



***Operator's Guide
to File System
Maintenance***

Revision 22.1

DOC9300-5LA

OPERATOR'S GUIDE TO FILE SYSTEM MAINTENANCE

Fifth Edition

George W. Gove

This guide documents the software operation of the Prime Computer and its supporting systems and utilities as implemented at Master Disk Revision Level 22.1 (Rev. 22.1).

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

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

Copyright © 1989 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, 4050, 4150, 4450, 6150, 6350, 6550, 9650, 9655, 9750, 9755, 9950, 9955, 9955II, DISCOVER, PRIME/SNA, PRIME EXL, FM+, Prime INFORMATION EXL, PRIME MEDUSA, INFO/BASIC, EDMS, MIDAS, MIDASPLUS, PERFORM, PERFORMER, Prime INFORMATION, INFORM, PRISAM, PRAMAN, PRIMELINK, PRIMIX, Prime INFORMATION CONNECTION, PRIMENET, MDL, PRIMEWAY, PRODUCER, Prime INFORMATION/pc, PRIME TIMER, PRIMEWORD, RINGNET, SIMPLE, PT25, PT45, PT65, PT200, PT250, PST 100, PW153, PW200, and PW250 are trademarks of Prime Computer, Inc.

Printing History

First Edition (DOC9300-1LA) December 1984 for Revision 19.4
Update 1 (UPD9300-11A) December 1985 for Revision 20.0
Second Edition (DOC9300-2LA) August 1986 for Revision 20.2
Third Edition (DOC9300-3LA) July 1987 for Revision 21.0
Fourth Edition (DOC9300-4LA) August 1988 for Revision 22.0
Fifth Edition (DOC9300-5LA) July 1989 for Revision 22.1

Credits

Editorial: Mary Skousgaard
Project Support: Steve Horowitz, Rob Perrine, Christopher Wroten
Illustration: Doug Gilbert, Myron Stein, Julie Cyphers
Production: Kathy Normington

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 1-800-343-2533,
Monday through Friday,
8:30 a.m. to 5:00 p.m. (EST).

International

Contact your local Prime
subsidiary or distributor.

Customer Support Center

Prime provides the following toll-free number for customers in the United States needing service:

1-800-343-2320

For other locations, 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

CONTENTS

About This Series	ix
About This Book	xiii

PART I INTRODUCTION AND CONCEPTS

1	THE FILE SYSTEM	
	Online Storage	1-1
	Disk Preparation	1-2
	Disk Maintenance	1-3
	FIXRAT (Pre-Rev. 19.0 Disks)	1-3
	Disk-error Messages	1-4
	File System Concepts	1-4
2	PHYSICAL DISKS	
	What Are Physical Disks?	2-1
	Types of Disks	2-1
	Badspots	2-2

PART II PARTITION MAINTENANCE

3	PHYSICAL DEVICE NUMBERS	
	Types of Disks	3-1
	Size of a Disk Partition	3-2
	Disk Drive Unit Numbers	3-3
	Construction of Physical Device Numbers	3-4
	Physical Device Numbers for SMDs and FMDs	3-5
	Example of Partitioning SMDs	3-10
	Cartridge Module Devices (CMDs)	3-14
	Partitioning Specific Disk Types	3-14
4	USING AND ASSIGNING DISKS	
	File System Disks and Assignable Disks	4-1
	Assigning and Unassigning Disks	4-3

5	FORMATTING DISKS — MAKE	
	Why You Need to Run MAKE	5-1
	Converting Partitions to Rev. 22.1	5-3
	Options Available for Running MAKE	5-5
	Recording Badspots	5-23
	What to Do Before Running MAKE	5-27
	Running MAKE	5-27
	What to Do After Running MAKE	5-30
	Examples of Running MAKE	5-32
	Booting MAKE.SAVE	5-37
6	REPAIRING PARTITIONS — FIX_DISK	
	What Is FIX_DISK?	6-1
	How Does FIX_DISK Work?	6-2
	When Should You Use FIX_DISK?	6-4
	What to Do Before Running FIX_DISK	6-7
	The Procedure for Running FIX_DISK	6-7
	Options Available for Running FIX_DISK	6-9
	What to Do After Running FIX_DISK	6-25
	Examples of Running FIX_DISK	6-26
	Invoking FIX_DISK From Magnetic Tape	6-38

PART III SPECIAL TOPICS

7	ROBUST PARTITIONS	
	What Is A Robust Partition?	7-1
	Understanding The -FAST Option of FIX_DISK	7-5
	Understanding The Robust Partition File System	7-7
	Evaluating The Use Of Robust Partitions	7-14
	Creating Robust Partitions	7-18
	Administration of the Files on a Robust Partition	7-20
	FIX_DISK and Robust Partitions	7-27
	ADDISK and Robust Partitions	7-30
	Mirroring Robust Partitions	7-31
8	DYNAMIC BADSPOT HANDLING	
	Requirements for Dynamic Badspot Handling	8-2
	Disk Errors	8-3
	Rev. 22.1 Disks and Dynamic Badspot Handling	8-5
	Need for Conversion to Rev. 22.1 Disks	8-5
	Using COPY_DISK, PHYSAV, and PHYRST	8-9
	Effect of Dynamic Badspot Handling on Operator Commands	8-10
	Messages Associated With Dynamic Badspot Handling	8-15

9	DISK MIRRORING	
	The Purpose of Mirroring	9-1
	Mirroring Requirements	9-2
	Configuration Directives for Mirroring	9-3
	Catch-up Copy	9-5
	Effect of Mirroring on Operator Commands	9-6
	PRIMOS Mirroring Commands	9-8
	Errors	9-13
	Mirroring Messages	9-16
10	RECORD ALLOCATION AND SECTORING	
	PRIMOS Record Allocation	10-1
	Allocation Order	10-3

APPENDICES

A	MAKE MESSAGES	A-1
B	FIX_DISK MESSAGES	B-1
	FIX_DISK Error Messages	B-2
C	MIRRORING MESSAGES	C-1
D	DISK ERRORS	D-1
	Nonintelligent Disk Controllers	D-1
	Intelligent Disk Controllers	D-4
E	MAKE_ROBUST MESSAGES	E-1
F	SUMMARY OF COMMAND SYNTAX FOR MAKE AND FIX_DISK	F-1
	Summary of MAKE Command Syntax	F-1
	Summary of FIX_DISK Command Syntax	F-5
G	GLOSSARY	G-1
	INDEX	Index-1

ABOUT THIS SERIES

The Operator's Guide series is designed to help you, as a System Operator or a System Administrator of a Prime® computer, do your job. This preface describes the eight Operator Guides and other Prime documentation that are particularly useful for System Operators and System Administrators. To display an online list of all Prime documentation, use the HELP DOCUMENTS command.

FOR THE SYSTEM OPERATOR

Before reading this book, you should have some familiarity with Prime systems. A good way to begin is to read the *PRIMOS User's Guide* (DOC4130-5LA), which explains the PRIMOS® file management system and provides introductory and tutorial information about essential commands and utilities. When you read any Prime documentation, be sure to consult the section entitled Prime Documentation Conventions, which follows the preface; this section is essential to understanding how information is presented.

After you are familiar with Prime systems, read the *Operator's System Overview*, which outlines the material in the Operator's Guide series. Then select the other books in the series as they apply to the tasks you must perform.

As you learn more about system operations, you will use the *Operator's Guide to System Commands* as a reference for many of the special system commands and arguments that you, as an Operator, will need to perform your job. The *Operator's Guide to System Commands* documents most of the commands described in the Operator's Guide series.

The Operator's Guide Series

The following books contain detailed information for the System Operator.

- *Operator's System Overview* (DOC9298-3LA) introduces the series and describes computer-room operation of Prime systems.
- *Operator's Guide to System Monitoring* (DOC9299-3LA) describes how to monitor system usage, activity, and messages.

- *Operator's Guide to File System Maintenance* (DOC9300-5LA) describes the PRIMOS file system and explains how to format disk partitions, run the disk partition maintenance program, determine physical device numbers, and interpret disk-error messages. In addition, this book discusses in detail dynamic badspot handling, mirroring, robust partitions, and record allocation.
- *Operator's Guide to the Batch Subsystem* (DOC9302-3LA) describes how to set up, monitor, and control the Batch subsystem.
- *Operator's Guide to the Spooler Subsystem* (DOC9303-3LA) describes how to set up, monitor, and control the Spooler subsystem.
- *Operator's Guide to System Commands* (DOC9304-4LA) serves as a reference guide for most of the commands described in the other books in the series.
- *Data Backup and Recovery Guide* (DOC10129-1LA) and its update package (UPD10129-11A) describe how to save information on disk or tape and how to restore that information when it is needed.
- *Operator's Guide to Prime Networks* (DOC10114-1LA) and its update package (UPD10114-11A) provide reference information about running network-related programs and monitoring network events.

Other Books for the Operator

- *Operator's Master Index* (DOC10110-4LA) indexes all the Operator and System Administrator Guides. Consulting this index is often the quickest way to find which manual has the information you need.
- The computer handbook for your particular CPU explains such topics as booting the system, shutting down PRIMOS, handling halts and hangs (including warm starts), performing tape dumps, and using the Virtual Control Panel (VCP).
- The *Using Your CPU* guide (available only for office CPUs) is intended for nontechnical users who are acting as System Operators, and covers system startup and shutdown, system backups, troubleshooting, and other day-to-day system management issues.
- *MAGNET User's Guide* (DOC10156-1LA) and its update package (UPD10156-11A) describe the MAGNET utility, used to transfer data by magnetic tape from other operating systems to PRIMOS and vice versa.

FOR THE SYSTEM ADMINISTRATOR

In addition to the documentation in the Operator Guide series, be sure to read the System Administrator Guide series, which describes how to set up, configure, and maintain PRIMOS:

- *System Administrator's Guide, Volume I: System Configuration* (DOC10131-2LA and RLN10131-21A) explains how to set up a system and allocate resources.

- *System Administrator's Guide, Volume II: Communication Lines and Controllers* (DOC10132-2LA and RLN10132-21A) explains how to configure communication lines.
- *System Administrator's Guide, Volume III: System Access and Security* (DOC10133-2LA and RLN10133-21A) explains PRIMOS security features and how to prevent unauthorized use of your system.
- *DSM User's Guide* (DOC10061-2LA) explains how to use the Distributed Systems Management (DSM) subsystem, including how to configure and operate DSM.

The System Administrator Guides also provide information about most of the commands necessary to operate your Prime system.

OTHER RECOMMENDED READING

In addition to the books listed above, you may find the following books useful:

- *New User's Guide to EDITOR and RUNOFF* (FDR3104-101B) is a basic reference for any user of a Prime system and provides information about the Prime text editor and formatter.
- *PRIMOS Commands Reference Guide* (DOC3108-7LA and RLN3108-71A) provides detailed information about user commands.
- *PRIMENET Planning and Configuration Guide* (DOC7532-4LA) describes how to plan, configure, and maintain PRIMENET™ software for a system.
- *User's Guide to Prime Network Services* (DOC10115-1LA) describes networking services that enable users to access files remotely, transfer files, and log in to other 50 Series™ systems on a network.
- *NTS User's Guide* (DOC10117-2LA) explains the Network Terminal Service (NTS).
- *50 Series Technical Summary* (DOC6904-2LA) describes the features of the 50 Series systems, including advanced architecture concepts and the software and hardware products the concepts support.
- *System Architecture Reference Guide* (DOC9473-3LA) describes internal functioning of all 50 Series computers.
- *Subroutines Reference II: File System* describes subroutines that deal with access to and management of the file system.

ABOUT THIS BOOK

The *Operator's Guide to File System Maintenance* is one of a series of books designed to help you, as an Operator or a System Administrator of Prime computers, to do your job. It discusses the tasks necessary to create and maintain the file system.

The purpose of this book is to provide the information you need to know in order to maintain your file system. The operations you must become familiar with include how to

- Determine physical device numbers for partitions
- Format partitions
- Repair partitions
- Convert the format of a partition from one revision to another
- Decode disk error messages

If you are a System Operator, this book is intended to help you become familiar with the tasks required to keep the file system operating efficiently.

If you are a System Administrator, this book is intended to help you gain insight into some of the tasks required of the Operator to maintain the file system.

In addition, this book discusses the concepts of

- The file system
- Using and assigning disks
- Robust partitions
- Dynamic badspot handling
- Mirroring
- File system record allocation

NEW FILE SYSTEM FEATURES AT REV. 22.1

The major new file system features at Rev. 22.1 involve the introduction of robust partitions. A **robust partition** is a new type of disk partition at Rev. 22.1 that is less vulnerable to errors due to system halts. In many cases when a system halt occurs, you may rapidly repair the file system on a robust partition by using the `-FAST` option of `FIX_DISK`. You create robust partitions by using the `MAKE_ROBUST` utility.

Robust partitions contain a subset of the file system objects on a standard (nonrobust) PRIMOS partition. All user files on a robust partition are physically organized as CAM files, although all three logical file types (SAM, DAM, and CAM) are supported on robust partitions. CAM files can contain an unlimited number of extents at Rev. 22.1. You may set and change the maximum and minimum extent sizes for CAM files with the `MAKE` and `FIX_DISK` options `-MAX_EXTENT_SIZE` and `-MIN_EXTENT_SIZE`.

Chapter 7 discusses robust partitions and the `MAKE_ROBUST` utility in detail in addition to the features and use of robust partitions. CAM files and the parameters associated with them and the concept of logical file type are also discussed in Chapter 7 as well as in Chapter 1 and the `MAKE` and `FIX_DISK` chapters, Chapters 5 and 6.

`FIX_DISK` has a new option, `-CONVERT_22.1`, to allow you to convert Rev. 22.0 partitions to Rev. 22.1 standard partitions. Note that this option converts only Rev. 22.0 partitions to Rev. 22.1 and these partitions become standard partitions; pre-Rev. 22.0 partitions must be converted to Rev. 22.1 by using `MAKE`.

`MAKE` has a new argument to the `-DISK_REVISION` option, `22.1`. `MAKE` creates a Rev. 22.1 partition if you do not specify a specific disk revision. `MAKE` also has a new option, `-INIT`, and the default method of operation is changed so that `MAKE`, by default, does not initialize file system records. Thus, it is now unnecessary for you to use the `-NO_INIT` option. If you do wish to initialize file system records, you must use the `-INIT` option.

ORGANIZATION OF THIS BOOK

This particular book, the *Operator's Guide to File System Maintenance*, is divided into four parts:

- Part I introduces you to the file system and discusses disks, the storage media used to keep files.
- Part II tells you what physical device numbers are and how to construct them. It also explains how to assign and use disks. It describes in detail how to create and format disk partitions and how to repair file structures.

- Part III describes in detail the concepts of robust partitions, dynamic badspot handling, mirroring of partitions, and file record allocation.
- The appendices explain the meaning of error messages related to disk operations, summarize the commands and options used for creating and repairing partitions, and include a glossary of terms used in this book. A comprehensive index lists all subjects discussed in this book.

You are expected to have some familiarity with Prime systems before reading this book. If you are not familiar with the PRIMOS operating system, you should read the *PRIMOS User's Guide* (DOC4130-5LA), which explains Prime's file management system and provides introductory and tutorial information about essential commands and utilities.

HOW THIS BOOK IS STRUCTURED

This book contains 10 chapters and 6 appendices:

- Chapter 1, The File System, explains terminology, the preparation and maintenance of disks used by PRIMOS, and concepts related to the file system.
- Chapter 2, Physical Disks, describes varieties of disks used on Prime systems, terminology related to these disks, badspots, and equivalence blocks.
- Chapter 3, Physical Device Numbers, explains physical device numbers and how to construct them and describes partitioning of disk types supported by PRIMOS.
- Chapter 4, Using and Assigning Disks, explains the procedures to assign disks to yourself in order to perform maintenance on them and the procedures to make disks available to users.
- Chapter 5, Formatting Disks — MAKE, explains how to use the MAKE utility, which formats and partitions disks for use on your system.
- Chapter 6, Repairing Partitions — FIX_DISK, explains how to use the FIX_DISK utility, which repairs file partitions.
- Chapter 7, Robust Partitions, describes robust partitions, the file system objects on them, allocation of CAM files, fragmentation, and the use of utilities in conjunction with robust partitions. This chapter also discusses the MAKE_ROBUST utility for creating robust partitions. In addition, this chapter describes the concept of logical file type.
- Chapter 8, Dynamic Badspot Handling, explains the concept of dynamic badspot handling introduced at Rev. 21.0, which makes use of intelligent disk controllers and allows partitions to be mirrored.
- Chapter 9, Disk Mirroring, explains the concept of partition mirroring introduced at Rev. 21.0, which allows you to dynamically make duplicate copies of partitions and which depends on dynamic badspot handling by intelligent disk controllers.

- Chapter 10, Record Allocation and Sectoring, explains the concepts of reverse and forward sectoring and interleave factors introduced at Rev. 21.0, which provide a method to allow efficient record allocation and use of intelligent disk controllers.
- Appendix A, Messages From MAKE, explains error messages and other messages from MAKE.
- Appendix B, FIX_DISK Messages, explains FIX_DISK messages.
- Appendix C, Mirroring Messages, explains messages related to the mirroring process.
- Appendix D, Disk Errors, explains error messages related to disk operations.
- Appendix E, explains error messages related to the MAKE_ROBUST utility.
- Appendix F, Summary of Command Syntax for MAKE and FIX_DISK, summarizes the command syntax for the MAKE and FIX_DISK utilities.
- Appendix G, Glossary, provides summary explanations of terms introduced in the main text.

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.	FIX_DISK -HELP
<i>italic</i>	In command formats, words in lowercase bold italic indicate variables for which you must substitute a suitable value. In text and in messages, variables are in non-bold lowercase italic.	MAKE -DISK <i>pdev</i> The <i>pdev</i> you used is invalid.
Abbreviations in format statements	If a command or option has an abbreviation, the abbreviation is placed immediately below the full form.	MIRROR_ON MON
Brackets []	Brackets enclose a list of one or more optional items. Choose none, one, or several of these items.	MAKE [-MINSIZ -MAXSIZ]
Braces { }	Braces enclose a list of items. Choose one and only one of these items.	MOFF {<i>-SP</i> <i>-SS</i>}
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.	FIX_DISK [{<i>-CONVERT_21</i> <i>-CONVERT_20</i>}]

<i>Convention</i>	<i>Explanation</i>	<i>Example</i>						
<u>Underscore</u> in examples	In examples, user input is underscored but system prompts and output are not.	OK, <u>ADDISK 460</u> Starting up revision 22.1 disk OK,						
Ellipsis ...	An ellipsis indicates that you have the option of entering several items of the same kind on the command line.	SHUTDN <i>pdev-1</i> [... <i>pdev-n</i>]						
Hyphen -	Wherever a hyphen appears as the first character of an option, it is a required part of that option.	MAKE -DISK						
Default indicator ●	In a list of options, a bullet indicates the default choice, if one exists. If you do not select an option, the system chooses the default option.	<table><tr><td>A</td><td>●</td></tr><tr><td>O</td><td></td></tr><tr><td>D</td><td></td></tr></table>	A	●	O		D	
A	●							
O								
D								
Subscript	A subscript after a number indicates that the number is not in base 10. For example, a subscript 8 is used for octal numbers.	200 ₈						
Key symbol	In examples and text, the name of a key enclosed by a rectangle indicates that you press that key.	Press Return						
Boldface terms	A term in boldface is defined for the first time in the text and is also included in the Glossary. Also used on first mention of an option.	A pdev is an octal number. Use the -INIT option.						

PART I
INTRODUCTION AND CONCEPTS

THE FILE SYSTEM

This chapter provides introductory information about the PRIMOS file system. (For a detailed discussion of files, see the *Advanced Programmer's Guide, Volume II: File System.*)

The sections that follow offer an overview of

- Online storage
- Disk preparation with the MAKE utility
- Disk maintenance and repair with the FIX_DISK utility
- The structure of the PRIMOS file system

ONLINE STORAGE

All Prime systems use **online storage**, or disks, to provide a storage area for files, or data, and to allow large volumes of data storage. Disks store much larger quantities of data than main memory can, and they retain the information whether power is on or off.

A disk drive includes a collection of physical disks (surfaces, platters) and data can be stored on the disk surfaces. The disks are generally divided into a number of sections, or **logical disks**, which use variable numbers of surfaces or the entire physical disk can be one logical disk. These logical disks are generally referred to as **partitions**, but may also be referred to as **disks**, **devices**, **volumes**, and **MFDs**. Generally, the term *partition* is used throughout this book to refer to the logical division of physical disks into file system partitions.

DISK PREPARATION

Unlike storage provided by main memory, disk storage must be prepared to conform to the addressing method used by PRIMOS and must be given a name before it is used. This preparation is called **formatting**, **partitioning**, or **creating** a disk or a partition. It is generally performed by an Operator using a PRIMOS utility called MAKE.

The MAKE Utility

This section presents a short, introductory description of how to use MAKE; more detailed information is presented in Chapter 5, Formatting Disks — MAKE.

To format (prepare) a disk, you must first determine how large the partition is to be and where it is within the physical disk so that you can determine the partition's physical device number. (Physical device numbers are discussed in detail in Chapter 3.) Use the DISKS command to add the the new partition to the **Assignable Disks Table**. This table, which is kept by PRIMOS, contains a list of physical device numbers of partitions that may be specifically assigned to a single user. Then use the ASSIGN command to assign the partition to yourself and use the MAKE utility to format and create the partition. Once you have formatted a user disk, or partition, that partition conforms to system requirements for use by PRIMOS. As part of the formatting process, you also name the partition. In addition, the formatting process creates these file system objects:

- The files BOOT, DSKRAT (which gets as its filename the name of the partition) and, conditionally BADSPT or DYNBSP if there are any defects, or **badspots**, on the partition being formatted
- The top-level directory MFD
- The empty top-level directories CMDNC0 and DOS

You may also have to create a partition that will be used for paging. PRIMOS uses **virtual memory** to run programs. This technique allows the computer to run a program that is larger than main memory. Portions of the program that are not being used may either reside in main memory or be moved out of main memory onto disk storage. This operation of moving parts of programs to and from main memory is performed automatically by PRIMOS and is called **paging**. Even a program that may fit into main memory when run by itself is paged by PRIMOS when other users are also running programs. Paging thus makes the available memory appear much larger than it actually is.

Chapter 5 tells you how to create a paging partition. For more detailed information on virtual memory and paging, see the *System Architecture Reference Guide*. For an introductory discussion of this subject, see the *Operator's System Overview*.

DISK MAINTENANCE

Disk maintenance consists of checking the quality of data on disks and using utilities designed to make any necessary restorations to the file structure.

The FIX_DISK Utility

To provide an efficient and thorough way to check the integrity of data on a PRIMOS disk, PRIMOS provides a file maintenance program, FIX_DISK. The purpose of FIX_DISK is to repair file structures that may have been damaged by a power loss or by an unexpected system halt or hang. In addition, FIX_DISK is also used to repair file structures that have been damaged by using PRIMOS II to make changes to files or directories on Rev. 19.0 partitions. Using PRIMOS II for these purposes results in incorrect **Access Control Lists (ACLs)** or quota information that require fixing. If PRIMOS finds that the file structure has been damaged, it asks you to run FIX_DISK.

