------------|---------------|
| <i>twiddle</i> | Ctrl-T           | LD *>BAR>@@   |

The desired command line is ready for submittal, now that your session of experimenting with various ECL commands has been completed.

### **Submit and Abort**

This section illustrates the ECL commands that allow you to submit a command to ECL and to abort an ECL command in progress.

| <i>Command</i>   | <i>Keystroke</i> | <i>Comment</i>             |
|------------------|------------------|----------------------------|
| <i>submit</i>    | Ctrl-J           | (Typed as carriage return) |
| <i>abort_cmd</i> | Ctrl-G           | (Aborts operation)         |

### **Case Conversion**

This section illustrates the case conversion ECL commands. Another simple example begins:

| <i>Command</i>       | <i>Keystroke</i> | <i>Result</i>    |
|----------------------|------------------|------------------|
|                      |                  | <u>f</u> ILE.Bin |
| <i>toggle_case</i>   | Ctrl-Ctrl        | FILE.Bin         |
| <i>downcase_word</i> | Esc-L            | File_Bin         |
| <i>upcase_word</i>   | Esc-U            | File.BIN_        |

### **Pathname and Abbreviation Expansion**

There are basically three types of pathname expansion in addition to the menu-selection capability:

| <i>Command</i>          | <i>Keystroke</i>                 | <i>Comment</i>                           |
|-------------------------|----------------------------------|------------------------------------------|
| <i>expand_wild</i>      | Ctrl-I                           | (Completes if possible)                  |
|                         | Ctrl-U-Ctrl-I                    | (Displays matches)                       |
|                         | Ctrl-U-Ctrl-U-Ctrl-I             | (Expands verbatim to full pathname)      |
| <i>expand_wild_menu</i> | Esc- <i>number</i><br>Esc-Ctrl-I | (Selects menu-list entry <i>number</i> ) |

You can expand an abbreviation four ways:

|                      |                     |                                          |
|----------------------|---------------------|------------------------------------------|
| <i>expand_abbrev</i> | Esc-A               | (Abbrev/gvar-expand single token)        |
|                      | Esc-0-Esc-A         | (Abbrev-only-expand single token)        |
|                      | Ctrl-U-Esc-A        | (Abbrev-only-expand entire command line) |
|                      | Ctrl-U-Ctrl-U-Esc-A | (Abbrev/gvar expand entire command line) |

### **History and Refresh**

This section illustrates the ECL commands that allow you to manipulate the command history and to refresh the display.

| <i>Command</i>        | <i>Keystroke</i>   | <i>Comment</i>                         |
|-----------------------|--------------------|----------------------------------------|
| <i>refresh</i>        | Ctrl-L             | (Performs display cleanup)             |
|                       | Esc- <i>number</i> | (Displays last <i>number</i> commands) |
|                       | Ctrl-L             |                                        |
| <i>prev_line</i>      | Ctrl-Z             | (Recalls last command)                 |
| <i>next_line</i>      | Ctrl-N             | (Moves forward in history)             |
| <i>goto_line</i>      | Esc- <i>number</i> | (Recalls command <i>number</i> )       |
|                       | Esc-G              |                                        |
| <i>reverse_search</i> | Esc-R              | (Searches backward through history)    |
| <i>forward_search</i> | Esc-S              | (Searches forward through history)     |

### **Keyboard Macros**

This section lists the ECL commands that allow you to begin and end a macro, and to execute a macro.

| <i>Command</i>       | <i>Keystroke</i> | <i>Comment</i>                  |
|----------------------|------------------|---------------------------------|
| <i>collect_macro</i> | Esc-(            | (Starts keystroke collection)   |
| <i>finish_macro</i>  | Esc-)            | (Finishes keystroke collection) |
| <i>execute_macro</i> | Esc-E            | (Executes collected keystrokes) |

### **Binding and PFkey Support**

These commands support the ability to change the default command bindings and provide flexibility in programming PF commands.



*Replace this page with the tab page labeled*

**Appendices**

# Appendices

.....

# RVEC Parameters

## Meaning of RVEC Parameters

The commands RESTOR, RESUME, SAVE, PM, and START process a group of optional parameters associated with the PRIMOS RVEC vector. These parameters are stored on disk for every runfile (executable program).

Initial values for the RVEC parameters are usually specified in the PRIMOS SAVE command, or by the LOAD or SEG SAVE command, when the program was stored on disk.

Each parameter is a 16-bit halfword, represented by as many as six octal digits.

| <i>Parameter</i> | <i>Memory Location</i> | <i>Definition</i>                                     |
|------------------|------------------------|-------------------------------------------------------|
| SA               | —                      | Starting Address (first memory word used by program)  |
| EA               | —                      | Ending Address (last memory halfword used by program) |
| PC               | 7                      | P Register (Program Counter)                          |
| A                | 1                      | A Register (Arithmetic)                               |
| B                | 2                      | B Register (Arithmetic)                               |
| X                | 0                      | Index Register                                        |
| Keys             | —                      | Status keys associated with INK and OTK instructions  |

The RVEC parameters are optional in the command string. Any item that is specified replaces the previous value in RVEC, which is saved with the program. Thus, for any parameters that are not specified, the value previously stored in RVEC is saved with the program.



### ***Slash Convention***

An ordinal value followed by a slash (/) and a value can be used to set a selected octal parameter without setting other octal parameters. For example, given the command format

**RESUME *pathname* [*pc*] [*a*] [*b*] [*x*] [*keys*]**

the command

```
R FILNAM 2/1000
```

skips the first two RVEC parameters and sets the value of the third (b) to 1000<sub>8</sub>.

### ***Supplying RVEC Parameters***

RVEC parameters specified in RESUME or START commands replace the previous values in RVEC. Also, when a program returns to PRIMOS through the EXIT subroutine, RVEC is loaded from the processor values in effect at the time of exit. Only the SAVE command alters the values of RVEC stored on disk with the program.

The RESTOR command returns a program from disk to memory and loads the SAVE parameters into RVEC in preparation for a START command.

The RESUME command combines the functions of RESTOR and START.

The PM command lists the current values of the RVEC parameters.

External commands have RVEC parameters that can be modified at the time the command is started (for example, PMA filename 1/740). Providing RVEC parameters to a command that does not need them causes unpredictable results.

## **Meaning of the Keys Parameter**

The RVEC parameter *keys* refers to the processor status keys handled by the INK and OTK instructions. (For these instructions, see the *System Architecture Reference Guide*.) *keys* is represented by a single 16-bit word in either the Keys (SR) or Keys (VI) format. S-mode and R-mode programs use the Keys (SR) format; V-mode and I-mode programs use the Keys (VI) format.

### ***Keys (SR) Format***

The keys format for S and R modes is as follows:

|      |     |   |      |     |     |      |
|------|-----|---|------|-----|-----|------|
| CBIT | DBL | — | MODE | FEX | IEX | VSC  |
| 1    | 2   | 3 | 4–6  | 7   | 8   | 9–16 |

The mnemonics and meanings of the bits are as follows:

| <i>Bit</i>             | <i>Description</i>                                                                                                                                                   |
|------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>CBIT (Bit 1)</b>    | Set by error conditions in arithmetic operations and by shifts                                                                                                       |
| <b>DBL (Bit 2)</b>     | Specifies the arithmetic mode:<br>0       Single precision<br>1       Double precision                                                                               |
| <b>— (Bit 3)</b>       | Reserved for future use                                                                                                                                              |
| <b>MODE (Bits 4–6)</b> | Specifies the current addressing mode:<br>000     16S<br>001     32S<br>010     64R<br>011     32R<br>100     32I<br>101     Unused<br>110     64V<br>111     Unused |
| <b>FEX (Bit 7)</b>     | Floating-point exception:<br>0       Set C bit to 1 and invoke fault handler on error<br>1       Set C bit to 1 only on error                                        |
| <b>IEX (Bit 8)</b>     | Integer exception:<br>0       Set C bit to 1 only on error<br>1       Set C bit to 1 and invoke fault handler on error                                               |
| <b>VSC (Bits 9–16)</b> | Visible Shift Count: bottom half of the floating-point exponent                                                                                                      |

**Keys (VI) Format**

Process status information is available in a 16-bit register known as the Keys. The register may be referenced by the LPSW, TKA, and TAK instructions.

The mnemonics and meanings of the bits are as follows:

| <i>Bit</i>             | <i>Description</i>                                                                                                                                                                                                                                                  |
|------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>CBIT (Bit 1)</b>    | Set by error conditions in arithmetic operations and by shifts                                                                                                                                                                                                      |
| <b>– (Bit 2)</b>       | Must be zero                                                                                                                                                                                                                                                        |
| <b>L (Bit 3)</b>       | Set by an arithmetic or shift operation except IRS, IRX, DRX. Equal to carry out of the most significant bit (Bit 1) of an arithmetic operation. Valuable for simulating many precision operations and for performing unsigned comparisons following a CAS or a SUB |
| <b>MODE (Bits 4–6)</b> | Specifies the current addressing mode:<br>000      16S<br>001      32S<br>010      64R<br>011      32R<br>100      32I<br>101      Unused<br>110      64V<br>111      Unused                                                                                        |
| <b>FEX (Bit 7)</b>     | Floating-point exception enable/disable:<br>0          Set C bit to 1 and invoke fault handler on error<br>1          Set C bit to 1 only on error                                                                                                                  |
| <b>IEX (Bit 8)</b>     | Integer exception enable/disable:<br>0          Set C bit to 1 only on error<br>1          Set C bit to 1 and invoke fault handler on error                                                                                                                         |
| <b>LT (Bit 9)</b>      | Less Than condition code: reflects the extended sign of the result (before truncation, if overflow); set to 1 if result is negative                                                                                                                                 |
| <b>EQ (Bit 10)</b>     | Equal To condition code: reflects the state of the result (after truncation, if overflow); set to 1 only if result is zero                                                                                                                                          |
| <b>DEX (Bit 11)</b>    | Decimal exception enable/disable:<br>0          Set C bit to 1 only on error<br>1          Set C bit to 1 and invoke fault handler on error                                                                                                                         |

|                         |                                                                                                                                                                                                                                                                                                                                                                                    |
|-------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>ASCII-8 (Bit 12)</b> | <p>ASCII character representation: specifies whether set or reset ASCII characters are generated (disregarded on the 2250 and earlier machines).</p> <p style="margin-left: 40px;">0        Most significant bit of characters is 1<br/>          (set format)</p> <p style="margin-left: 40px;">1        Most significant bit of characters is 0<br/>          (reset format)</p> |
| <b>RND (Bit 13)</b>     | <p>Floating-point round: specifies the form of rounding in floating-point operations (disregarded on the 2250 and earlier machines).</p> <p style="margin-left: 40px;">0        No rounding</p> <p style="margin-left: 40px;">1        Rounding</p>                                                                                                                                |
| <b>P850 (Bit 14)</b>    | <p>P850 bit used only by the P850 processor: set/cleared only by process exchange</p>                                                                                                                                                                                                                                                                                              |
| <b>IN (Bit 15)</b>      | <p>In dispatcher: specifies if the current process associated with the register is in the dispatcher</p> <p style="margin-left: 40px;">0        Process is in the dispatcher</p> <p style="margin-left: 40px;">1        Process is not in the dispatcher</p>                                                                                                                       |
| <b>SD (Bit 16)</b>      | <p>Save done: specifies if PXM (process exchange mechanism) has saved the values of the current register set</p> <p style="margin-left: 40px;">0        Save must be done before this register set<br/>          can be used</p> <p style="margin-left: 40px;">1        Save has been done and this register set is<br/>          available</p>                                    |

# DUMP\_STACK Format

## DUMP\_STACK Command

The DUMP\_STACK command, explained in Chapter 2 of this guide, traces the sequence of calls and returns by which the user's process arrived at its current state. The user's command stack preserves machine states for internal commands, condition frames, and fault frames. In addition, the most recent activation of a static-mode program or command is preserved on the static-mode stack. DUMP\_STACK displays the stack dump on the terminal or writes it to a command output file. Because DUMP\_STACK is an internal command, it does not overwrite the static-mode stack, thereby permitting reentry into the faulting program.

## Format of Stack Frames

DUMP\_STACK lists each stack frame in the following general format:

*(nn) offset. Owner= proc-name (LB= owner-lb).*  
Called from *pcl-addr*; returns to *return-addr*.

The meaning of each parameter is listed below. See the *System Architecture Reference Guide* for an explanation of the registers and rings involved.

| <i>Parameter</i> | <i>Definition</i>                                                                                |
|------------------|--------------------------------------------------------------------------------------------------|
| <i>nn</i>        | Frame index number of the stack frame                                                            |
| <i>offset</i>    | The word number in the current stack segment where the stack frame of this activation begins     |
| <i>proc-name</i> | The name (if available) of the procedure that owns the stack frame                               |
| <i>owner-lb</i>  | The value of the LB (linkage base) register belonging to the procedure that owns the stack frame |

*pcl-addr* Address of the PCL instruction that caused the procedure to be invoked

*return-addr* The address to which the procedure returns

**Note**

A called-from or returns-to value such as 0(0) or 0(0)/177776 usually means that the stack frame has an invalid return point and can never return. An example of such a frame is the first frame set up by SEG in a V-mode static mode program.

If the stack switches to a different segment during the trace, DUMP\_STACK displays the following message:

STACK SEGMENT IS *n*

where *n* is the octal segment number of the new stack segment.

**Format of a Fault Frame**

If the frame is a fault frame, the following format is used:

```
(nn) offset: FAULT FRAME; fault type = fault-type.
      Fault returns to ret-pb; LB= fault-lb, keys= fault-keys.
      Fault code= f-code; fault addr= f-addr.
      Registers at time of fault in inner ring:
                Save Mask= ssssss; XB= xb-value
      GR0      0      0      0      GR1      0      0      0
      L,GR2    0      0      0      E,GR3    0      0      0
      GR4      0      0      0      Y,GR5    0      0      0
      GR6      0      0      0      X,GR7    0      0      0
      FAR0 0(0)/0      FLR0      0      FRO 0.00000000E 00
      FAR1 0(0)/0      FLR1      0      FRI 0.00000000E 00
```

| <i>Parameter</i>     | <i>Definition</i>                                                                                                     |
|----------------------|-----------------------------------------------------------------------------------------------------------------------|
| <i>fault-type</i>    | Location in the fault table of the type of fault that occurred                                                        |
| <i>ret-pb</i>        | Address to which the fault returns                                                                                    |
| <i>fault-lb</i>      | LB register belonging to the procedure in which the fault occurred                                                    |
| <i>fault-keys</i>    | CPU keys at the time of the fault                                                                                     |
| <i>register data</i> | If present, a direct dump of the register save area (in the same format as that produced by the CPU RSAV instruction) |
| <i>f-code</i>        | Fault code generated by this particular fault                                                                         |

*f-addr*                      Fault address generated by this particular fault

**Format of a Condition Frame**

If the activation is a condition frame, the following format is used:

```
(nn) offset: CONDITION FRAME for "condition-name"; returns to ret-pb.
Condition raised at sigloc; LB= siglb; keys= sigkeys.
(Crawlout to outerpb; LB= outerlb; keys= outerkeys.)
Inner ring fault: type "PROCESS" (4); code= 000200; addr= 0(0)/0
Registers at time of fault in inner ring:
          Save Mask= ssssss; XB= xb-value
GR0      0      0      0      GR1      0      0      0
L,GR2    0      0      0      E,GR3    0      0      0
GR4      0      0      0      Y,GR5    0      0      0
GR6      0      0      0      X,GR7    0      0      0
FAR0 0(0)/0      FLR0      0 FR0 0.00000000E 00
FAR1 0(0)/0      FLR1      0 FR1 0.00000000E 00
```

The latter two items are displayed only if the condition was signaled in an inner ring and if subsequently a crawlout to the current ring occurred.

# Obsolete Commands

## *Introduction*

This appendix describes commands that have been made obsolete either by changing technology or by new commands that supercede their functionality.

## ARCHIVE

The Backup and Recovery Management Service (BRMS) command ARCHIVE copies file system objects from disk to tape. File system objects include files, directories, access categories, and Recovery Based Files.

## *Format*

```
ARCHIVE { pathname -MT n -VOLID tapename }  
        { -HELP [option] }
```

## *Options and Arguments*

|                         |                                                                                                                                                                                               |
|-------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>pathname</i>         | Specifies the file or directory to be copied to tape.                                                                                                                                         |
| -MT <i>n</i>            | Specifies the unit number of the tape drive on which the ARCHIVE tape is mounted. <i>n</i> is the physical device number of the tape drive.                                                   |
| -VOLID <i>tapename</i>  | Specifies the volume name of the ARCHIVE tape on which the objects are to be stored. <i>tapename</i> is the name of the ARCHIVE volume. The ARCHIVE tape can be a current tape or a new tape. |
| -HELP [ <i>option</i> ] | Displays the online information about the syntax and options of the ARCHIVE command.                                                                                                          |





## ARCHIVE\_RESTORE

The Backup and Recovery Management Service (BRMS) command ARCHIVE\_RESTORE copies file system objects from an ARCHIVE tape to disk.

Before using ARCHIVE\_RESTORE, you must assign the tape drive with the ASSIGN command and mount the correct reel.

### *Format*

```
ARCHIVE_RESTORE { pathname [new-pathname] -MT n }
                 { -HELP [option] }
```

### *Options and Arguments*

|                         |                                                                                                                                                       |
|-------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>pathname</i>         | Specifies the file or directory to be copied to tape.                                                                                                 |
| <i>new-pathname</i>     | Specifies the pathname of objects when restored to disk. If omitted, the files are restored to the current directory, using the object name as saved. |
| -MT <i>n</i>            | Specifies the unit number of the tape drive on which the ARCHIVE tape is mounted. <i>n</i> is the physical device number of the tape drive.           |
| -HELP [ <i>option</i> ] | Displays the online information about the syntax and options of the ARCHIVE command.                                                                  |

### *Usage*

After you save file system objects on tape with the ARCHIVE command, you can restore the data back to disk with ARCHIVE\_RESTORE. File system objects include files, directories, access categories, and Recovery Based Files.

For detailed information on the options and use of ARCHIVE\_RESTORE, see the *Data Backup and Recovery Guide*.

See also ARCHIVE; ARCHIVE\_RELEASE; GENERATE\_CATALOG; LIST\_CATALOG; LIST\_TAPE.

## ASRCWD

ASRCWD sets a virtual control word that selects the input or output device for diverted terminal I/O.

### Format

ASRCWD *number*

### Argument

*number* An octal value which specifies the input and output devices for diverted terminal I/O.

### Usage

ASRCWD functions only on a system that has a serial asynchronous interface with serial printers, serial card readers, and/or serial card punches. After the command is given, input is taken from and output is sent to the two devices selected by bits 11 through 16 of the octal value *number*.

|                              |      |                    |
|------------------------------|------|--------------------|
| INPUT BITS ( <i>xx</i> ):    | 00   | User Terminal      |
|                              | 01   | (Reserved)         |
|                              | 10   | (Reserved)         |
|                              | 11   | Serial Card Reader |
| OUTPUT BITS ( <i>yyyy</i> ): | 0000 | User Terminal      |
|                              | 1000 | User Terminal      |
|                              | 0100 | Serial Printer 1   |
|                              | 0010 | Serial Printer 2   |
|                              | 0001 | Serial Card Punch  |

For example, to choose the serial card reader for input and the user terminal for output, select the following bits:

11 0000

which can be regrouped as 110 000 to become octal 60. The command would be

OK, ASRCWD 60

The virtual control word is usually set to the appropriate value by programs. ASRCWD is used only when a program exits abnormally (for example, by pressing

the BREAK or CONTROL-P keys), leaving input or output diverted away from the terminal. The command

```
ASRCWD 0
```

corrects this condition.

## ATTACH

### Obsolete syntax

ATTACH finds the disk file location of *directory*, checks *password* (if any), and places this information into two storage areas associated with the user. These two areas define the current directory and the home directory. Each area contains the name of the directory, its disk location, and a status flag that indicates whether the user is an owner or a nonowner of the directory. As an option, the user can specify that the information be recorded to redefine only the current directory, leaving the home directory information unchanged.

ATTACH operates only on file directories, not segment directories.

The ATTACH command was rewritten to provide greater functionality, superceding the format documented here. See the ATTACH command entry in Chapter 2.

### Format

```
ATTACH directory [password] [ldisk] [key]
```

### Arguments

|                  |                                                                                                                                                            |
|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>directory</i> | Specifies the directory to make the new current directory.                                                                                                 |
| <i>password</i>  | Specifies the directory's password                                                                                                                         |
| <i>ldisk</i>     | <i>ldisk</i> is an octal integer identifying the logical disk to search for the directory. See the section below called Logical Disk for more information. |
| <i>key</i>       | Specifies the home key value. See the section below called Home Key Values for more information.                                                           |

## Passwords

A directory can have a pair of passwords to provide security. Only users who know the owner password have owner access to the files in the directory. The nonowner password likewise restricts nonowner access. See the **PASSWD** and **PROTECT** commands in Chapter 2 of this guide.

If a directory has both owner and nonowner passwords, a correct password — owner or nonowner — is required, and the user obtains owner or nonowner status appropriately. An incorrect or missing password results in the **NO UFD ATTACHED** status.

If a directory has only an owner password, then the correct password gives the user owner status. An incorrect password, or none, gives nonowner status.

If a directory has no password, owner status is given to an attaching user, whether the user supplies a password or not.

## Logical Disk

The user can specify which logical disk is to be searched for the directory by using the command format

**ATTACH** *directory password ldisk*

The value of *ldisk* is specified as follows:

- |          |                                                                                                                                                                                                                    |
|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>n</i> | Searches the MFD of logical disk <i>n</i> . The <b>LDEV</b> column of the <b>STATUS DISKS</b> printout shows the logical disk number for each disk. <i>n</i> can be from 0 to the maximum number of disks allowed. |
| 100000   | Searches MFDs of all disks in order of increasing <i>ldisk</i> number. (Default)                                                                                                                                   |
| 177777   | Searches the MFD of the disk to which the user is currently attached.                                                                                                                                              |

You can accomplish the same actions by using pathnames. For example,

**ATTACH** *<n>pathname*

**ATTACH** *ordinary-pathname*

**ATTACH** *<\*>pathname*

correspond exactly to disk values of *n*, 100000, and 177777, respectively.

## Home Key Values

**ATTACH** normally sets both the current and the home directories to the target directory. However, in the command format

**ATTACH** *directory password ldisk key*

you can choose a value of *key* to allow or prevent the setting of the home directory to be *directory*. Specify *key* as follows:

| <i>Key</i> | <i>Meaning</i>                                                                                  |
|------------|-------------------------------------------------------------------------------------------------|
| 177777     | Attaches to a directory in the MFD on <i>ldisk</i> but does not set it as the home directory.   |
| 0          | Attaches to a directory in the MFD on <i>ldisk</i> and sets it as the home directory. (Default) |
| 1          | Attaches to a subdirectory in the current directory but does not set it as the home directory.  |
| 2          | Attaches to a subdirectory in the current directory and sets it as the home directory.          |

To specify a key without specifying *ldisk*, use the slash convention, as in the following three formats:

```
ATTACH directory password l/key
ATTACH directory l/key
ATTACH pathname l/key
```

See also the ATTACH command function in Chapter 3.

## CPMPC

CPMPC punches the contents of a file onto cards in an assigned card reader/punch.

### **Format**

```
CPMPC pathname [-CRn] [-PRINT]
```

### **Options**

You may specify the CPMPC parameters in any order.

|                 |                                                                                                                                                                                                                                                                      |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>pathname</i> | Specifies the name of the file to be punched. The value of <i>n</i> in the -CR option is 0 or 1, depending on whether you specify the first (CR0) or second (CR1) card reader/punch. You must assign the card reader/punch with an ASSIGN CR0 or ASSIGN CR1 command. |
| -CR             | Specifies the number of the card reader/punch. <i>n</i> must be 0 or 1.                                                                                                                                                                                              |

**-PRINT** Causes punched data to be printed on the card if the punch has that capability.

### Usage

CPMPC does not punch an end-of-file (\$E) card at the end of the output deck of punched cards. For further information on \$E cards, see the CRMPC command.

## CRMPC

CRMPC reads cards from the parallel interface card reader connected to the MPC controller and loads card image ASCII data into the file specified by *pathname*.

### Format

CRMPC *pathname* [-CR*n*] [-PRINT]

### Options

You may specify the CRMPC parameters in any order.

|                 |                                                                                                                                                                                                                                                                      |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>pathname</i> | Specifies the name of the file to be punched. The value of <i>n</i> in the -CR option is 0 or 1, depending on whether you specify the first (CR0) or second (CR1) card reader/punch. You must assign the card reader/punch with an ASSIGN CR0 or ASSIGN CR1 command. |
| -CR             | Specifies the number of the card reader/punch. <i>n</i> must be 0 or 1.                                                                                                                                                                                              |
| -PRINT          | Prints the contents of each card on the card, if the card reader has that capability.                                                                                                                                                                                |

### Usage

Reading of the cards continues until one of the following occurs:

- A card is read that has \$E in columns 1 and 2 (the recommended way to stop).
- There are no more cards in the reader.
- The STOP button on the card reader is pressed.
- BREAK or CONTROL-P is pressed on the terminal.

Cards are expected to be in 029 (EBCDIC) representation. Control cards may be inserted into the card deck to instruct the card reader, as follows:

| <i>Columns 1 and 2 of<br/>deck control card</i> | <i>Instruction</i>                                                                                                                 |
|-------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------|
| \$6                                             | Placed before a deck of cards in 026 (BCD) format. Instructs the card reader to interpret 026 cards as if they were in 029 format. |
| \$9                                             | Instructs the card reader to resume reading in 029 format.                                                                         |
| \$E                                             | Placed last in the deck and signals the end of the deck. Control returns to PRIMOS and the file is closed.                         |

## EDB

EDB loads and starts EDB, the Prime binary editor.

### *Format*

**EDB *input-filename* [*output-filename*]**

### *Usage*

EDB is used primarily for creating and maintaining libraries of subroutines. When EDB starts, it displays an ENTER prompt and waits for an EDB subcommand.

The input and output files may be on disk or paper tape. *input-filename* should be an existing library or the binary output of a Prime language translator. If *output-filename* is an existing file, it is overwritten by the new output file.

If paper tape is used for either the input or output file, use **-PTR** as the filename. In the following example EDB is started with a paper tape as the input file and NEWLIB as the name of the output file.

```
OK, EDB -PTR NEWLIB
```

If a terminal is used as either the input or output file, use **-ASR** as the filename. In the following example PASLIB2 is a PRIMOS input file and the output file is your terminal.

```
OK, EDB PASLIB2 -ASR
```

A PRIMOS input or output file is assumed if neither **-ASR** or **-PTR** is specified.





### GENERATE\_CATALOG -HELP *[option]*

For full information on the operation, arguments, and options of GENERATE\_CATALOG, see the *Data Backup and Recovery Guide*.

## LIBEDB

LIBEDB is used for editing bypass information into library files. The BIND linker uses the bypass information to skip an unnecessary routine efficiently instead of reading and discarding all the unwanted object text.

LIBEDB has been replaced by the EDIT\_BINARY command.

For details see *Advanced Programmer's Guide 1: BIND and EPFs*.

## LISTF

LISTF displays information on the current directory.

### *Format*

LISTF

### *Usage*

The LISTF default output displays a header followed by the names of the files, access categories, subdirectories, and segment directories in the current directory. The display does not list each object's type. The information in the header is the same as for the LD command.

LISTF has been replaced by the LD command.

## LIST\_CATALOG

The Backup and Recovery Management Service (BRMS) command LIST\_CATALOG lists the contents of catalogs created by the ARCHIVE and BACKUP commands.

### **Format**

LIST\_CATALOG [*pathname*] [*options*]

### **Usage**

By default, LIST\_CATALOG lists the contents of ARCHIVE tape catalogs in the subdirectory CATS\* which is located in the user's origin directory. If CATS\* is located elsewhere, its pathname must be specified. Only the tape owner or System Administrator can list an ARCHIVE catalog.

BACKUP tape catalogs are maintained in the system catalog directory BACKUP\*>CATS\* and are listed with the -BACKUP option.

LIST\_CATALOG supports the iteration, wildcarding, and treewalking command-line features. To display online help information on LIST\_CATALOG, use the command

LIST\_CATALOG -HELP

For details on LIST\_CATALOG and its options, see the *Data Backup and Recovery Guide*.

## LIST\_TAPE

The Backup and Recovery Management Service (BRMS) command LIST\_TAPE lists information about the contents of tapes created by the ARCHIVE, BACKUP, or TRANSPORT commands. Before using LIST\_TAPE, you must assign a tape drive unit with the ASSIGN command and mount the correct reel on the drive. The number of the drive unit is specified by *n* in the -MT *n* keyword. By default, the LIST\_TAPE display includes the types and pathnames of files, directories, segment directories, and access categories recorded on the tapes. More details can be listed with LIST\_TAPE options. Only system operators or the System Administrator can list BACKUP tapes. Only the tape owner or the System Administrator can list ARCHIVE tapes. Any user can list TRANSPORT tapes.

### **Format**

LIST\_TAPE [*pathname*] -MT *n* [*options*]

### **Usage**

To display online help information on LIST\_TAPE, use the following command:

OK, LIST\_TAPE -HELP

For details on LIST\_TAPE and its options, see the *Data Backup and Recovery Guide*.

## PRMPC

PRMPC prints a file on an MPC parallel interface printer configured to PRIMOS.

### **Format**

PRMPC *pathname*

*pathname* is the name of the file to be printed.

### **Usage**

Before issuing the PRMPC command, you must assign the printer (PRO) with the ASSIGN command.

## PROTEC

PROTEC sets access rights for files in a password directory. These rights are for an owner of the file and for nonowners. To set these rights, you must have owner access to the file.

### Format

PROTEC *pathname key1 key2*

### Arguments

*pathname* specifies the name of the file to be protected. *key1* is an integer that specifies the owner's access rights to *pathname*. *key2* is an integer that specifies the nonowner's access rights to *pathname*.

The values and meanings for *key1* and *key2* are as follows:

| Value | Rights                        |
|-------|-------------------------------|
| 0     | No access of any kind allowed |
| 1     | Read only                     |
| 2     | Write only                    |
| 3     | Read and write                |
| 4     | Delete and truncate           |
| 5     | Delete, truncate, and read    |
| 6     | Delete, truncate, and write   |
| 7     | All access                    |

### Usage

Here are two examples of the PROTEC command.

**Example 1:** The owner is given all access rights to JULY.MEMO and read-only rights to nonowners.

```
OK, PROTEC JULY.MEMO 7 1
```

**Example 2:** Both owners and nonowners receive all access rights to the file SALES\_REPORT.

```
OK, PROTEC SALES_REPORT 7 7
```

---

**Note**

The default protection keys associated with any newly created file or directory are 7 and 0. (Owner is given all rights and nonowner is given none.) If you issue PROTEC without arguments, values are set to 0 and 0. This means the owner has no rights and therefore cannot access the file unless you issue another PROTEC command with an appropriate value for key-1.

---

**PRSER**

PRSER prints a file on the serial interface printer configured to PRIMOS.

Before using PRSER, you must assign the printer (CENPR or CE2PR) with the ASSIGN command.

***Format***

PRSER *pathname*

***Argument***

*pathname* Specifies the file to be printed.

***Usage***

The following example assigns the plotter and then prints the file SALES.GRAPH:

```
OK, ASSIGN PLOT
OK, PRSER SALES.GRAPH
```

## PRVER

PRVER prints a file on a printer/plotter configured to PRIMOS.

Before using PRVER, you must assign the plotter with the command ASSIGN PLOT.

### Format

PRVER *pathname*

### Argument

*pathname* Specifies the file to be printed.

### Usage

The following example assigns the plotter and then prints the file SALES.GRAPH:

```
OK, ASSIGN PLOT
OK, PRVER SALES.GRAPH
```

## TRANSPORT

The Backup and Recovery Management Service (BRMS) command TRANSPORT copies file system objects from disk to magnetic tape for transporting to another Prime site.

### Format

TRANSPORT *pathname* -MT *n* [*options*]

### Arguments and Options

*pathname* Specifies the directory, segment directory, or file to be copied. (RBF files cannot be copied with this command.) *pathname* can be a wildcard name and can use the treewalking option.

- MT *n*                      Specifies the tape drive on which the tape is mounted. *n* is the tape drive's unit number. Before using TRANSPORT, you must assign the tape drive with the ASSIGN command.
- HELP                        Starts the online help facility.

**Usage**

The TRANSPORT command is similar to the ARCHIVE command except that TRANSPORT does not save ACLs and password protection (unless you specify the -SAVE\_PROTECTION option). The TRANSPORT tape can therefore be restored at another site by any user.

For details on the options and operation of TRANSPORT, see the *Data Backup and Recovery Guide*.

See also LIST\_TAPE; TRANSPORT\_RELEASE; TRANSPORT\_RESTORE.

**TRANSPORT\_RELEASE**

The Backup and Recovery Management Service (BRMS) command TRANSPORT\_RELEASE releases a TRANSPORT tape so that it can be overwritten.

Before using TRANSPORT\_RELEASE, you must assign the tape drive with the ASSIGN command.

**Format**

TRANSPORT\_RELEASE -MT *n* [*options*]

**Arguments and Options**

- MT *n*                      Specifies the tape drive on which the tape is mounted. *n* is the tape drive's unit number.
- HELP                        Starts the online help facility.

**Usage**

For details on the options and operation of TRANSPORT\_RELEASE, see the *Data Backup and Recovery Guide*.

See also LIST\_TAPE; TRANSPORT; TRANSPORT\_RESTORE.



## TRANSPORT\_RESTORE

The Backup and Recovery Management Service (BRMS) command **TRANSPORT\_RESTORE** restores file system objects to disk from a tape created by the **TRANSPORT** command.

Before using **TRANSPORT\_RESTORE**, you must assign the tape drive with the **ASSIGN** command.

### Format

**TRANSPORT\_RESTORE** *pathname* [*new-pathname*] **-MT** *n* [*options*]

### Arguments and Options

|                     |                                                                                                                                                |
|---------------------|------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>pathname</i>     | Specifies the directory, segment directory, or file to be restored. <i>pathname</i> can be a wildcard name and can use the treewalking option. |
| <i>new-pathname</i> | Renames an object as it is restored or specifies a location for the object other than your current attach point.                               |
| <b>-MT</b> <i>n</i> | Specifies the tape drive on which the tape is mounted. <i>n</i> is the tape drive's unit number.                                               |
| <b>-HELP</b>        | Starts the online help facility.                                                                                                               |

### Usage

A **TRANSPORT** tape created at one Prime site with the **TRANSPORT** command can be copied to disk at another Prime site with the **TRANSPORT\_RESTORE** command.

For details on the options and operation of **TRANSPORT\_RESTORE**, see the *Data Backup and Recovery Guide*.

See also **LIST\_TAPE**; **TRANSPORT**; **TRANSPORT\_RELEASE**.

## ECS and EBCDIC Character Sets

As of Rev. 21.0, Prime has expanded its character set. The basic character set remains the same: it is the ANSI ASCII 7-bit set (called ASCII-7), with the 8th bit turned on. However, the 8th bit is now significant; when it is turned off, it signifies a different character. Thus, the size of the character set has doubled, from 128 to 256 characters. This expanded character set is called the Prime Extended Character Set (Prime ECS).

The pre-Rev. 21.0 character set is a proper subset of Prime ECS. These characters have not changed. Software written before Rev. 21.0 will continue to run exactly as it did before. Software written at Rev. 21.0 that does not use the new characters needs no special coding to use the old ones.

Prime ECS support is automatic at Rev. 21.0. You may begin to use characters that have the 8th bit turned off. However, the extra characters are not available on most printers and terminals. Check with your System Administrator to find out whether you can take advantage of the new characters in Prime ECS.

Table D-1 shows the complete Prime Extended Character Set. The pre-Rev. 21.0 character set consists of the characters with decimal values 128 through 255 (octal values 200 through 377). (The pre-Rev. 21.0 character set is shaded in Table D-1.) The characters added at Rev. 21.0 all have decimal values less than 128 (octal values less than 200).

### Specifying Prime ECS Characters

#### *Direct Entry*

On terminals that support Prime ECS, you can enter the characters directly. For information on how to do this, see the appropriate manual for your terminal.

#### *Octal Notation*

If you use the Editor (ED), you can enter any Prime ECS character on any terminal by typing:

`^octal-value`



*Table D-1 The Prime Extended Character Set*

| <i>Graphic</i> | <i>Mnemonic</i> | <i>Description</i>                  | <i>Binary</i> | <i>Decimal</i> | <i>Hex</i> | <i>Octal</i> |
|----------------|-----------------|-------------------------------------|---------------|----------------|------------|--------------|
|                | RES1            | Reserved for future standardization | 0000 0000     | 000            | 00         | 000          |
|                | RES2            | Reserved for future standardization | 0000 0001     | 001            | 01         | 001          |
|                | RES3            | Reserved for future standardization | 0000 0010     | 002            | 02         | 002          |
|                | RES4            | Reserved for future standardization | 0000 0011     | 003            | 03         | 003          |
|                | IND             | Index                               | 0000 0100     | 004            | 04         | 004          |
|                | NEL             | Next line                           | 0000 0101     | 005            | 05         | 005          |
|                | SSA             | Start of selected area              | 0000 0110     | 006            | 06         | 006          |
|                | ESA             | End of selected area                | 0000 0111     | 007            | 07         | 007          |
|                | HTS             | Horizontal tabulation set           | 0000 1000     | 008            | 08         | 010          |
|                | HTJ             | Horizontal tab with justify         | 0000 1001     | 009            | 09         | 011          |
|                | VTS             | Vertical tabulation set             | 0000 1010     | 010            | 0A         | 012          |
|                | PLD             | Partial line down                   | 0000 1011     | 011            | 0B         | 013          |
|                | PLU             | Partial line up                     | 0000 1100     | 012            | 0C         | 014          |
|                | RI              | Reverse index                       | 0000 1101     | 013            | 0D         | 015          |
|                | SS2             | Single shift 2                      | 0000 1110     | 014            | 0E         | 016          |
|                | SS3             | Single shift 3                      | 0000 1111     | 015            | 0F         | 017          |
|                | DCS             | Device control string               | 0001 0000     | 016            | 10         | 020          |
|                | PU1             | Private use 1                       | 0001 0001     | 017            | 11         | 021          |
|                | PU2             | Private use 2                       | 0001 0010     | 018            | 12         | 022          |
|                | STS             | Set transmission state              | 0001 0011     | 019            | 13         | 023          |
|                | CCH             | Cancel character                    | 0001 0100     | 020            | 14         | 024          |
|                | MW              | Message waiting                     | 0001 0101     | 021            | 15         | 025          |
|                | SPA             | Start of protected area             | 0001 0110     | 022            | 16         | 026          |
|                | EPA             | End of protected area               | 0001 0111     | 023            | 17         | 027          |
|                | RES5            | Reserved for future standardization | 0001 0000     | 024            | 18         | 030          |
|                | RES6            | Reserved for future standardization | 0001 1001     | 025            | 19         | 031          |
|                | RES7            | Reserved for future standardization | 0001 1010     | 026            | 1A         | 032          |
|                | CSI             | Control sequence introducer         | 0001 1011     | 027            | 1B         | 033          |
|                | ST              | String terminator                   | 0001 1100     | 028            | 1C         | 034          |
|                | OSC             | Operating system command            | 0001 1101     | 029            | 1D         | 035          |
|                | PM              | Privacy Message                     | 0001 1110     | 030            | 1E         | 036          |
|                | APC             | Application program command         | 0001 1111     | 031            | 1F         | 037          |
|                | NBSP            | No-break space                      | 0010 0000     | 032            | 20         | 040          |
| ¡              | INVE            | Inverted exclamation mark           | 0010 0001     | 033            | 21         | 041          |
| ¢              | CENT            | Cent sign                           | 0010 0010     | 034            | 22         | 042          |
| £              | PND             | Pound sign                          | 0010 0011     | 035            | 23         | 043          |