Note

PRIMOS II (DOS) will *not* write on partitions formatted with Rev. 20.0 or subsequent revisions of MAKE. It will read these partitions and will continue to both read and write Rev. 19.0 and earlier partitions.

FIX_DISK reads every physical record in every file, top-level directory, and segment directory (SEGDIR), and checks to see that the information is consistent. If discrepancies are found, FIX_DISK generates error messages. You can request FIX_DISK to repair damaged files and quota information. You can also request FIX_DISK to convert a disk made as a pre-Rev. 19.0 partition to a Rev. 19.0 partition; to convert a disk made as a pre-Rev. 20.0 partition to a Rev. 20.0 partition; to convert a disk made as a pre-Rev. 21.0 partition to a Rev. 21.0 partition; and to convert a disk made as a Rev. 22.0 partition to a Rev. 22.1 partition. For details and examples, refer to Chapter 6, Repairing Partitions — FIX_DISK.

FIXRAT (PRE-REV. 19.0 DISKS)

FIXRAT is an obsolete file maintenance program that is not shipped on the master disk at Rev. 22.1 but you may have FIXRAT on your system if you are upgrading from pre-Rev. 22.1 PRIMOS. FIXRAT checks file integrity on pre-Rev. 19.0 disks. However, FIX_DISK can be used on pre-Rev. 19.0 disks, and it is recommended that you use FIX_DISK to repair partitions of any revision.

WARNING

FIXRAT *must not* be used on any partition that is made as a Rev. 19.0 or later partition. Use of this command on Rev. 19.0 or subsequent revision partitions may result in the loss of data. To repair partitions of any revision, use FIX_DISK.

DISK-ERROR MESSAGES

When disk errors are detected by a disk controller because of read errors or write errors, messages are sent to the supervisor terminal and/or to a log file, as configured by DSM. The format and meaning of these messages are discussed in Appendix D, Disk Errors.

FILE SYSTEM CONCEPTS

The remainder of this chapter gives you a brief overview of the structure of the PRIMOS file system. Although this information is not necessary for you to perform your duties as an Operator, it will help you to understand how the file system works.

Purpose of the File System

The purpose of the file system is to simplify the manipulation of large quantities of data. The file system

- Automatically allocates disk storage space for files
- Efficiently stores and retrieves data on behalf of the user
- Provides a file system object management service to allow users to cluster related information together and to refer to the information in files by name

Allocating Disk Storage Space — the DSKRAT: In order to automatically allocate disk storage space, PRIMOS maintains a file on each disk, the **Disk Record Availability Table (DSKRAT)**, whose name is the partition name. This file contains a header, which describes the partition, and a bit map, which indicates the status (used or unused) of every physical record on the disk and, thus, is a record of available space on that disk.

The DSKRAT file gets the partition name when you create the disk with the MAKE command. For example, if the name of a partition is ZEPHYR, the name of the DSKRAT file for this partition is also ZEPHYR. Each record on the partition is represented in the file by a single binary bit; a one (1) means the record is available, and a zero (0) means it is in use. PRIMOS uses this information to allocate disk space automatically.

Referring to Files by Name: In order to reference files by name, you must select the desired file by giving PRIMOS or its utility programs a string of alphanumeric characters that constitute the filename. The file system reserves one special file as a directory, the **Master File Directory (MFD)**. The MFD contains the names of other files and their locations on the partition. PRIMOS can find the MFD readily, because both its name (MFD) and its location on the partition are always the same.

Clustering Related Information: Clustering of related information is achieved in two ways. The first way is to have many file directories; this approach allows similar files to have their names and locations saved in one file directory. The second way is to allow nested file directories so that a file directory may contain names, not only of files, but also of other file directories. Thus, each user may divide files into appropriate groups and subgroups. File directories also provide some degree of access protection to the files contained within them. To examine the files in a directory, the user must have appropriate access to that directory. File directories are discussed in the next section.

File Directories

Directories are specialized files containing entries that point to files or to other directories. PRIMOS uses a hierarchical file structure, also known as a **tree structure**, that allows directories to be nested as the user wishes. See Figure 1-1.

Directories are the nodes in the file system tree structure; files are the branches. Each partition has one special directory called a Master File Directory (MFD), which contains an entry for each top-level directory in the MFD. Directories are either top-level directories or segment directories. (See the following discussion.) In turn, each top-level directory contains an entry for every file and subdirectory in that top-level directory. Top-level directories and MFDs are accessed by PRIMOS in much the same way as other files.

Master File Directory (MFD): Each disk partition contains one MFD file as an index to the first physical record of each top-level directory and file in the MFD. The MFD has the same format as any directory. The first record of the MFD begins at physical record 1 of the partition. Figure 1-5, at the end of this chapter, shows a simple schematic of the relationship among the MFD and top-level directories, subdirectories, segment directories, and SAM, DAM, and CAM files.

MFD entries include an identification for some special files that have unique use in the file system and are not normally accessed by a user. These files are BOOT, the DSKRAT, BADSPT, DYNBSP, and MFD, and are discussed in Chapter 5.

Top-level Directory: A file that contains a list of subordinate directories and files by name is a **top-level directory**. The top-level directory is the point from which the user accesses those subordinate directories and files.

Each user is assigned a directory, and that directory is the initial attach point (IAP) to the system when the user logs in. The directory contains an entry for every file and subdirectory created by the user. Each entry includes a filename and the address of the first physical record of the file. This address is the **beginning record address (bra)** of the entry.

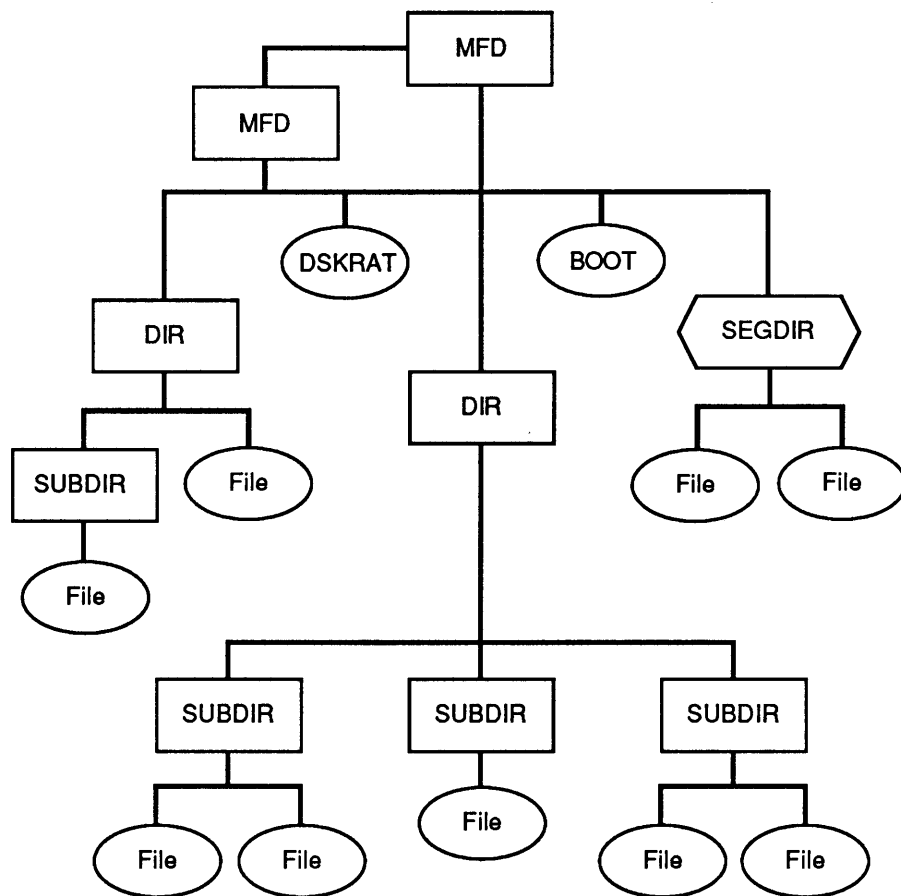


FIGURE 1-1. Prime File System Tree Structure

Segment Directories: Segment directories (SEGDIR) are similar to file directories except that entries in segment directories are identified by single entry numbers ranging from 0 through 65535. Segment directories are therefore limited to 65536 (64K) entries. These directories are generally referenced by programs rather than by users.

Segment directories differ from top-level directories in one fundamental respect: they contain file numbers rather than file names. In terms of the file system, the files in a segment directory have no symbolic names. However, the user may refer to files within a segment directory by their entry number, such as

1
2
99
185

The preceding four numbers are names of files in segment directories. Segment directory entry numbers are decimal numbers.

Hashed Directories: Since Rev. 20.0, the MAKE utility creates disk partitions that support hashed directories. The term **hashed** refers to the way in which PRIMOS gains access to data controlled by that directory. Hashing speeds up processing because the system does not have to perform a sequential search to locate a file system object.

The process of hashing involves converting the name of a file or directory to a number, known as its **hash index**. The hash index in turn is used to look up the list of names that have this index. Then, the names are compared. The system can then locate the file system object using this hash index. All hashed directories are Direct Access Method (DAM) files with an index block for the records in the directory.

Only ACL directories can be hashed. Password directories can still be created but they will not be hashed.

Note

Rev. 22.1 PRIMOS continues to support pre-Rev. 22.1 format partitions, but a Rev. 22.1 format partition must not be locally connected to a pre-Rev. 22.1 system because of differences in the DSKRAT headers. A Rev. 22.1 partition, however, may be accessed remotely by a pre-Rev. 22.1 system.

File Structures

File storage on a disk is composed of many separate blocks of data-recording space called disk records, or sectors. How these records are put together to make a file can greatly affect the efficiency of data retrieval from a file. To allow for different uses of files, the file system has three different ways of linking disk records to form a file:

- The **Sequential Access Method (SAM)** is very efficient for sequential access to a file. However, this method is much slower for repeated random positioning over a file because each record must be accessed sequentially.

- The **Direct Access Method (DAM)** is efficient for random file access to medium size files (less than 512 records) but uses additional disk records for file index information to allow direct access to each record.
- The **Contiguous Access Method (CAM)** is efficient for random file access to large size files (more than 512 records). A CAM file requires one more disk record than a SAM file requires for file index information. The primary benefit of CAM files, however, is that records for them are preallocated and fragmentation of files on disk is reduced considerably.

Thus, SAM, DAM, and CAM files differ in performance and storage efficiency but are functionally equivalent in all other respects. The structural differences among these file types are transparent to the user.

SAM Files: SAM file organization is the basic way of structuring disk records into a sequentially ordered set. A SAM file is a threaded list of physical disk records in which each record points to the preceding record, points to the next sequential record, and points to the first record in the file, as shown in Figure 1-2.

Thus, a SAM file is a collection of disk records chained together by forward and backward pointers. Each record in a SAM file (or in any other type of file) contains a pointer to the Beginning Record Address (BRA) of the file. The first record has a pointer (root, or parent, pointer) to the directory in which this file is an entry. The file system maintains the record headers and is responsible for the structure of the records on the disk.

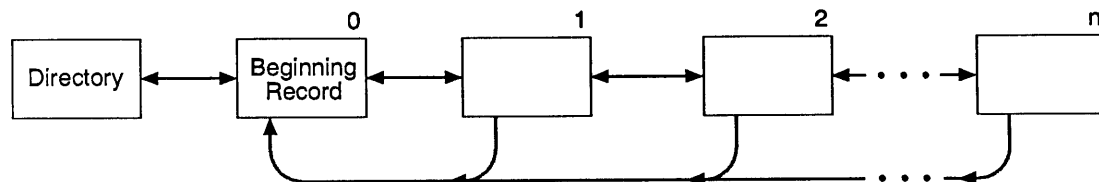


FIGURE 1-2. SAM File Structure

DAM Files: DAM file organization uses the SAM file method of making a sequentially ordered set but uses a special technique to rapidly access any data record directly:

- The first record of a DAM file is reserved for use by the system. No user data is ever written in this record, which is always the top-level index. An index is a list of pointers either to subindexes or to the records in a DAM file. See Figure 1-3 for a graphic representation.

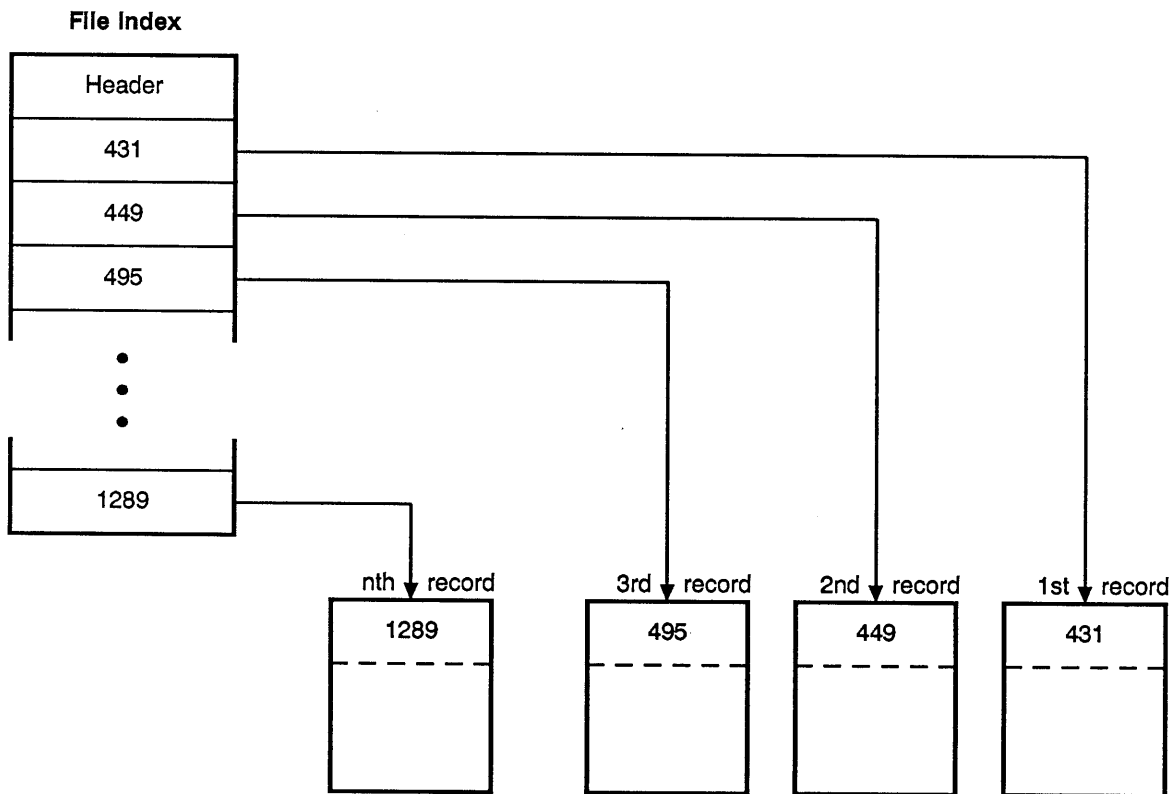


FIGURE 1-3. DAM File Structure

- The top-level index is always exactly one record long. If the file is short (less than 512 records), the record address pointers point to records containing user data. If there are more than 512 data records, a multilevel index is maintained so that any record in the file can be directly accessed.

CAM Files: Whereas each SAM and DAM file contains records that are allocated wherever space is available (and therefore may not be contiguous), a CAM file contains groups of contiguous records only. These groups of contiguous records are called **extents** and are indexed by the CAM file's **extent map**. The extents themselves are not contiguous; only the records within each extent are contiguous. An extent is a variable number of contiguous 2048-byte records.

Figure 1-4 illustrates the structure of a CAM file. For each extent, the extent map records the address of the first record in the extent and the length of the extent, that is, the number of records in the extent. The extent map occupies the first record in the file.

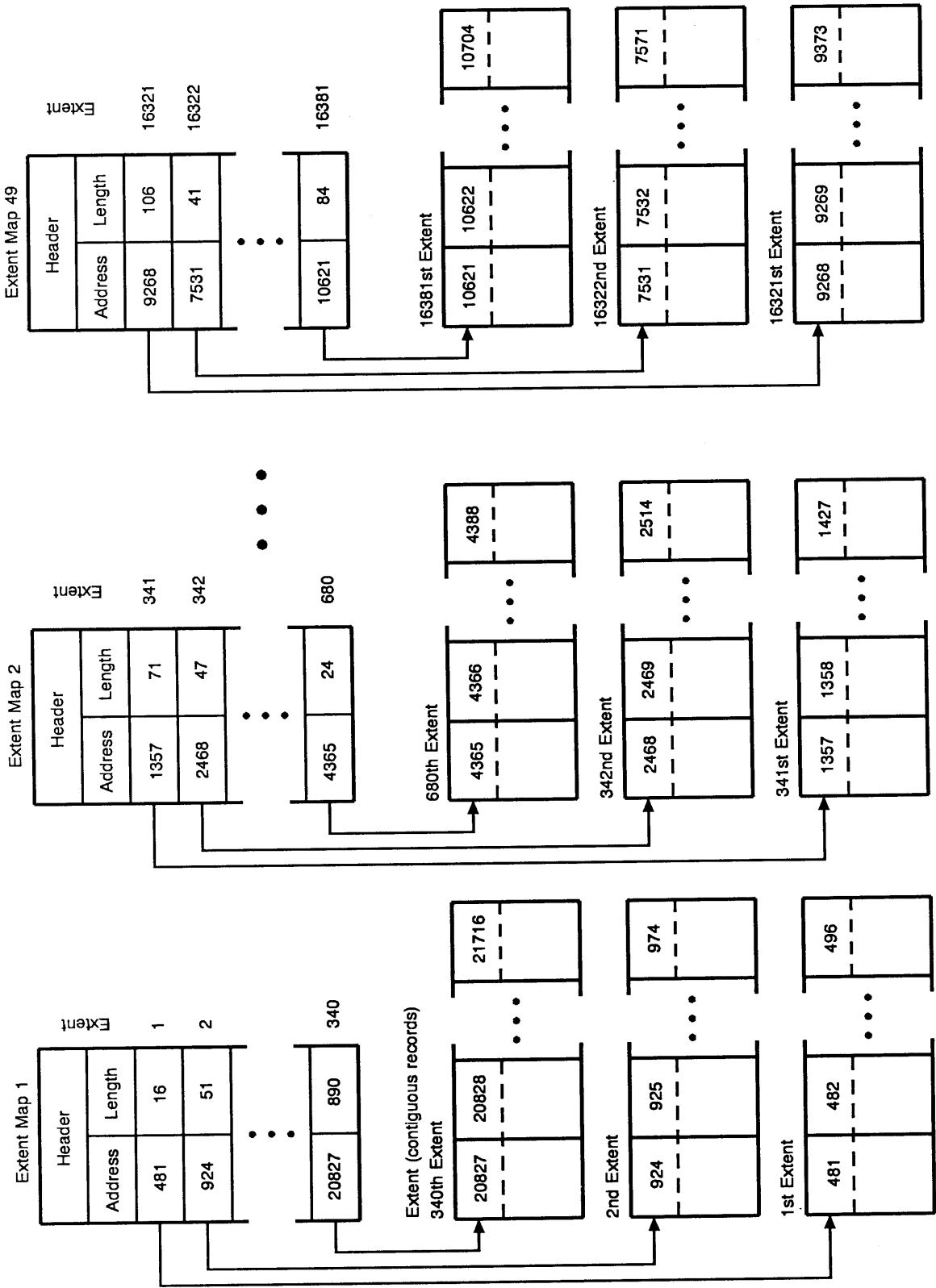


FIGURE 1-4. CAM File Structure

Within each extent, the records exist in sequential order. Because of this order, CAM files require less indexing information than DAM files unless the CAM file is fragmented into many small extents. A DAM file index contains one disk address for each record. The CAM file's extent map contains one disk address for each extent, which means there is one pointer for many records.

By using fewer pointers, PRIMOS can access a record in a CAM file more efficiently than in a DAM file that contains more than 512 randomly placed records. For example, if a program needs data in record 974, as shown in Figure 1-4, PRIMOS reads the extent map into memory when the file is opened and then the effect is as if PRIMOS reads only two entries in the extent map and then moves to the second extent. Because extents contain only sequentially ordered records, PRIMOS moves automatically to the 51st record in the extent to find record 974. If this file were a DAM file, PRIMOS would have to read two index blocks and then find the pointer to record 974 in the second index block.

The real benefit of CAM files is that, since records are allocated in contiguous blocks, there is less fragmentation of CAM files and, thus, less seeking is required of the disk drive.

CAM files have various numeric parameters. Although all records are equal in size, the number of records per extent can vary from 1 to 32,767. The maximum of 32,767 records per extent is a software limitation. In actual use, however, the maximum number of records is also limited by the number of consecutive records that are available on the disk.

The number of extents per file can range from 0 to a maximum of 340, inclusive, on systems running Rev. 20.0 through Rev. 22.0 PRIMOS. At Rev. 22.1, the number of extents per file is unlimited for both standard and robust partitions if the partition is created by using Rev. 22.1 MAKE or if you convert a Rev. 22.0 partition to Rev. 22.1 with Rev. 22.1 FIX_DISK. (The actual maximum number of extents at Rev. 22.1 is 16,381.) Each extent map for a CAM file can index a maximum of 340 extents; thus, a CAM file can have multiple extent maps. As shown in Figure 1-4, a file with the maximum number of extents with 340 extents indexed in each extent map would have 49 extent maps.

It is not possible to create or access CAM files by using pre-Rev. 20.0 versions of PRIMOS.

Figure 1-5 schematically shows the relationship among the various file system object types discussed in the preceding sections.

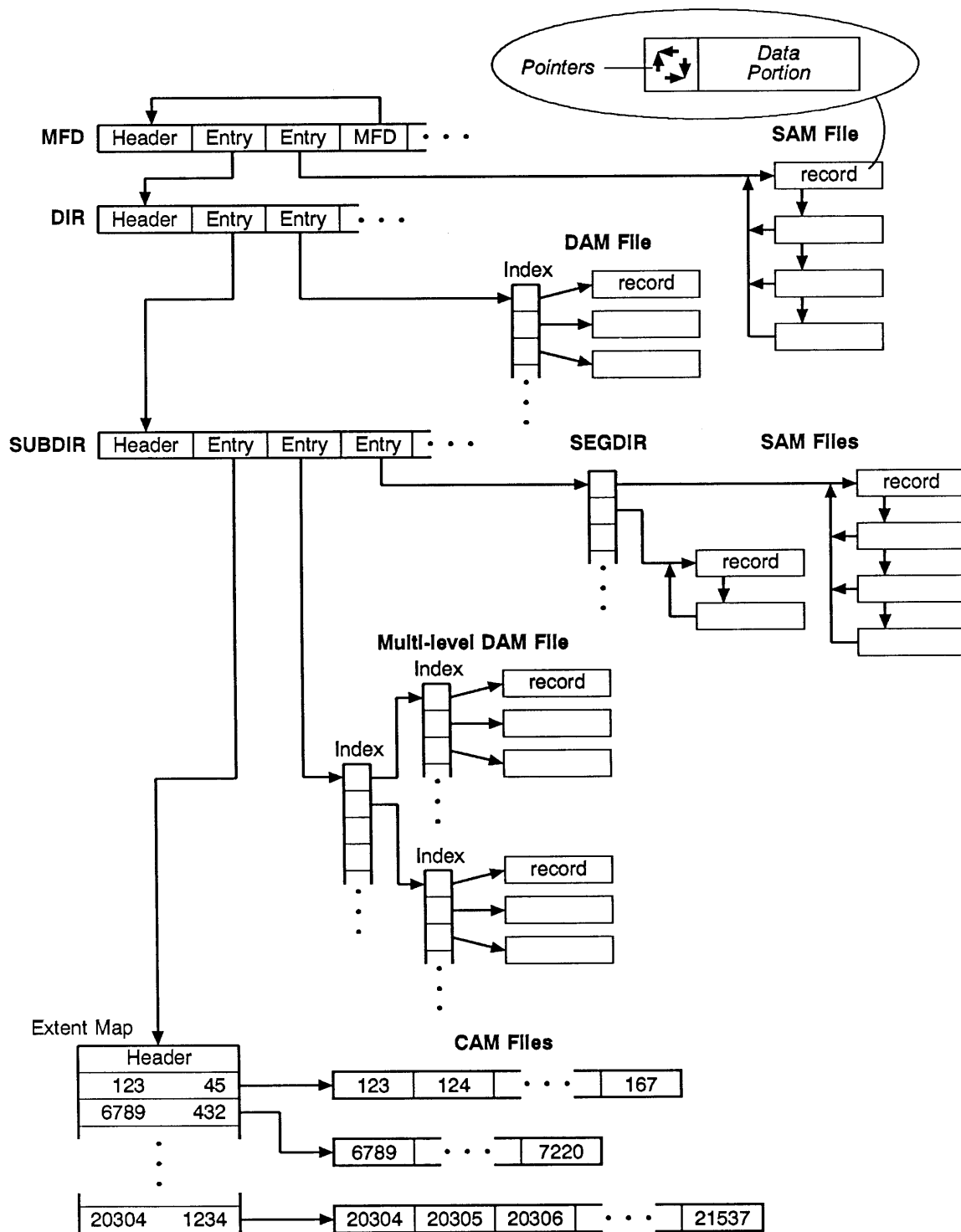


FIGURE 1-5. Relation Among File Structures

PHYSICAL DISKS

This chapter briefly describes the types of disks that Prime systems support and introduces some of the terminology associated with the use of these disks. It then goes on to describe badspots and the utilities that use badspot information.

WHAT ARE PHYSICAL DISKS?

The term **physical disk** refers to an entire multi-surface disk pack or disk drive. **Disk geometry** refers to the physical attributes of a physical disk such as the number of tracks, or cylinders, per surface, the number of records, or sectors, per track, and the number of surfaces making up the disk. The physical disk has a medium on which data is written. The medium may have badspots, or physical defects, on it that prevent data from being written to or read from those areas of the disk.

Several of the MAKE and FIX_DISK options involve handling badspots. The following sections explain what badspots are and how MAKE and FIX_DISK handle them. For specific information on each of MAKE's badspot handling options, see Chapter 5. Similarly, for specific information on each of FIX_DISK's badspot handling options, see Chapter 6.

TYPES OF DISKS

Prime systems support three basic types of physical disks:

- Storage Module Disks (SMDs)
- Fixed-Media Disks (FMDs)
- Cartridge Module Devices (CMDs)

Storage Module Disks (SMDs)

SMDs consist of two units: the **disk drive** and the **disk pack**. The disk pack is a covered stack of disks, or platters, that contain the actual data on their surfaces. This disk pack is mounted in the disk drive so that the data can be accessed and changed (read and written). The disk drive contains all of the circuitry and equipment to manipulate data on the disk pack. Because the disk pack may be removed from the disk drive, vast amounts of data can be stored on many disk packs, which can then be alternated among a few drives. In addition, because of this removal capability, disk-to-disk backups are possible.

Fixed-Media Disks (FMDs)

Fixed-Media Disks (**FMDs**), also called **Winchester disks**, consist of essentially one unit: the disk drive. The platters within these disk drives are not removable. Permanent sealing of the disk platters into the disk drive enables FMDs to store larger amounts of data than SMDs or CMDs can store in the same amount of space. This increased storage density is possible because tolerances can be made much smaller on these sealed drives.

Cartridge Module Devices (CMDs)

CMDs include a removable pack of one usable surface. However, the disk drive for a CMD also includes one or more nonremovable platters. CMDs generally store less data than either SMDs or FMDs, because they have fewer surfaces.

BADSPOTS

Vast quantities of data can be stored on a disk. Portions of a disk, however, are sometimes unable to store data reliably. These portions of a disk are called **badspots** or **flaws**. PRIMOS does not store data onto badspots.

While formatting a disk, MAKE searches it for badspots by writing test data onto every record on the disk, reading it back, and verifying that it is the same data that was written. In addition, if you (or your System Administrator) already know of badspots on the disk and their location, MAKE allows you to enter the locations of the badspots. Some disks also contain a **flaw map** written to a reserved area of the disk; MAKE can read the flaw map and then create a **badspot file** with the information from the flaw map.

MAKE uses the test and verification results, user input, and the disk flaw map, if it exists, to build a list of badspots on the disk. MAKE writes this list on the newly created partition in a file named **BADSPT** (the badspot file). On partitions that have no badspots, MAKE does not write a badspot file.

MAKE also creates a file called the **DSKRAT** (Disk Record Availability Table), which contains a list of all records on the partition. The information in this file indicates

whether a record is in use by the file system. A single bit in this file corresponds to each record in the partition and is set (=1) to indicate that the record is available or reset (=0) to indicate that the record is in use. PRIMOS uses the DSKRAT either to determine the status of a record or to change the status when users create, delete, extend, or shorten files.

When MAKE creates the DSKRAT, most of the records on the disk are free for use. To prevent PRIMOS from using badspots on the disk, MAKE initializes the DSKRAT so that all badspots are marked as being in use. When PRIMOS needs to find an unused record, it skips over badspots.

Dynamic Badspot Handling

Generally, all physical disks have badspots. Historically, **static badspots** (badspots that exist on a disk when it is first partitioned) have been handled in these ways:

- Found by MAKE when you partition a disk
- Entered by you when running MAKE or FIX_DISK
- Read by MAKE from a vendor flaw map

Dynamic badspots (DBS) are those static badspots that are marginally defective and are missed by MAKE or those badspots that are actually developing due to disk media degradation. FIX_DISK handles these badspots only when a partition is shut down for repair, not while the partition is in operation.

It is now possible for a disk controller to handle all badspots and to do so while the disk is in operation. This is referred to as **dynamic badspot handling**. Dynamic badspot handling allows PRIMOS to access an apparently error-free partition. Dynamic badspot handling can be done only by **intelligent disk controllers**, that is, those disk controllers that are microprocessor-based.

Note

Intelligent disk controllers are programmable and are downline loaded at cold start with Intelligent Channel Order Protocol (ICOP) software. This software enables the intelligent disk controller to operate either in Dynamic Badspot Handling (-IC) mode or in Nondynamic Badspot Handling (-AC) mode. In either mode, the controller acts as an intelligent disk controller. If the controller is not downline loaded with ICOP software, it functions as a nonintelligent disk controller.

The intelligent disk controller also handles original, or static, badspots (those supplied by a vendor flaw map on the disk, those supplied by the badspot file BADSPT, or those that have been manually entered with MAKE or FIX_DISK). The controller remaps these badspots using the same technique as for dynamically occurring badspots. The controller gets these original badspots from the dynamic badspot, or DBS, file on the disk. The DBS file contains the addresses of all known badspots on the entire physical disk. It also contains the addresses of the records in the **remapped area**, or **RMA**, that are available for remapping badspots.

The RMA is a reserved area of the physical disk designed to contain file records that are originally addressed to a badspot. The RMA also contains records that are available for new, or dynamically occurring, badspots.

For a further discussion of dynamic badspot handling, see Chapter 8, Dynamic Badspot Handling.

Caution

The need to remap a badspot on a Model 4719 disk associated with a 2450™ system is extremely rare. If, however, the controller must remap a badspot on a Model 4719 disk while the system is in operation, the system may appear to be hung. The reason is that PRIMOS allows the controller as much time to do the remapping operation as the controller may need in a worst-case situation. Allow up to sixty seconds to pass before you attempt to do anything. When the controller has finished remapping a badspot, a message is displayed indicating that a record has been remapped. If no message is displayed and more than a minute has passed, the system is hung for some other reason and you will have to take action. See the section on halts and hangs in your CPU handbook.

Utilities That Use Badspot Information

When a copy of a partition is made for backup purposes, the copy process may or may not utilize the file system management of PRIMOS. A **logical save** (for example, using MAGSAV) utilizes this management; a **physical save** (for example, using PHYSAV) does not. A physical save may be faster than a logical save if the partition is nearly full.

Physical Saves: Unlike a logical save, which copies files and directories individually, a physical save copies the entire partition, record by record, as illustrated in Figure 2-1. It does not involve the DSKRAT on the disk to which the data are being copied (the *target* partition), because all information on that partition is overwritten with the data on the partition from which the data are being copied (the *source* partition). However, the BADSPT file must be preserved on the target partition, because the lists of badspots on the source and target partitions will differ.

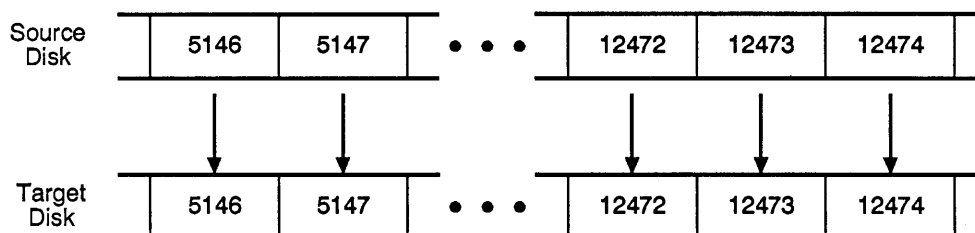


FIGURE 2-1. Ideal Physical Copy

If there are badspots on the target partition, the program performing the copy (COPY_DISK or PHYRST) cannot copy records from the source partition to the corresponding records on the target partition when those records correspond to badspots on the target partition. Therefore, another location for these records must be found on the target partition.

To find this location, when COPY_DISK and PHYRST need to copy a record whose corresponding record on the target partition is a badspot, they temporarily adopt the characteristics of a logical save operation. They use the DSKRAT of the source partition to find a record that is not being used on the source partition, and that therefore does not need to be copied. Then they use the corresponding record on the target partition as a home for the record that they could not copy before, as illustrated in Figure 2-2. If these utilities cannot find a free record, they display an error message.

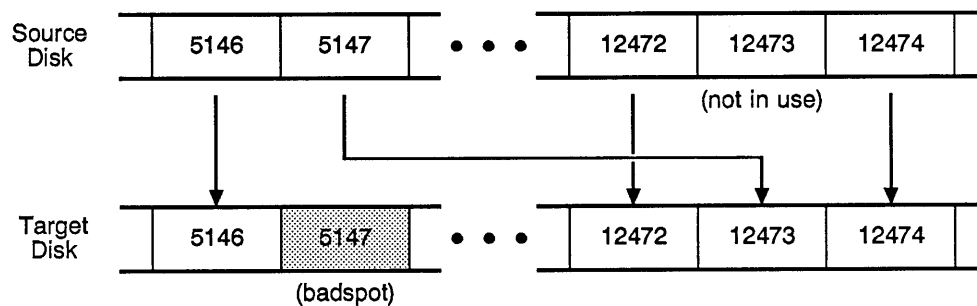


FIGURE 2-2. Physical Copy With Badspot Handling

Note that once a record has been redirected in this manner, the target partition must not be used under PRIMOS. As shown in Figure 2-2, this limitation exists because there are other records on the partition that point to, for example, record number 5147, when in fact the data stored in record number 5147 on the source partition was copied to record number 12473 on the target partition. Before the target partition can be used, all references to remapped records (such as record number 5147) must be changed to point to the actual records (such as record number 12473). FIX_DISK performs the task of changing these pointers as illustrated in Figure 2-3.

For FIX_DISK to know where remapped records (such as record number 5147) were copied to (such as record number 12473), COPY_DISK and PHYRST create **equivalence blocks**. These equivalence blocks tell FIX_DISK which records were copied and where the records were copied.

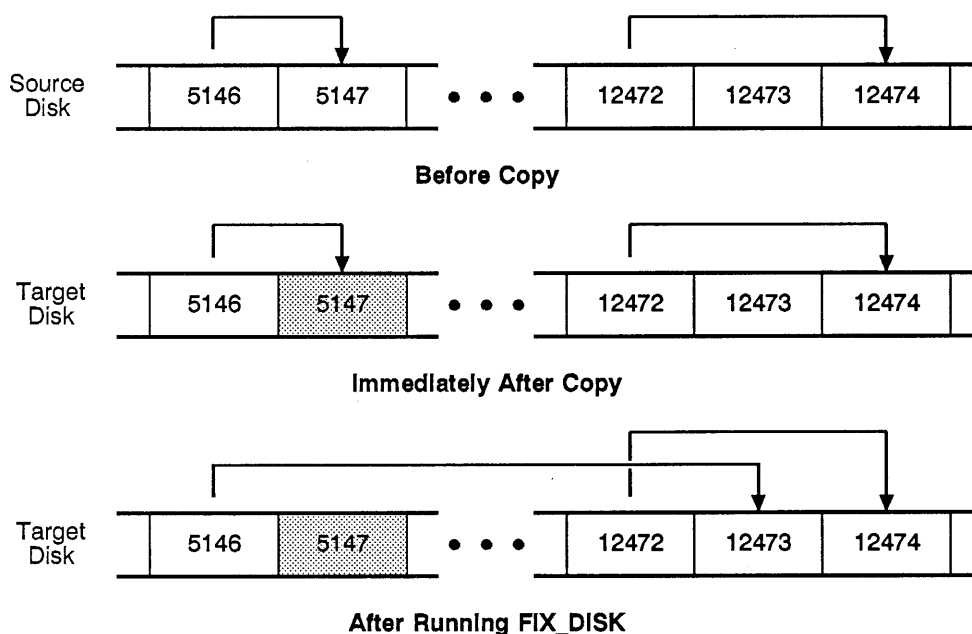


FIGURE 2-3. Badspot Handling and File Pointers

Equivalence Blocks

An **equivalence block** describes where one record is actually stored on the partition. There is one equivalence block for every record remapped by COPY_DISK or PHYRST 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.

Need for Creating Equivalence Blocks: When COPY_DISK and PHYRST are copying to a disk, they copy record number 1 from the source partition to record number 1 on the target partition, record number 2 to record number 2, and so on. (Even though PHYRST is restoring data from tape, the tape contains a record-by-record image of the source partition; therefore this explanation still applies.) If a record is a badspot on the *source* partition, the record is not copied. However, a record that is to be copied to a badspot on the *target* partition presents two basic problems:

- The data in the record must be saved. The record is not a badspot on the source partition; therefore, it contains data if it is in use (as indicated by the source partition DSKRAT). However, it cannot be written to the same place on the target partition, because that spot is bad and will not hold data.
- If the record is in use, other records on the source partition (and their copies on the target partition) contain pointers to the record. These pointers are record numbers.

To solve the first problem, COPY_DISK and PHYRST try to find a different place in which to store the record on the target partition — an alternate record. This alternate record must satisfy the following requirements:

- It must not be a badspot on the target partition (although it can correspond to a badspot on the source partition).
- It must not correspond to a record that is in use by the source partition (as indicated in the DSKRAT of the source partition).
- It cannot already have been used as an alternate record for some other badspot on the target partition.

If an alternate record that satisfies these requirements is not found, COPY_DISK or PHYRST issues an error message. If an alternate record is found, the data in the original record is copied there (remapped).

Thus, when COPY_DISK and PHYRST perform this operation, they create an equivalence block, indicating that the data intended for the original record has been stored in the alternate record instead. This brings us to the second problem: although the data are stored in the alternate record, all of the other records on the target partition identify the alternate record by its original record number (a badspot). There are no records that point to the alternate record number. FIX_DISK solves the problem by changing all file system pointers to point to this new record.

Results of the Creation of Equivalence Blocks: When COPY_DISK or PHYRST has finished copying to a partition on which it created one or more equivalence blocks, it displays the message

Badspots handled on partition *pdev*, please run FIX_DISK

This message tells you that before you try to use the ADDISK or PHYSAV commands on *pdev*, and before you try to copy *pdev* to another disk by using COPY_DISK, you must run FIX_DISK on *pdev*, using the -FIX option.

When you run FIX_DISK with the -FIX option on a disk that contains equivalence blocks in the BADSPT file, FIX_DISK updates all records on the disk that point to the original (bad) records so that they point to the corresponding alternate (good) records instead. FIX_DISK then deletes the equivalence blocks. When FIX_DISK is finished, the disk can be used.

PART II
PARTITION MAINTENANCE

PHYSICAL DEVICE NUMBERS

Each disk or disk partition has a **physical device number (pdev)** that identifies the type of storage device, the drive unit on which it is mounted, the controller to which the drive unit is attached, the size of the partition, and the location of the partition on the disk. These physical device numbers are used in the following commands:

ADDISK	FIX_DISK	SHUTDN
ASSIGN DISK	MAKE	UNASSIGN DISK
COPY_DISK	PHYRST	
DISKS	PHYSAV	

Physical device numbers are also used in configuration directives such as COMDEV, PAGING, and others.

A physical device number is an octal number representing the setting of bits in a 16-bit halfword. The halfword is the address of a disk partition and the address allows PRIMOS to access the partition. The physical device number makes each disk partition physically and logically unique.

This chapter describes how to determine a physical device number for a given disk.

TYPES OF DISKS

As explained in Chapter 2, PRIMOS supports three varieties of disks: Storage Module Disks (SMDs), Fixed-Media Disks (FMDs), and Cartridge Module Devices (CMDs). PRIMOS can use these disks for file storage and for paging area. Each variety is available in two or more sizes.

SIZE OF A DISK PARTITION

The size of a disk partition is measured in records. The number of records on a disk (and the amount of usable storage per surface) depends on the type of disk. Table 3-1 summarizes these values for all disks that are fully supported at Rev. 22.1.

TABLE 3-1. Disk Size Data

Disk Type	Model Number	Number of Surfaces	Records per Surface	Total Number of Records	Usable MB	Removable
80MB SMD		5	7407	37035	75.85	yes
300MB SMD		19	7407	140733	288.22	yes
60MB FMD	4711	4	7140	28560	58.49	no
68MB FMD		3	10071	30213	61.88	no
84MB FMD	4714	5	8120	40600	83.15	no
120MB FMD	4715	8	7140	57120	116.98	no
158MB FMD		7	10071	70497	144.38	no
160MB FMD		10	7389	73890	151.33	no
258MB FMD	4719	17	7320	124440	254.85	no
315MB FMD*	4475	19	7407	140733	288.22	no
328MB FMD	4721	12	13128	157536	322.63	no
496MB FMD	4735	24	9954	238896	489.26	no
675MB FMD**		40	7569	302760	620.05	no
770MB FMD	4845	23	16112	370576	758.94	no
817MB FMD	4860	15	26201	393015	804.89	no
32MB CMD		2	7407	14814	30.34	1 surface
64MB CMD		4	7407	29628	60.68	1 surface
96MB CMD		6	7407	44442	91.02	1 surface

Notes to Table 3-1

* Sometimes referred to as a 300MB FMD.

** Sometimes referred to as a 600MB disk or as a 630MB disk.

The official name is 675MB disk.

For all of the disks in Table 3-1, a record holds 2048 bytes of user data plus 32 bytes for housekeeping information. The amount of usable data in megabytes (*Usable MB*) is based on 2048 bytes per record and is always less than the rated size of the disk. Approximately 5% to 8% of a disk's capacity is used for housekeeping information.

The *Removable* column in Table 3-1 specifies whether the disk drive supports removable disk packs. SMDs allow the entire disk pack to be removed. FMDs allow no such removal. CMDs allow the removal of one platter, which has one surface of usable information. The remaining surfaces (1, 3, or 5 surfaces) are not removable.

The size of a disk partition is specified as an integral number of surfaces. A disk partition may contain 1 through 31 surfaces. Because a partition cannot contain more than 31 surfaces, the 40 surfaces available on a 675MB disk cannot be contained by one partition. You must define at least two partitions in order to utilize the disk fully.

The surface offset, or the starting surface number, for a partition is always an *even* number, ranging from 0 through 30, inclusive.

DISK DRIVE UNIT NUMBERS

Individual disk drive units are identified by drive unit numbers. A drive unit number is selected on the disk drive unit. During disk-related activities, the Operator supplies this number to PRIMOS as part of the physical device number. PRIMOS uses this drive unit number to identify the particular drive unit being referenced.

Disk drives are connected to disk controllers. The **disk controller** is a circuit board containing hardware and microcode that allow communication between the CPU and the drive unit, or disk storage device. A maximum of four or eight disk drives may be connected to each disk controller, depending on the particular controller, and one system may have a maximum of eight disk controllers. It is therefore possible for a system to have a total of sixty-four disk drives.

No two disk drive units connected to the same disk controller may have the same drive unit number. In addition, the disk controller has a specific address that becomes part of the identification PRIMOS uses to reference the disk media.

It is important to keep a record in the system logbook of disk controller addresses and numbers, drive unit numbers, and the physical device numbers for disk partitions mounted on these drives so that you have a ready reference to a permanent record of them.

Disk Drive Unit Numbers for SMDs and CMDs

Drive unit numbers for SMDs and CMDs are indicated and set by the removable buttons, or logical address plugs, on the front of the disk drive. These buttons, which have the drive unit number printed on them, are numbered 0 through 7. These removable buttons can be interchanged among drives, thus changing the logical addresses of the drives.

Disk Drive Unit Numbers for FMDs

A drive unit number for a FMD can range from 0 through 7, inclusive. The drive unit numbers for the Model 4475 FMD and the Model 4735 FMD are set by a removable button on the front of the drive. The Model 4845 FMD has a slotted-screw rotary switch behind the front removable access panel.

The Model 4860 has a button marked ADDRESS on the drive control panel. You hold this button in to cycle through the hexadecimal numbers displayed in the four LEDs marked 8 4 2 1 above the button. Since drive unit numbers can range only from 0 to 7, only numbers indicated by the LEDs marked 4 2 1 are valid. The drive unit numbers are indicated by addition of the numbers above lit LEDs; thus if LEDs 1 and 4 are lit, this indicates drive unit 5; if all three are lit, this indicates drive unit 7; and if none of the LEDs are lit, this indicates drive unit 0.

Drive unit numbers for other FMDs cannot be set by the Operator. They are set internally by a representative from your Customer Support Center.