Table D-1 The Prime Extended Character Set (continued)

| Graphic | Mnemonic | Description                          | Binary    | Decimal | Hex | Octal |
|---------|----------|--------------------------------------|-----------|---------|-----|-------|
| ¤       | CURR     | Currency sign                        | 0010 0100 | 036     | 24  | 044   |
| ¥       | YEN      | Yen sign                             | 0010 0101 | 037     | 25  | 045   |
| ¡       | BBAR     | Broken bar                           | 0010 0110 | 038     | 26  | 046   |
| §       | SECT     | Section sign                         | 0010 0111 | 039     | 27  | 047   |
| ¨       | DIA      | Diaeresis, umlaut                    | 0010 1000 | 040     | 28  | 050   |
| ©       | COPY     | Copyright sign                       | 0010 1001 | 041     | 29  | 051   |
| ª       | FOI      | Feminine ordinal indicator           | 0010 1010 | 042     | 2A  | 052   |
| «       | LAQM     | Left angle quotation mark            | 0010 1011 | 043     | 2B  | 053   |
| ¬       | NOT      | Not sign                             | 0010 1100 | 044     | 2C  | 054   |
| –       | SHY      | Soft hyphen                          | 0010 1101 | 045     | 2D  | 055   |
| ®       | TM       | Registered trademark sign            | 0010 1110 | 046     | 2E  | 056   |
| ˆ       | MACN     | Macron                               | 0010 1111 | 047     | 2F  | 057   |
| °       | DEGR     | Degree sign                          | 0011 0000 | 048     | 30  | 060   |
| ±       | PLMI     | Plus/minus sign                      | 0011 0001 | 049     | 31  | 061   |
| ²       | SPS2     | Superscript two                      | 0011 0010 | 050     | 32  | 062   |
| ³       | SPS3     | Superscript three                    | 0011 0011 | 051     | 33  | 063   |
| ´       | AAC      | Acute accent                         | 0011 0100 | 052     | 34  | 064   |
| µ       | LCMU     | Lowercase Greek letter m, micro sign | 0011 0101 | 053     | 35  | 065   |
| ¶       | PARA     | Paragraph sign, Pilgrow sign         | 0011 0110 | 054     | 36  | 066   |
| ·       | MIDD     | Middle dot                           | 0011 0111 | 055     | 37  | 067   |
| ¸       | CED      | Cedilla                              | 0011 1000 | 056     | 38  | 070   |
| ¹       | SPS1     | Superscript one                      | 0011 1001 | 057     | 39  | 071   |
| º       | MOI      | Masculine ordinal indicator          | 0011 1010 | 058     | 3A  | 072   |
| »       | RAQM     | Right angle quotation mark           | 0011 1011 | 059     | 3B  | 073   |
| ¼       | FR14     | Common fraction one-quarter          | 0011 1100 | 060     | 3C  | 074   |
| ½       | FR12     | Common fraction one-half             | 0011 1101 | 061     | 3D  | 075   |
| ¾       | FR34     | Common fraction three-quarters       | 0011 1110 | 062     | 3E  | 076   |
| ¿       | INVQ     | Inverted question mark               | 0011 1111 | 063     | 3F  | 077   |
| À       | UCAG     | Uppercase A with grave accent        | 0100 0000 | 064     | 40  | 100   |
| Á       | UCAA     | Uppercase A with acute accent        | 0100 0001 | 065     | 41  | 101   |
| Â       | UCAC     | Uppercase A with circumflex          | 0100 0010 | 066     | 42  | 102   |
| Ã       | UCAT     | Uppercase A with tilde               | 0100 0011 | 067     | 43  | 103   |
| Ä       | UCAD     | Uppercase A with diaeresis           | 0100 0100 | 068     | 44  | 104   |
| Å       | UCAR     | Uppercase A with ring above          | 0100 0101 | 069     | 45  | 105   |
| Æ       | UCAE     | Uppercase diphthong Æ                | 0100 0110 | 070     | 46  | 106   |
| Ç       | UCCC     | Uppercase C with cedilla             | 0100 0111 | 071     | 47  | 107   |

*Table D-1 The Prime Extended Character Set (continued)*

| <i>Graphic</i> | <i>Mnemonic</i> | <i>Description</i>                      | <i>Binary</i> | <i>Decimal</i> | <i>Hex</i> | <i>Octal</i> |
|----------------|-----------------|-----------------------------------------|---------------|----------------|------------|--------------|
| È              | UCEG            | Uppercase E with grave accent           | 0100 1000     | 072            | 48         | 110          |
| É              | UCEA            | Uppercase E with acute accent           | 0100 1001     | 073            | 49         | 111          |
| Ê              | UCEC            | Uppercase E with circumflex             | 0100 1010     | 074            | 4A         | 112          |
| Ë              | UCED            | Uppercase E with diaeresis              | 0100 1011     | 075            | 4B         | 113          |
| Ì              | UCIG            | Uppercase I with grave accent           | 0100 1100     | 076            | 4C         | 114          |
| Í              | UCIA            | Uppercase I with acute accent           | 0100 1101     | 077            | 4D         | 115          |
| Î              | UCIC            | Uppercase I with circumflex             | 0100 1110     | 078            | 4E         | 116          |
| Ï              | UCID            | Uppercase I with diaeresis              | 0100 1111     | 079            | 4F         | 117          |
| Ð              | UETH            | Uppercase Icelandic letter <u>Eth</u>   | 0101 0000     | 080            | 50         | 120          |
| Ñ              | UCNT            | Uppercase N with tilde                  | 0101 0001     | 081            | 51         | 121          |
| Ò              | UCOG            | Uppercase O with grave accent           | 0101 0010     | 082            | 52         | 122          |
| Ó              | UCOA            | Uppercase O with acute accent           | 0101 0011     | 083            | 53         | 123          |
| Ô              | UCOC            | Uppercase O with circumflex             | 0101 0100     | 084            | 54         | 124          |
| Õ              | OCOT            | Uppercase O with tilde                  | 0101 0101     | 085            | 55         | 125          |
| Ö              | UCOD            | Uppercase O with diaeresis              | 0101 0110     | 086            | 56         | 126          |
| ×              | MULT            | Multiplication sign used in mathematics | 0101 0111     | 087            | 57         | 127          |
| Ø              | UCOO            | Uppercase O with oblique line           | 0101 1000     | 088            | 58         | 130          |
| Ù              | UCUG            | Uppercase U with grave accent           | 0101 1001     | 089            | 59         | 131          |
| Ú              | UCUA            | Uppercase U with acute accent           | 0101 1010     | 090            | 5A         | 132          |
| Û              | UCUC            | Uppercase U with circumflex             | 0101 1011     | 091            | 5B         | 133          |
| Ü              | UCUD            | Uppercase U with diaeresis              | 0101 1100     | 092            | 5C         | 134          |
| Ý              | UCYA            | Uppercase Y with acute accent           | 0101 1101     | 093            | 5D         | 135          |
| Þ              | UTHN            | Uppercase Icelandic letter <u>Thorn</u> | 0101 1110     | 094            | 5E         | 136          |
| ß              | LGSS            | Lowercase German double <u>s</u>        | 0101 1111     | 095            | 5F         | 137          |
| à              | LCAG            | Lowercase a with grave accent           | 0110 0000     | 096            | 60         | 140          |
| á              | LCAA            | Lowercase a with acute accent           | 0110 0001     | 097            | 61         | 141          |
| â              | LCAC            | Lowercase a with circumflex             | 0110 0010     | 098            | 62         | 142          |
| ã              | LCAT            | Lowercase a with tilde                  | 0110 0011     | 099            | 63         | 143          |
| ä              | LCAD            | Lowercase a with diaeresis              | 0110 0100     | 100            | 64         | 144          |
| å              | LCAR            | Lowercase a with a ring above           | 0110 0101     | 101            | 65         | 145          |
| æ              | LCAE            | Lowercase diphthong <u>æ</u>            | 0110 0110     | 102            | 66         | 146          |
| ç              | LCCC            | Lowercase c with cedilla                | 0110 0111     | 103            | 67         | 147          |
| è              | LCEG            | Lowercase e with grave accent           | 0110 1000     | 104            | 68         | 150          |
| é              | LCEA            | Lowercase e with acute accent           | 0110 1001     | 105            | 69         | 151          |
| ê              | LCEC            | Lowercase e with circumflex             | 0110 1010     | 106            | 6A         | 152          |
| ë              | LCED            | Lowercase e with diaeresis              | 0110 1011     | 107            | 6B         | 153          |

Table D-1 The Prime Extended Character Set (continued)

| Graphic | Mnemonic  | Description                             | Binary    | Decimal | Hex | Octal |
|---------|-----------|-----------------------------------------|-----------|---------|-----|-------|
| ì       | LCIG      | Lowercase i with grave accent           | 0110 1100 | 108     | 6C  | 154   |
| í       | LCIA      | Lowercase i with accute accent          | 0110 1101 | 109     | 6D  | 155   |
| î       | LCIC      | Lowercase i with circumflex             | 0110 1110 | 110     | 6E  | 156   |
| ï       | LCID      | Lowercase i with diaeresis              | 0110 1111 | 111     | 6F  | 157   |
| ð       | LETH      | Lowercaxe Icelandic letter <u>Eth</u>   | 0111 0000 | 112     | 70  | 160   |
| ñ       | LCNT      | Lowercase n with tilde                  | 0111 0001 | 113     | 71  | 161   |
| ò       | LCOG      | Lowercase o with grave accent           | 0111 0010 | 114     | 72  | 162   |
| ó       | LCOA      | Lowercase o with acute accent           | 0111 0011 | 115     | 73  | 163   |
| ô       | LCOC      | Lowercase o with circumflex             | 0111 0100 | 116     | 74  | 164   |
| õ       | LCOT      | Lowercase o with tilde                  | 0111 0101 | 117     | 75  | 165   |
| ö       | LCOD      | Lowercase o wiht diaeresis              | 0111 0110 | 118     | 76  | 166   |
| +       | DIV       | Division sign used in mathematics       | 0111 0111 | 119     | 77  | 167   |
| ø       | LCOO      | Lowercase o with oblique line           | 0111 1000 | 120     | 78  | 170   |
| ù       | LCUG      | Lowercase u with grave accent           | 0111 1001 | 121     | 79  | 171   |
| ú       | LCUA      | Lowercase u with acute accent           | 0111 1010 | 122     | 7A  | 172   |
| û       | LCUC      | Lowercase u with circumflex             | 0111 1011 | 123     | 7B  | 173   |
| ü       | LCUD      | Lowercase u wiht diaeresis              | 0111 1100 | 124     | 7C  | 174   |
| ý       | LCYA      | Lowercase y with acute accent           | 0111 1101 | 125     | 7D  | 175   |
| þ       | LTHN      | Lowercase Icelandic letter <u>Thorn</u> | 0111 1110 | 126     | 7E  | 176   |
| ÿ       | LCYD      | Lowercase y with diaeresis              | 0111 1111 | 127     | 7F  | 177   |
|         | NUL       | Null                                    | 1000 0000 | 128     | 80  | 200   |
| ^A      | SOH/TC1   | Start of heading                        | 1000 0001 | 129     | 81  | 201   |
| ^B      | STX/TC2   | Start of text                           | 1000 0010 | 130     | 82  | 202   |
| ^C      | ETX/TC3   | End of text                             | 1000 0011 | 131     | 83  | 203   |
| ^D      | EOT/TC4   | End of transmission                     | 1000 0100 | 132     | 84  | 204   |
| ^E      | ENQ/TC5   | Enquiry                                 | 1000 0101 | 133     | 85  | 205   |
| ^F      | ACK/TC6   | Acknowledge                             | 1000 0110 | 134     | 86  | 206   |
| ^G      | BEL       | Bell                                    | 1000 0111 | 135     | 87  | 207   |
| ^H      | BS/FE0    | Backspace                               | 1000 1000 | 136     | 88  | 210   |
| ^I      | HT/FE1    | Horizontal tab                          | 1000 1001 | 137     | 89  | 211   |
| ^J      | LF/NL/FE2 | Line feed                               | 1000 1010 | 138     | 8A  | 212   |
| ^K      | VT/FE3    | Vertical tab                            | 1000 1011 | 139     | 8B  | 213   |
| ^L      | FF/FE4    | Form feed                               | 1000 1100 | 140     | 8C  | 214   |
| ^M      | CR/FE5    | Carriage return                         | 1000 1101 | 141     | 8D  | 215   |
| ^N      | SO/LS1    | Shift out                               | 1000 1110 | 142     | 8E  | 216   |
| ^O      | SI/LSO    | Shift in                                | 1000 1111 | 143     | 8F  | 217   |

*Table D-1 The Prime Extended Character Set (continued)*

| <i>Graphic</i> | <i>Mnemonic</i> | <i>Description</i>        | <i>Binary</i> | <i>Decimal</i> | <i>Hex</i> | <i>Octal</i> |
|----------------|-----------------|---------------------------|---------------|----------------|------------|--------------|
| ^P             | DLE/TC7         | Data link escape          | 1001 0000     | 144            | 90         | 220          |
| ^Q             | DC1/XON         | Device control 1          | 1001 0001     | 145            | 91         | 221          |
| ^R             | DC2             | Device control 2          | 1001 0010     | 146            | 92         | 222          |
| ^S             | DC3/XOFF        | Device control 3          | 1001 0011     | 147            | 93         | 223          |
| ^T             | DC4             | Device control 4          | 1001 0100     | 148            | 94         | 224          |
| ^U             | NAK/TC8         | Negative acknowledge      | 1001 0101     | 149            | 95         | 225          |
| ^V             | SYN/TC9         | Synchronous idle          | 1001 0110     | 150            | 96         | 226          |
| ^W             | ETB/TC10        | End of transmission block | 1001 0111     | 151            | 97         | 227          |
| ^X             | CAN             | Cancel                    | 1001 1000     | 152            | 98         | 230          |
| ^Y             | EM              | End of medium             | 1001 1001     | 153            | 99         | 231          |
| ^Z             | SUB             | Substitute                | 1001 1010     | 154            | 9A         | 232          |
| ^[             | ESC             | Escape                    | 1001 1011     | 155            | 9B         | 233          |
| ^\             | FS/IS4          | File separator            | 1001 1100     | 156            | 9C         | 234          |
| ^]             | GS/IS3          | Group separator           | 1001 1101     | 157            | 9D         | 235          |
| ^^             | RS/IS2          | Record separator          | 1001 1110     | 158            | 9E         | 236          |
| ^_             | US/IS1          | Unit separator            | 1001 1111     | 159            | 9F         | 237          |
|                | SP              | Space                     | 1010 0000     | 160            | A0         | 240          |
| !              |                 | Exclamation mark          | 1010 0001     | 161            | A1         | 241          |
| "              |                 | Quotation mark            | 1010 0010     | 162            | A2         | 242          |
| #              | NUMB            | Number sign               | 1010 0011     | 163            | A3         | 243          |
| \$             | DOLR            | Dollar sign               | 1010 0100     | 164            | A4         | 244          |
| %              |                 | Percent sign              | 1010 0101     | 165            | A5         | 245          |
| &              |                 | Ampersand                 | 1010 0110     | 166            | A6         | 246          |
| '              |                 | Apostrophe                | 1010 0111     | 167            | A7         | 247          |
| (              |                 | Left parenthesis          | 1010 1000     | 168            | A8         | 250          |
| )              |                 | Right parenthesis         | 1010 1001     | 169            | A9         | 251          |
| *              |                 | Asterisk                  | 1010 1010     | 170            | AA         | 252          |
| +              |                 | Plus sign                 | 1010 1011     | 171            | AB         | 253          |
| ,              |                 | Comma                     | 1010 1100     | 172            | AC         | 254          |
| -              |                 | Minus sign                | 1010 1101     | 173            | AD         | 255          |
| .              |                 | Period                    | 1010 1110     | 174            | AE         | 256          |
| /              |                 | Slash                     | 1010 1111     | 175            | AF         | 257          |
| 0              |                 | Zero                      | 1011 0000     | 176            | B0         | 260          |
| 1              |                 | One                       | 1011 0001     | 177            | B1         | 261          |
| 2              |                 | Two                       | 1011 0010     | 178            | B2         | 262          |
| 3              |                 | Three                     | 1011 0011     | 179            | B3         | 263          |



Table D-1 The Prime Extended Character Set (continued)

| <i>Graphic</i> | <i>Mnemonic</i> | <i>Description</i> | <i>Binary</i> | <i>Decimal</i> | <i>Hex</i> | <i>Octal</i> |
|----------------|-----------------|--------------------|---------------|----------------|------------|--------------|
| 4              |                 | Four               | 1011 0100     | 180            | B4         | 264          |
| 5              |                 | Five               | 1011 0101     | 181            | B5         | 265          |
| 6              |                 | Six                | 1011 0110     | 182            | B6         | 266          |
| 7              |                 | Seven              | 1011 0111     | 183            | B7         | 267          |
| 8              |                 | Eight              | 1011 1000     | 184            | B8         | 270          |
| 9              |                 | Nine               | 1011 1001     | 185            | B9         | 271          |
| :              |                 | Colon              | 1011 1010     | 186            | BA         | 272          |
| ;              |                 | Semicolon          | 1011 1011     | 187            | BB         | 273          |
| <              |                 | Less than sign     | 1011 1100     | 188            | BC         | 274          |
| =              |                 | Equal sign         | 1011 1101     | 189            | BD         | 275          |
| >              |                 | Greater than sign  | 1011 1110     | 190            | BE         | 276          |
| ?              |                 | Question mark      | 1011 1111     | 191            | BF         | 277          |
| @              | AT              | Commercial at sign | 1100 0000     | 192            | C0         | 300          |
| A              |                 | Uppercase A        | 1100 0001     | 193            | C1         | 301          |
| B              |                 | Uppercase B        | 1100 0010     | 194            | C2         | 302          |
| C              |                 | Uppercase C        | 1100 0011     | 195            | C3         | 303          |
| D              |                 | Uppercase D        | 1100 0100     | 196            | C4         | 304          |
| E              |                 | Uppercase E        | 1100 0101     | 197            | C5         | 305          |
| F              |                 | Uppercase F        | 1100 0110     | 198            | C6         | 306          |
| G              |                 | Uppercase G        | 1100 0111     | 199            | C7         | 307          |
| H              |                 | Uppercase H        | 1100 1000     | 200            | C8         | 310          |
| I              |                 | Uppercase I        | 1100 1001     | 201            | C9         | 311          |
| J              |                 | Uppercase J        | 1100 1010     | 202            | CA         | 312          |
| K              |                 | Uppercase K        | 1100 1011     | 203            | CB         | 313          |
| L              |                 | Uppercase L        | 1100 1100     | 204            | CC         | 314          |
| M              |                 | Uppercase M        | 1100 1101     | 205            | CD         | 315          |
| N              |                 | Uppercase N        | 1100 1110     | 206            | CE         | 316          |
| O              |                 | Uppercase O        | 1100 1111     | 207            | CF         | 317          |
| P              |                 | Uppercase P        | 1101 0000     | 208            | D0         | 320          |
| Q              |                 | Uppercase Q        | 1101 0001     | 209            | D1         | 321          |
| R              |                 | Uppercase R        | 1101 0010     | 210            | D2         | 322          |
| S              |                 | Uppercase S        | 1101 0011     | 211            | D3         | 323          |
| T              |                 | Uppercase T        | 1101 0100     | 212            | D4         | 324          |
| U              |                 | Uppercase U        | 1101 0101     | 213            | D5         | 325          |
| V              |                 | Uppercase V        | 1101 0110     | 214            | D6         | 326          |
| W              |                 | Uppercase W        | 1101 0111     | 215            | D7         | 327          |