CONSTRUCTION OF PHYSICAL DEVICE NUMBERS

The remainder of this chapter discusses the construction of physical device numbers (pdevs). Tables 3-3 and 3-6 present pdevs for the disks supported by PRIMOS, and the text presents examples of their construction and use. This information is presented in the following tables and figures:

Table 3-2	Additions to pdev for Disk Drive Unit Number and Disk Controller Address
Table 3-3	Basic Physical Device Numbers for Storage Modules and Fixed-Media Devices
Table 3-4	Example Combinations of Controller Addresses and Disk Drive Unit Numbers for Basic pdev 103020
Table 3-5	Example of Basic and Adjusted pdevs
Table 3-6	Basic Physical Device Numbers for Cartridge Module Devices
Figure 3-1	pdev Worksheet Example
Figure 3-2	Example of Storage Module Partitions
Figure 3-3	Construction of a Physical Device Number

Information Needed for Physical Device Numbers

Generally, the System Administrator supplies you with the following information necessary to construct a physical device number. Write this information in the system logbook.

- Size or model and type of disk (SMD and FMD, or CMD)
- Starting surface number
- Number of surfaces in the partition
- Drive unit number
- Controller address or number

From this information, you can construct a physical device number. The method used to construct a physical device number depends on the type of disk. Physical device numbers are specified in octal.

PHYSICAL DEVICE NUMBERS FOR SMDS AND FMDS

Table 3-3 summarizes all of the legitimate physical device numbers for SMDs and FMDS. It does not list numbers that, although valid, will result in an unused surface. These numbers result from using an odd number of surfaces in a partition that is not the last partition of a physical disk, or a disk pack. If there is a partition with an odd number of surfaces in the middle of a physical disk, the next surface is not used because the next partition must have an even numbered starting surface.

Note

The numbers in Table 3-3 appear different than versions of this table prior to Rev. 21.0 because the basic pdev is now computed for controller 0 at address 24₈, whereas versions prior to Rev. 21.0 used controller 1 at address 26₈ to compute the basic pdev. Any basic pdevs you may already be using based on controller 1 at address 26₈ are still valid. You do not have to change basic pdevs.

Table 3-3 shows all the valid physical device numbers for the following disks.

60MB (Model 4711)	158MB	328MB (Model 4721)
68MB	160MB	496MB (Model 4735)
80MB SMD	258MB (Model 4719)	675MB
84MB (Model 4714)	300MB SMD	770MB (Model 4845)
120MB (Model 4715)	315MB (Model 4475)	817MB (Model 4860)

Numbers in Table 3-3 ending with 1 result in specification of an odd number of surfaces in a partition. These numbers should be used only as the *last* partition on the following disks:

- 80MB and 300MB SMDs
- Models 4475, 4714, 4719, 4845, and 4860 FMDs
- 68MB and 158MB FMDs

Using these numbers for partitions on other disks reduces the storage capacity of the disk by 15MB to 33MB per unused surface. Note also that no partitions can have a starting surface number greater than 30, and no partition can be composed of more than 31 surfaces.

To obtain the values you need to construct a pdev for SMDs and FMDs, use the following steps. Also consult Table 3-2 and Figure 3-1.

The steps for constructing a pdev are

1. Determine the number of surfaces in the partition.
2. Determine the surface number of the first surface in the partition — the starting surface number.
3. Look up the basic physical device number in Table 3-3.
4. For the disk drive unit, add the value shown in Table 3-2.
5. For the disk controller address, add the value shown in Table 3-2.

TABLE 3-2. Additions to pdev for Disk Drive Unit Number and Disk Controller Address

<i>Number to Add for Disk Drive Unit Number</i>		<i>Number to Add for Disk Controller Address</i>		
<i>Disk Drive Unit</i>		<i>Disk Controller</i>		
<i>Number</i>	<i>Add</i>	<i>Address</i>	<i>Number</i>	<i>Add</i>
0	0	24 ₈	0	0
1	2	26 ₈	1	40
2	4	25 ₈	2	100
3	6	22 ₈	3	140
4	10	45 ₈	4	200
5	12	27 ₈	5	240
6	14	46 ₈	6	300
7	16	23 ₈	7	340

***Number
of
Surfaces
in
Partition***

All partitions must begin on an even starting surface number. Use an odd number of surfaces only when the partition is the last partition on a physical disk having an odd number of surfaces; otherwise disk space will be wasted.

All partitions must begin on an even starting surface number. Use an odd number of surfaces only when the partition is the last partition on a physical disk having an odd number of surfaces; otherwise disk space will be wasted.

Example of Determining a pdev

Suppose that you are creating a partition with these characteristics:

Starting surface number	16
Number of surfaces	12

From Table 3-3, the basic pdev is 103020. By adding the appropriate values from Table 3-2, you obtain all the possible pdevs for this partition and for all the controller and disk drive unit number combinations, as shown in Table 3-4. For example, the pdev for this partition on disk drive unit number 2 associated with disk controller at address 27_8 (controller 5) is 103264.

*TABLE 3-4. Example Combinations of Controller Addresses
and Disk Drive Unit Numbers for Basic pdev 103020*

		Disk Drive Unit Number							
		0	1	2	3	4	5	6	7
<i>C</i>									
<i>o A</i>	24_8	103020	103022	103024	103026	103030	103032	103024	103036
<i>n d</i>	26_8	103060	103062	103064	103066	103070	103072	103074	103076
<i>t d</i>	25_8	103120	103122	103124	103126	103130	103132	103134	103136
<i>r r</i>	22_8	103160	103162	103164	103166	103170	103172	103174	103176
<i>o e</i>	45_8	103220	103222	103224	103226	103230	103232	103234	103236
<i>l s</i>	27_8	103260	103262	103264	103266	103270	103272	103274	103276
<i>l s</i>	46_8	103320	103322	103324	103326	103330	103332	103334	103336
<i>e</i>	23_8	103360	103362	103364	103366	103370	103372	103374	103376
<i>r</i>									

Figure 3-1 is a worksheet that you can fill out to arrive at a pdev. It includes an example.

	<i>Example</i>
1. Starting surface number:	16
2. Number of surfaces in partition:	12
3. Basic pdev from Table 3-3:	103020
4. Controller address: <u>26</u>	
From Table 3-2, add:	40
5. Drive Unit number: <u>2</u>	
From Table 3-2, add:	4
6. pdev to use:	<hr/> 103064

FIGURE 3-1. pdev Worksheet Example

EXAMPLE OF PARTITIONING SMDs

Figure 3-2 shows an example of using Table 3-3. The system in this example contains three drive units connected to the disk controller with address 26₈ (controller 1); drives 0 and 1 have 300MB SMDs, and drive 2 has an 80MB SMD.

As shown in Figure 3-2, the physical disks are to be partitioned as follows:

Drive 0	Partitions of 2, 2, 6, 2, 2, 2, and 3 surfaces Starting surface numbers — 0, 2, 4, 10, 12, 14, and 16
Drive 1	Partitions of 14 and 5 surfaces Starting surface numbers — 0 and 14
Drive 2	Partitions of 2 and 3 surfaces Starting surface numbers — 0 and 2

Since each of these disk packs has an odd number of surfaces, the last partition in each disk pack has a 1 in the basic pdev. Since there are three disk drives involved, a number must be added to each of the basic pdevs to designate the disk drive units. From Table 3-2, these numbers are 0 for drive unit 0, 2 for drive unit 1, and 4 for drive unit 2. Also, from Table 3-2, 40₈ must be added for disk controller 1 at address 26₈.

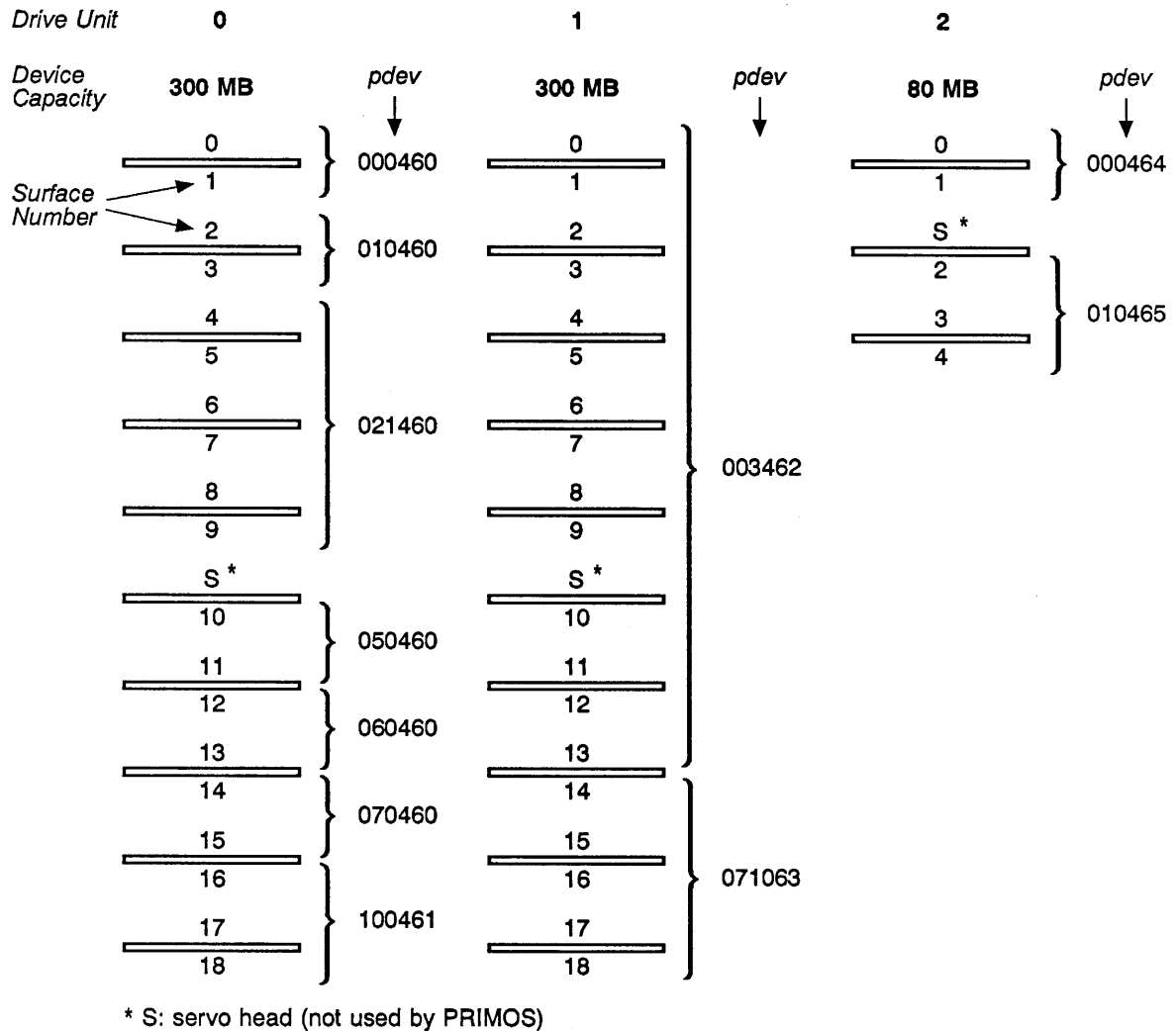


FIGURE 3-2. Example of Storage Module Partitions

The basic physical device numbers for this example, as taken from Table 3-3, are shown on the left side of Table 3-5. The appropriate numbers to designate the disk controller and the disk drives are added and the adjusted pdevs, as they are to be used, are shown on the right side of the table.

TABLE 3-5. Example of Basic and Adjusted pdevs

Basic pdev			Adjusted pdev		
Drive 0	Drive 1	Drive 2	Drive 0	Drive 1	Drive 2
000420	003420	000420	000460	003462	000464
010420	071021	010421	010460	071063	010465
021420			021460		
050420			050460		
060420			060460		
070420			070460		
100421			100461		

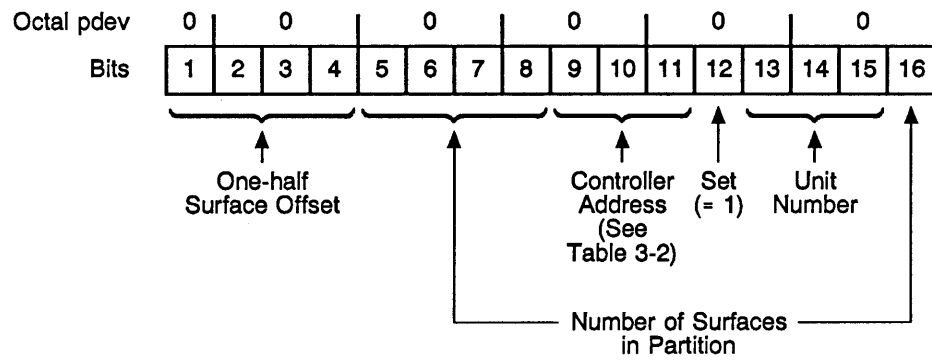
Binary Breakdown of Physical Device Numbers

Figure 3-3 shows the binary breakdown of a physical device number. Physical device numbers (pdevs) tell the system the type of storage device being used, the drive unit on which the device is mounted, the controller to which the drive unit is connected, and the size of the partition and its location on the disk pack. The meaning of the bits in the halfword making up the physical device number are shown in the figure and discussed here:

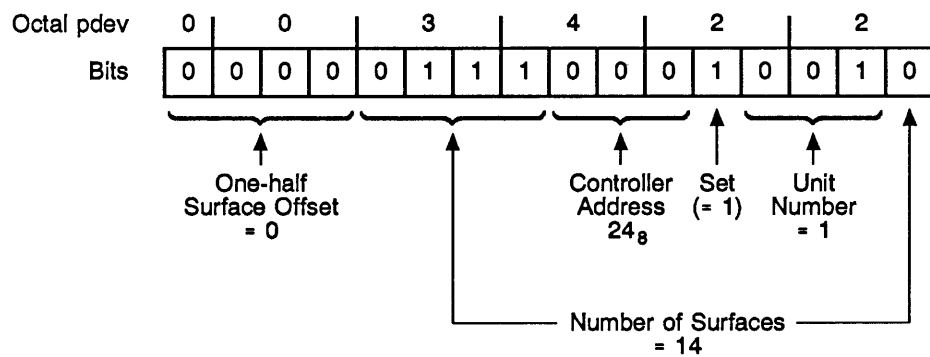
- Bits 1 through 4 represent the starting surface number divided by two; thus the highest starting surface number can be no more than 30 since the highest number represented by four bits is 15.
- Bits 5 through 8 plus bit 16 (the odd surface bit) represent the value for the number of surfaces in a partition; thus no more than 31 surfaces can be in a partition since the highest number represented by five bits is 31.
- Bits 9, 10, and 11 represent the controller number associated with the address of the disk controller; thus there can be eight disk controllers (0 - 7) represented by three bits.
- Bit 12 is reserved and is set (=1).
- Bits 13, 14, and 15 represent the disk drive unit number; thus there can be eight disk drive units (0 - 7) represented by three bits.

Figure 3-3 also shows the meanings of two of the pdevs from the previous example. These are the two pdevs for the partitions on Drive 1. For pdev 003462, the octal-to-binary conversion indicates a surface offset (starting surface number) of 0, 14 surfaces in the partition, controller 1 at address of 26₈, and a drive unit number of 1.

For pdev 071063, the conversion indicates the same drive unit number and controller address as above. These portions of the pdevs for the two partitions are the same because the partitions are part of the same physical disk. The remaining bits indicate that there are five surfaces in this partition with a starting surface number of 14.



Drive 1, 14 Surfaces, Starting Surface = 0, pdev = 003422



Drive 1, 5 Surfaces, Starting Surface = 14, pdev = 071023

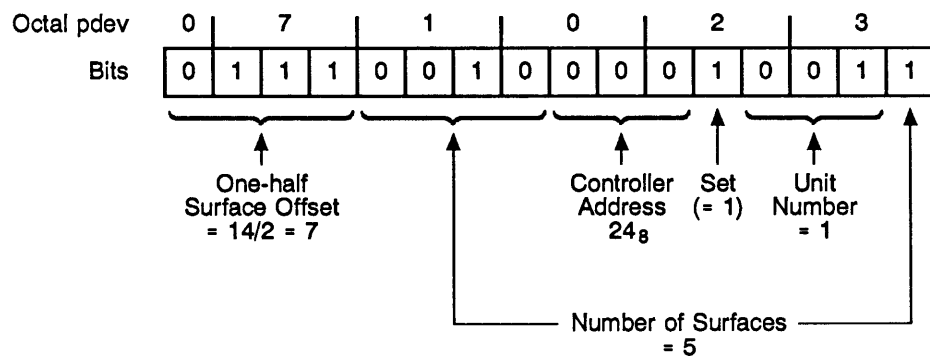


FIGURE 3-3. Construction of a Physical Device Number

CARTRIDGE MODULE DEVICES (CMDs)

Physical device numbers for CMDs are similar to those for SMDs and FMDs. However, a peculiarity exists in the assignment of surface offsets: the removable portion of the disk is surface offset 0, but the nonremovable portion starts at surface offset 16; thus pdevs for the nonremovable portions of CMDs always start with a 1.

This peculiarity renders the specification of surfaces 1 through 15 ineffective because no actual surfaces correspond to the specification. Therefore, a CMD must contain at least two partitions. The removable portion of the disk is always one partition, and the nonremovable portion is one or more partitions.

Valid pdevs for CMDs are summarized at the end of the next section, Partitioning Specific Disk Types, and are shown in Table 3-6 at the end of this chapter.

PARTITIONING SPECIFIC DISK TYPES

This section describes each disk type and size that may be partitioned and tells how it may be partitioned. All sizes of SMDs, CMDs, and FMDs may be partitioned. Partitioning a disk allows you to use it as several logical disks, each with its own name, rather than as one physical disk. Table 3-1, earlier in this chapter, summarizes the disk characteristics.

Various methods are available to partition the different types and sizes of disks. The following sections discuss the methods for each type of disk.

80MB Storage Module Disk

The 80MB SMD has five surfaces. Table 3-3 presents the legitimate basic physical device numbers for 80MB SMDs. The only valid entries are those entries for which adding the number of surfaces to the starting surface number produces a total of 5 or less. Because the disk has an odd number of surfaces, the last partition must have an odd number of surfaces, or else some disk space is not utilized. Therefore, use one of the valid entries for 80MB disks ending in 1 in Table 3-3 as the last partition, to fully utilize the disk. However, use only the entries ending in 1 for which adding the number of surfaces to the starting surface number results in 5. Do not use those entries resulting in 3, because they are for 68MB FMDs only and will result in unused space on 80MB SMDs. An 80MB disk can be set up as a single partition.

300MB Storage Module Disk

The 300MB SMD has 19 surfaces. Table 3-3 presents the legitimate basic physical device numbers for 300MB SMDs. The only valid entries are those entries for which adding the number of surfaces to the starting surface number produces a total of 19 or less. Because the disk has an odd number of surfaces, the last partition must have an odd number of surfaces, or else some disk space is not utilized. Therefore, use one of the valid entries for 300MB SMDs ending in 1 in Table 3-3 as the last partition, to fully utilize the disk. However, use only the entries ending in 1 for which adding the number of surfaces to the starting surface number results in 19. Do not use 1 entries resulting in 3, 5, 7, 15, and 17 because they are for 68MB FMDs, 80MB SMDs, 84MB FMDs (Model 4714), 158MB FMDs, 817MB FMDs (Model 4860), and 258MB FMDs (Model 4719) only and will result in unused space on 300MB SMDs. A 300MB disk can be set up as a single partition.

60MB Fixed-Media Disk (Model 4711)

The 60MB FMD has four surfaces. Table 3-3 presents the legitimate basic physical device numbers for 60MB FMDs. The only valid entries are those entries for which adding the number of surfaces to the starting surface number produces a total of 4 or less. Because the disk has an even number of surfaces, no partition should have an odd number of surfaces, or else some disk space is not utilized. Therefore, do not use any of the entries ending in 1 in Table 3-3. A 60MB disk can be set up as a single partition.

68MB Fixed-Media Disk

The 68MB FMD has three surfaces. Table 3-3 presents the legitimate basic physical device numbers for this disk. The only valid entries are those entries for which adding the number of surfaces to the starting surface number produces a total of 3 or less. Because the disk has an odd number of surfaces, the last partition must have an odd number of surfaces, or else some disk space is not utilized. Therefore, Table 3-3 shows only two valid entries for this disk ending in 1 (000421 and 010021), and you must use one of these entries as the last partition, to fully utilize the disk. A 68MB disk can be set up as a single partition.

84MB Fixed-Media Disk (Model 4714)

The 84MB FMD has five surfaces. Table 3-3 presents the legitimate basic physical device numbers for 84MB FMDs. The only valid entries are those entries for which adding the number of surfaces to the starting surface number produces a total of 5 or less. Because the disk has an odd number of surfaces, the last partition must have an odd number of surfaces, or else some disk space is not utilized. Therefore, use one of the valid entries for 84MB FMDs ending in 1 in Table 3-3 as the last partition, to fully utilize the disk. However, use only the entries ending in 1 for which adding the number of surfaces and the starting surface number results in 5. Do not use those resulting in 3, because they are for 68MB FMDs only.

120MB Fixed-Media Disk (Model 4715)

The 120MB FMD has eight surfaces. Table 3-3 presents the legitimate basic physical device numbers for 120MB FMDs. The only valid entries are those entries for which adding the number of surfaces to the starting surface number produces a total of 8 or less. Because the disk has an even number of surfaces, no partition should have an odd number of surfaces, or else some disk space is not utilized. Therefore, do not use any of the entries ending in 1 in Table 3-3. A 120MB disk can be set up as a single partition.

158MB Fixed-Media Disk

The 158MB FMD has seven surfaces. Table 3-3 presents the legitimate basic physical device numbers for 158MB FMDs. The only valid entries are those entries for which adding the number of surfaces to the starting surface number produces a total of 7 or less. Because the disk has an odd number of surfaces, the last partition must have an odd number of surfaces, or else some disk space is not utilized. Therefore, use one of the valid entries for 158MB FMDs ending in 1 in Table 3-3 as the last partition, in order to fully utilize the disk. However, use only the entries ending in 1 for which adding the number of surfaces and the starting surface number results in 7. Do not use those resulting in 3 or 5, because they are for 68MB FMDs and 80MB SMDs only.

160MB Fixed-Media Disk

The 160MB FMD has 10 surfaces. Table 3-3 presents the legitimate basic physical device numbers for 160MB FMDs. The only valid entries are those entries for which adding the number of surfaces to the starting surface number produces a total of 10 or less. Because the disk has an even number of surfaces, no partition should have an odd number of surfaces, or else some disk space is not utilized. Therefore, do not use any of the entries ending in 1 in Table 3-3. A 160MB disk can be set up as a single partition.

258MB Fixed-Media Disk (Model 4719)

The 258MB FMD has 17 surfaces. Table 3-3 presents the legitimate basic physical device numbers for Model 4719 (258MB) FMDs. The only valid entries are those entries for which adding the number of surfaces to the starting surface number produces a total of 17 or less. Because the disk has an odd number of surfaces, the last partition must have an odd number of surfaces, or else some disk space is not utilized. Therefore, use one of the valid entries for Model 4719 FMDs ending in 1 in Table 3-3 as the last partition, to fully utilize the disk. However, use only the entries ending in 1 for which adding the number of surfaces and the starting surface number results in 17. Do not use those resulting in 3, 5, 7, and 15 because they are for 68MB FMDs, 80MB SMDs, 84MB FMDs (Model 4714), 158MB FMDs, and 817MB FMDs (Model 4860) only. A Model 4719 disk can be set up as a single partition.

315MB Fixed-Media Disk (Model 4475)

The 315MB FMD has 19 surfaces. Table 3-3 presents the legitimate basic physical device numbers for Model 4475 (315MB) FMDs. The only valid entries are those entries for which adding the number of surfaces to the starting surface number produces a total of 19 or less. Because the disk has an odd number of surfaces, the last partition must have an odd number of surfaces, or else some disk space is not utilized. Therefore, use one of the valid entries for Model 4475 FMDs ending in 1 in Table 3-3 as the last partition, to fully utilize the disk. However, use only the entries ending in 1 for which adding the number of surfaces and the starting surface number results in 19. Do not use those resulting in 3, 5, 7, 15, and 17 because they are for 68MB FMDs, 80MB SMDs, 84MB FMDs (Model 4714), 158MB, 817MB FMDs (Model 4860), and 258MB (Model 4719) FMDs only. A Model 4475 disk can be set up as a single partition.

328MB Fixed-Media Disk (Model 4721)

The 328MB FMD has 12 surfaces. Table 3-3 presents the legitimate basic physical device numbers for Model 4721 (328MB) FMDs. The only valid entries are those entries for which adding the number of surfaces to the starting surface number produces a total of 12 or less. Because the disk has an even number of surfaces, no partition should have an odd number of surfaces, or else some disk space is not utilized. Therefore, do not use any of the entries ending in 1 in Table 3-3. A 328MB disk can be set up as a single partition.

496MB Fixed-Media Disk (Model 4735)

The 496MB FMD has 24 surfaces. Table 3-3 presents the legitimate basic physical device numbers for Model 4735 (496MB) FMDs. The only valid entries are those entries for which adding the number of surfaces to the starting surface number produces a total of 24 or less. Because the disk has an even number of surfaces, no partition should have an odd number of surfaces, or else some disk space is not utilized. Therefore, do not use any of the entries ending in 1 in Table 3-3. A 496MB disk can be set up as a single partition.

675MB Fixed-Media Disk

The 675MB FMD has 40 surfaces. Table 3-3 presents the legitimate basic physical device numbers for 675MB disks. Because the disk has an even number of surfaces, no partition should have an odd number of surfaces, or else some disk space is not utilized. Therefore, do not use any of the entries ending in 1 in Table 3-3. A 675MB disk cannot be set up as a single partition, because a partition cannot contain more than 31 surfaces. Therefore, you must create at least two partitions in order to fully utilize a 675MB disk. In addition, the last partition on a 675MB disk must use at least ten surfaces; thus it is desirable to use the last partition for a file system that is very large.

770MB Fixed-Media Disk (Model 4845)

The 770MB FMD has 23 surfaces. Table 3-3 presents the legitimate basic physical device numbers for Model 4845 (770MB) FMDs. The only valid entries are those entries for which adding the number of surfaces to the starting surface number produces a total of 23 or less. Because the disk has an odd number of surfaces, the last partition must have an odd number of surfaces, or else some disk space is not utilized. Therefore, use one of the valid entries for Model 4845 FMDs ending in 1 in Table 3-3 as the last partition, to fully utilize the disk. However, use only the entries ending in 1 for which adding the number of surfaces and the starting surface number results in 23. Do not use those resulting in 3, 5, 7, 15, 17, and 19 because they are for 68MB FMDs, 80MB SMDs, 84MB FMDs (Model 4714), 158MB FMDs, 817MB FMDs (Model 4860), 258MB FMDs (Model 4719), 315MB FMDs (Model 4475), and 300MB SMDs only. A Model 4845 disk can be set up as a single partition.

817MB Fixed-Media Disk (Model 4860)

The 817MB FMD has 15 surfaces. Table 3-3 presents the legitimate basic physical device numbers for Model 4860 (817MB) FMDs. The only valid entries are those entries for which adding the number of surfaces to the starting surface number produces a total of 15 or less. Because the disk has an odd number of surfaces, the last partition must have an odd number of surfaces, or else some disk space is not utilized. Therefore, use one of the valid entries for Model 4860 FMDs ending in 1 in Table 3-3 as the last partition, to fully utilize the disk. However, use only the entries ending in 1 for which adding the number of surfaces and the starting surface number results in 15. Do not use those resulting in 3, 5, and 7 because they are for 68MB FMDs, 80MB SMDs, 84MB FMDs, and 158MB FMDs only. A Model 4860 disk can be set up as a single partition with 15 surfaces.

Cartridge Module Devices (CMDs)

CMDs exist in three sizes: 32, 64, and 96 megabytes. They may be partitioned as indicated in Table 3-6 following.

Partitioning of CMDs is done in a manner similar to that of SMDs and FMDs. However, the removable portion of a CMD is always a separate partition. The nonremovable portion is partitioned separately from the removable portion. The nonremovable surfaces are treated as if they start at a surface offset, or starting surface number, of 16.

TABLE 3-6. Basic Physical Device Numbers for Cartridge Module Devices

<i>CMD Type Surfaces</i>		<i>Platter(s)</i>	<i>First Controller</i>
32MB	1	Removable	21 (16MB)
	1	Nonremovable	100021 (16MB)
64MB	1	Removable	21 (16MB)
	3	Nonremovable	100420 (32MB)
			110021 (16MB)
			or 100421 (48MB)
96MB	1	Removable	21 (16MB)
	5	Nonremovable	100420 (32MB)
			110420 (32MB)
			120021 (16MB)
			or 101020 (64MB)
			120021 (16MB)
			or 101021 (80MB)
			or 100420 (32MB)
			110421 (48MB)

Note

See Table 3-2 for numbers to add to the basic pdev if the disk is mounted in a disk drive unit other than unit 0, that is, drive units 1 through 3.

See Table 3-2 for numbers to add to the basic pdev if the disk drive is connected to a disk controller other than controller 0 at address 24₈.

The removable surface of all CMDs is organized as one partition.

The nonremovable surface of the 32MB CMD is also organized as one partition.

The nonremovable surfaces of the 64MB CMD can be organized as one or two partitions.

The nonremovable surfaces of the 96MB CMD can be organized as one, two, or three partitions.

USING AND ASSIGNING DISKS

This chapter explains the procedure by which you

- Make disks available for general use
- Assign disks to yourself (or to other users), so that you can perform maintenance tasks on them

FILE SYSTEM DISKS AND ASSIGNABLE DISKS

Think of disks (that is, partitions, or logical disks) as being in one of the two categories shown in Figure 4-1: File System Disks and Assignable Disks. **File system disks** are disks that are available to system users for shared access. They are turned on and made available for use when you give the ADDISK command followed by their physical device numbers (pdev). While the disks are on (available), users may perform all normal operations on directories and files, such as creating and listing directories and editing, copying, deleting, and renaming files.

When it becomes necessary for you to perform some operation on a logical disk, such as copying it or repairing it, use the SHUTDN command to shut the disk down. This action makes the disk unavailable to users.

So that you can assign the disk to yourself for exclusive, direct access, you must place the disk in the Assignable Disks Table. **Assignable disks** are partitions that a single user can assign for exclusive access.

Note

Assignable disks can be either formatted or unformatted. File system disks must be formatted so that they contain at least the basic elements of the PRIMOS file system.

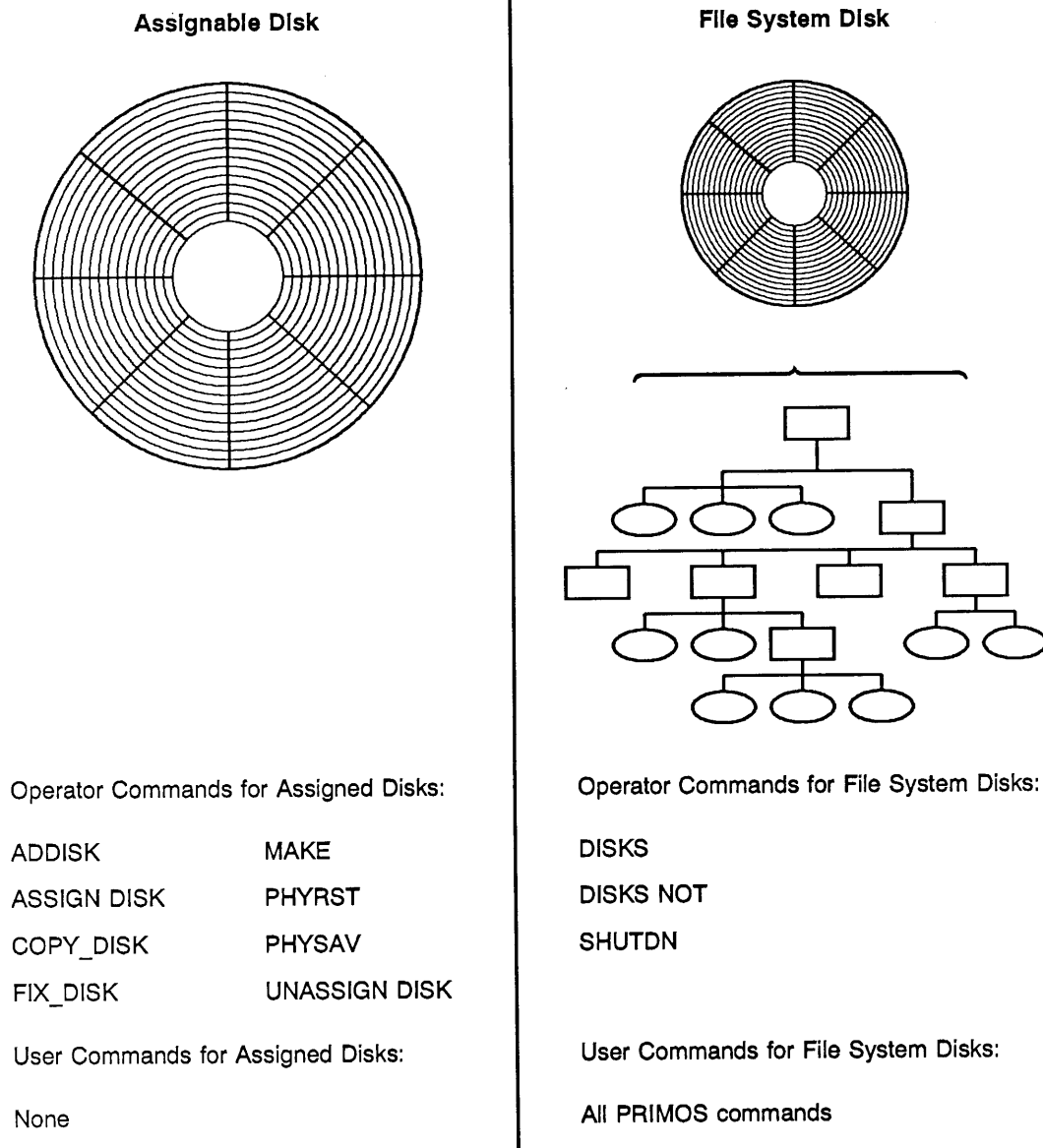


FIGURE 4-1. Assignable Disks and File System Disks

Physical Device Number

The common link between file system disks and assignable disks is the **physical device number (pdev)**. The pdev is the partition's address; that is, the pdev tells PRIMOS what disk drive and disk controller are associated with the partition and where on the physical disk the partition is located. If there are any changes to the partition's address, its pdev must change. For example, if a partition has been associated with a particular disk drive, and the disk pack containing that partition is then placed in a different drive or the original drive's designation is changed, the partition's address changes; therefore, its pdev must be changed.

To make the disk available or unavailable, to repair it, or to otherwise operate on it, you must use the pdev to specify which logical disk you mean. Although users identify disks by their partition names, you often perform operations before the relationship between a partition and its disk drive has been defined to the system. To identify the partition, you must use the pdev. See the descriptions and examples in Chapter 3 to determine physical device numbers.

ASSIGNING AND UNASSIGNING DISKS

Before you run MAKE or FIX_DISK to create or repair a partition, you must shut down the partition to make it unavailable to users. When you have finished creating or repairing a partition, you should reverse the shutdown process and make the partition again available to users.

Shutting Down a Partition

Before you shut down a partition, a disk drive, or a system, give users sufficient time to finish their work and stop working on that partition or on that system. Use the MESSAGE command to give adequate warnings. You may need to use the -FORCE option. For example:

```
OK, MESSAGE -ALL -NOW -FORCE
Disk PTUSER will be shutting down in 5 minutes  
OK,
```

Press to ring the bell at each terminal.

If the you are going to run `FIX_DISK` on the command device, do the following:

1. Use the `MAXUSR 0` command to prevent any users from logging in.
2. Use the `STOP_LSR` command to stop the Login server.
3. Use the `STOP_DSM` command to stop DSM.
4. Stop any other system servers that depend on the command device.
5. Log out out any other phantoms and any users remaining on the system after the appropriate warning time has expired.

For example:

```
OK, MAXUSR 0
OK, STOP_LSR
Really? YES
OK, STOP_DSM
OK, LOGOUT ALL
OK,
```

Messages appear indicating that the Login server, the DSM phantoms, and any other phantoms are logging out. It is necessary to log out users and system servers because, when a partition is shut down, all files on the partition, including runfiles for EPFs, are closed. Thus, if a process attempts to access memory, it fails because the EPF is unmapped.

Procedure for Assigning and Unassigning Disks

To create new partitions or to perform maintenance on an existing partition, use the following procedure, which is summarized in Figure 4-2. Note that most of this procedure can be carried out only from the supervisor terminal. Remember to warn users before shutting down a partition.

1. Use the `SHUTDN` command to shut down the partition to be repaired.

```
OK, SHUTDN 1060
OK,
```

Note

If you are running `FIX_DISK` on the command device, stop the Login server with the `STOP_LSR` command and stop DSM with the `STOP_DSM` command. Log out other system phantoms. Use the `-COMDEV` option to shut the partition down; do not use the `SHUTDN` command. Remember to start the Login server (`START_LSR`), DSM (`START_DSM`), and any other system phantoms when you finish running `FIX_DISK` on the command device.

2. Use the `DISKS` command followed by the logical disk's physical device number (pdev) to place the disk in the Assignable Disks Table.

```
OK, DISKS 1060
OK,
```

3. After placing the disk in the Assignable Disks Table, use the `ASSIGN DISK pdev` command to assign it to yourself.

```
OK, ASSIGN DISK 1060  
OK,
```

4. You can now perform these operations:

- Use `MAKE` to format a partition.
- Use `FIX_DISK` to repair a partition.
- Use `COPY_DISK` to physically copy a partition.
- Use `PHYSAV` to physically save a partition.
- Use `PHYRST` to physically restore a partition.

5. After you have completed the operations in step 4, use the `UNASSIGN DISK pdev` command to unassign the disk.

```
OK, UNASSIGN DISK 1060  
OK,
```

6. Use the `DISKS NOT pdev` command to remove the disk from the Assignable Disks Table.

```
OK, DISKS NOT 1060  
OK,
```

7. Use the `ADDISK` command to make the disk available to users again.

```
OK, ADDISK 1060  
Starting up revision 22.1 partition "RASCAL".  
OK,
```

Once a partition has been placed in the Assignable Disks Table, any user can assign the partition and use many of the commands shown in Figure 4-2 at any user terminal. However, it is assumed for the purposes of this document that only an Operator would carry out the processes described here at the supervisor terminal.

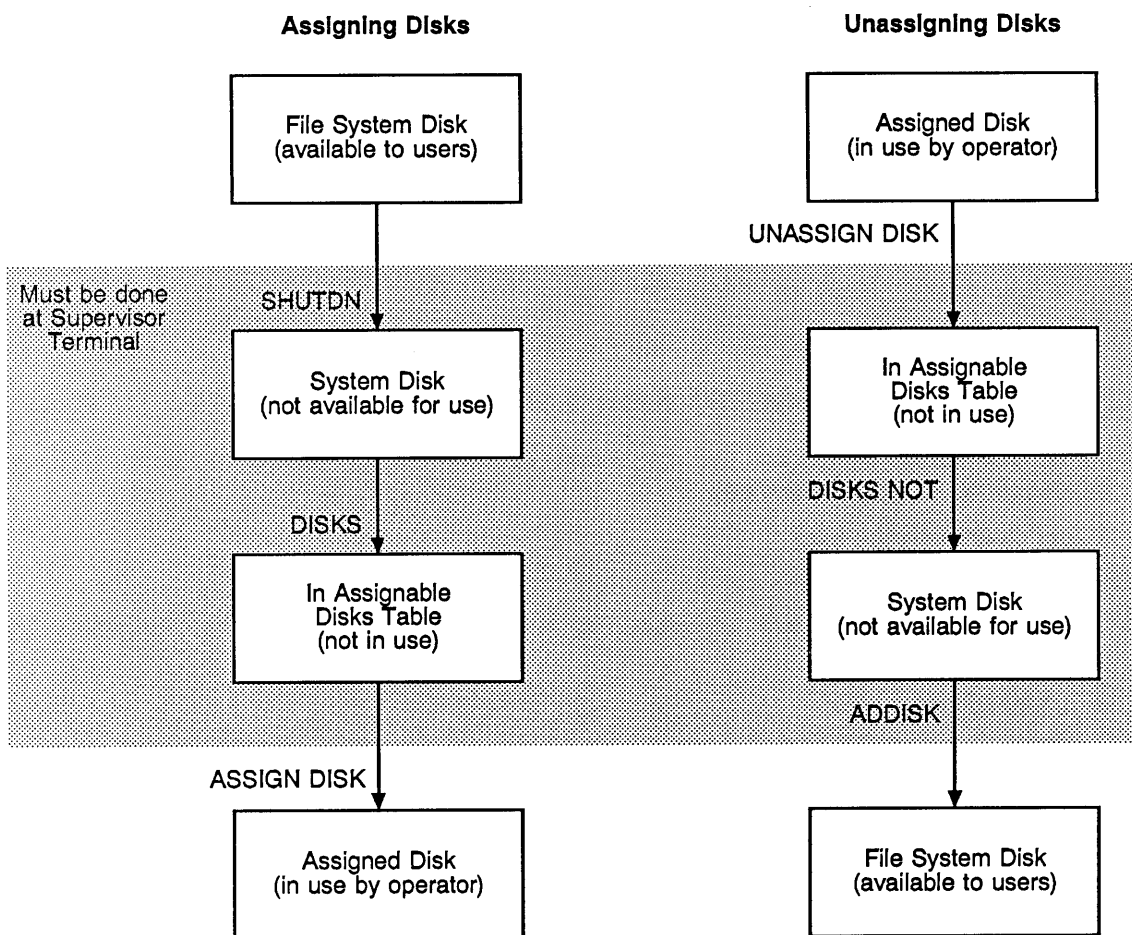


FIGURE 4-2. Assigning and Unassigning Disks

FORMATTING DISKS — MAKE

This chapter describes the MAKE utility, used for formatting and partitioning physical disks. The first part tells why you use MAKE, explains how to choose the options you want for particular jobs, and discusses handling of badspots by PRIMOS. The second part takes you step-by-step through the actions that may precede, accompany, and follow the normal use of MAKE, and provides examples of typical sessions with MAKE. The final part discusses how to use MAKE when PRIMOS is not running.

WHY YOU NEED TO RUN MAKE

Before a disk can be used by your computer system, it must conform to PRIMOS' method of organizing data on the disk. Bringing a disk into conformity with your system's requirements is called **formatting**, **making**, or **creating** a disk. Your System Administrator may require you to subdivide a disk's memory area into two or more partitions; this action is referred to as **partitioning** a disk.

MAKE is the system utility for formatting and partitioning disks. MAKE can format and create both user disks and paging disks. **User disks** provide the actual storage for users' files. **Paging disks** are areas of the disk that PRIMOS uses when managing virtual memory.

After a user partition is created, PRIMOS writes the Master File Directory (MFD) on it. The MFD is the top-level directory of the file system and contains the list of all directories and files stored on the partition. When the partition is first created, the following two or three files and two directories are in the MFD.

- The BOOT file, which you use to start, or bootstrap, the system from disk.
- The Disk Record Availability Table (DSKRAT), which contains information about the physical structure of the partition. The DSKRAT file gets the same name as the partition when the partition is created.

- Either a badspot file (BADSPT) or a dynamic badspot (DBS) file, which records the locations of any physical defects on the disk. (The BADSPT file is created only if badspots exist on the partition and the partition is in -AC mode.) The DBS file is not visible but there is a file named DYNBSP that controls access to the DBS file.
- An empty directory, CMDNCO, used to hold system files.
- An empty directory, DOS, used to hold a copy of the PRIMOS II single-user operating system.

The MAKE command starts the MAKE utility, which is stored as CMDNCO>MAKE.SAVE. You normally invoke MAKE directly with the MAKE command. You can also run MAKE from a command input (COMINPUT) file. In addition, you can boot MAKE.SAVE as a standalone program either from disk or from tape; this is discussed at the end of this chapter.

Caution

To avoid mistakenly invoking a pre-Rev. 22.1 version of MAKE, delete any older versions of MAKE. When you invoke MAKE, a copyright line with a revision stamp is displayed. If you invoke an earlier version than Rev. 22.1, you may not get the results that you expect. See also the following two warnings.

The MAKE Utility and Hashed Directories

Starting at Rev. 20.0 PRIMOS, the MAKE utility creates ACL-protected partitions that support **hashed directories**. The term *hashed* refers to the way PRIMOS gains access to data controlled by a directory. Hashing speeds up processing because the system does not have to perform a sequential search to locate a file system object.

Only ACL directories are hashed. If, for some reason, you create password directories on Rev. 20.0 and later disks, they will not be hashed. In addition,

if you create robust partitions with Rev. 22.1 MAKE_ROBUST, the directories on these partitions are also not hashed but they are ACL-protected. (See Chapter 7 for a discussion of robust partitions, a new feature of Rev. 22.1.)

The protection placed on the MFD when a partition is created is SYSTEM:ALL and \$REST:LUR. All directories created on these partitions are ACL-protected by default.

In order to use hashed directories on existing pre-Rev. 20.0 partitions, you must first convert those partitions to Rev. 20.0 or later format. Conversion of partitions is discussed later in this chapter.

Note

Because Rev. 22.1 partitions have file system attributes that pre-Rev. 22.1 partitions do not, a Rev. 22.1 partition can *not* be added locally to a pre-Rev. 22.1 system. A Rev. 22.1 partition may be accessed remotely by pre-Rev. 22.1 systems, however, through PRIMENET.

The First Partition of a Rev. 22.1 Disk

The first partition (the partition containing surface 0) of a Rev. 21.0 or later physical disk defines how all the partitions on the disk are to be treated by the disk controller. (See also Chapter 8.)

Note

Starting at Rev. 21.0, MAKE creates a DBS file and an RMA on the first partition (logical device 0) of a physical disk. MAKE uses the DBS file and the RMA in handling badspots on all other partitions on that physical disk when the partitions on the disk are in intelligent controller (-IC) mode. Thus, you must create the first partition of a Rev. 21.0 or later disk *before* creating other partitions on that disk. The DBS file and the RMA are accessed when the partition is in Dynamic Badspot Handling (-IC) mode.