*Table D-1 The Prime Extended Character Set (continued)*

| <i>Graphic</i> | <i>Mnemonic</i> | <i>Description</i>              | <i>Binary</i> | <i>Decimal</i> | <i>Hex</i> | <i>Octal</i> |
|----------------|-----------------|---------------------------------|---------------|----------------|------------|--------------|
| X              |                 | Uppercase X                     | 1101 1000     | 216            | D8         | 330          |
| Y              |                 | Uppercase Y                     | 1101 1001     | 217            | D9         | 331          |
| Z              |                 | Uppercase Z                     | 1101 1010     | 218            | DA         | 332          |
| [              | LBKT            | Left bracket                    | 1101 1011     | 219            | DB         | 333          |
| \              | REVS            | Reverse slash, backslash        | 1101 1100     | 220            | DC         | 334          |
| ]              | RBKT            | Right bracket                   | 1101 1101     | 221            | DD         | 335          |
| ^              | CFLX            | Circumflex                      | 1101 1110     | 222            | DE         | 336          |
| _              |                 | Underline, underscore           | 1101 1111     | 223            | DF         | 337          |
| `              | GRAV            | Left single quote, grave accent | 1110 0000     | 224            | E0         | 340          |
| a              |                 | Lowercase a                     | 1110 0001     | 225            | E1         | 341          |
| b              |                 | Lowercase b                     | 1110 0010     | 226            | E2         | 342          |
| c              |                 | Lowercase c                     | 1110 0011     | 227            | E3         | 343          |
| d              |                 | Lowercase d                     | 1110 0100     | 228            | E4         | 344          |
| e              |                 | Lowercase e                     | 1110 0101     | 229            | E5         | 345          |
| f              |                 | Lowercase f                     | 1110 0110     | 230            | E6         | 346          |
| g              |                 | Lowercase g                     | 1110 0111     | 231            | E7         | 347          |
| h              |                 | Lowercase h                     | 1110 1000     | 232            | E8         | 350          |
| i              |                 | Lowercase i                     | 1110 1001     | 233            | E9         | 351          |
| j              |                 | Lowercase j                     | 1110 1010     | 234            | EA         | 352          |
| k              |                 | Lowercase k                     | 1110 1011     | 235            | EB         | 353          |
| l              |                 | Lowercase l                     | 1110 1100     | 236            | EC         | 354          |
| m              |                 | Lowercase m                     | 1110 1101     | 237            | ED         | 355          |
| n              |                 | Lowercase n                     | 1110 1110     | 238            | EE         | 356          |
| o              |                 | Lowercase o                     | 1110 1111     | 239            | EF         | 357          |
| p              |                 | Lowercase p                     | 1111 0000     | 240            | F0         | 360          |
| q              |                 | Lowercase q                     | 1111 0001     | 241            | F1         | 361          |
| r              |                 | Lowercase r                     | 1111 0010     | 242            | F2         | 362          |
| s              |                 | Lowercase s                     | 1111 0011     | 243            | F3         | 363          |
| t              |                 | Lowercase t                     | 1111 0100     | 244            | F4         | 364          |
| u              |                 | Lowercase u                     | 1111 0101     | 245            | F5         | 365          |
| v              |                 | Lowercase v                     | 1111 0110     | 246            | F6         | 366          |
| w              |                 | Lowercase w                     | 1111 0111     | 247            | F7         | 367          |
| x              |                 | Lowercase x                     | 1111 1000     | 248            | F8         | 370          |
| y              |                 | Lowercase y                     | 1111 1001     | 249            | F9         | 371          |
| z              |                 | Lowercase z                     | 1111 1010     | 250            | FA         | 372          |
| {              | LBCE            | Left brace                      | 1111 1011     | 251            | FB         | 373          |



Table D-2. EBCDIC Character Set

| Decimal | Octal | Hex. | Char. | Decimal | Octal | Hex. | Char. |
|---------|-------|------|-------|---------|-------|------|-------|
| 000     | 000   | 00   | NUL   | 048     | 060   | 30   |       |
| 001     | 001   | 01   | SOH   | 049     | 061   | 31   |       |
| 002     | 002   | 02   | STX   | 050     | 062   | 32   | SYN   |
| 003     | 003   | 03   | ETX   | 051     | 063   | 33   |       |
| 004     | 004   | 04   | PF    | 052     | 064   | 34   | PN    |
| 005     | 005   | 05   | HT    | 053     | 065   | 35   | RS    |
| 006     | 006   | 06   | LC    | 054     | 066   | 36   | UC    |
| 007     | 007   | 07   | DEL   | 055     | 067   | 37   | EOT   |
| 008     | 010   | 08   |       | 056     | 070   | 38   |       |
| 009     | 011   | 09   |       | 057     | 071   | 39   |       |
| 010     | 012   | 0A   | SMM   | 058     | 072   | 3A   |       |
| 011     | 013   | 0B   | VT    | 059     | 073   | 3B   | CU3   |
| 012     | 014   | 0C   | FF    | 060     | 074   | 3C   | DC4   |
| 013     | 015   | 0D   | CR    | 061     | 075   | 3D   | NAK   |
| 014     | 016   | 0E   | SO    | 062     | 076   | 3E   |       |
| 015     | 017   | 0F   | SI    | 063     | 077   | 3F   | SUB   |
| 016     | 020   | 10   | DLE   | 064     | 100   | 40   | Sp    |
| 017     | 021   | 11   | DC1   | 065     | 101   | 41   |       |
| 018     | 022   | 12   | DC2   | 066     | 102   | 42   |       |
| 019     | 023   | 13   | TM    | 067     | 103   | 43   |       |
| 020     | 024   | 14   | RES   | 068     | 104   | 44   |       |
| 021     | 025   | 15   | NL    | 069     | 105   | 45   |       |
| 022     | 026   | 16   | BS    | 070     | 106   | 46   |       |
| 023     | 027   | 17   | IL    | 071     | 107   | 47   |       |
| 024     | 030   | 18   | CAN   | 072     | 110   | 48   |       |
| 025     | 031   | 19   | EM    | 073     | 111   | 49   |       |
| 026     | 032   | 1A   | CC    | 074     | 112   | 4A   |       |
| 027     | 033   | 1B   | CU1   | 075     | 113   | 4B   |       |
| 028     | 034   | 1C   | IFS   | 076     | 114   | 4C   | <     |
| 029     | 035   | 1D   | IGS   | 077     | 115   | 4D   | (     |
| 030     | 036   | 1E   | IRS   | 078     | 116   | 4E   | +     |
| 031     | 037   | 1F   | IUS   | 079     | 117   | 4F   |       |
| 032     | 040   | 20   | DS    | 080     | 120   | 50   | &     |
| 033     | 041   | 21   | SOS   | 081     | 121   | 51   |       |
| 034     | 042   | 22   | FS    | 082     | 122   | 52   |       |
| 035     | 043   | 23   |       | 083     | 123   | 53   |       |

Table D-2. EBCDIC Character Set (continued)

| <i>Decimal</i> | <i>Octal</i> | <i>Hex.</i> | <i>Char.</i> | <i>Decimal</i> | <i>Octal</i> | <i>Hex.</i> | <i>Char.</i> |
|----------------|--------------|-------------|--------------|----------------|--------------|-------------|--------------|
| 036            | 044          | 24          | BYP          | 084            | 124          | 54          |              |
| 037            | 045          | 25          | LF           | 085            | 125          | 55          |              |
| 038            | 046          | 26          | ETB          | 086            | 126          | 56          |              |
| 039            | 047          | 27          | ESC          | 087            | 127          | 57          |              |
| 040            | 050          | 28          |              | 088            | 130          | 58          |              |
| 041            | 051          | 29          |              | 089            | 131          | 59          |              |
| 042            | 052          | 2A          | SM           | 090            | 132          | 5A          | !            |
| 043            | 053          | 2B          | CU2          | 091            | 133          | 5B          | \$           |
| 044            | 054          | 2C          |              | 092            | 134          | 5C          | *            |
| 045            | 055          | 2D          | ENQ          | 093            | 135          | 5D          | )            |
| 046            | 056          | 2E          | ACK          | 094            | 136          | 5E          | ;            |
| 047            | 057          | 2F          | BEL          | 095            | 137          | 5F          |              |
| 096            | 140          | 60          | -            | 143            | 217          | 8F          |              |
| 097            | 141          | 61          | /            | 144            | 220          | 90          |              |
| 098            | 142          | 62          |              | 145            | 221          | 91          | j            |
| 099            | 143          | 63          |              | 146            | 222          | 92          | k            |
| 100            | 144          | 64          |              | 147            | 223          | 93          | l            |
| 101            | 145          | 65          |              | 148            | 224          | 94          | m            |
| 102            | 146          | 66          |              | 149            | 225          | 95          | n            |
| 103            | 147          | 67          |              | 150            | 226          | 96          | o            |
| 104            | 150          | 68          |              | 151            | 227          | 97          | p            |
| 105            | 151          | 69          |              | 152            | 230          | 98          | q            |
| 106            | 152          | 6A          |              | 153            | 231          | 99          | r            |
| 107            | 153          | 6B          | ,            | 154            | 232          | 9A          |              |
| 108            | 154          | 6C          | %            | 155            | 233          | 9B          |              |
| 109            | 155          | 6D          | _            | 156            | 234          | 9C          |              |
| 110            | 156          | 6E          | >            | 157            | 235          | 9D          |              |
| 111            | 157          | 6F          | ?            | 158            | 236          | 9E          |              |
| 112            | 160          | 70          |              | 159            | 237          | 9F          |              |
| 113            | 161          | 71          |              | 160            | 240          | A0          |              |
| 114            | 162          | 72          |              | 161            | 241          | A1          | ~            |
| 115            | 163          | 73          |              | 162            | 242          | A2          | s            |
| 116            | 164          | 74          |              | 163            | 243          | A3          | t            |
| 117            | 165          | 75          |              | 164            | 244          | A4          | u            |
| 118            | 166          | 76          |              | 165            | 245          | A5          | v            |
| 119            | 167          | 77          |              | 166            | 246          | A6          | w            |

*Table D-2. EBCDIC Character Set (continued)*

| <i>Decimal</i> | <i>Octal</i> | <i>Hex.</i> | <i>Char.</i> | <i>Decimal</i> | <i>Octal</i> | <i>Hex.</i> | <i>Char.</i> |
|----------------|--------------|-------------|--------------|----------------|--------------|-------------|--------------|
| 120            | 170          | 78          |              | 167            | 247          | A7          | x            |
| 121            | 171          | 79          | '            | 168            | 250          | A8          | y            |
| 122            | 172          | 7A          | :            | 169            | 251          | A9          | z            |
| 123            | 173          | 7B          | #            | 170            | 252          | AA          |              |
| 124            | 174          | 7C          | @            | 171            | 253          | AB          |              |
| 125            | 175          | 7D          | ,            | 172            | 254          | AC          |              |
| 126            | 176          | 7E          | =            | 173            | 255          | AD          |              |
| 127            | 177          | 7F          | "            | 174            | 256          | AE          |              |
| 128            | 200          | 80          |              | 175            | 257          | AF          |              |
| 129            | 201          | 81          | a            | 176            | 260          | B0          |              |
| 130            | 202          | 82          | b            | 177            | 261          | B1          |              |
| 131            | 203          | 83          | c            | 178            | 262          | B2          |              |
| 132            | 204          | 84          | d            | 179            | 263          | B3          |              |
| 133            | 205          | 85          | e            | 180            | 264          | B4          |              |
| 134            | 206          | 86          | f            | 181            | 265          | B5          |              |
| 135            | 207          | 87          | g            | 182            | 266          | B6          |              |
| 136            | 210          | 88          | h            | 183            | 267          | B7          |              |
| 137            | 211          | 89          | i            | 184            | 270          | B8          |              |
| 138            | 212          | 8A          |              | 185            | 271          | B9          |              |
| 139            | 213          | 8B          |              | 186            | 272          | BA          |              |
| 140            | 214          | 8C          |              | 187            | 273          | BB          |              |
| 141            | 215          | 8D          |              | 188            | 274          | BC          |              |
| 142            | 216          | 8E          |              | 189            | 275          | BD          |              |
| 190            | 276          | BE          |              | 223            | 337          | DF          |              |
| 191            | 277          | BF          |              | 224            | 340          | E0          |              |
| 192            | 300          | C0          | {            | 225            | 341          | E1          |              |
| 193            | 301          | C1          | A            | 226            | 342          | E2          | S            |
| 194            | 302          | C2          | B            | 227            | 343          | E3          | T            |
| 195            | 303          | C3          | C            | 228            | 344          | E4          | U            |
| 196            | 304          | C4          | D            | 229            | 345          | E5          | V            |
| 197            | 305          | C5          | E            | 230            | 346          | E6          | W            |
| 198            | 306          | C6          | F            | 231            | 347          | E7          | X            |
| 199            | 307          | C7          | G            | 232            | 350          | E8          | Y            |
| 200            | 310          | C8          | H            | 233            | 351          | E9          | Z            |
| 201            | 311          | C9          | I            | 234            | 352          | EA          |              |
| 202            | 312          | CA          |              | 235            | 353          | EB          |              |

Table D-2. EBCDIC Character Set (continued)

| Decimal | Octal | Hex. | Char.  | Decimal | Octal | Hex. | Char.  |
|---------|-------|------|--------|---------|-------|------|--------|
| 203     | 313   | CB   |        | 236     | 354   | EC   | (bank) |
| 204     | 314   | CC   | (bank) | 237     | 355   | ED   |        |
| 205     | 315   | CD   |        | 238     | 356   | EE   |        |
| 206     | 316   | CE   | (bank) | 239     | 357   | EF   |        |
| 207     | 317   | CF   |        | 240     | 360   | F0   | 0      |
| 208     | 320   | D0   | )      | 241     | 361   | F1   | 1      |
| 209     | 321   | D1   | J      | 242     | 362   | F2   | 2      |
| 210     | 322   | D2   | K      | 243     | 363   | F3   | 3      |
| 211     | 323   | D3   | L      | 244     | 364   | F4   | 4      |
| 212     | 324   | D4   | M      | 245     | 365   | F5   | 5      |
| 213     | 325   | D5   | N      | 246     | 366   | F6   | 6      |
| 214     | 326   | D6   | O      | 247     | 367   | F7   | 7      |
| 215     | 327   | D7   | P      | 248     | 370   | F8   | 8      |
| 216     | 330   | D8   | Q      | 249     | 371   | F9   | 9      |
| 217     | 331   | D9   | R      | 250     | 372   | FA   |        |
| 218     | 332   | DA   |        | 251     | 373   | FB   |        |
| 219     | 333   | DB   |        | 252     | 374   | FC   |        |
| 220     | 334   | DC   |        | 253     | 375   | FD   |        |
| 221     | 335   | DD   |        | 254     | 376   | FE   |        |
| 222     | 336   | DE   |        | 255     | 377   | FF   |        |

*Replace this page with the tab page labeled*  
**Acronyms**



*Replace this page with the tab page labeled*

**Glossary**

# Glossary

■ ■ ■ ■ ■ ■

## **absolute pathnames**

Pathnames that begin with root (<). Absolute pathnames are fully qualified pathnames.

## **access control list (ACL)**

A list of users and their access rights to file system objects as produced by the LIST\_ACCESS command.

## **ACL**

*See* access control list.

## **added partition**

A PRIMOS file system partition that is added to the system, or started, by the ADDISK command for user input.

## **Assignable Disks Table**

A table kept by PRIMOS that lists the pdevs of disks that may be assigned by a single user.

## **assignable disks**

Disks listed in the Assignable Disks Table by pdev and that may be assigned by a single user.

## **assigned partition**

A partition that has been assigned to one user for that user's exclusive use and is unavailable as a file system partition.

## **badspot**

A physical defect in the disk media that prevents data from being correctly read from or written to the disk. It is identified by either a record address within the partition or by a combination of head, cylinder, and sector number.

## **BADSPT file**

The Nondynamic Badspot Handling (-DBS OFF) mode file listing badspots on a partition. There is one MFD>BADSPT file per partition if there are badspots on



**DBS**

The dynamic badspot file. This file contains addresses of all the known badspots for an entire physical disk. It also contains a list of all of the available remapping records. All badspots are matched to a remapping record. Additional remapping records are available for new, or dynamically occurring, badspots. *See also* dynamic badspot.

**Direct Access Method (DAM)**

A method of allocating and storing records in a file. PRIMOS accesses the records by reading an index to them.

**disk**

Generally used to refer to a partition but also used as a term for a disk drive and disk pack.

**disk drive**

The peripheral device that contains the physical disks and the hardware and electronic circuitry to accomplish reading and writing on the physical disk surfaces. The disk drive may be external to the system or it may be internal (as in the case of 2455 systems). Also referred to as a disk storage device and drive unit.

**disk geometry**

The physical attributes of a physical disk such as the number of cylinders, or tracks, per surface; the numbers of sectors, or records, per track; and the number of surfaces. These attributes are defined by a physical device number for disk partitions.

**disk formatting**

Preparing disks for use by PRIMOS. Disk controllers store the location and any auxiliary information in the disk header for each sector on the disk. Cylinder, head (or surface), and sector values are stored with each sector. Cylinders are numbered from the outermost to the innermost. Heads are numbered from the top surface to the bottom surface on the disk. Sectors within a track are currently numbered in a clockwise, or forward, order from sector 0 to the maximum number of sectors per track minus one.

**Disk Information Table**

A table of logical device numbers (ldevs) consisting of four arrays of 62 words each containing this information for each partition:

- pdev for each ldev
- Sectors per track
- Total number of records
- Number of words per record

**disk mirroring**

The creation of two logically equivalent partitions that store the same data such that, if either partition fails, the other can be used in its place.

**disk pack**

The physical disks that are removable from a disk drive as in the case of the 80MB and 300MB SMDs. Also used to refer to physical disks in general. *See also* disk drive.

**DSKRAT**

The Disk Record Availability Table. The DSKRAT contains disk geometry information for each partition and bit positions for each record on the disk. These bits are either set (=1) indicating that a record is available for file system use or are reset (=0) indicating that a record is in use.

**dual-ported disk drive**

A disk drive that can be attached to two systems simultaneously although only one system has control over the disk drive at any moment.

**dynamic badspot**

A badspot on a disk that was either not found and remapped previously or developed dynamically due to progressive media degradation. *See also* DBS.

**dynamic badspot handling**

The process whereby a disk controller, upon detecting a badspot, remaps the record containing the badspot to another good record on the partition.

**Dynamic Badspot Handling (-DBS ON) mode**

A state of a disk that allows intelligent disk controllers to handle badspots and to allow mirroring on these partitions. This disk mode is not compatible with nonintelligent controllers.

**equivalence blocks**

Describes where one record is actually stored on the partition. There is one equivalence block for every remapped record on a partition. These equivalence blocks are stored in the BADSPT file of the target partition. They are created by COPY\_DISK and PHYRST in order to indicate that badspot handling has taken place for the partition to which data were copied (the target partition). Until the equivalence blocks are deleted by FIX\_DISK, the partition must not be used for any purpose.

**extent**

Groups of contiguous records in CAM files.

**extent map**

An index of the extents in a CAM file used by PRIMOS to locate and retrieve CAM file records.

**external commands**

Programs that are stored in a special top-level named CMDNCO.

**fast FIX\_DISK**

FIX\_DISK with the --FAST option. Fast FIX\_DISK should be used only on robust partitions. The use of the --FAST option causes FIX\_DISK to check only directory entries, including CAM file extent maps, the DSKRAT, and the quota system on robust partitions.

**file system disk**

A logical disk, or partition, used by PRIMOS to store system and user files.

**first partition**

The partition of a physical disk that contains the first surface (starting surface 0) of the disk; thus, the first four bits of its pdev are 0. On a physical disk partitioned by MAKE at Rev. 21.0 and later, the first partition contains the dynamic badspot file (DBS) and the remapped area (RMA) for all the partitions on that physical disk.

**Fixed-Media Disk**

*See FMD.*

**flaw**

A badspot; an area of the physical disk that cannot store data.

**flaw map**

A list of flaws provided by the disk manufacturer and written on an unused cylinder of the disk. The flaw map is available for MAKE to read. Also sometimes refers to a list of badspots written on paper and affixed to the physical disk by the disk manufacturer. The Operator can then enter these badspots manually by using the appropriate MAKE or FIX\_DISK options.

**FMD**

Fixed-Media Disk; a type of physical disk for file system storage that includes the sealed storage media and the disk drive. Sometimes referred to as a Winchester disk.

**formatting**

Using MAKE to prepare a physical disk for file system or paging use. MAKE writes physical record headers onto the partition that are recognizable to PRIMOS.

**forward sectoring**

A method of file record allocation used by PRIMOS in which the next record to be allocated is three sectors forward of the last record; the interleave factor is 3. *See also* interleaving; interleave factor; reverse sectoring.

**full FIX\_DISK**

FIX\_DISK without the -FAST option. Full FIX\_DISK checks (and repairs if you use the -FIX option) the entire file system.

**fully qualified pathnames**

Any pathname that explicitly or implicitly starts with the root entry name or disk partition name. Fully qualified pathnames are unambiguous and do not use search rules. In a multi-rooted file system, pathnames begin with the name of the disk partition. In a singly-rooted file system, pathnames begin with the name of the root directory.

**Global Mount Table (GMT)**

Contains a list of all disk partitions and their mount-point pathnames.

**GMT**

See Global Mount Table.

**hashing**

The use of an algorithm by PRIMOS to rapidly access data or records within a partition. Directories on robust partitions and non-ACL directories are not hashed; only ACL-protected directories are hashed.

**head**

The physical device that reads data from the disk surfaces; newer disks may contain more than one read head per disk surface. This term is also used synonymously with *surface* when referring to the number of surfaces in a partition.

**intelligent disk controller**

A microprocessor-based disk controller that is capable of buffering data, of using algorithms to perform the read and write operations on a disk, and of dynamically remapping badspots that occur on the disk. A nonintelligent disk controller does not have these capabilities. An intelligent disk controller must be used for disk mirroring because it provides dynamic badspot handling.

**interleave factor**

The sector gap between consecutively allocated records. It is 3 for forward sectoring and 1 for reverse sectoring. See also interleaving; forward sectoring; reverse sectoring.

**interleaving**

The order of writing records to disk so as to maximize the potential for the sequentially next record of a file to be under the read head of the disk after processing of the current record is complete. See also interleave factor.

**internal commands**

Part of PRIMOS itself.

**Idev**

An octal number from 0 through 355 (0 through 237, decimal) that is assigned to a partition when the partition is started by the ADDISK or the STARTUP command. It also indicates the location of the pdev of the added PRIMOS file system partition in the Disk Information Table.

**List File**

A file you create prior to making a system boot tape. It contains the pathnames of all the directories and files necessary to restore your system to normal working order.

**local partitions**

Partitions that are connected to your system. *See also* partition.

**logical device number**

*See* Idev.

**logical disk**

Synonymous with partition or logical device. A logical division of a physical disk used for file storage or for paging.

**logical file type**

What a subroutine or utility creating a file sets the file type to be as opposed to how the file is physically arranged on the storage media (the *physical* file type). For example, all user files on a robust partition are physically arranged as CAM files but the software creating the file may set the file type to SAM or DAM; thus, the files are logically created as SAM or DAM files.

**logical save**

Saving of records as logical entities such as files as opposed to a physical save. The MAGSAV and MAGRST utilities save and restore records logically. *See also* physical save.

**Master File Directory (MFD)**

The highest level directory on a partition; each partition contains one MFD. The MFD contains a file that is an index to each top-level directory and file in the partition, or MFD. Also refers to the partition itself.

**MFD**

*See* Master File Directory.

**mount-point pathnames**

Disk partition directory names of up to 32 characters. Although entries in the root directory represent disk partitions, they can have names other than the disk name. Therefore, the disk name is no longer a component in the pathname.





**partitioning**

Use of MAKE to format physical disks. The disks may be segregated into logical divisions called partitions. A partition may contain the entire physical disk or the disk may be divided into many partitions.

**pdev**

A 16-bit octal number that defines to the file system a range of surfaces as a logical partition of a physical disk and that specifies the disk controller address and a disk drive unit number. The location and size of a partition are described by starting surface (surface offset), number of surfaces, drive unit number, and controller address.

**physical device number**

See pdev.

**physical disk**

An entire multi-surface disk (SMD, CMD, or FMD) containing 1 through  $n$  partitions.

**physical file type**

How the file is physically organized on the disk as opposed to what the logical file type is set to by the routine creating the file. When a command such as LD lists file type, the type listed is the *logical* file type.

**physical save**

Saving of records in the order that they are stored on the disk without consideration for what file they belong to. The utilities PHYSAV and COPY\_DISK use a physical save. See also logical save.

**portal**

A file system object that serves as a gateway to another file system name space.

**primary partition**

The main partition of a mirrored pair of partitions; the partition from which a catch-up copy is made. See also secondary partition.

**RAT**

The Record Availability Table, which contains a header that describes the partition and a bit map that indicates which records are available for use and which records are in use. Synonymous with DSKRAT.

**Recovery Based File (RBF)**

A type of ROAM file.

**remapped area (RMA)**

An area of the first partition on a physical disk that is set aside to contain records that would be written into badspots but that are instead written to the RMA by an

intelligent disk controller. This area of the disk is normally accessed only by the intelligent controller but is also accessed by FIX\_DISK when converting to Nondynamic Badspot Handling (-AC) mode, in which case these records are read directly before their pointers are restrung into their parent file. The RMA records are marked as in-use in the DSKRAT and are never directly accessed by PRIMOS. The RMA records are full disk records that contain parts of various files that the file system initially attempted to write to badspots.

**RMA**

See remapped area.

**remote partitions**

Partitions that are connected to other systems in the network of which your system is a part. See also partition.

**reverse sectoring**

A method of file record allocation used by PRIMOS in which the next record to be allocated is one sector behind the last record such that logically contiguous records are adjacent to one another. The interleave factor in this case is 1. See also interleaving; interleave factor; forward sectoring.

**robust partition**

A PRIMOS file system partition that contains CAM files only and that is designed to be less subject to disk errors resulting from system halts and that can generally be rapidly repaired by using fast FIX\_DISK.

**root directory**

A directory, designated as (<), that resides at the uppermost level of the file system hierarchy (above the MFD). It contains only directories that correspond to the MFDs of local and remote disk partitions.

**root entryname**

The mount-point directory of a mounted disk partition (local or remote). It corresponds to the symbolic PRIMOS directory referred to as an MFD. All directory names following the root symbol in an absolute pathname are root entrynames.

**SAM**

See Sequential Access Method.

**secondary partition**

The alternate partition of a mirrored pair of partitions and the partition that the primary partition is copied during the catch-up copy process. See also primary partition.

**sector**

A portion of a track on the surface of a disk. A sector contains one record, or

block, of data and, on PRIMOS disks, contains 2048 bytes of user data and 32 bytes of housekeeping data.

**SEGDIR**

*See* Segment Directory.

**Segment Directory (SEGDIR)**

Contains entries referenced by file numbers from 1 through 6535 rather than by file names. File are referred to as subfiles. Generally used by programs rather than by users.

**Sequential Access Method (SAM)**

A method of allocating and storing records in a file. The file records are accessed sequentially such that to get to a record in a file, all previous records in the file must be read by PRIMOS.

**singly-rooted file system name space**

The PRIMOS file system structure at Rev. 23.0 and subsequent revisions. It is a structure where all file system objects, no matter where they are located, stem from a single root directory instead of many disk partitions.

**SMD**

Storage module disk; a type of physical disk for file system storage that can be removed from the disk drive.

**split partition**

A logical partition that that has part of its storage space reserved for file system use and part reserved for paging use. *See also* paging partition.

**standalone**

Refers to a program that can be booted to run by itself without the services of PRIMOS. An example is MAKE.SAVE.

**standard partition**

A nonrobust partition. The type of PRIMOS file system partition always created prior to Rev. 22.1. Full FIX\_DISK must be used to repair standard partitions.

**static badspot**

A badspot that is present on the disk surface and that is detected by MAKE when the partition is first created.

**Storage Module Disk**

*See* SMD.

**supervisor terminal**

A term that refers to both the physical supervisor terminal and a logical supervisor terminal that has been enabled by the RESUS command.

**surface**

The magnetic area of a disk where data is actually stored (written to) and retrieved (read from). Groups of surfaces constitute partitions and all partitions on a physical disk except the last must contain an even number of surfaces.

**survivor**

Describes the most up-to-date and usable partition of a mirrored pair when the mirror breaks.

**tape index**

A list of files that have been saved by MAGSAV on a tape.

**top-level directory**

The directories at the highest level in the file system tree structure immediately inferior to the Master File Directory (MFD). These directories contain files and other directories referred to as subdirectories. CMDNCO is a top-level directory on the command device.

**user commands**

Commands that are generally available to any user and which affect only that user's work.

**user disks**

Disks or partitions used for the storage and retrieval of user files. These disks are prepared for use by the MAKE utility.

**virtual memory**

Disk storage memory that is used by PRIMOS in the paging process with the result that the system appears to have considerably more physical memory than actually exists. Use of virtual memory provides each user with 512MB of virtual address space. *See also* paging.

**volume**

Term used synonymously with disks and partitions. *Volume* may also refer to an entire physical disk as one logical disk.

**Winchester disk**

A sealed disk subsystem in which the physical disks and their associated disk drive and circuitry are all contained. A Fixed-Media Disk (FMD).

Replace this page with

tab page labeled

Index

Index

.....





ACL protection, displaying at prompt, 2-286

**ACLs**

- creating, 2-99, 2-314
- editing, 2-98
- listing, 2-178
- listing priority, 2-195
- setting, 2-311
- specific, 2-311, 2-312
- users, deleting from, 2-99

ADD\_REMOTE\_ID command, 2-14

Address space, user, clearing, 2-123

AFTER function, 3-4

-AFTER wildcard option, 4-10

ALL access right, 2-178

Angle brackets, in pathnames, 1-4

ARCHIVE command, C-1

ARCHIVE tapes, C-3, C-10, C-12, C-13

ARCHIVE\_RELEASE command, C-2

ARCHIVE\_RESTORE command, C-3

Argument abbreviations, 2-8

Arguments, command line format, 1-1

ARID command, 2-14

Arithmetic command functions, summary of, 1-26

ASCII files, writing text into, 3-49

ASRCWD command, C-4

Assembler, invoking, 2-273

ASSIGN command, 2-15

Assignable Disks Table, 2-21

Assigned devices, assigned to users, listing of, 2-211

Assigning asynchronous lines, 2-22

Assigning disks, 2-21

Assigning lines, 2-22

Assigning magnetic tapes, 2-17

Assigning NTS lines, 2-22

Assigning peripheral devices, 2-15

Assigning plotters, C-16

Assigning SMLC lines, 2-22

Asterisk (\*) command, 2-405

Asterisk in pathnames, 1-5

Asynchronous lines

- assigning, 2-22
- characteristics of, 2-318
- configuring, 2-317

- default settings for, 2-322
- setting characteristics of, 2-386
- unassigning, 2-391

ATM command, 2-23

ATTACH command, 2-23, C-5

pathname qualification for, 1-5

Attach points

- changing projects, 2-43
- current, function for returning, 3-5
- displaying at prompt, 2-285
- function for returning, 3-5
- origin, 2-43

ATTACH\_POINT function, 3-5

Attaching to origin directory, 2-263

ATTRIB function, 3-5

using on files, 3-5

Attribute files, 2-357

Attributes, definition of, 2-357

Automatic pathname completion, 2-111, 6-3

AVAIL command, 2-28

**B**

B command, 2-36

back\_char, 6-12, 6-37

back\_word, 6-37

-BACKEDUP\_AFTER wildcard option, 4-10

-BACKEDUP\_BEFORE wildcard option, 4-10

Backslash character (card option), 6-11

BACKUP tapes, C-10, C-12, C-13

Basename, 1-6

BASIC command, 2-31

double-precision interpreter, 2-83

BASIC interpreter, 2-31

BASIC programs

- executing, 2-31, 2-83
- renumbering, 2-258
- running, 2-31, 2-83

BASICV command, 2-32

Batch jobs

- aborting, 2-148
- canceling, 2-148
- displaying, 2-212

information on, 2-33

managing submitted, 2-150

modifying, 2-151

monitoring, 2-151

running, listing of, 2-212

storing submittal options, 2-150

submitting, 2-149

Batch monitor, passing parameters to, 2-150

Batch subsystem, project assignment, 2-43

BATGEN command, 2-34

BEFORE function, 3-7

-BEFORE wildcard option, 4-10

begin\_line, 6-9, 6-37

BINARY command, 2-36

Binary editor, C-9

Binary libraries, creating and modifying of, 2-101

Binary output files, opening, 2-36

BIND command, 2-37

Bindings

- finding current, 6-27
- reestablishing fundamental, 6-26

Bindings file

- compiling, 6-25
- definition of, 6-9
- PST 100, 6-18
- PT200, 6-18
- PT45, 6-24
- restoring, 6-26
- sample, 6-17
- saving, 6-26
- saving and restoring simultaneously, 6-27
- syntax of, 6-9

Boot tapes, creating, 2-38

-BOTTOM\_UP treewalking option, 4-19

Brackets, square, 3-1

BREAK key, enabling or disabling, 2-385, 6-2

Breaks in ring, finding, 2-125

Buffers

- See also Locate buffers
- terminal, discarding, 2-302
- terminal, emptying, 2-302

Building concatenated files, 2-62

**C**

- C command, 2-48
- C compiler, 2-40
- CAD/CAM commands, summary of, 1-25
- CALC function, 3-7
- Canceling spooled files, 2-359
- Card punch
  - assigning, 2-16
  - punching files onto, C-7
- Card reader
  - assigning, 2-16
  - reading cards from, C-8
- caret (^), 6-11
- Carriage return
  - in MESSAGE command, 2-238
  - in prompts, 2-286
- Case in command line, 1-2
- Catalogs, tape, C-12
- Categoryname, 1-6
- CBL command, 2-38
- CBLDML command, 2-38
- CBLSUBS command, 2-39
- CC command, 2-40
- Chaining COMI files, 2-57
- CHANGE\_PASSWORD command, 2-41
- CHANGE\_PROJECT command, 2-43
  - command line restriction, 5-4
- Changing to another directory, 2-23
- CHAP command, 2-45
- Character strings
  - character substitution function, 3-40
  - content, checking validity of, 3-44
  - function for checking characters
    - absence in, 3-45
  - function to return length of, 3-25
  - function to return substrings of, 3-36
  - function to test for empty (NULL), 3-26
  - function to trim characters from, 3-40
  - indexing, 3-35
  - reversing, 3-35
  - string substitution in, 3-36
- Child processes
  - running, display of, 2-212
  - running, list of, 2-212
- Clearing user address space, 2-123
- CLOSE command, 2-48
- Closing files, 2-48
  - summary of commands for, 1-14
- CLUP command, 2-49
- CMDNC0 directory, 1-7
- CMPF command, 2-50
- CN\_RBF command, 2-53
- CNAME command, 2-52
- CND\_INFO function, 3-9
- CO command, 2-55
- COBOL 74
  - compiler, 2-38
  - DDL compiler, 2-38
  - DML preprocessor, 2-38
- COBOL85 command, 2-54
- COBOL85 compiler, 2-54
- COBOL85 DDL compiler, 2-54
- Cold starts
  - disks inactive since, 3-37
  - time since, 3-38
- collect\_macro, 6-37
- COMI files
  - \$\$ command in, 2-150
  - as phantoms, 2-269
  - chaining, 2-57
  - closing, 2-55
  - executing, 2-56
  - pausing, 2-55
  - running, 2-55
  - used to submit batch jobs, 2-144
- COMINPUT command, 2-55
  - command line restriction, 5-4
- Command abbreviations, defining, 2-8
- Command directory (CMDNC0), 1-7
- Command environment
  - listing attributes of, 2-190
  - resetting, 2-140
  - summary of commands to define, 1-18
- Command execution, summary of
  - commands to get information about, 1-20
- Command functions
  - characters, square brackets [ ], 3-1
  - command line processing, 5-5
  - containing abbreviations, 5-3
  - containing variables, 4-3, 5-5
  - format, 3-1
  - nested, 5-5
  - string-handling, summary, 1-27
  - summary of, 1-26
  - suppressing, 4-24
  - within iteration lists, 4-24
- Command history, 2-109, 6-41
- Command levels, releasing, 2-290
- Command line
  - ; character, 4-3, 5-4
  - % characters, 4-3
  - ~ character, 5-4
  - ABBREV command, 5-4
  - case in, 1-2
  - combining features, 4-24
  - command execution, 5-7
  - editor, 2-102
  - errors during processing, 4-3, 5-5
  - execution sequence, 5-1
  - execution sequence flowchart, 5-2
  - format, 1-1
  - functions, 5-5
  - iteration lists, 4-4, 5-5
  - length, 1-2
  - multiple commands, 4-3, 5-4
  - name generation, 5-7
  - processing example, 5-7
  - processing features, 4-1
  - prompts, 2-283
  - special characters, 4-2
  - specifying CHANGE\_PROJECT, 2-44
  - suppressing special characters on, 4-24
  - suppressing syntax processing, 5-4
  - syntax suppression character, 5-4
  - treewalking, 5-6
  - variables, 4-2, 5-5
  - wildcarding, 5-6
- Command line features
  - COPY command, use with, 2-78
  - DELETE command, use with, 2-88
- Command sequences, multiple, 6-14
- Command-line editing, 6-1
- Commands
  - at mini-command level, 2-192
  - deferring execution of, 2-159

- execution of, 5-7
- execution sequence, 5-1
- list of documented, 1-9
- multiple on a single command line, 5-4
- null command, 2-405
- obsolete, C-1
- online help text for, 2-135
- opening and closing files, 1-14
- selection criteria execution, 5-6
- treewalking limitation, 5-6
- user, by function, 1-11
- wildcard options, 5-6
- with iteration lists, 4-4
- Comment lines, designating, 2-406
- Common file system name space, 1-3
- Communicating with other users, 2-235
- Communications, summary of commands for, 1-22
- COMO -NTTY command
  - function for returning current output state, 3-9
  - WATCH session, 2-404
- COMO command, 2-58
- COMO files
  - closing, 2-60
  - dating, 2-83
  - function for returning current state, 3-9
  - opening, 2-59
  - pausing, 2-59
  - using, 2-60
  - WATCH session, 2-403
- COMO\_INFO function, 3-9
- COMOUTPUT command, 2-58
- Comparing files
  - with CMPF command, 2-50
  - with FILVER command, 2-124
- Compilers
  - CBL (COBOL 74), 2-38
  - CBLSUBS (COBOL 74 DDL), 2-39
  - CC (C), 2-40
  - COBOL85, 2-54
  - F77 (FORTRAN 77), 2-118
  - F77SUBS (FORTRAN 77 DDL), 2-120
  - FSUBS (FORTRAN IV DDL), 2-126
  - FTN (FORTRAN IV), 2-126
  - PASCAL, 2-264
  - PL1, 2-271
  - PL1G, 2-271
  - summary of, 1-15
  - VRPG (V mode), 2-401
- Compute time, displaying, 2-388
- CONCAT command, 2-62
- Concatenating files
  - with CONCAT command, 2-62
  - with MRGF command, 2-249
  - with RJQ command, 2-301
  - with SORT command, 2-350
- Concurrency locks on files, setting, 2-304
- Condition frame format, B-3
- Condition on the stack, function for returning, 3-9
- Conditional branching, IF function, 3-19
- Configuring a PRIME MEDUSA installation, 2-233
- Configuring asynchronous lines, 2-317
- Connect time, displaying, 2-388
- Connecting to other networked systems, 2-253
- Control characters in ECL, 6-1
- Control key sequences. *See* Ctrl-
- Controllers, disk and tape, number on system, 3-37
- Controllers, disk and tape, never used, 3-37
- Converting
  - decimal to hexadecimal function for, 3-39
  - decimal to octal function for, 3-39
  - hexadecimal to decimal function, 3-18
  - octal to decimal function, 3-27
- Converting ASCII files to uppercase, 2-393
- COPY command, 2-66
- Copy operations, incremental, 2-77
- COPY\_RBF command, 2-79
- copy\_region, 6-38
- Copying access categories, 2-70
- Copying to an open target file, 2-73
- Copying with password directories, 2-78
- CPL command, 2-80
- CPL programs
  - executing with CPL command, 2-80
  - executing with RESUME command, 2-297
  - restarting, 2-372
  - running, 2-80
  - running as phantoms, 2-269
  - use for submitting batch jobs, 2-144
  - use of SET\_VAR command in, 2-334
  - user-defined abbreviations in, 2-1
- CPMPC command, C-7
- CPU, returning processor type, 3-37
- CPU time
  - displaying, 2-388
  - displaying at prompt, 2-284, 2-285
  - since boot, 3-38
  - since login, 3-43
- CPW command, 2-41
- CREATE command, 2-81
- CREATED\_AFTER wildcard option, 4-10
- CREATED\_BEFORE wildcard option, 4-10
- Creating, binary libraries, 2-101
- Creating boot tapes, 2-38
- Creating directories, 2-81
- CREATK command, 2-82
- CRLF feature, 2-238
- CRMPC command, C-8
- Ctrl-P, enabling or disabling, 2-385, 6-2
- Ctrl-S/Ctrl-Q, enabling, 2-107, 2-386
- Current directory, 2-160, 2-263
- Current environment, commands for information about, 1-19