All partitions on a physical disk must be in the same mode: either Dynamic Badspot Handling (-IC) mode or Nondynamic Badspot Handling (-AC) mode. It is also recommended that all partitions on a disk be of the same revision.

WARNING

Do not use a pre-Rev. 22.1 version of MAKE on an existing first partition of a Rev. 22.1 disk. Do not use any version of MAKE with the -FORMAT option on the first partition of a Rev. 22.1 disk if you have created other partitions on the disk. If you do either, the entire physical disk (all partitions on the disk) may be corrupted because MAKE overwrites the DBS file in these cases. Consequently, the data for records that initially were to be written to badspots on any partition of the physical disk, and that were thus written to the RMA, are lost. Be sure all the data on a disk is backed up before using MAKE.

In addition, do not change the number of surfaces on an existing Rev. 22.1 first partition. If you do, the basic pdev of the first partition is changed and all other partitions on that disk lose any data that was remapped to the RMA.

CONVERTING PARTITIONS TO REV. 22.1

Because Rev. 22.1 partitions are new format PRIMOS partitions, it is necessary for you to convert existing partitions to Rev. 22.1 format so that your system can use all of the features of Rev. 22.1. You must use MAKE to convert pre-Rev. 22.0 partitions to Rev. 22.1 partitions; you cannot convert pre-Rev. 22.0 partitions to Rev. 22.1 with FIX_DISK. You can convert Rev. 22.0 partitions to Rev. 22.1 with FIX_DISK or you can create Rev. 22.1 partitions with MAKE. If you want to use robust partitions, you must use Rev. 22.1 MAKE to create them. (For a discussion of robust partitions, see Chapter 7.)

Conversion Procedure

The procedure below explains how to convert existing partitions to Rev. 22.1 by using MAKE.

1. Save all of the data on the existing partition to magnetic tape by using one of the logical backup utilities, such as MAGSAV or BACKUP. (See the *Data Backup and Recovery Guide*.)
2. Run Rev. 22.1 MAKE on the partition, specifying all the necessary options to create the particular Rev. 22.1 partition that you want. (See the description of the options in the next section.)
3. Restore the data saved in step 1 by using a compatible logical backup utility, such as MAGRST if you used MAGSAV in step 1.

When converting existing partitions to Rev. 22.1, you must create the first partition (logical device 0) first since MAKE must write the DBS and RMA on the first partition. The DBS and RMA handle badspots for all partitions on the disk when the partition is in Dynamic Badspot Handling (-IC) mode.

File System Features

The file system features available on the various revision partitions are summarized in Table 5-1. The features are additive from left to right; that is, Rev. 22.1 supports all features of the previous revisions plus robust partitions and unlimited numbers of extents for CAM files.

*TABLE 5-1. Partition Features Available at
Rev. 20.0 Through the Current Revision*

<i>Rev. 20.0</i>	<i>Rev. 21.0</i>	<i>Rev. 22.0</i>	<i>Rev. 22.1</i>
Hashed directories (Not available on robust partitions)	Mirroring	Setting minimum and maximum extent sizes for CAM files	Robust partitions
ACL-protected directories	Dynamic badspot handling		Unlimited CAM file extents
Reverse/forward sectoring			

OPTIONS AVAILABLE FOR RUNNING MAKE

MAKE resides in CMDNCO as CMDNCO>MAKE.SAVE. You invoke MAKE with this command format:

MAKE -DISK *pdev* [*options*]

pdev is the physical device number of the disk you are creating. See Chapter 3 for directions on constructing physical device numbers.

This list summarizes the MAKE command-line options in the order they are discussed in the next section of this chapter.

1. Specify the *pdev* of the partition that you are creating. You must specify the -DISK option and specify the *pdev* with it.
2. Specify the type of physical disk that the partition is on. This option is required. (-DISK__TYPE)
3. Create the partition as a Rev. 21.0 or later disk that can be used on any type of disk controller. This is the default when using MAKE standalone. (-ALL__CONTROLLER)
4. Specify the level of verification to be used in checking for badspots. (-BADSPOT__LEVEL)
5. Specify the baud rate of the supervisor terminal, if the disk will be used for booting a system. (-BAUD__RATE)
6. Use an existing badspot file on another partition if the partitions are in Nondynamic Badspot Handling (-AC) mode or are pre-Rev. 21.0. (-COPY__BADSPOTS__BY__DEVICE and -COPY__BADSPOTS__BY__NAME)
7. Create the partition as a specific revision disk. (-DISK__REVISION)
8. Format a new disk and write a new badspot file if badspots exist on the disk. (-FORMAT and -NEW__DISK)
9. Display the command line and a complete list of options. (-HELP)
10. Initialize file system records. (-INIT)
11. Create the partition as a Rev. 21.0 or later disk, taking into consideration the capabilities of an intelligent disk controller to do dynamic badspot handling and disk mirroring. You cannot use this option when running MAKE standalone. (-INTELLIGENT__CONTROLLER)
12. Save records having correctable errors, not considering them as badspots. (-MAP__UNCORR)
13. Specify maximum and minimum extent sizes for CAM files for Rev. 22.0 and later partitions. (-MAX__EXTENT__SIZE, -MIN__EXTENT__SIZE)
14. Implicitly use vendor flaw maps on disks that have them by specifying the disk model numbers with the -DISK__TYPE option.

15. Do not use vendor flaw maps on disks that have them and use the default level of badspot checking. (-NO_FLAW_MAP)
16. Use the reverse sectoring method (-OVERRIDE_DEFAULT_INTERLEAVE) or the forward sectoring method (-RESTORE_DEFAULT_INTERLEAVE) of file record allocation for post-Rev. 19.0 partitions.
17. Specify the name of the partition. This option is required. (-PARTITION)
18. Input a list of known badspots. (-QUERY_BADSPOTS)
19. Create a robust partition. (-ROBUST)
20. Specify the number of records to be used for paging, if the partition is to be a paging partition. (-SPLIT)

Deciding Which Options to Use

This section contains the twenty items listed above that explain the options you need to use to create partitions under various circumstances. As you read the next twenty items, make a list of options that you need to achieve the desired results. All of the MAKE options are also summarized in Appendix F, Summary of Command Syntax for MAKE and FIX_DISK, for your convenience.

1. Specify the Physical Device Number: You must use the **-DISK** option (abbreviation **-DSK**) to specify the physical disk and the logical disk, or partition, on which MAKE is to operate. (See Chapter 3 for information on determining physical device numbers.) The **-DISK** option *must* be specified and *must* be the first option on the command line. (However, it is not necessary to use **-DISK** with the **-HELP** option if all you want is the **-HELP** display.) For example,

```
OK, MAKE -DISK 21220
```

Caution

The **-DISK** option *must precede* the physical device number on the command line. If you omit the **-DISK** option or enter it after the physical device number, the program may halt or unpredictable behavior may occur. If the latter happens, do the following:

- Press **Ctrl P** to stop the program. If you have assigned only one disk (the one to be formatted) to your terminal, it is unlikely that any harm has occurred.
- Use the **RELEASE_LEVEL -ALL (RLS -ALL)** command, then invoke **MAKE** again. Include the **-DISK** option on the command line in the correct place.

If you did not previously assign the partition, MAKE informs you and aborts:

```
DISK 21220 NOT ASSIGNED  
ERI
```

If you do not specify the physical device number (pdev) following the -DISK option, MAKE prompts you for it. If the pdev that you enter is invalid, MAKE tells you that it is invalid and again prompts you for the pdev:

```
Invalid physical device number.
Physical device?
```

If you attempt to recreate or overlap the first partition of a Rev. 21.0 or later physical disk by specifying a pdev that includes an existing first partition, MAKE warns you and you are queried to be sure you want to continue:

```
Disk DBS file changed! All other partitions on this disk
will be corrupted if you continue.
Do you wish to continue?
```

If you answer YES, MAKE continues; if you answer NO, MAKE aborts.

WARNING

If you change the number of surfaces in the first partition of the physical disk (that is, you change the basic pdev of the first partition), you change the RMA of the disk and all other partitions on that disk lose any data that was remapped to the RMA.

2. Specify the Disk Type: Starting at Rev. 21.0, you must specify the type of disk with the -DISK_TYPE option (abbreviation -DT). If you do not, MAKE displays a list of disk types and prompts you for one as in the example below. Follow the -DISK_TYPE option with one of the arguments shown in the following example and in Table 5-2. If you do not follow the -DISK_TYPE option with the disk type argument or if you specify an invalid disk type (one that is not known to MAKE), MAKE displays a list of valid disk types and prompts you for a correct one:

```
OK, MAKE -DISK 21260 -PARTITION FALCON -DISK_TYPE
**** MAKE <Rev. 22.1> **** Copyright (c) 1988, Prime Computer, Inc.
```

Disk type must be one of the following:

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

Disk type?

Caution

Be sure that you specify the correct disk type when creating a Rev. 21.0 or later partition. MAKE uses the geometry information of the physical disk to construct the DBS file and the RMA. They will not be the correct size if MAKE has incorrect information.

Be sure that you specify the correct disk type in any case. If you do not, MAKE will attempt to create the type of disk you have specified and will encounter an error condition and abort. You can recover by rerunning MAKE, using the correct disk type argument, but you will have wasted time.

TABLE 5-2. Disk Types and Arguments

<i>Disk Type</i>	<i>Argument</i>	<i>Comments</i>
Storage Module Disks: 80MB or 300MB	SMD	Removable disk packs
Cartridge Module Devices: 32MB, 64MB, or 96MB	CMD	
Fixed-Media Disks: 60MB FMD	MODEL_4711	2455™ system internal disk
68MB FMD	68MB	
84MB FMD	MODEL_4714	2455 system
120MB FMD	MODEL_4715	internal disks
158MB FMD	158MB	
160MB FMD	160MB	
258MB FMD	MODEL_4719	2455 system internal disk
315MB FMD	MODEL_4475	
328MB FMD	MODEL_4721	2455 system internal disk
496MB FMD	MODEL_4735	Internal flaw maps
675MB FMD	600MB	Same as 600MB FMD
770MB FMD	MODEL_4845	Internal flaw maps
817MB FMD	MODEL_4860	Internal flaw maps

3. Create a Rev. 22.1 Partition to Use on Any Disk Controller: If you wish to use a Rev. 22.1 disk on a drive that is associated with a nonintelligent disk controller, use the **-ALL_CONTROLLER** option (abbreviation **-AC**). This causes MAKE to create the file BADSPT for badspot handling. The DBS file and the RMA are created but are inactive. You can use **FIX_DISK** to switch between Nondynamic Badspot Handling (**-AC**) mode and Dynamic Badspot Handling (**-IC**) mode.

The format for creating a partition in Nondynamic Badspot Handling mode is

MAKE -DISK *pdev* -DISK_TYPE *type* -ALL_CONTROLLER

You must specify the **-DISK_TYPE** option so MAKE can set aside the correct number of records for the DBS file and the RMA. You must use the **-AC** option when the disk type is CMD because you cannot use the CMD type with dynamic badspot handling. See Table 5-2 for disk types. You must also specify the **-ALL_CONTROLLER** option when running MAKE standalone because intelligent disk controllers do not get downloaded with the necessary software until PRIMOS is booted.

Note

You can specify either the **-INTELLIGENT_CONTROLLER** or the **-ALL_CONTROLLER** option with Rev. 21.0 and later partitions. If you do not specify either option, MAKE defaults to the proper mode based on the type of disk controller in your system. Conversely, you cannot specify either of those options when making pre-Rev. 21.0 partitions; if you do, MAKE tells you that these modes are not available with partitions earlier than Rev. 21.0 and aborts.

4. Specify the Level of Verification: MAKE can perform a maximum of four levels of media verification while creating a logical disk. Verification consists of writing a data pattern onto each record of the disk, reading the data back, and checking to see that the data being read is the same data that was written. This provides an automatic check for badspots on the disk.

Normally, one (1) level of badspot checking is performed for SMDs and CMDs, two (2) levels of badspot checking are performed for Model 4711, Model 4714, Model 4715, Model 4719 and Model 4721 FMDs (disks on 2350™, 2450, and 2455 systems), and four (4) levels of badspot checking are performed for all other FMDs. To override these defaults or to specify that no verification is to take place, include the **-BADSPOT_LEVEL** option (abbreviation **-BADLEV**) on the command line, followed by an integer ranging from 0 through 4, inclusive. This number represents the level of verification to be performed by MAKE. The 0 means no verification; 4 means four levels of verification (the most thorough and the most time-consuming). For example:

OK, MAKE -DISK 1062 -PARTITION IONIA -DISK_TYPE SMD -BADSPOT_LEVEL 4

If you include the **-BADSPOT_LEVEL** option on the command line, but you do not include the number representing the level of verification, MAKE prompts you for it like this, in the case of an FMD:

Level of bad spot checking? (default = 4)

If you do not enter a level at this prompt or if you do not include the `-BADSPOT_LEVEL` option on the command line, MAKE defaults to the appropriate level of verification for the type of disk involved.

If you specify `-BADLEV 0`, MAKE does not do any badspot checking. If the partition has an existing badspot file, MAKE reads the existing badspot file. If the disk also has a flaw map, MAKE reads the flaw map. Specifying `-BADLEV 0` may be useful if you are recreating an existing partition because MAKE uses less time if no badspot checking is done. However, do not use the `-NEW_DISK` option in this case. If you do, MAKE will write a new badspot file, overwriting the existing badspot file. Since you specified `-BADLEV 0`, the new badspot file will be empty.

5. Set the Baud Rate: MAKE writes the boot program onto the disk when it creates a partition. The baud rate of the supervisor terminal is set in the boot program. If the partition being made will be used for booting PRIMOS, you can set or change the supervisor terminal baud rate when you create the partition.

If the baud rate setting of the partition disagrees with the actual baud rate set on the supervisor terminal and your machine does not have a VCP III or later Diagnostic Processor, you may be unable to use the supervisor terminal after booting from the newly created partition. Set the baud rate to the baud rate of your supervisor terminal with the `-BAUD_RATE` option (abbreviation `-BAUD`) if the partition will be used for booting PRIMOS.

The default baud rate is 300 bits per second (bps). To set a different baud rate, include the `-BAUD_RATE` option on the command line, followed by the desired baud rate in decimal bits per second. Valid baud rates are: 110, 300, 1200, 9600.

For example, to set the baud rate to 9600, use a command line like this:

```
OK, MAKE -DISK 1060 -BAUD_RATE 9600 -DT MODEL_4715
Partition name? PAYROL
```

If you include the `-BAUD_RATE` option on the command line but do not specify the baud rate, MAKE prompts for a valid rate. If you specify an invalid baud rate, MAKE displays a list of valid baud rates and prompts you for a valid rate. If you do not include the `-BAUD_RATE` option on the command line, MAKE sets the baud rate to 300 bps.

6. Copy Badspot Information From Another Partition: If you have already created a pre-Rev. 21.0 partition or you have created a Rev. 21.0 or later partition in Nondynamic Badspot Handling (`-AC`) mode on the current physical disk and have entered all badspots known on the entire disk, you may copy the badspot information from that partition to a new partition. (For a detailed example, see the section Keeping Redundant Badspot Files later in this chapter.)

To copy badspot information from a started (named) partition on the current physical disk, use the `-COPY_BADSPOTS_BY_NAME` option (abbreviation `-CPYNAM`). The partition from which you are copying badspot information must have been started up with the `ADDISK` command.

For example:

```
OK, MAKE -DISK 20063 -PART GOLDEN -COPY_BADSPOTS_BY_NAME SILVER
```

If you do not supply the name or the name you do supply is not that of an added partition, MAKE displays this message and continues:

```
Partition not found: SILVER
```

```
Ignoring copy_badspots_by_name option. (MAKE)
```

To copy badspot information from an assigned partition (specified by a physical device number), use the **-COPY_BADSPOTS_BY_DEVICE** option (abbreviation **-CPYDEV**). The partition must have been assigned with the ASSIGN DISK command. For example:

```
OK, ASSIGN DISK 1062
```

```
OK, MAKE -DISK 20063 -PART GOLDEN -COPY_BADSPOTS_BY_DEVICE
```

If you do not include *pdev*, MAKE prompts you for the pdev:

```
Copy badspots from physical device? 1062
```

If the partition from which you want MAKE to copy the badspot file has not been assigned, MAKE informs you and aborts.

7. Create a Specific-Rev. Partition: Normally, MAKE creates a Rev. 22.1 partition. If the partition being created is to be run on a pre-Rev. 22.1 system, include the **-DISK_REVISION** option (abbreviation **-DSKREV**) on the command line. If you do not specify this option, MAKE creates a Rev. 22.1 partition by default. The valid arguments to this option are 18, 19, 20, 21, 22, and 22.1. (You can use a decimal point and zero combination with any of the major revisions, 21.0, for example.) If you do not specify one of these six arguments on the command line, MAKE prompts you for one and checks the validity of the argument you enter. If the argument is out of the range of the arguments listed here, MAKE informs you that you entered a bad revision and reprompts until you enter a valid disk revision.

```
MAKE -DISK 41062 -DT SMD -PART JUPITR -DSKREV
**** MAKE <Rev. 22.1> **** Copyright (c) 1988, Prime Computer, Inc.
Disk revision? 24
```

```
Bad disk revision, must be either 18, 19, 20, 21, 22, or 22.1.
```

```
Disk revision? 22
```

```
Making 4 head partition JUPITR
```

If you specify an invalid disk revision on the command line, MAKE informs you that you entered a bad argument and reprompts as above.

8. Format a New Disk: If the disk has never been used on a Prime system before, include the options **-FORMAT** (abbreviation **-FMT**) and **-NEW_DISK** (abbreviation **-NEWDSK**) on the command line. Including these options tells MAKE to initialize the record headers on the disk (**-FORMAT**) and create a new badspot file if badspots exist (**-NEW_DISK**) without looking for an existing badspot file. For example:

```
OK, MAKE -DISK 1060 -DISK_TYPE 600MB -FORMAT -NEW_DISK
```

If you are recreating and changing the geometry (changing the basic pdev) of a first partition on a disk that was previously partitioned in -IC mode, include the -FORMAT option on the command line. You must use the -FORMAT option of MAKE on the first partition to remodify the record headers. If you do not specify the -FORMAT option when repartitioning the first partition as a Rev. 21.0 or later partition, MAKE defaults to that option and informs you with this message:

MAKE will enable -FORMAT to change the DBS file.

WARNING

Do not use a pre-Rev. 22.1 version of MAKE on the first partition of an existing Rev. 22.1 disk. Do not use any revision of MAKE with the -FORMAT option on the first partition of an existing Rev. 22.1 disk. If you do, the entire physical disk (all partitions on the disk) may be corrupted because MAKE overwrites the DBS file in these cases. Consequently, the data for records that initially were to be written to badspots on any partition of the physical disk, and that were thus written to the RMA, are lost. Be sure all the data on a disk is backed up before using MAKE.

You must therefore use the -FORMAT option of MAKE in these circumstances:

- The partition is Rev. 21.0 or later format and is in Dynamic Badspot Handling (-IC) mode and you are repartitioning and changing its geometry, or basic pdev.
- The disk is new or has not been used on a Prime system before.

The -FORMAT option is ineffective if the disk type is one of the following FMDs:

Model 4711 (60MB) Model 4719 (258MB)
Model 4714 (84MB) Model 4721 (328MB)
Model 4715 (120MB)

These disks are preformatted when they are manufactured. It is recommended that you not use the -FORMAT option with these disk types to save processing time.

If you are creating a new partition on a disk that has been used previously (for example, you are creating one 4-surface partition from two 2-surface partitions), and you do not include the -NEW_DISK option, MAKE displays this message:

```
Unable to get badspots from device: <pdev>
Continuing with make. (MAKE)
```

This means that MAKE cannot find a badspot file on the partition *pdev* you are creating; MAKE constructs a badspot file. If you use the -NEW_DISK and -FORMAT options on an existing Rev. 21.0 or later first partition (ldev 0), you get this message:

```
Disk DBS file changed! All other partitions on this disk
will be corrupted if you continue.
Do you wish to continue?
```

If you specify the -NEWDSK option or if you are changing the first partition geometry, but do not specify the -FORMAT option, you get this message:

```
Disk DBS file change! All other partitions on this disk
will be corrupted if you continue.
MAKE will enable -FORMAT to change the DBS file.
Do you wish to continue?
```

This is a caution to let you know that a change in the DBS file results in the loss of any remapped records from other partitions on the disk. If you answer NO, MAKE aborts. Answer YES if your intent is to repartition the disk. If you do not specify the -FORMAT option, MAKE defaults to that option as indicated in the second message in order to remodify record headers.

9. Get Online Help With the MAKE Options: To see a display of the MAKE options and a brief explanation of each option, use the **-HELP** option in this command line:

```
OK, MAKE -HELP
**** MAKE <Rev. 22.1> **** Copyright (c) 1988, Prime Computer, Inc.
USAGE: MAKE -DISK pdev -PARTITION name [ options ]
        MAKE -DSK pdev -PART name [ options ]
```

OPTIONS:

A list of the valid disk types and a summary of the operation of each option are displayed in four screens. Note that you do not need the -DISK option to get help and that you can use this command at any terminal.

10. Initialize File System Records: Formerly it was necessary for MAKE to initialize the records in a partition. Initialization consists of clearing out any header information in the records, such as pointers to other records if these records were previously used in a file. PRIMOS now initializes records when creating file system objects, so it is not necessary for MAKE to do it. At Rev. 22.1, MAKE does not initialize records by default, thus saving some processing time. Prior to Rev. 22.1, you had to use the **-NO_INIT** option to prevent MAKE from initializing the records.

WARNING

Although MAKE does not initialize file system records by default at Rev. 22.1 and previously by the use of the **-NO_INIT** option, MAKE does initialize the file system. All file system objects are effectively deleted when you run MAKE on a partition. Use of **-NO_INIT**, specifically or by default, merely reduces the amount of time needed to create a particular partition. Never use MAKE on a partition if it contains the only copy of any data that you want.

However, having MAKE initialize records may help if you are trying to locate file system errors, particularly errors resulting from hardware problems. If you want MAKE to initialize records, use the **-INIT** option, which is new at Rev. 22.1. This causes MAKE to clear the header information from all records. You cannot use both the **-INIT** and the **-NO_INIT** options on the same command line. If you do, MAKE displays this error message and aborts:

```
-NO_INIT option cannot be used in conjunction with the -INIT option.
```

11. Create a Partition Capable of Dynamic Badspot Handling: If your system has an intelligent disk controller and the software to download the controller, this is indicated by the following message displayed at cold start.

DLL and init ICOP complete (26) - (disk_init)

Use the **-INTELLIGENT_CONTROLLER** option (abbreviation **-IC**) so that dynamic badspot handling can take place on the partitions associated with intelligent disk controllers. If you intend to mirror partitions, you must use this option. Mirroring can be done only with Rev. 21.0 or later revision partitions associated with intelligent disk controllers in Dynamic Badspot Handling (**-IC**) mode. (See Chapter 8 for details on dynamic badspot handling and Chapter 9 for details on mirroring partitions.) The format for this purpose is

MAKE -DISK *pdev* -DT *type* -INTELLIGENT_CONTROLLER -FMT -NEWDSK

You must use the **-DISK_TYPE** (**-DT**) option and *type* must be one of the types listed in Table 5-5 under item 20. Because SMDs can be moved from one disk drive to another and, thus, from one disk controller to another, it is recommended that you not use the **-IC** option with SMDs. In addition, you cannot use this option with the **CMD** type or the disks (Models 4711, 4714, 4715, 4719, or 4721) associated with the 2350, 2450, and 2455 systems. Thus you can use **-IC** only with the disk types listed in Table 5-5.

You must also use the **-FORMAT** and **-NEW_DISK** options when you create the first partition of a disk in **-IC** mode so that **MAKE** initializes record headers and recreates the badspot file for the intelligent disk controller.

Note

You must specify the **-ALL_CONTROLLER** option with Rev. 22.1 partitions when running **MAKE.SAVE** standalone. You can then convert a partition to **-IC** mode with **FIX_DISK**. If you do not specify either **-AC** or **-IC** when running **MAKE** with the system booted and running (that is, under **PRIMOS**), **MAKE** is able to default to the proper mode based on the disk controller in the system. You cannot specify either the **-AC** or the **-IC** option with pre-Rev. 21.0 partitions.

Starting at Rev. 21.0, **MAKE** creates a DBS file and an RMA on the first partition (logical device 0) of a physical disk. (See Chapter 8) **MAKE** uses the DBS file and the RMA to handle badspots on all other partitions on that physical disk. Thus, you must create the first partition of an **-IC** mode disk *before* creating other partitions on that disk. The DBS file and the RMA are activated when the partition is in Dynamic Badspot Handling (**-IC**) mode.

12. Save Records With Correctable Errors: When checking for badspots, **MAKE** writes data patterns in each record on the partition and reads those data patterns back. If the data pattern read does not match the pattern written, that record is marked as a badspot and is not used in the file system. While doing badspot checking, **MAKE** uses an error-correcting algorithm to attempt to correct the data pattern read. If you want to save records having correctable errors and want **MAKE** to mark only uncorrectable errors as badspots, include the **-MAP_UNCORR** option (abbreviation **-UNCORR**). This will save marginally usable space on the partition.

Note that use of this option is not recommended unless you absolutely need the space and have no other choice, and unless you frequently back up your file system. The reason this option is not recommended is that, although the data pattern may be corrected at the time you use `-MAP_UNCORR`, it may not be corrected when PRIMOS reads and writes data to that record in the future. You may thus lose data.

13. Set the Maximum and Minimum Extent Sizes for CAM Files: Starting at Rev. 22.0, you can set the maximum and minimum extent sizes for all CAM files on a partition. An extent size is the number of contiguous 2048-byte records in one extent of the CAM file. (See the discussions of robust partitions and CAM files in Chapter 7 and the description of CAM files in Chapter 1.) You set the maximum and minimum extent sizes using the options `-MAX_EXTENT_SIZE` (abbreviation `-MAXSIZ`) and `-MIN_EXTENT_SIZE` (abbreviation `-MINSIZ`) in a command line like this:

```
MAKE -DISK pdev [-MAXSIZ  
                -MINSIZ] size
```

size is a decimal number representing the maximum and minimum number of records per extent in a CAM file. *size* must be greater than zero and the minimum extent size must be less than or equal to the maximum. You can set the maximum extent size to any value up to 32767.

When you set the maximum and minimum extent sizes with the `MAKE -MAXSIZ` and `-MINSIZ` options, these values are written in the header information for the partition and remain for the life of the partition. However, you can later change these values with the `FIX_DISK -MAXSIZ` and `-MINSIZ` options.

If you do not specify the maximum and minimum extent sizes by using the `-MAXSIZ` and `-MINSIZ` options, MAKE uses the default maximum and minimum extent sizes, in number of records, for Rev. 22.0 and Rev. 22.1 standard partitions as shown in Table 5-3. If you convert a Rev. 22.1 standard partition to a robust partition by using the `MAKE_ROBUST` utility, `MAKE_ROBUST` uses the defaults for robust partitions shown unless you use the `MAKE_ROBUST -MAXSIZ` and `-MINSIZ` options (see Chapter 7).

TABLE 5-3. Rev. 22.1 Default Maximum and Minimum Extent Sizes

	<i>Robust Partition</i>	<i>Standard Partition</i>
<i>Maximum</i>	256	32
<i>Minimum</i>	64	16

If you do use the `-MAXSIZ` and `-MINSIZ` options and you do not include the sizes, you are prompted for these values and they are checked for proper ranges of size.

For example,

```
MINIMUM_EXTENT_SIZE? 70
```

```
MAXIMUM_EXTENT_SIZE? 250
```

You enter the appropriate values at the prompts. See Chapter 7 for a discussion of setting minimum and maximum extent sizes. If you intend to convert a Rev. 22.1 standard partition to a robust partition by using the MAKE_ROBUST utility, as described in Chapter 7, do not use the MAKE -MAXSIZ and -MINSIZ options. To set the maximum and minimum extent sizes on a robust partition, you must use those options with MAKE_ROBUST. If you do not, MAKE_ROBUST sets the extent sizes to the default values.

If the partition is a Rev. 20.0 or a Rev. 21.0 partition, the default minimum and maximum extent sizes for CAM files are set by PRIMOS to 16 records and 32 records, respectively.

If you attempt to use either the -MAXSIZ or -MINSIZ option but the partition you are creating is an earlier revision than Rev. 22.0, you see an appropriate error message and MAKE aborts:

```
MAXIMUM_EXTENT_SIZE is only available starting at Rev. 22.  
ER!
```

You should then restart MAKE with the START command or reenter the MAKE command line, using the proper options in either case.

14. Use the Flaw Map on Fixed-Media Disks That Have This feature: The Model 4735 (496MB), the Model 4845 (770MB), and the Model 4860 (817MB) FMDs contain flaw maps, or lists of defects, written on a designated area of the disk medium during manufacturing. You should use this flaw map when creating partitions on Model 4735, Model 4845, or Model 4860 disks. MAKE identifies the Model 4735, 4845, or 4860 disks by the MODEL_4735, the MODEL_4845, or the MODEL_4860 argument to the -DISK_TYPE option. To initially format and partition a Model 4735 disk, use a command line like the following:

```
OK, MAKE -DISK 1060 -DISK_TYPE MODEL_4735 -FORMAT -NEW_DISK  
Partition name? SYSCMD
```

This command line causes MAKE to read the flaw map because the argument to the -DISK_TYPE option identifies this disk type as having a flaw map. MAKE converts the information contained in the flaw map to badspot information, and to writes the badspot information to the standard badspot file, MFD>BADSPT, or into the DBS file.

After you use any of these arguments (MODEL_4735, MODEL_4845, or MODEL_4860) to the -DISK_TYPE option, and MAKE successfully reads and processes the flaw map, you see the following two messages relating to flaw map processing.

```
OK, MAKE -DISK 1060 -DISK_TYPE MODEL_4735 -FORMAT -NEW_DISK
Partition name? SYSCMD
```

```
**** MAKE <Rev. 22.1> **** Copyright (c) 1988, Prime Computer, Inc.
```

```
      .      .      .
      .      .      .
      .      .      .
```

```
Processing flaw map.
Processing of flaw map completed.
```

```
      .      .      .
      .      .      .
      .      .      .
```

If MAKE successfully processes the flaw map, it performs no further badspot checking. The system responds as if the `-BADSPOT_LEVEL 0` (`-BADLEV 0`) option had been used on the MAKE command line. When a valid flaw map is found and read, the default to `-BADLEV 0` considerably reduces the time necessary for MAKE to process a partition.

If MAKE cannot find a valid flaw map on the disk, MAKE displays a message similar to the following and continues processing.

```
Flaw map is unreadable. Continuing with MAKE.
```

You should use the `MODEL_4735`, the `MODEL_4845`, or the `MODEL_4860` argument with the `-DISK_TYPE` option when you are partitioning any of these disks. Using these arguments ensures that MAKE includes all badspots in the `BADSPT` or the `DBS` file. If MAKE is instructed to do the badspot checking on these disks, or if MAKE does the badspot checking by default, MAKE may miss marginally defective areas that are listed in the flaw map.

Note

The Model 4735, the Model 4845, and the Model 4860 disks are the only disk types that presently support flaw map processing in this manner. Use the flaw maps for these disks.

15. Ignore the Flaw Map on Fixed-Media Disks That Have This Feature: Use the command-line option `-NO_FLAW_MAP` (abbreviation `-NOFLMP`) if you want to prevent MAKE from reading and processing the flaw map on the Model 4735, 4845, and 4860 FMDs. When you use the `-NOFLMP` option, MAKE does not attempt to find a flaw map on these disks, and thus no flaw map processing messages are displayed. You may want to use this option if it is known that the flaw map is bad or inaccessible.

MAKE also reverts to the default level of badspot checking for the disk involved in this case. For the Model 4735, 4845, and 4860 disks, the default level of badspot checking is 4 (`-BADLEV 4`). To override the default level of badspot checking, use the `-BADLEV` option, specifying the level of badspot checking you want. For example, if you do not want MAKE to read the flaw map and you want two levels of badspot checking, use a command line like this:

```
OK, MAKE -DISK 1060 -DT MODEL_4735 -NO_FLAW_MAP -BADLEV 2
Partition name? SYSCMD
```

You may use the `-BADLEV` option regardless of the disk type or the presence or absence of flaw maps. `MAKE` defaults to a badspot-checking level of 0 if flaw maps exist. If flaw maps do not exist, if they cannot be read, or if you use the `-NOFLMP` option, `MAKE` defaults to a badspot-checking level of 4 for FMDs. If there are flaw maps on the disk and you do not specify the `-NOFLMP` option but you do specify the `-BADLEV` option, `MAKE` reads the flaw map and does the badspot checking specified. Table 5-4 summarizes these combinations:

- Whether or not flaw maps exist
- Whether or not you specify the `-BADSPOT_LEVEL` option
- Whether or not you specify the `-FLAW_MAP` option

TABLE 5-4. Use of Options With Model 4735 and Model 4845 Disks

<i>Flaw Map Exists</i>	<i>Specify -BADLEV</i>	<i>Specify -NOFLMP</i>	<i>Result</i>
Yes	No	No	Read the flaw map No badspot checking performed
Yes	Yes	No	Read the flaw map Check badspots
Yes	No	Yes	Default level of badspot checking
Yes	Yes	Yes	Check badspots to level <code>-BADLEV n</code>
No	Yes	No	Check badspots to level <code>-BADLEV n</code>
No	No	Yes	Default level of badspot checking

16. Set the Method of File Record Allocation: At Rev. 21.0 and beyond, PRIMOS can use either of two methods of file record allocation: reverse sectoring with an interleave factor of 1 or forward sectoring with an interleave factor of 3. (See Chapter 10 for an explanation.) The method of record allocation for Rev. 20.0, Rev. 21.0, and Rev. 22.0 partitions and Rev. 22.1 standard (nonrobust) partitions depends on the combination of type of CPU and type of disk controller in your system. `MAKE` determines this combination and sets a bit in the `DSKRAT` to indicate the direction of record allocation and the interleave factor.

Note

The method of file allocation on Rev. 20.0 partitions can be set and changed only if Rev. 21.0 or later utilities (MAKE and FIX_DISK) are used on them and only if they are run under Rev. 21.0 or later PRIMOS.

Table 10-2 in Chapter 10 presents the record allocation schemes based on the combination of the type of disk controller and the type of CPU in your system. Basically, the method is forward sectoring with an interleave factor of 3 for systems with a nonintelligent disk controller and a CPU in the 9950™ class. The method is reverse sectoring with an interleave factor of 1 for all other combinations of CPU and disk controller.

Note

CPUs in the 9950 class consist of the 4000 and 6000 series CPUs, the 2850™, the 2950™, the 9755™, and those with model numbers numerically equal to or larger than 9950, such as a 9955™. All other CPUs have model numbers numerically smaller than 9950.

You can change these recommended methods. To set the allocation method to forward, use the **-OVERRIDE_DEFAULT_INTERLEAVE** option (abbreviation **-ODI**). The MAKE command format is

MAKE -DISK pdev -DT type -OVERRIDE_DEFAULT_INTERLEAVE

Use the **-RESTORE_DEFAULT_INTERLEAVE** option (abbreviation **-RDI**) to set the allocation method to reverse. The MAKE command format is

MAKE -DISK pdev -DT type -RESTORE_DEFAULT_INTERLEAVE

The options and their associated methods of sectoring then are

<i>Option</i>	<i>Method of Sectoring</i>
-ODI	Forward
-RDI	Reverse

If you are running MAKE.SAVE standalone, MAKE prompts you for the method of record allocation with the following query if you are creating a standard partition and have not specified the method of file record allocation with either the -ODI or the -RDI options.

```
Which file sectoring scheme would you like?
Type "0" (Reverse Sector)
  or "1" (Forward Sector)
>
```

Enter either 0 or 1 at the angle-bracket (>) prompt.

When you create robust partitions by using the MAKE_ROBUST utility, the method of file allocation is set by MAKE_ROBUST to forward. No reverse sectoring takes place on a robust partition.

17. Name the Partition: Use the **-PARTITION** option (abbreviation **-PART**) to specify the partition name. The partition name may contain a maximum of six characters. The first character must not be a digit or a dash (-); the name can contain only alphabetic characters, digits, and the special characters `_ # $ & * - . /`. For example:

```
OK, MAKE -DISK 21220 -PARTITION SYS_24
```

The partition name serves as the name of both the partition and the file containing the DSKRAT. Make sure that each partition has its own unique partition name.

You may change the partition name later by using the **-RENAME** option of either the **ADDISK** or of the **SHUTDN** command.

If you do not specify the **-PARTITION** option on the command line or if you specify an invalid partition name, **MAKE** prompts you for a partition name:

```
Invalid partition name.  
Partition name?
```

18. Enter Known Badspots: If you are creating a pre-Rev. 21.0 partition or you are creating a Rev. 21.0 or later partition in Nondynamic Badspot Handling (**-AC**) mode, **MAKE** assembles a list of badspots on the partition as part of its verification procedure and places the list in the badspot file, **BADSPT**. If you are creating a Rev. 21.0 or later partition in Dynamic Badspot Handling (**-IC**) mode, **MAKE** puts the list in the **DBS** file. However, you may already be aware of badspots on the disk.

On some disks, badspot locations are recorded in a printed list physically attached to the disk. Such a list is referred to as a **flaw map** (see Badspots in Chapter 2). Look for a list of pairs of numbers representing the track and the head, or surface, of those parts of the disk determined by the disk manufacturer to be probable badspots. Some other disks list track, head, and sector numbers of badspots. (Some disks — for example, the Model 4735 — have flaw maps recorded on the disk medium that are read by **MAKE**. In that case this discussion does not apply. See item 14.)

If you intend to enter known badspots, you must include the **-QUERY_BADSPOTS** option (abbreviation **-QBADS**) on the command line. For example:

```
OK, MAKE -DISK 1060 -PART GEODE -DT 600MB -QUERY_BADSPOTS
```

When you invoke **MAKE** with the **-QUERY_BADSPOTS** option, you are asked to identify the locations of the badspots. A later section, Recording Badspots, shows you how.

19. Create a Robust Partition: At Rev. 22.1, if you use the **-ROBUST** option (abbreviation **-ROB**), **MAKE** issues this messages and aborts.

```
OK, MAKE -DISK 462 -DT SMD -PART FALCON -ROBUST  
**** MAKE <Rev. 22.1> **** Copyright (c) 1988, Prime Computer, Inc.  
See the Operator's Guide to File System Maintenance for  
information on how to create a robust partition.  
For options type: MAKE -HELP (MAKE)  
ER!
```

If you wish to use robust partitions at your site, you must first read Chapter 7. Chapter 7 describes and discusses robust partitions and contains a Robust Partition Evaluation Form that you should fill out. In addition, there are directions on how you can obtain and use the MAKE_ROBUST utility to convert newly-created Rev. 22.1 partitions to robust partitions.

20. Split the Disk: If you are creating a partition that is to be used either for paging or for both paging and storage of user files, specify the **-SPLIT** option. MAKE displays the total number of available records and then prompts you for the number of records to be used for paging. Both of these numbers are decimal. For example:

```
OK, MAKE -DISK 20021 -PARTITION PAGING -SPLIT
**** MAKE <Rev. 22.1> **** Copyright (c) 1988,
Prime Computer, Inc.
```

```
Total decimal number of records available:      7407
```

```
7376
```

The number of records to be used for paging should be evenly divisible by 16. If it is not divisible by 16, any remaining records are not used either for paging storage or for file system storage. Use the MOD function to determine the remainder of dividing by 16. (See the *PRIMOS Commands Reference Guide*.) For example, using the functions TYPE [MOD 7407 16] gives a remainder of 15. If you subtract 15 from 7407, the result is 7392, which is evenly divisible by 16. If you subtract another 16 from this, you have 7376 records for paging and 31 records for the file system on a one-surface SMD.

Table 5-5 shows the minimum number of records to be used for the file system portion of the split partition if it is the first partition of a physical disk. If it is not the first partition or if the physical disk is not listed in Table 5-5, leave 16 to 32 records for the file system portion.

If the paging partition is the first partition of the physical disk and you have not specified enough records for the DBS file and the RMA, MAKE aborts after displaying a message that tells you the minimum number of records that must be in the file system portion of the partition in order to have room for the RMA. Use this number for the number of records in the file system portion or use the numbers in Table 5-5.

The message displayed is

```
Insufficient size for minimum file system. Leave at least nn records.
```

See the *System Administrator's Guide, Volume I: System Configuration* for a discussion of determining the amount of paging space needed.

If you specify an invalid number for the paging records — for example, if you specify more records than are in the partition — MAKE tells you and prompts you again for the number of paging records. If you enter a non-numeric character for the number of paging records, MAKE displays a question mark (?) and awaits proper input. All preceding input for the number of paging records is lost so you must enter the complete decimal number.

Operator's Guide to File System Maintenance

If you do not and instead press or , MAKE does not split the partition. If you attempt to use this partition for paging later, you are queried to be sure that you want to page on this partition.

If you know how many records are to be used for paging and you do not want to be prompted for this information, specify the number of records to be used for paging following the -SPLIT option on the command line.

For example:

```
OK, MAKE -DISK 20061 -PARTITION PAGING -SPLIT 7376
```

TABLE 5-5. *Recommended Total File System Records
for Split First Partition*

Disk Type	Records for			Total
	DBS File	RMA	Other	
80MB SMD	1	32	11	44
300MB SMD	1	32	17	50
68MB	1	100	10	111
158MB	2	200	13	215
160MB	2	200	13	215
600MB	6	1500	27	1533
MODEL_4475	2	400	17	419
MODEL_4735	3	750	23	776
MODEL_4845	4	1000	31	1035
MODEL_4860	4	1000	32	1036

Note

The above numbers represent the minimum number of records that should be set aside to accommodate the DBS file, the RMA, the DSKRAT, BOOT, the MFD and the BADSPT file.

If you do not specify the -SPLIT option on the command line, MAKE does not set aside room on the disk for the badspot file. In addition, you will not be able to add the partition to the system with the ADDISK command, if this is desired, because only the file system portion of a split disk can be added. If you do not specify the -SPLIT option on the command line and later specify this partition as a paging partition by including its pdev with the PAGING directive, PRIMOS queries you at cold start as follows:

```
PAGDEV pdev does not point to the beginning of a valid file system partition.
```

```
Are you SURE you want to page on PAGDEV pdev?
```

If you answer NO, PRIMOS prompts for another paging device. If you answer YES, you are not prompted again when the system is cold started. However, if you intend to use a partition as a paging partition, use the -SPLIT option.

Note

The file system portion of a split disk should be used only for storing information on badspots on the disk. The system suffers a performance penalty if you use the file system portion of a split disk for user file storage, particularly when the paging portion of the disk is used heavily.

This rule applies to all systems, with the possible exceptions of the 2350, 2450, 2455, and 2250™ systems. On these systems, split disks with heavy user and paging use are often necessary, particularly on those systems that have only one 60MB or 68MB disk.

RECORDING BADSPOTS

When you invoke MAKE with the `-QUERY_BADSPOTS` option, MAKE prompts you to enter the locations of badspots on the disk. You then select one of the following two formats for entering badspots.

<i>Format of Badspot Information</i>	<i>Number System</i>	<i>Offset From</i>
Track, head, sector	Decimal	Beginning of physical disk
Record number	Octal	Beginning of partition

In most cases you will probably prefer to use the track, head, and sector format. Here, head zero corresponds to head zero of the physical disk, not necessarily to the first head, or surface, of the partition being made. (See Figure 3-1 in Chapter 3.) This correspondence allows you to input known badspots anywhere on the physical disk, not just in the partition currently being created. Both formats are described below.

Entering Badspots by Record Number

If you choose to enter badspots by record number, answer YES when MAKE prompts

Enter badspots by record number?

In this case, you enter each record number in octal, as offset from the start of the partition. MAKE continues prompting for record numbers with the `Record =` prompt until you enter a record number of zero to indicate that you are done. See the example later in this chapter.

Entering Badspots by Track, Head, and Sector

Answer NO to the Enter badspots by record number? prompt. MAKE repeatedly prompts for three values: Track, Head, and Sector. (If the -DISK_REVISION 18 option is included on the command line, MAKE does not prompt for Sector, and the entire track is considered a badspot.)

Enter the CYL (cylinder) number from the flaw map when MAKE prompts for Track. Enter the Head number from the flaw map when MAKE prompts for it. If the flaw map lists sector numbers as byte offsets, use Table 5-6 below to convert the byte offset from the flaw map to the Sector number; for example, byte number 9987 is in Sector 4. If sector numbers are listed as decimal numbers, enter them as they are listed.

To terminate the dialog, enter 0 at the Track =, Head =, and Sector = prompts. MAKE then prompts

Verify data?

If you answer YES, MAKE displays a list of the badspot track, head, and sector numbers that you entered.

TABLE 5-6. Conversion of Byte Numbers to Sectors

<i>Sector</i>	<i>Byte Numbers</i>
0	0000 - 2239
1	2240 - 4479
2	4480 - 6719
3	6720 - 8959
4	8960 - 11199
5	11200 - 13439
6	13440 - 15679
7	15680 - 17919
8	17920 - 20159

Note

The preceding conversion table is not applicable to the following FMDs. These disks have flaw maps written on the disks' surfaces that are handled by the disk subsystem. Badspots found by MAKE are written to the badspot file. You may enter any new badspots that are noted by PRIMOS or other utilities on these disks.

Model 4711	Model 4721
Model 4714	Model 4735
Model 4715	Model 4845
Model 4719	Model 4860

After Known Badspots Have Been Entered and Verified

At this point, MAKE asks Parameters OK? Check the list of badspot locations displayed by MAKE (assuming that you have requested the list by answering YES to the Verify data? prompt). If the list is correct, type YES. MAKE includes this information in the badspot file, which appears in the MFD with the filename BADSPT, or in the DBS file. MAKE also updates the DSKRAT file to ensure that badspot records are not available for file system use.

If the list is incorrect, a NO answer returns you to the Enter badspots by record number? prompt. Enter the badspots again or enter 0 to terminate the dialog.