## D

- D, Delete access right, 2-178
- Data management commands, summary of, 1-23
- Data Terminal Ready signal, dropping, 2-95
- Date
  - displaying at prompt, 2-285
  - displaying current calendar, 2-82
  - displaying function for, 3-10
  - function for validating, 3-44
- DATE command, 2-82
- DATE function, 3-10

- DBASIC command, 2-83
- DBG command, 2-84
- DBMS commands, summary of, 1-23
- DBMS schema DDL compiler, 2-308
- DBMS Schema Decompiler (SCHDEC), 2-307
- DBMS Schema Editor (SCHED), 2-307
- DBUTL command, 2-84
- Debuggers
  - DBG (Source Level), 2-84
  - HPSD, 2-139
  - PSD, 2-143
  - PSD (R-mode version), 2-280
  - VPSD, VPSD16, 2-401
- Debugging, commands for, 1-17
- Debugging obsolete programs, 2-273
- Decimal integers
  - decimal to hexadecimal function, 3-39
  - decimal to octal function, 3-39
- Decoding physical device numbers, 2-266
- Decompiler, schema, 2-307
- Default system prompts, changing, 2-283
- Deferring execution of commands, 2-159
- Deferring messages, 2-236
- Deferring phantom logout notification, 2-224
- DEFINE\_GVAR command, 2-85
- Defining global variables, 2-333
- Defining user abbreviations, 2-5
- DELAY command, 2-86
- Delaying execution of commands, 2-159
- DELETE command, 2-87
- Delete protection on file system objects, 2-324
- delete\_char, 6-38
- DELETE\_RBF command, 2-90
- DELETE\_VAR command, 2-91
- delete\_word, 6-38
- Deleting global variables, 2-91
- DELSEG command, 2-92
- Dependency list, for registered EPFs, listing, 2-199
- Devices
  - assigned to users, displaying, 2-211
  - assigning, 2-15
  - status information on, 2-374
  - unassigning, 2-391
- DIAG command, 2-93
- Diagnostic Utility (DIAG), 2-93
- DIR function, 3-11
- Directories
  - See also* Password directories
  - attaching to, 2-23
  - ATTRIB function, using on, 3-5
  - changing passwords in, 2-264
  - changing to another, 2-23
  - converting to password, 2-297
  - copying, 2-70
  - creating, 2-81
  - current, 2-160
  - deleting, 2-87
  - DIR function, using on, 3-11
  - displaying ACL protection, 2-286
  - home, C-7
  - listing contents, 2-166
  - listing quotas, 2-196
  - listing size of, 2-337
  - managing, summary of commands, 1-12
  - name, changing, 2-52
  - nested, copying, 2-73
  - nonquota, information on, 2-196
  - origin, 2-263
  - protecting, summary of commands, 1-12, 1-13
  - quotas, listing, 2-196
  - replacing, 2-75
  - root, 1-3, 1-4
  - setting delete protection on, 2-324
  - setting quotas on, 2-325
  - treewalking, using in, 4-15
  - wildcarding, 5-6
- Directory, command (CMDNC0), 1-7
- DIRECTORY wildcard option, 4-10
- Directoryname, 1-6
- Discarding stack history, 2-290
- Discarding terminal buffers, 2-302
- Disconnecting from lines, 2-95
- DISCOVER command, 2-93
- Disk and tape controllers, number on system, 3-37
- Disk drives
  - never used, 3-37
  - number on system, 3-37
- Disk I/O time, displaying, 2-388
- Disk partitions
  - as a file system object, 1-3
  - listing, in file system name space, 2-193
  - remote, displaying, 2-193
- Disks
  - assigning, 2-21
  - displaying space on, 2-28
  - in use by users, listing, 2-211
  - listing priority ACL, 2-195
  - number on system, 3-37
  - restoring data from tapes, 2-228
  - restoring data to, C-3
  - retrieving data from tapes, 2-228
  - saving data to tape, 2-227, C-1
  - status information on, 2-375
  - unassigning, 2-392
- Display manager, 6-30
- Displaying batch jobs running, 2-212
- Displaying child processes running, 2-212
- Displaying current date and time, 2-82
- Displaying current user project assignments, 2-212
- Displaying date, function for, 3-10
- Displaying date and time, function for, 3-10
- Displaying devices assigned to users, 2-211
- Displaying disks in use by users, 2-211
- Displaying ISC server names, 2-206
- Displaying ISC session information, 2-207
- Displaying MESSAGE command parameters, 2-236
- Displaying mounted disk partitions, 2-193
- Displaying phantom processes, 2-212
- Displaying registered EPFs, 2-199
- Displaying remote disk partitions, 2-193
- Displaying search rules use, 2-243
- Displaying server processes, 2-212
- Displaying slave processes running, 2-212
- Displaying spoolers running, 2-212
- Displaying time, function for, 3-10
- Displaying user process information, 2-211

Diverted terminal I/O, C-4  
 Dividing numbers with MOD function, 3-26  
 DMSTK command, 2-96, B-1  
     dump stack format, B-1  
 do\_echo, 6-2, 6-12, 6-16, 6-38  
 do\_quote, 6-2, 6-13, 6-16, 6-28, 6-38  
 Double-precision BASIC interpreter, 2-83  
 downcase\_word, 6-38  
 DPTX queues, information on, 2-94  
 DPTX/DSC commands  
     DPTXMTR, 2-94  
     OWLDSC, 2-263  
     PRTDSC, 2-280  
     PT45DSC, 2-281  
     PT46DSC, 2-281  
     PTDSC, 2-282  
     TCF, 2-383  
 DPTX/TCF facility, 2-383  
 DPTXMTR command, 2-94  
 Drives, tape, unassigning, 2-392  
 DROPDTR command, 2-95  
 Dropping the DTR signal, 2-95  
 DTAR2 private segments, information on, 2-204  
 DTR signal, dropping, 2-95  
 Dump stack format, B-1  
 DUMP\_STACK command, 2-96, 2-273, 2-274  
     dump stack format, B-1  
 Dumping MIDASPLUS files, 2-233  
 Duplex, full or half, 2-385  
 Dynamic segments, private, information on, 2-204

## E

echo\_raw, 6-13, 6-38  
 ECL  
     automatic pathname completion in, 6-3  
     command history, 2-110  
     command history options, 2-109  
     command output (COMO) files, 6-3  
     commands reference, 6-37  
     comment character, 6-11  
     completing pathnames with, 6-3

control characters, 6-1  
 display manager, 6-30  
 editing characteristics, 2-112  
 introduction, 6-1  
 invoked as EPF command function, 6-3  
 key bindings, 6-9  
 kill ring, 6-2  
 options, description of, 2-102  
 options, list of, 2-103  
 pathname completion, 2-111  
 programmable function commands, 6-13  
 prompt handling, 6-31  
 prompts, with RDY prompts, 2-288  
 replacing pathnames with, 6-5  
 search ring, 6-2  
 terminal bindings and characteristics, 2-105  
 terminal I/O options, 2-108  
 terminal type, setting of, 2-107  
 TERMINFO capabilities, 6-34  
 treewalk wildcard expansion, using, 6-5  
     using commands of, 6-2  
     WATCH monitoring, 2-404  
 ECL command, 2-102  
 ED command, 2-97  
 EDAC command, 2-98  
 EDB command, C-9  
 EDIT\_ACCESS command, 2-98  
 EDIT\_BINARY command, 2-101  
 EDIT\_CMD\_LINE command, 2-102  
 Editing ACLs, 2-98  
 Editing binary libraries, 2-101  
 Editors  
     ED (line), 2-97  
     EDB (binary), C-9  
     EMACS (screen), 2-114  
     FED (FORMS), 2-123  
     NSED (nonshared ED), 2-256  
     SCHEM (schema), 2-307  
 EMACS, WATCH monitoring, 2-404  
 EMACS command, 2-114  
 Emptying terminal buffers, 2-302  
 Encoding physical device numbers, 2-266  
 end\_line, 6-39

ENTRY\$ search rules, creating optimized, 2-247  
 ENTRYNAME function, 3-12  
 Entrypoints  
     as a command argument, 1-2  
     description of, 1-6  
     in a pathname, 1-4  
 Entrypoint search lists, listing contents of, 2-202  
 Entrypoint search rules, expanding objectname to fully-qualified pathname, 2-115  
 Entrypoints  
     for registered EPFs, listing, 2-199  
     listing library EPF, 2-189  
 Environment, commands for information about current, 1-19  
 Environment directives, printer, 2-278  
 EPF command function, ECL invoked as, 6-3  
 EPF commands  
     ICE, 2-140  
     LIST\_EPF, 2-182  
     LIST\_LIBRARY\_ENTRIES, 2-189  
     LIST\_LIMITS, 2-190  
     LIST\_MINI\_COMMANDS, 2-192  
     LIST\_REGISTERED\_EPF, 2-199  
     LIST\_SEARCH\_RULES, 2-202  
     LIST\_SEGMENT, 2-204  
     REMOVE\_EPF, 2-291  
 EPFs  
     creating, 2-37  
     executing, 2-295  
     library, listing entrypoints, 2-189  
     listing entrypoints, 2-189  
     output, 2-182  
     registered, listing of, 2-199  
     removing from address space, 2-291  
     REPLACE files, 2-74  
     replacing mapped, 2-74  
     restarting, 2-372  
     X access right for, 2-178  
 ER, prompt, changing, 2-284  
 Erase character, 2-385  
 Erasing files, 2-87  
 Error codes, displaying at prompt, 2-286  
 Error messages, RDY command, 2-287  
 ERRSET subroutine, 2-274

ERRVEC, displaying contents of, 2-273  
 esc\_digit, 6-2, 6-37, 6-39, 6-42  
 ESR command, 2-115  
 Establishing your identity, summary of  
 commands for, 1-12  
 Evaluating expressions, function for, 3-7  
 exchange\_mark, 6-39  
 execute\_macro, 6-14, 6-15, 6-37, 6-39  
 Executing and Linking programs,  
 commands for, 1-16  
 Executing BASIC programs, 2-31  
 Executing COMI files, 2-56  
 Executing CPL programs, 2-80  
 Executing EPFs, 2-295  
 Executing PRIME MEDUSA utility  
 programs, 2-234  
 Execution priority level, 2-45  
 EXISTS function, 3-12  
 expand\_abbrev, 6-39  
 EXPAND\_SEARCH\_RULES command,  
 2-115  
 EXPAND\_SEARCH\_RULES function,  
 3-14  
 expand\_wild, 2-106, 2-111, 6-3, 6-5, 6-39  
 expand\_wild\_menu, 2-111, 6-7, 6-40  
 Expanding partial pathnames, 6-6  
 explain\_key, 6-13, 6-28, 6-40  
 extend\_command, 6-40  
 Extents, maps  
   described, 2-175  
   listing, 2-175  
 External command, definition of, 1-7

## F

F77 command, 2-118  
 F77DML command, 2-118, 2-122  
 F77SUBS command, 2-120  
 Failed login, 2-219  
 FAP command, 2-120  
 FAU command, 2-121  
 Fault Frame format, B-2  
 FED command, 2-123  
 Fetching files from remote systems, 2-132  
 File Administrator Utility, invoking,  
 2-121

File system command functions,  
 summary of, 1-26  
 File system name space, 1-3  
   listing disk partitions and portals in,  
   2-193  
 File system objects, 1-3, 1-4  
   changing names, 2-52  
   copying, 2-66  
   delete protection on, 2-324  
   function for returning attributes of, 3-5  
   function to check existence of, 3-12  
   listing size of, 2-336  
   names, 1-4  
   protected, forced deleting of, 2-88  
   protecting from deletion, 2-324  
   replacing, 2-75  
   selecting by date, 4-11  
   selecting by similar name, 4-7  
   selecting by suffix, 4-7  
   selecting by type, 4-11  
   setting access rights on, 2-279  
 File transfer assistance (online), 2-133  
 File transfer queues, information on,  
 2-134  
 File Transfer Request utility, invoking,  
 2-127  
 File transfer requests, 2-130, 2-131, 2-132  
 File Transfer Service, summary of  
 commands, 2-134  
 File transfers, requesting, 2-131  
 File types, sorting different, 2-342  
 File units, status of, 2-376  
 File utility (FUTIL), C-10  
 -FILE wildcard option, 4-10  
 Filenames, 1-6  
   changing, 2-52  
   simple, 1-6  
 Files  
   archiving ROAM, 2-306  
   ATTRIB function, using on, 3-5  
   attribute, 2-357  
   changing names of, 2-52  
   closing, 2-48  
   closing, summary of commands, 1-13  
   combining, with MRGF command,  
   2-249  
   COMI (command input), 2-55

commands for protecting, 1-15  
 COMO (command output), 2-58  
 comparing, 2-50, 2-124  
 concatenating, 2-62  
 concatenating RJE, 2-301  
 concurrency locks, setting, 2-304  
 converting to uppercase, 2-393  
 copying, 2-66  
 copying across networks, 2-131, 2-301  
 deleting, 2-87  
 displaying on terminal, 2-340  
 dumping MIDASPLUS, 2-233  
 fetching from other systems, 2-132  
 formatting ASCII, 2-303  
 function for opening, 3-27  
 function for returning current COMO  
 file state, 3-9  
 function returning an entryname, 3-12  
 global variable, 2-85, 2-86, 2-333  
 keyed-index, creating or modifying,  
 2-82  
 listing, C-11  
 listing attributes of ROAM, 2-326  
 listing size of, 2-336  
 managing, summary of commands,  
 1-13  
 merging, 2-249, 2-341, 2-350  
 merging with COPY command, 2-76  
 MIDASPLUS, building, 2-153  
 MIDASPLUS, deleting, 2-154  
 moving, 2-66  
 open, checking for, 2-376  
 opening, 2-36, 2-142, 2-177, 2-261  
 opening and closing, commands for,  
 1-14  
 plot, printing, 2-272, C-16  
 plotting, 2-352  
 printing, 2-357  
 printing on MPC parallel printer, C-13  
 printing on serial interface printers,  
 C-15  
 PRISAM, verifying, 2-93  
 protecting, 2-279, 2-311  
 reading next line in, 3-27  
 renaming, 2-52  
 replacing, 2-75  
 restoring ROAM, 2-294

- RJE, queued, 2-299
- ROAM, deleting, 2-90
- ROAM, listing attributes, 2-198
- sending, 2-131
- setting access rights in password directories, 2-279
- setting access rights on, 2-311, C-14
- setting concurrency locks on, 2-304
- setting delete protection on, 2-324
- setting read/write locks, 2-304
- sorting, 2-341, 2-378
- spooled, cancelling, 2-359
- spooled, listing, 2-360
- spooled, modifying, 2-363
- temporary destinations, 2-131
- transferring, 2-131, 2-301, 2-389
- transferring to other systems, 2-127, 2-301
- using ATTRIB function on, 3-5
- writing text with WRITE\_FILE function, 3-49
- FILMEM command, 2-123
- FILVER command, 2-124
- finish\_macro, 6-37, 6-40
- Fonts, PostScript, 2-365, 2-370
- Forced deletion of protected objects, 2-88
- Forced logout, 2-223
- Format of command line, 1-1
- Formatting ASCII files, 2-303
- FORMS Administrative Processor, 2-120
- FORMS commands, summary of, 1-24
- FORMS editor, invoking, 2-123
- FORTRAN 77 compiler, 2-118
- FORTRAN 77 DDL compiler, 2-120
- FORTRAN 77 DML preprocessor, 2-118, 2-122
- FORTRAN DDL compiler, 2-126
- FORTRAN IV compiler, 2-126
- forward\_char, 6-12, 6-40
- forward\_search, 2-112, 6-2, 6-40
- forward\_word, 2-112, 6-41
- FSUBS command, 2-126
- FTN command, 2-126
- FTR command, 2-127
- FTS (File Transfer Service), 2-134
- Full pathname, definition of, 1-5

- Full-duplex modes, setting, 2-385
- Fully-qualified pathnames, ESR command output of, 2-117
- Fully-qualified pathnames, 1-3
  - definition of, 1-5
  - function for returning, 3-14, 3-28
- Function keys, 6-15
  - codes for, 6-28
- Functions
  - displaying at prompt, 2-287
  - miscellaneous, 1-27
  - REVERSE, 3-35
- FUTIL command, C-10

**G**

- GENERATE\_CATALOG command, C-10
- GET\_VAR function, 3-16
- Global Mount Table, listing contents of, 2-193
- Global variable files
  - creating, 2-85
  - deleting, 2-86
  - returning pathname of, 3-16
- Global variables
  - creating, 2-333
  - defining, 2-333
  - deleting, 2-91
  - displaying at prompt, 2-286
  - function returning global variable file pathname, 3-18
  - function returning value of, 3-16
  - in abbreviations, 2-7
  - listing, 2-215
  - returning value of, 3-16
  - use in CPL programs, 2-334
  - use of, 4-2
- Global variables files, activating, 2-85
- goto\_line, 6-2, 6-41
- GROUP\_LIST function, 3-17
- Groups, function for listing. *See* ACL groups
- GVPATH function, 3-18

**H**

- Half-duplex
  - networks, information on, 2-134
  - setting terminal line to, 2-385
  - WATCH sessions, 2-404
- Handling terminal I/O, summary of commands for, 1-21
- HDX sites, monitoring, 2-134
- HDXSTAT command, 2-134
- HELP command, 2-135
- Help facility, 2-135
  - adding text, 2-138
- HEX function, 3-18
- Hexadecimal numbers, function for validating, 3-44
- Home directory, C-7
- [HOME\_DIR] search rule, 2-247, 2-329
- HP LaserJet printers, SPOOL support for, 2-370
- HPSD command, 2-139

**I**

- I/O device handling, summary of commands for, 1-21
- I/O time
  - displaying, 2-388
  - displaying at prompt, 2-286
  - ready prompt, display in, 2-284
- I/O, diverted terminal, C-4
- ICE command, 2-140, 2-331
  - command line restriction, 5-4
- IF function, 3-19
- Implicit login, 2-218
- Incremental copy operations, 2-77
- INDEX function, 3-21
- INFO command, 2-139
- Initial Attach Point
  - at login, 2-217
  - returning to, 2-263
- INITIALIZE\_COMMAND\_ENVIRONM ENT command, 2-140, 2-331
- INPUT command, 2-142
- INSERT search rule, 2-329
- Internal command, definition of, 1-7
- Interpreters

BASIC, 2-31  
 BASICV, 2-32  
 DBASIC, 2-83  
 summary of, 1-15  
 InterServer Communications (ISC),  
 2-206, 2-207  
 I/O buffers, monitoring, 2-403  
 I/O operations, metering, 2-396  
 I/O time  
   since boot, 3-38  
   since login, 3-43  
 IPSD command, 2-143  
 ISC sessions, listing of, 2-207  
 Iteration, DELETE command, use with,  
 2-88  
 Iteration lists  
   combined with other features, 4-24  
   command line processing, 5-5  
   containing abbreviations, 5-3  
   creating, 4-4  
   cross-product, 4-6  
   explanation of, 4-4  
   multiple, 4-5  
   specifying members of, 4-5  
   suppressing, 4-24

**J**

JOB command, 2-144  
 Job processing, commands for, 1-17

**K**

KBUILD command, 2-153  
 Key bindings in ECL, 6-9  
 Keyed-index files, creating or modifying,  
 2-82  
 KIDDEL command, 2-154  
 Kill buffer, 6-41, 6-45  
 Kill character, setting, 2-386  
 Kill ring, 2-110, 6-2  
 kill\_line, 6-41  
 kill\_region, 2-108, 6-38, 6-41  
 KLMD function, 3-22  
 KLMF function, 3-23

KLMT function, 3-24

**L**

L, List access right, 2-178  
 L command, C-11  
 LAB command, 2-155  
 LABEL command, 2-157  
 Labels  
   list of kinds for tapes, 2-158  
   tape, 2-157  
 LAC command, 2-178  
 LATE command, 2-159  
 LCB command, 2-180  
 LD command, 2-160, 2-166  
 LE command, 2-182  
 Leaving the system, 2-222  
 LEM command, 2-175  
 LENGTH function, 3-25  
 Length of command-line, 1-2  
 Levels of priority, displaying, 2-46  
 LG command, 2-188  
 LIBEBD command, C-11  
 Library EPFs, listing entrypoints of,  
 2-189  
 Line editors  
   ED, 2-97  
   NSED, 2-256  
 Line printers  
   assigning, 2-16  
   printing on, 2-357  
   recovering output from, C-4  
 LINEFEED  
   echoing, 2-385  
   in MESSAGE command, 2-238  
 Lines  
   *See also* Asynchronous  
   assigning, 2-22  
   disconnecting from, 2-95  
 Linker, BIND, 2-37  
 Linking and executing programs,  
   commands for, 1-16  
 Linking EPF runfiles, 2-37  
 Linking loader  
   LOAD, 2-216  
   SEG, 2-309  
 LIST\_ACCESS command, 2-178  
   access rights for (table), 2-178  
 LIST\_CATALOG command, C-12  
 LIST\_CONTIGUOUS\_BLOCKS  
   command, 2-180  
 LIST\_EPF command, 2-182  
 LIST\_GROUP command, 2-188  
 LIST\_LIBRARY\_ENTRIES command,  
 2-189  
 LIST\_LIMITS command, 2-190  
 LIST\_MINI\_COMMANDS command,  
 2-192  
 LIST\_MOUNTS command, 2-193  
 LIST\_PRIORITY\_ACCESS command,  
 2-195  
 LIST\_QUOTA command, 2-196  
 LIST\_RBF command, 2-198  
 LIST\_REGISTERED\_EPF command,  
 2-199  
 LIST\_REMOTE\_ID command, 2-200  
 LIST\_SCHEDULER\_ATTRIBUTES  
   command, 2-201  
 LIST\_SEARCH\_RULES command,  
 2-202  
 LIST\_SEGMENT command, 2-204  
 LIST\_SERVER\_NAMES command,  
 2-206  
 LIST\_SESSIONS command, 2-207  
 LIST\_TAPE command, C-13  
 LIST\_USERS command, 2-211  
 LIST\_VAR command, 2-215  
 LISTF command, C-11  
 Listing abbreviations, 2-10  
 Listing attributes on ROAM files, 2-198  
 LISTING command, 2-177  
 Listing priority ACL, 2-195  
 Listing spooled files, 2-360  
 LL command, 2-190  
 LLENT command, 2-189  
 LMC command, 2-192  
 LO command, 2-222, 5-4  
 LOAD command, 2-216  
 Loader  
   LOAD, 2-216  
   SEG, 2-309  
 Locate buffers, metering, 2-397  
 Locks, setting concurrency, 2-304  
 Log in, time of, 3-43



Logging out. *See* LOGOUT command  
 LOGIN command, 2-217  
   command line restriction, 5-4  
 Login password, changing, 2-41  
 Logins  
   failed, 2-219  
   implicit, 2-218  
   multiple, 2-218  
   origin directory, 2-263  
   project assignment, 2-43  
   remote, 2-218  
   validation codes for, 2-218  
 LOGOUT command, 2-222  
   command line restriction, 5-4  
 Logout messages, deferring, 2-224  
 Logout notification, receiving, 2-224  
 Logouts  
   forced, 2-223  
   phantom notification, 2-224  
 LON command, 2-224  
 LPAC command, 2-195  
 LQ command, 2-196  
 LRE command, 2-199  
 LRID command, 2-200  
 LS command, 2-204  
 LSR command, 2-202

## M

M command, 2-235  
 MAGNET command, 2-227  
 MAGRST command, 2-228  
 MAGSAV command, 2-230  
 Making new directories, 2-81  
 Managing directories, summary of  
   commands for, 1-12  
 Managing files, summary of commands  
   for, 1-13  
 Managing ROAM files, summary of  
   commands for, 1-14  
 Managing submitted batch jobs, 2-150  
 Mapped EPFs  
   removing, 2-291  
   replacing, 2-74  
 mark, 6-42  
 Master File Directory (MFDs), 1-3

  attaching to, 2-24  
 MDUMP command, 2-233  
 MEDCONFIG command, 2-233  
 MEDUSA command, 2-234  
 MEDUTIL command, 2-234  
 Memory  
   amount configured, 3-37  
   pages in use by process, 3-43  
   pages in use by system, 3-38  
   wired pages, 3-38, 3-43  
   zeroing, 2-123  
 Merging files, 2-249, 2-341  
   with COPY command, 2-76  
 Merging segment directories with COPY  
   command, 2-77  
 Merging sorted files, 2-350  
 MESSAGE command, 2-235  
 Message receive states, 2-236  
   setting, 2-241  
 Messages, 2-236, 2-237, 2-238  
   receive states, checking, 2-239  
   sending, summary of commands, 1-21  
   sending, to other users, 2-236  
 Metafiles, plotting, 2-272  
 Metering system usage, 2-394  
 MFDs, 1-3  
   attaching to, 2-24  
 MIDASPLUS  
   building files, 2-153  
   files, cleaning up, 2-248  
   files, deleting, 2-154  
   files, dumping, 2-233  
   files, editing, 2-154  
   files, packing, 2-248  
   files, restructuring, 2-248  
 MIDASPLUS commands, summary of,  
   1-24  
 Mini-command level, commands at,  
   2-192  
 MOD function, 3-26  
 Modes  
   full-duplex, setting, 2-385  
   setting, 2-385  
 -MODIFIED\_AFTER wildcard option,  
   4-10  
 -MODIFIED\_BEFORE wildcard option,  
   4-10

Modifying ACLs, 2-98  
 Modifying batch jobs, 2-151  
 Modifying spooled files, 2-363  
 Modulus, returning, 3-26  
 MONITOR\_NET command, 2-242  
 MONITOR\_SEARCH\_RULES  
   command, 2-243  
 Monitoring batch jobs, 2-151  
 Monitoring DPTX queues, 2-94  
 Monitoring file transfer requests, 2-132  
 Monitoring system usage, 2-394  
 Mount-point pathname, 1-3  
   disk partitions and portals, listing of,  
   2-193  
 Mount-points, 1-3  
 MPACK command, 2-248  
 MPC parallel interface printer, C-13  
 MPLUSCLUP command, 2-248  
 MRGF command, 2-249  
 MT, argument of LABEL command,  
   2-157  
 Multiline messages, 2-238  
 Multiple command sequences, 6-14  
 Multiple login sessions, 2-218  
 multiplier, 6-2, 6-42

## N

Name generation  
   adding name components, 4-23  
   command line processing, 5-7  
   deleting name components, 4-23  
   explanation of, 4-20  
   specifying, 4-21  
   suppressing, 4-24  
   using with wildcarding, 4-24  
   within iteration lists, 4-24  
 Name Server, listing disk partitions when  
   running, 2-193  
 Name space, common file system, 1-3  
 Negation character  
   name generation, 4-23  
   wildcarding, 4-7  
 Nested directory copying, 2-73  
 NETLINK command, 2-253  
 Networks  
   connecting to other systems, 2-253

half-duplex, 2-134  
 local area networks, 2-190  
 status information on, 2-375  
 NEWLINE, echoing, 2-385  
 next\_line, 2-110, 6-2, 6-42  
 -NO\_VERIFY wildcard option, 4-10  
 Non-printable characters, WATCH  
 monitoring, 2-404  
 NONE access right, 2-178  
 Nonprinting characters, 6-10  
 Nonquota directories, information on,  
 2-196  
 Nonshared version of EDITOR, 2-256  
 NONTAG sort, 2-341  
 NSED command, 2-256  
 NTS  
   assigning lines, 2-22  
   status information on, 2-375  
   unassigning lines for, 2-392  
 NTS associations, listing, 2-257  
 NTS terminal, placing into Command  
 mode, 2-256  
 NTS\_LINE command, 2-256  
 NTS\_LIST\_ASSOCIATE command,  
 2-257  
 Null command, 2-405  
 NULL function, 3-26  
 NUMBER command, 2-258  
 Number-sign characters  
   #, 6-11, 6-31  
   ###, 6-42  
 Numbers  
   dividing with MOD function, 3-26  
   function for validating, 3-44

## O

O, Owner access right, 2-178  
 O command, 2-261  
 OAS command, 2-260  
 OAS commands, summary of, 1-25  
 Objectname, 1-6  
 Obsolete commands, C-1  
 OCTAL function, 3-27  
 Octal numbers

converting to decimal function for,  
 3-27  
 function for validating, 3-44  
 Office Automation System (OAS)  
   commands for, 1-25  
   starting up, 2-260  
   Telephone Inquiry function, 2-282  
 OK, prompt, changing, 2-284  
 OPEN command, 2-261  
 Open files, checking for, 2-376  
 OPEN\_FILE function, 3-27  
 open\_line, 6-42  
 Opening files, 2-36  
   summary of commands for, 1-14  
   with BINARY command, 2-36  
   with INPUT command, 2-142  
   with LISTING command, 2-177  
   with OPEN command, 2-261  
   with OPEN\_FILE function, 3-27  
 Operator, sending messages to, 2-237  
 Options, command line format, 1-1  
 OR command, 2-263  
 ORIGIN command, 2-263  
 Origin directory  
   attaching to, 2-263  
   changing projects, 2-43  
   changing to, 2-263  
   function for returning, 3-5  
   login process, 2-217  
   returning to, 2-263  
 [ORIGIN\_DIR] search rule, 2-329  
 OWLDSC command, 2-263

## P

P, Protect access right, 2-178  
 P command, 2-272  
 Packet Switched Data Network (PSDN),  
   using, 2-253  
 Packing MIDASPLUS files, 2-248  
 Paper tape files, C-9  
 Partial pathname, expanding, 6-6  
 PASCAL command, 2-264  
 Passing parameters to batch monitor,  
   2-150  
 PASSWD command, 2-264  
 password, 6-42  
 Password directories  
   *See also* Directories  
   attaching to, 2-25  
   copying with, 2-78  
 Passwords  
   changing, 2-219  
   changing in directories, 2-264  
   changing login, 2-41  
   changing projects, 2-43, 2-44  
   description of login, 2-218  
   expiration, 2-219  
   function for validating, 3-44  
   WATCH sessions, 2-404  
 Pathname completion, 6-3  
 PATHNAME function, 3-28  
 Pathnames  
   absolute, definition of, 1-4  
   asterisk in, 1-5  
   automatic completion, 2-111, 6-3  
   description of, 1-4  
   directory portion, function for  
   returning, 3-11  
   directory wildcard, 5-6  
   entryname portion, function for  
   returning, 3-12  
   entryname wildcard, 5-6  
   expanding partial, 6-6  
   full, function for returning, 3-28  
   fully-qualified, 1-3  
   fully-qualified, ESR command output  
   of, 2-117  
   fully-qualified, function for returning,  
   3-14, 3-28  
   fully-qualified, definition, 1-5  
   generating multiple, 4-20  
   mount-point, 1-3, 2-193  
   relative, definition of, 1-5  
   replacing, 6-5  
   strings, function to check validity, 3-44  
   unqualified, definition, 1-5  
 Pausing COMI files, 2-55  
 Pausing COMO files, 2-59  
 pdev. *See* Physical device number  
 PDEV command, 2-266  
 PDEV function, 3-29  
 Per-user search rules use, 2-244

- Peripheral devices, assigning, 2-15
- Permissions. *See* Access rights
- PF commands
  - introduction, 6-13
  - multiple command sequences, 6-14
  - using, 6-14
- PFkey commands (pfkey), 6-43
- PH command, 2-268
- PHANTOM command, 2-268
- Phantoms
  - cleaning up ROAM, 2-49
  - controlling system printers, 2-276
  - identifying process as, 3-42
  - logging out, 2-223, 2-224, 2-270
  - logout notification, 2-224
  - monitoring, 2-270
  - operation of, 2-269
  - project assignment, 2-43
  - receiving logout messages, 2-224
  - running, list of, 2-212
  - slave, 2-14
  - spooler, 2-276
- Physical device number
  - encoding/decoding, 2-266
  - function for displaying/encoding, 3-29
- PL1 command, 2-271
- PL1G command, 2-271
- Placing quotation marks around text,
  - function for, 3-31
- PLOT command, 2-272
- Plot files, 2-272
  - printing, 2-272, 2-357, C-16
- Plot queue, 2-272
- Plotters
  - assigning, C-16
  - printing files on, 2-272, C-16
- Plotting files, 2-352
- PM command, 2-272
- PMA command, 2-273
- Port selector, disconnecting from, 2-95
- Portals
  - listing mount-point pathnames, 2-193
  - listing, in file system name space, 2-193
- PostScript printers, SPOOL support for, 2-365
- PRERR command, 2-273
- prev\_line, 2-110, 6-2, 6-11, 6-43
- PRIMAN command, 2-274
- Prime INFORMATION, invoking, 2-139
- Prime Macro Assembler (PMA), 2-273
- PRIME MEDUSA, 2-233, 2-234
  - commands, summary, 1-25
- Prime RPG II V-mode compiler, 2-401
- Prime Symbolic Debugger (PSD), 2-143, 2-280, 2-401
- PRIME/SNA commands, summary of, 2-275
- PRIMENET commands
  - FTR, 2-127
  - HDXSTAT, 2-134
  - MONITOR\_NET, 2-242
  - NETLINK, 2-253
- PRIMEWORD Word Processing system, 2-405
- PRIMON command, 2-276
- PRIMOS
  - validating treename function, 3-44
  - version of, 2-376
- PRIMOS commands. *See* Commands
- PRIMOS revision
  - displaying at prompt, 2-286
  - returning to function, 3-37
- Printer, serial interface, printing files on, C-15
- Printer environment directives, 2-278
- Printers
  - assigning, 2-16
  - attributes, 2-353, 2-357
  - displaying information on, 2-276
  - HP LaserJet, support for, 2-370
  - PostScript, support for, 2-365
  - status of, 2-277
  - summary of commands for, 1-22
- Printing characters, 6-10
- Printing files, 2-357
- Printing plot files, 2-272, 2-357, C-16
- Priority ACLs, listing, 2-195
- Priority levels
  - displaying, 2-46
  - execution, 2-45
- PRISAM commands, summary of, 1-24
- PRISAM File Administrator Utility, 2-121
- PRISAM File Diagnostic Utility, 2-93
- PRISAM query language and report writer, 2-93
- Private dynamic segments, information on, 2-204
- Private static segments, information on, 2-205
- PRMPC command, C-13
- Processes
  - See also* User processes
  - idle, 3-42
  - killing, 2-223
  - returning ID, 3-42
  - slave, listing of running, 2-212
  - suspended, 3-42
  - user information, 3-41
- Programs
  - executing CPL, 2-297
  - executing with CPL command, 2-80
  - executing with RESUME command, 2-295
  - executing with SEG command, 2-309
  - linking, 2-37
  - loading, 2-216
  - loading with SEG command, 2-309
  - obsolete, debugging, 2-273
  - restarting, 2-372
- Project assignments
  - changing, 2-43
  - default, 2-43
  - function for validating ID, 3-44
  - listing, 2-43
  - listing current, 2-212
  - returning ID, 3-42
- Project IDs
  - at login, 2-217
  - status of, 2-375
- Prompts
  - carriage return, 2-286
  - error prompts, 2-283
  - handling, 6-31
  - ready prompt, 2-283
  - sort, 2-345
  - system, 2-283
  - variables, 2-285

warning, changing, 2-284  
 PROP command, 2-276  
 PROTEC command, C-14  
 PROTECT command, 2-279  
 Protected objects, forced deletion of, 2-88  
 Protecting directories, summary of  
 commands for, 1-13  
 Protecting files  
 summary of commands for, 1-15  
 with RWLOCK command, 2-304  
 with SET\_ACCESS command, 2-311  
 Protecting objects from deletion, 2-324  
 PRSER command, C-15  
 PRTDSC command, 2-280  
 PRVER command, C-16  
 PSD. *See* Prime Symbolic Debugger  
 PSD command, 2-280  
 PSDN, using, 2-253  
 PT45DSC command, 2-281  
 PT46DSC command, 2-281  
 PTDSC command, 2-282  
 PTELE command, 2-282  
 Punched cards  
 reading from, C-8  
 writing to, C-7

## Q

QUERY function, 3-30  
 Query/response function, 3-30, 3-33  
 Queueing files for printing, 2-358  
 Queueing files via RJQ, 2-301  
 Queues  
 DPTX, 2-94  
 FTS, 2-134  
 plot, 2-272  
 spool, 2-358, 2-360, 2-363  
 Quitting system, 2-222  
 Quotas, directory  
 listing, 2-196  
 setting, 2-325  
 Quotation marks  
 function for placing around text, 3-31  
 removing from text, 3-31, 3-41  
 quote, 6-13, 6-16, 6-43

QUOTE function, 3-31

## R

R, Read access right, 2-178  
 R command, 2-295, 4-21  
 R-mode runfiles  
 executing, 2-296  
 restoring, to memory, 2-294  
 -RBF wildcard option, 4-10  
 RDY command, 2-283, 2-290  
 error messages, 2-287  
 prompts, with ECL prompts, 2-288  
 Read/write concurrency locks, setting,  
 2-304  
 READ\_FILE function, 3-32  
 Reading from punched cards, C-8  
 Reading next line in a file, function for,  
 3-32  
 Reading tape labels, 2-157  
 READY prompt, changing, 2-284  
 Reattaching to origin directory, 2-263  
 Receive state of terminals, 2-241  
 Receiving phantom logout messages,  
 2-224  
 Recording terminal sessions, 2-58  
 REENTER command, 2-289  
 reexecute, 6-43  
 [REFERENCING\_DIR] search rule,  
 2-329  
 refresh, 6-41, 6-43  
 Registered EPFs, dependency list, 2-199  
 Registers, contents of, 2-272, 2-273  
 Reinitializing stack, 2-290  
 Reinitializing the default search list,  
 2-331  
 Rejecting messages, 2-236  
 Relative pathname, definition of, 1-5  
 RELEASE\_LEVEL command, 2-290  
 command line restriction, 5-4  
 Releasing command levels, 2-290  
 Releasing magnetic tapes, with  
 ARCHIVE\_RELEASE command,  
 C-2  
 REMEPF command, 2-291  
 Remote IDs, deleting, 2-293  
 Remote Job Entry (RJE) Phase II  
 commands, 2-298  
 Remote logins, 2-218  
 Remote system name, displaying at  
 prompt, 2-286  
 Remote systems, fetching files from,  
 2-132  
 Remote user IDs  
 adding, 2-14  
 listing, 2-200  
 Remote users  
 sending messages to, 2-237  
 user information, 3-41  
 WATCH sessions, 2-404  
 REMOVE\_EPF command, 2-291  
 REMOVE\_REMOTE\_ID command,  
 2-293  
 Removing files or directories, 2-87  
 REN command, 2-289  
 Renaming files or directories, 2-52  
 Renumbering BASIC programs, 2-258  
 Replacing pathnames, 6-5  
 Request numbers, 2-358  
 RESCAN function, 3-33  
 Resetting system prompts, 2-283  
 RESPONSE function, 3-33  
 \$REST identifier, setting rights for, 2-312  
 REST\_RBF command, 2-294  
 Restarting EDITOR, 2-97  
 Restarting programs, 2-372  
 RESTOR command, 2-294  
 RESTORE\_RBF command, 2-294  
 Restoring tape data to disks, from  
 ARCHIVE tapes, C-3  
 Restructuring MIDASPLUS files, 2-248  
 RESUME command, 2-295  
 name generation, 4-21  
 Retrieving disk data from tapes, with  
 MAGRST command, 2-228  
 Returning to origin directory, 2-263  
 REVERSE function, 3-35  
 reverse\_search, 2-112, 6-2, 6-43  
 REVERT\_PASSWORD command, 2-297  
 Ring, finding breaks in, 2-125  
 RJE commands, summary of, 2-298  
 RJE files, queued, 2-299  
 RJQ, concatenating files, 2-301  
 RJQ command, 2-299

RLS command, 2-290  
 ROAM, files, listing attributes of, 2-198  
 ROAM Command Processor Clean-up Program, 2-49  
 ROAM commands, summary of, 1-14  
 ROAM files  
   activating, 2-294  
   attributes, listing, 2-198, 2-326  
   changing names of, 2-53  
   copying, 2-79  
   listing attributes of, 2-198  
   managing, summary of commands, 1-14  
   restoring, 2-294  
   saving to disk, 2-306  
   setting attributes on, 2-326  
 Root directory, 1-3, 1-4  
 Root entry, 1-3, 1-4  
 RPG II V-mode compiler, 2-401  
 RRID command, 2-293  
 RSTERM command, 2-302  
 rubout\_char, 2-108, 6-44  
 rubout\_word, 2-112, 6-44  
 Runfiles  
   creating EPF (with BIND), 2-37  
   creating R-mode (with LOAD), 2-216  
   creating SEG, 2-309  
   executing R-mode, 2-296  
   restarting, 2-372  
   restoring R-mode, 2-294  
 Running CPL programs, as phantoms, 2-269  
 RUNOFF command, 2-303  
 RVEC parameters  
   displaying, 2-272  
   meaning of, A-1  
   supplying, A-2  
 RVEC use register, displaying contents of, 2-272  
 RWLOCK command, 2-304

## S

S command, 2-372, 5-4  
 SA command, 2-305  
 SAC command, 2-311

SAVE command, 2-305  
 SAVE\_RBF command, 2-306  
 Saving disk data to tapes  
   with ARCHIVE command, C-1  
   with MAGNET command, 2-227  
   with MAGSAV command, 2-230  
 SCHDEC command, 2-307  
 SCHED command, 2-307  
 Scheduler, returning priority, 3-42  
 SCHEMA command, 2-308  
 Schema decompiler, 2-307  
 Schema editor, 2-307  
 Screen editor, 2-114  
 SEARCH function, 3-35  
 Search lists  
   creating, 2-329  
   reinitializing to the default, 2-331  
   resetting to system defaults, 2-332  
   specifying, 3-14  
 Search ring, 2-110, 6-2, 6-41  
 Search rules  
   file, use of, 2-328  
   file, creating, 2-329  
   monitoring, 2-243  
   optimizing, 2-243  
   per-user, 2-244  
   setting, 2-328  
 SEG command, 2-309  
   name generation, 4-21  
 SEG runfiles, 2-309  
 Segment 4000(octal), saving, 2-305  
 Segment directories  
   changing name of, 2-52  
   copying, 2-70  
   creating, 2-309  
   deleting, 2-87  
   listing size of, 2-337  
   merging, with COPY command, 2-77  
   protecting, 2-279  
   setting access rights on, 2-311  
   setting delete protection on, 2-324  
 -SEGMENT\_DIRECTORY wildcard option, 4-10  
 Segmented programs, 2-309  
 Segments  
   freeing, 2-92

  in use by process, 3-43  
   information on, 2-204  
   number in use, 3-38  
   private dynamic, 2-205  
 Self-terminating function keys, 6-15  
 self\_insert, 6-9, 6-13, 6-16, 6-44  
 Semaphores, status information on, 2-376  
 Sending files, 2-301  
 Sending messages  
   summary of commands for, 1-21  
   to other users, 2-236  
 Sense switches, virtual, 2-402  
 Serial interface printer, printing files on, C-15  
 Servers  
   listing names of ISC, 2-206  
   listing of running, 2-212  
   user information, 3-41  
 Session, logging in, 2-217  
 Sessions, ISC, listing of, 2-207  
 SET\_ACCESS command, 2-311  
 SET\_ASYNC command, 2-317  
 SET\_DELETE command, 2-324  
 SET\_QUOTA command, 2-325  
 SET\_RBF command, 2-326  
 SET\_SEARCH\_RULES command, 2-328  
 SET\_VAR command, 2-333  
 Setting  
   access rights on passworded objects, 2-279  
   attributes on ROAM files, 2-326  
   delete protection on objects, 2-324  
   erase character, 2-385  
   global variables, 2-333  
   kill character, 2-386  
   message receive state, 2-241  
   quotas on directories, 2-325  
   read/write concurrency locks, 2-304  
   terminal characteristics, 2-87  
   terminal characteristics, summary of commands for, 1-17  
   terminal line characteristics, 2-317  
 SHOW command, 2-334  
   monitoring with WATCH, 2-403  
 Simple filename, 1-6  
 SIZE command, 2-336  
 Slash (/) character, 2-102

- Slash-asterisk command, 2-406
  - Slave phantom, 2-14
  - Slave processes, listing of running, 2-212
  - SLIST command, 2-340
  - SMLC lines, assigning, 2-22
  - SNADSC command, 2-340
  - Software
    - function for returning attribute of, 3-24
    - function for returning information about, 3-22, 3-23
    - information about, function for returning, 3-23
  - SORT command, 2-341, 2-389
    - sort keys, 2-343
  - Specific ACLs, 2-311
  - SPOOL command, 2-352
    - PROC option, 2-366
  - Spool queues, 2-358
    - listing files in, 2-360
    - modifying files in, 2-363
  - Spooled files
    - canceling, 2-359
    - listing, 2-360
    - modifying, 2-363
  - Spooler phantoms, status of, 2-277
  - Spoolers, listing of running, 2-212
  - SPY command, 2-371
  - SQ command, 2-325
  - Square brackets [ ], as command function characters, 3-1
  - SSR command, 2-328
  - Stack
    - condition, function for returning, 3-9
    - discarding, 2-290
    - dump format of, B-1
    - information on, 2-96
  - START command, 2-372
    - command line restriction, 5-4
  - STAT command, 2-47, 2-374
  - Static mode execution, displaying at prompt, 2-286
  - Static program segments, assignment of, 2-310
  - Static segments, private, information on, 2-205
  - Static-mode programs, restarting, 2-373
  - STATUS command, 2-374
    - used to display user priority levels, 2-46
  - Status information on
    - abbreviation files, 3-3
    - access groups, 2-188
    - communication controllers, 2-374
    - devices, 2-374
    - disks, 2-375
    - DPTX queues, 2-94
    - EPFs, 2-184
    - file units, 2-376
    - half-duplex networks, 2-134
    - ISC sessions, 2-207
    - MESSAGE command parameters, 2-236
    - networks, 2-375
    - NTS, 2-375
    - printers, 2-277
    - project IDs, 2-375
    - remote user IDs, 2-200
    - segments, 2-204
    - semaphores, 2-376
    - terminal, 2-385
    - user IDs, 2-375
    - user processes, 2-211
  - String-handling command functions, summary of, 1-27
  - Strings
    - See also* Character strings
    - function to return length of, 3-25
    - function to test for empty (NULL), 3-26
    - removing characters from, 3-40
  - submit, 6-13, 6-15, 6-37, 6-44
  - Submitting batch jobs, 2-149
  - Subschema tables
    - producing with F77SUBS command, 2-120
    - producing with FSUBS command, 2-126
  - SUBST function, 3-36
  - SUBSTR function, 3-36
  - Substring indexing function, 3-21
  - Subsystems, reentering, 2-289
  - Suffixes, 1-6
  - SVC switch, setting, 2-378
  - SVCSW command, 2-378
  - SYNCSORT command, 2-378
  - System
    - accessing, 2-217
    - leaving, 2-222
  - System Administrator, WATCH access permission, 2-403
  - System console, project assignment, 2-43
  - System information
    - displaying, 2-374
    - function to return, 3-37
    - summary of commands for getting, 1-19
  - System name
    - displaying at prompt, 2-286
    - returning to function, 3-37
  - System prompts, changing, 2-283
  - SYSTEM search rule, 2-329
  - System settings commands, summary of, 1-25
  - System usage, monitoring, 2-394
  - System users, listing names of, 2-376
  - SYSTEM\_INFO function, 3-37
  - SYSTEM\_USAGE function, 3-38
- ## T
- T command, 2-388
  - Tab stops, in ECL, 6-36
  - TAG sort, 2-341
  - Tape catalogs, listing, C-12
  - Tape commands
    - ARCHIVE, C-1
    - ARCHIVE\_RELEASE, C-2
    - ARCHIVE\_RESTORE, C-3
    - ASSIGN, 2-15
    - GENERATE\_CATALOG, C-10
    - LABEL, 2-157
    - LIST\_CATALOG, C-12
    - MAGNET, 2-227
    - MAGRST, 2-228
    - MAGSAV, 2-230
    - TRANSPORT, C-16
    - TRANSPORT\_RELEASE, C-17
    - TRANSPORT\_RESTORE, C-18
    - UX\_TAPE for UNIX formats, 2-398
  - Tape devices, summary of commands for accessing, 1-21

- Tape drives
    - assigning, 2-17
    - information on, 2-374
    - unassigning, 2-392
  - Tapes
    - information on contents, C-13
    - labels, reading, 2-157
    - releasing with ARCHIVE\_RELEASE command, C-2
    - restoring data to disks, 2-228
    - retrieving disk data with MAGRST command, 2-228
    - saving disk data to, C-1
    - transport, listing contents of, C-13
    - writing, 2-157
  - TCF command, 2-383
  - TERM command, 2-107, 2-385
  - Terminal Emulation program, 2-340
  - Terminal line characteristics, setting, 2-317
  - Terminal type, specifying, 6-31
  - .TERMINAL\_TYPES\$, 6-26, 6-31
    - ECL, use by, 2-107
  - Terminals
    - deferring execution of commands at, 2-159
    - delaying output to, 2-86
    - display during WATCH session, 2-404
    - displaying files on, 2-340
    - displaying line characteristics of, 2-385
    - emptying buffers of, 2-302
    - printing text at, 2-390
    - receive state, 2-241
    - recovering from diverted I/O, C-4
    - sessions, recording, 2-58
    - setting characteristics, 1-17
    - setting delay padding for, 2-87
    - setting line characteristics of, 2-385
  - TERMINFO, using, 6-35
  - TERMINFO database, 2-107, 6-33
  - Text formatting, 2-303
  - THEMIS command, 2-387
  - Time
    - displaying at prompt, 2-286
    - displaying clock, 2-82
    - log in, 3-43
    - using date function to return, 3-10
  - TIME command, 2-388
  - Timeslice, major, setting, 2-47
  - TO\_HEX function, 3-39
  - TO\_OCTAL function, 3-39
  - toggle\_case, 6-44
  - toggle\_overlay, 6-44
  - TRAMLC command, 2-389
  - Transferring files, 2-131, 2-301, 2-389
  - TRANSLATE function, 3-40
  - Translators, summary of, 1-15
  - TRANSPORT command, C-16
  - Transport tapes, listing contents of, C-13
  - TRANSPORT\_RELEASE command, C-17
  - TRANSPORT\_RESTORE command, C-18
  - Treenames, function for validating PRIMOS, 3-44
  - Treewalk wildcard expansion, 6-5
  - Treewalking
    - command line processing, 5-6
    - explanation of, 4-15
    - options, 4-18
    - within iteration lists, 4-24
  - TRIM function, 3-40
  - twiddle, 6-44
  - TYPE command, 2-390
- U**
- U, Use access right, 2-178
  - U command, 2-391
  - UNASSIGN command, 2-391
  - unbound, 2-105, 6-13, 6-44
  - UNIX tapes, accessing, 2-398
  - Unqualified pathname, definition of, 1-5
  - UNQUOTE function, 3-41
  - UPCASE command, 2-393
  - upcase\_word, 6-45
  - Uppercase letters, converting ASCII files to, 2-393
  - Usage
    - function to return system, 3-38
    - function to return user, 3-43
  - USAGE command, 2-394
  - User, re-initializing, 2-140
  - User abbreviations, creating, 2-4
  - User commands
    - by function, summary of, 1-11
    - list of documented, 1-9
  - User IDs
    - commands for establishing, 1-12
    - displaying at prompt, 2-286
    - function for validating, 3-44
    - names, description of, 2-217
    - remote, 2-14
    - remote, listing, 2-200
    - returning, 3-41
    - status information on, 2-375
  - User number
    - displaying at prompt, 2-286
    - returning, 3-41
  - User processes
    - function to return information about, 3-41
    - listing of, 2-211
  - User segments, 2-204
  - USER\_INFO function, 3-41
  - USER\_USAGE function, 3-43
  - Users
    - children, listing, 2-212
    - deleting from ACLs, 2-99
    - disabling WATCH access, 2-335
    - displaying at prompt, 2-286
    - displaying the number of, 2-397
    - displaying WATCH monitoring, 2-404
    - information on, 2-376
    - logging out, 2-223
    - multiple logins and WATCH, 2-403
    - phantom, 2-14, 2-268
    - phantoms, listing, 2-212
    - remote, sending messages to, 2-237
    - remote, user information, 3-41
    - remote, WATCH sessions, 2-404
    - resetting, 2-140
    - returning number logged in, 3-37
    - returning type, 3-41
    - servers, listing, 2-212
    - slave, 2-14
    - slaves, listing, 2-212
    - spoolers, listing, 2-212
  - USERS argument of STATUS command, 2-46

USERS command, 2-397  
 UX\_TAPE command, 2-398

## V

V command, 2-402  
 VALIDATE function, 3-44  
 Validation codes for login, 2-218  
 Variable evaluation, function to force, 3-33  
 Variables  
   command line processing, 4-2, 5-5  
   containing functions, 5-5  
   function returning value of, 3-16  
   listing global, 2-215  
   within iteration lists, 4-24  
 Verification options for wildcarding, 4-13  
 VERIFY function, 3-45  
 Verify mode, abbreviation, 2-4  
 -VERIFY wildcard option, 4-10  
 version, 6-45  
 Virtual sense switches, 2-402  
 VPSD command, 2-401  
 VPSD16 command, 2-401  
 VRPG command, 2-401  
 VRTSSW command, 2-402

## W

W, Write access right, 2-178  
 -WALK\_FROM treewalking option,, 4-19  
 -WALK\_TO treewalking option,, 4-19  
 Warning prompt, changing, 2-284  
 WATCH command, 2-334  
   multiple sessions, 2-404  
 WATCH monitoring, 2-403  
   control codes, 2-404  
 .WATCH\$ ACL group, 2-403  
   disabling, 2-335  
 WILD function, 3-46  
 Wildcard options, use with LD command, 2-160  
 Wildcarding  
   command line processing, 5-6  
   DELETE command, use with, 2-88

  explanation of, 4-6  
   selecting by type of object, 4-11  
   selection sequence, 4-8  
   suppressing, 4-24  
   table of wildcard characters, 4-7  
   use with ABBREV command, 2-12  
   use with LD command, 2-160, 2-170  
   use with LIST\_MINI\_COMMANDS command, 2-192  
   use with SIZE command, 2-339  
   use with the WILD function, 3-46  
   use with treewalking, 4-15  
   using with name generation, 4-24  
   verification options, 4-13  
   wildcard name, 4-6  
   wildcard options, 4-10  
   within iteration lists, 4-24

## Wildcards

  function for validating in strings, 3-44  
   listing groups, 3-17  
 WORD command, 2-405  
 WRITE\_FILE function, 3-49  
 Writing tape labels, 2-157  
 Writing text into ASCII files, 3-49  
 Writing to punched cards, C-7

## X

X, Execute access right, 2-178  
 XOFF/XON feature, enabling or disabling, 2-107, 2-386

## Y

yank, 6-41, 6-45  
 yank\_replace, 6-41, 6-45



# Surveys

.....

**Reader Response Form**  
**PRIMOS Commands Reference Guide**  
**DOC3108-8LA**

Your feedback will help us continue to improve the quality, accuracy, and organization of our user publications.

1. How do you rate this document for overall usefulness?

- excellent*    *very good*    *good*    *fair*    *poor*

2. What features of this manual did you find most useful?

---

---

---

---

3. What faults or errors in this manual gave you problems?

---

---

---

---

4. How does this manual compare to equivalent manuals produced by other computer companies?

- Much better*    *Slightly better*    *About the same*  
 *Much worse*    *Slightly worse*    *Can't judge*

5. Which other companies' manuals have you read?

---

---

Name: \_\_\_\_\_

Position: \_\_\_\_\_

Company: \_\_\_\_\_

Address: \_\_\_\_\_

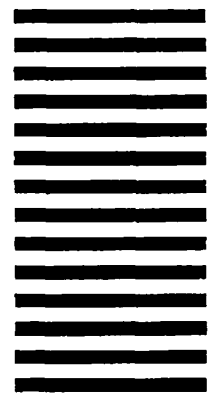
Postal Code: \_\_\_\_\_

Telephone no: \_\_\_\_\_ Ext: \_\_\_\_\_ Best time to call: \_\_\_\_\_

Fold, tape and mail (DO NOT STAPLE)



NO POSTAGE  
NECESSARY  
IF MAILED  
IN THE  
UNITED STATES



**BUSINESS REPLY MAIL**  
FIRST CLASS PERMIT NO. 531 FRAMINGHAM, MA 01701

POSTAGE WILL BE PAID BY ADDRESSEE



**Prime Computer, Inc.**  
**Prime Park/10-3**  
**500 Old Connecticut Path**  
**Framingham, MA 01701-9429**



**Reader Response Form**  
**PRIMOS Commands Reference Guide**  
**DOC3108-8LA**

Your feedback will help us continue to improve the quality, accuracy, and organization of our user publications.

1. How do you rate this document for overall usefulness?

- excellent*    *very good*    *good*    *fair*    *poor*

2. What features of this manual did you find most useful?

---

---

---

---

---

3. What faults or errors in this manual gave you problems?

---

---

---

---

---

4. How does this manual compare to equivalent manuals produced by other computer companies?

- Much better*    *Slightly better*    *About the same*  
 *Much worse*    *Slightly worse*    *Can't judge*

5. Which other companies' manuals have you read?

---

---

Name: \_\_\_\_\_

Position: \_\_\_\_\_

Company: \_\_\_\_\_

Address: \_\_\_\_\_

\_\_\_\_\_

Postal Code: \_\_\_\_\_

Telephone no: \_\_\_\_\_ Ext: \_\_\_\_\_ Best time to call: \_\_\_\_\_

Fold, tape and mail (DO NOT STAPLE)



NO POSTAGE  
NECESSARY  
IF MAILED  
IN THE  
UNITED STATES



**BUSINESS REPLY MAIL**  
FIRST CLASS PERMIT NO. 531 FRAMINGHAM, MA 01701

POSTAGE WILL BE PAID BY ADDRESSEE



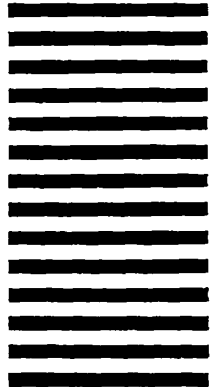
**Prime Computer, Inc.**  
**Prime Park/10-3**  
**500 Old Connecticut Path**  
**Framingham, MA 01701-9429**



Fold, tape and mail (DO NOT STAPLE)



NO POSTAGE  
NECESSARY  
IF MAILED  
IN THE  
UNITED STATES



**BUSINESS REPLY MAIL**  
FIRST CLASS PERMIT NO. 531 FRAMINGHAM, MA 01701

POSTAGE WILL BE PAID BY ADDRESSEE



**Prime Computer, Inc.**  
**Prime Park/10-3**  
**500 Old Connecticut Path**  
**Framingham, MA 01701-9429**



Fold, tape and mail (DO NOT STAPLE)



NO POSTAGE  
NECESSARY  
IF MAILED  
IN THE  
UNITED STATES



**BUSINESS REPLY MAIL**

FIRST CLASS PERMIT NO. 531 FRAMINGHAM, MA 01701

POSTAGE WILL BE PAID BY ADDRESSEE



**Prime Computer, Inc.**  
**Prime Park/10-3**  
**500 Old Connecticut Path**  
**Framingham, MA 01701-9429**