Keeping Redundant Badspot Files

As a result of a trend toward higher data storage densities, newer varieties of disks tend to be more susceptible to badspots. For example, a 675MB disk might have as many as 1200 badspots. To guard against loss of badspot information, MAKE allows you to maintain redundant badspot files for a physical disk with multiple partitions. MAKE supports redundant badspot files by allowing the badspot file on each partition to contain information about all the badspots on the disk, regardless of the partitions the badspots are in. If you have a downloaded intelligent disk controller, all badspots on the physical disk are kept in the DBS file on the first partition of Rev. 21.0 and later disks. Use the following procedure only on pre-Rev. 21.0 disks or Rev. 21.0 and later disks in -AC mode.

Procedure for Copying Badspot Files: It is advisable to run MAKE on an entire physical disk before allowing users to access and store data on any partition of the disk. To do so, follow the steps shown next.

1. Make a list of the physical device numbers for all partitions that you will create on the physical disk.
2. Create the first partition, entering the track, head, and sector values of all badspots from the flaw map. (Typically, MAKE is run on a disk starting with the partition at surface offset zero (the first partition), but pre-Rev. 21.0 partitions can be made in any order. You must create the first partition first when making Rev. 21.0 and later partitions.)
3. Create the second partition on your list. This time, you must include the -COPY_BADSPOTS_BY_DEVICE option on the command line in order to copy the badspot information from the first partition.
4. Create the third partition, including -COPY_BADSPOTS_BY_DEVICE in order to copy information from the second partition, and so on, always using the most recently created badspot file.

MAKE may find additional badspots as it creates the additional partitions. For example, if MAKE finds an additional badspot on the third partition of a five-partition disk, the badspot files for partitions 3, 4, and 5 will then contain the entry for that new badspot.

Operator's Guide to File System Maintenance

The badspot files for partitions 1 and 2, however, will lack information on the new badspot. If MAKE subsequently finds a badspot on partition 5, only the badspot file on partition 5 will be complete.

Once all partitions on the disk have been made, but *before* any users have been allowed to access the disk, you can update the badspot files on the earlier partitions. To do this, remake the earlier partitions using the `-COPY_BADSPOTS_BY_DEVICE` option. To minimize the time needed, use the `-BADSPOT_LEVEL 0` option because badspots have already been found and recorded for the partitions you are remaking.

Example of Copying Badspot Files: To make a 675MB disk on drive unit 2 of the controller at address 26₈ as a four-partition disk with 10 surfaces per partition, construct a list of physical device numbers such as the following. (See Chapter 3 for details.)

<i>pdev</i>	<i>Starting Surface</i>	<i>Number of Surfaces</i>
2464	0	10
52464	10	10
122464	20	10
172464	30	10

Then assign these four partitions and use MAKE to create them using the following command lines. (Command lines using the option abbreviations are shown.)

```
MAKE -DSK 002464 -PART FALCON -NEWDSK -FMT -DT 600MB -QBADS
MAKE -DSK 052464 -PART CORVID -NEWDSK -FMT -DT 600MB -CPYDEV 2464
MAKE -DSK 122464 -PART WILLET -NEWDSK -FMT -DT 600MB -CPYDEV 52464
MAKE -DSK 172464 -PART PLOVER -NEWDSK -FMT -DT 600MB -CPYDEV 122464
```

Now remake partition 2464, copying the badspot file from the last device, 172464. Use the `-BADLEV 0` option in remaking these three partitions because MAKE has already found the badspots on these partitions. Using the `-BADLEV` option saves some MAKE processing time.

```
OK, MAKE -DSK 2464 -BADLEV 0 -CPYDEV 172464 -DT 600MB
Partition name? FALCON
```

Similarly, remake partition 52464 copying the badspot file from the last device, 172464.

```
OK, MAKE -DSK 52464 -BADLEV 0 -CPYDEV 172464 -DT 600MB
Partition name? CORVID
```

Finally, remake partition 122464, again copying the badspot file from the last device, 172464.

```
OK, MAKE -DSK 122464 -BADLEV 0 -CPYDEV 172464 -DT 600MB
Partition name? WILLET
```

WHAT TO DO BEFORE RUNNING MAKE

Take the following actions before you run MAKE:

1. Ensure that the new partition contains no important data that is not already backed up onto another disk or onto magnetic tape. When you create a partition, MAKE deletes all existing data on that partition.
2. Determine the physical device number (pdev) of the logical disk, or partition, that is to be created. Chapter 3 explains how to determine this number. The physical device number tells the system the disk drive unit on which the disk is mounted, the address of the controller to which the drive unit is connected, and, for partitions, the size of the partition and its location on the physical disk.
3. Then, use the following procedure, which is shown schematically in Figure 5-1:
 - a. If you are creating a partition that replaces existing partitions on the disk, go to the supervisor terminal and shut down the partitions that are to be replaced. For example, assume two partitions are being replaced:

```
OK, SHUTDN 460
OK, SHUTDN 10460
OK,
```

The above example shuts down the first two-surface partition with a surface offset of 0 and the second two-surface partition with a surface offset of 2, allowing a new four-surface partition, which has a pdev of 1060, to be created in their place.

- b. Add the new, four-surface partition you are creating to the assignable disks table by using the DISKS command at the supervisor terminal:

```
OK, DISKS 1060
```

- c. Assign the partition to be created to your terminal. Use the ASSIGN DISK command with the pdev:

```
OK, ASSIGN DISK 1060
```

- d. Now run MAKE with the appropriate options.

RUNNING MAKE

You invoke MAKE with a number of options and arguments on the command line in order to specify the details of how the partition is to be created. MAKE *requires* that you specify three options and their arguments in order to obtain the following information:

- The physical device number of the partition to create (-DISK *pdev*)
- The type of physical disk to partition (-DISK__TYPE *type*)
- The name to give to the newly created partition (-PARTITION *diskname*)

Note

MAKE cannot run under PRIMOS II (DOS). Attempting to use Rev. 22.1 MAKE under PRIMOS II results in the error message T00 BIG.

Invoking MAKE

After you determine the options you wish to specify, invoke MAKE as follows:

MAKE -DISK *pdev* -PARTITION *name* -DISK_TYPE *type* [*options*]

In order to create a partition, you must include -DISK on the command line. If you do not include the *pdev*, MAKE prompts you for it. You can determine the physical device number, *pdev*, from the information in Chapter 3.

If you do not include the -PARTITION option, MAKE prompts you for the name. *name* is the name that you assign to the partition and consists of as many as six alphanumeric characters.

If you do not include the -DISK_TYPE option, MAKE displays a list of valid disk types and prompts you for one. *type* is one of the valid types in Table 5-2 under item 2.

The physical disk you specify with the -DISK_TYPE option is partitioned as directed by the -DISK and -PARTITION options and by the other options you specify on the command line. If you specify an option more than once on the command line, the following message is displayed and MAKE aborts:

```
An option appeared more than once (MAKE)
ER!
```

When MAKE Finishes

MAKE signals its completion by displaying

```
Done checking for badspots.
  nn lost records.
Disk created.
OK,
```

MAKE then returns you to the PRIMOS OK, prompt. *nn* is the number of records recorded in the badspot file (and therefore "lost" to badspots). If no badspots exist or if you are creating a partition in -IC mode, the *nn* lost records. message is not displayed.

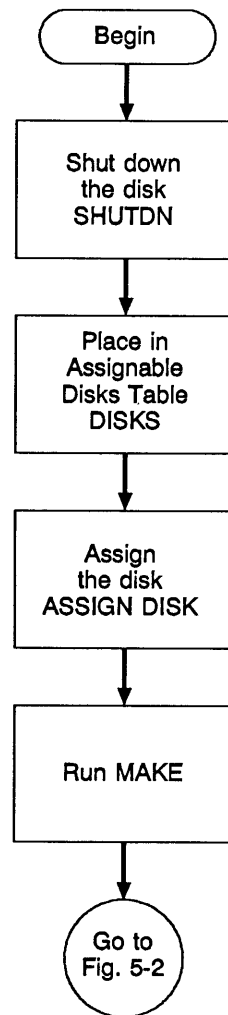


FIGURE 5-1. What to Do Before Running MAKE

WHAT TO DO AFTER RUNNING MAKE

After running MAKE, you should add the partition to the system so that it can be used. Then set any additional ACLs or the owner password for the partition. Follow these steps, which are shown in Figure 5-2:

1. Unassign the partition with the UNASSIGN DISK command, and remove it from the Assignable Disks Table with the DISKS NOT command:

```
UNASSIGN DISK pdev
DISKS NOT pdev
```

2. Add the partition to the file system by typing

```
ADDISK pdev
```

Use the same *pdev* in these commands as the one you constructed to create the partition.

3. Rev. 22.1 MAKE sets the ACLs SYSTEM:ALL, \$REST:LUR on the MFD. If you use the -DSKREV 18 option, MAKE creates a password directory. If you wish to change the ACLs or the password, use either the SET_ACCESS command or the EDIT_ACCESS command to change ACLs, or use the PASSWD command to set an owner password other than XXXXXX. To do either, attach to the MFD of the newly created partition as follows:

```
ATTACH <name>MFD [XXXXXXX]
```

where *name* is the name of the partition and the angle brackets (<>) are necessary in the command line. Use the password, XXXXXX, only if you created a password partition.

4. Issue either SET_ACCESS MFD or EDIT_ACCESS MFD to change ACLs or the PASSWD command to change the password. (If you use the PASSWD command, be sure to set the nonowner password to XXXXXX; otherwise AVAIL will not be able to size the partition.)

PRIMOS may be bootstrapped from this partition because MAKE writes the bootstrap program, BOOT, onto the MFD. If the partition is to be used only as a user partition, the empty directories CMDNC0 and DOS may be deleted from the newly created partition.

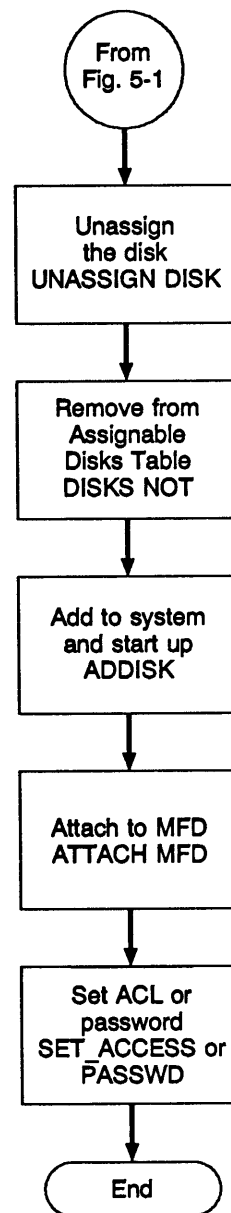


FIGURE 5-2. What to Do After Running MAKE

EXAMPLES OF RUNNING MAKE

The following examples show various uses of MAKE when PRIMOS is running. Examples of booting MAKE.SAVE from disk and from magnetic tape are presented at the end of this chapter under Booting MAKE.SAVE.

Example of Normal MAKE Display

In the first example, MAKE formats a new first partition that has four surfaces, with starting surface 0, and is on drive unit 1. The example shows the use of the -FORMAT and -NEW_DISK options.

```
OK, DISKS 1062
OK, ASSIGN DISK 1062
OK, MAKE -DISK 1062 -DISK_TYPE SMD -FORMAT -NEW-DISK
**** MAKE <Rev. 22.1> **** Copyright (c) 1988, Prime Computer, Inc.
Unrecognizable TOKEN: -NEW-DISK
For options type: MAKE -HELP (MAKE)
ERI MAKE -DISK 1062 -DISK_TYPE SMD -FORMAT -NEW_DISK
Partition name? GOLD
**** MAKE <Rev. 22.1> **** Copyright (c) 1988, Prime Computer, Inc.

Making      4 head partition GOLD
Tracks:                823
Sectors per track:      9
Partition is using Reverse Sectoring
Partition is in -Intelligent_Controller Mode
Partition size in decimal records:      29595

Beginning format.
Format completed.

Checking for bad spots.

Done checking for bad spots.
Disk created.
OK, UNASSIGN DISK 1062
OK, DISKS NOT 1062
OK,
```

The partition is first placed in the Assignable Disks Table and assigned by the Operator at the supervisor terminal. In the first command line, an error is made in entering the -NEW_DISK option and MAKE aborts. The command line is reentered correctly.

The method of file record allocation and the partition mode were not specified, so MAKE determines the CPU type and the disk controller type and sets the allocation method to reverse sectoring and the partition to Dynamic Badspot Handling (-IC) mode. These settings are based on a CPU in the 9950 class with an intelligent disk controller as indicated in Table 10-2 in Chapter 10. Note that the size of the partition in decimal records for a first partition is reduced by the sizes of the DBS file and the RMA, a total of 33 records for an SMD (see Table 5-5 under item 20).

After MAKE has formatted the partition and checked for badspots, the partition is unassigned and removed from the Assignable Disks Table.

Example of Remaking a First Partition

In this example, a new four-surface first partition is being created where a two surface partition existed previously on an -INTELLIGENT_CONTROLLER mode disk.

```
OK, DISKS 1062
OK, ASSIGN DISK 1062
OK, MAKE -DISK 1062 -DT MODEL_4475 -FORMAT -NEW-DISK
**** MAKE <Rev. 22.1> **** Copyright (c) 1988, Prime Computer, Inc.
Unrecognizable TOKEN: -NEW-DISK
For options type: MAKE -HELP (MAKE)
ERI MAKE -DISK 1062 -DT MODEL_4475 -FORMAT -NEW_DISK
**** MAKE <Rev. 22.1> **** Copyright (c) 1988, Prime Computer, Inc.
Partition name? GOLD

Making      4 head partition GOLD
Tracks:                823
Sectors per track:      9
Partition is using Reverse Sectoring
Partition is in -Intelligent_Controller Mode
Partition size in decimal records:      29226

Disk DBS file change! All other partitions on this disk
will be corrupted if you continue
Do you wish to continue? YES

Beginning format.
Format completed.

Checking for bad spots.
      6 lost records.
Done checking for bad spots.
Disk created.
OK, UNASSIGN DISK 1062
OK, DISKS NOT 1062
OK,
```

The method of file record allocation and the partition mode were not specified, so MAKE determines the CPU type and the disk controller type and sets the allocation method to reverse sectoring and the partition to Dynamic Badspot Handling (-IC) mode. These settings are based on a CPU in the 9950 class with an intelligent disk controller as indicated in Table 10-2 in Chapter 10. Note that the size of the partition in decimal records for a first partition is reduced by the sizes of the DBS file and the RMA, a total of 402 records for a Model 4475 disk (see Table 5-5 under item 20).

Because you are recreating an existing first partition, the geometry of the first partition will change, changing the DBS file and the RMA. As shown in the example, you are warned that any other partitions on the disk will lose their badspot data if you continue. If you answer NO at the Do you wish to continue? prompt, MAKE aborts. YES allows MAKE to continue. If there are other partitions on this disk, they will have to be recreated.

Operator's Guide to File System Maintenance

If you did not specify the -FORMAT option, MAKE would enable the -FORMAT option to change the DBS file and inform you as follows:

```
Disk DBS file changed! All other partitions on this disk
will be corrupted if you continue.
MAKE will enable -FORMAT to change the DBS file.
Do you wish to continue?
```

If you use the -AC option or the disk is created in Nondynamic Badspot Handling (-AC) mode by default, you are not queried.

Example of Inputting Known Badspots

This example shows the use of the -QUERY_BADSPOTS option and illustrates the input of known badspots and the verification of the input data.

```
OK, DISKS 20063
OK, ASSIGN DISK 20063
OK, MAKE -DISK 20063 -DT MODEL_4475 -NEWDISK -QUERY_BADSPOTS
**** MAKE <Rev. 22.1> **** Copyright (c) 1988, Prime Computer, Inc.
Partition name? SILVER
```

```
Making      1 head partition SILVER
Tracks:                823
Sectors per track:      9
Partition is using Reverse Sectoring
Partition is in -All_Controller Mode
Partition size in decimal records:      7407
```

```
Enter badspots by record number? YES
Record numbers are OCTAL.
Use a record number of 0 to end input.
Record = 456
Record = 567
Record = 4332
Record = 44556
Record = 443
Record = 0
Verify data? YES
Record = 00000000456
Record = 00000000567
Record = 00000004332
Record = 00000044556
Record = 00000000443
```

```
Parameters OK? YES
```

```
Checking for bad spots.
```

```
Done checking for bad spots.
  5 lost records.
```

```
Disk created.
```

```
OK, UNASSIGN DISK 20063
OK, DISKS NOT 20063
OK,
```

The partition is first placed in the Assignable Disks Table and assigned. At the end of the session, the partition is unassigned and removed from the Assignable Disks Table. After the known badspots are entered, MAKE checks for additional badspots. No more badspots are found and the only badspots are those entered.

The method of file record allocation and the partition mode were not specified, so MAKE determines the CPU type and the disk controller type and sets the allocation method to forward sectoring and the partition to Nondynamic Badspot Handling (-AC) mode. These settings are based on a CPU in the 9950 class with an nonintelligent disk controller as indicated in Table 10-2.

Example of Creating a Paging Partition

The following example illustrates the use of the MAKE command with the -SPLIT option to make a paging partition and with the -BADSPOT_LEVEL option in order to have MAKE use the highest level of badspot checking. If this partition is on an FMD, the same amount of badspot checking could be accomplished by using the -DISK_TYPE option because the default level for an FMD is 4.

First, the user determines the number of records available on the three surface partition from the information in Table 3-1 and by using the CALC function.

```
OK, TYPE [CALC 7407 * 3]
22221
```

Then the user determines the remainder of dividing the records available by 16 by using the MOD function, adds the remainder (13) to 16 to get a reasonable amount of records for the file system portion of the partition (29) and still have the paging records divisible by 16 ($22192/16 = 1387$), and then uses the CALC function to arrive at the number of records to use for paging (22192).

```
OK, TYPE [MOD 22221 16]
13
OK, TYPE [CALC 22221 - 29]
22192
OK,
```

The user then assigns the disk and invokes MAKE with the -SPLIT option and enters the number of records to be used for paging.

```
OK, DISKS 100463
OK, ASSIGN DISK 100463
OK, MAKE -DISK 100463 -DT MODEL_4475 -SPLIT -BADLEV 4
**** MAKE <Rev. 22.1> **** Copyright (c) 1988, Prime Computer, Inc.
Partition name? PAGER1
```

```
Total decimal number of records available:      22221
Decimal records for paging? 23192
```

```
Invalid number of paging records!
```

```
Total decimal number of records available:      22221
Decimal records for paging? 22192
```

Operator's Guide to File System Maintenance

```
Making      3 head partition PAGER1
Tracks:           823
Sectors per track: 9
Partition is using Reverse Sectoring
Partition is in -Intelligent_Controller Mode
```

```
File system records:      29
Paging records:          22192
```

Checking for bad spots.

Done checking for bad spots.

Disk created.

OK, UNASSIGN DISK 100463

OK, DISKS NOT 100463

OK,

The number of records to set aside for paging is mistyped resulting in an error because the number represents more records than are available on the partition. MAKE displays an error message and reprompts for the number of paging records. The number of paging records is divisible by 16.

The method of file record allocation and the partition mode were not specified, so MAKE determines the CPU type and the disk controller type and sets the allocation method to reverse sectoring and the partition to Dynamic Badspot Handling (-IC) mode. These settings are based on a CPU in the 9950 class with an disk controller as indicated in Table 10-2.

MAKE looks for an existing badspot file on the partition because the -NEW_DISK option was not used and this partition had not been formatted by MAKE previously. The badspot file cannot be found; MAKE displays a message and continues. MAKE checks for badspots and creates a new badspot file.

Example of Copying Badspots From Another Partition

The example below shows the use of the -COPY_BADSPOTS_BY_NAME option to copy the badspots from another partition of the same physical disk. The partition is created in nondynamic badspot handling mode (-AC) since the -CPYNAM option can be used only if the partition is in that mode. If the partition from which you are copying the badspot file is not an added partition, MAKE displays the following error message and continues

```
Partition not found: GOLD
```

```
Ignoring copy_badspots_by_name option. (MAKE)
```

If the partition from which you are copying the badspot file is a Dynamic Badspot Handling (-IC) mode partition, MAKE displays the following error message and continues

```
No badspot file on Partition: GOLD
```

```
Ignoring copy_badspots_by_name option. (MAKE)
```

The entire process is as in this example.

```

OK, DISKS 10422
OK, ASSIGN DISK 10422
OK, MAKE -DISK 10422 -DT MODEL_4715 -AC -COPY_BADSPOTS_BY_NAME -NEWDSK
**** MAKE <Rev. 22.1> **** Copyright (c) 1988, Prime Computer, Inc.
Partition name? HEART
Copy badspots from partition? GOLD

Making      2 head partition HEART
Tracks:           1020
Sectors per track:      7
Partition is using Reverse Sectoring
Partition is in -All_Controller Mode
Partition size in decimal records:      14280

Checking for bad spots.

Done checking for bad spots.
Disk created.
OK, DISKS NOT 10422
OK, UNASSIGN DISK 10422
OK,

```

The method of file record allocation was not specified, so MAKE determines the CPU type and the disk controller type and sets the allocation method to reverse sectoring. This is based on a CPU not in the class of the 9950 with a nonintelligent disk controller as indicated in Table 10-2.

BOOTING MAKE.SAVE

Rev. 22.1 MAKE will run under PRIMOS or it may be booted to run by itself without PRIMOS. The latter mode of operation, sometimes referred to as running **standalone**, is useful for creating new systems or for recovering from halts.

MAKE is run standalone by giving the Rev. 22.1 system boot the pathname of MAKE when the system boot prompts for the runfile pathname. The pathname of Rev. 22.1 MAKE is CMDNCO>MAKE.SAVE. MAKE may be booted from disk or from MAGSAV-format magnetic tape.

Booting MAKE.SAVE From Disk

When MAKE is run under PRIMOS, command-line options are typed immediately following the MAKE command and on the same command line. When MAKE is run standalone, MAKE prompts you for the command-line options because the boot does not accept command-line options following the pathname. When MAKE is running standalone, the CPU halts when MAKE has finished. See your CPU handbook for an explanation of the BOOT command and its options.

Note

MAKE cannot run under PRIMOS II (DOS). Attempting to use Rev. 22.1 MAKE under PRIMOS II results in the error message T00 BIG.

The following example illustrates booting Rev. 22.1 MAKE from disk. In this example, the COMDEV is created by MAKE, the partition is split with 4992 records allocated to paging with the -SPLIT option, and four surfaces are used for the new partition. MAKE is booted from the second disk drive (disk unit 1) to create a partition on disk drive unit 0.

In the BOOT 10314 command, the 10000_g switch is set as MAKE is booted from the partition on drive 1 and controller 0, the same device that the boot is read from. Because the 4000_g switch is not set, the RUN FILE TREENAME= prompt is displayed and you enter the pathname of MAKE. After MAKE starts, you are prompted for command-line options. You must enter the -DISK argument with the pdev of the partition you are creating. You then enter the appropriate options to instruct MAKE on how you want the partition created. If you do not enter the -PARTITION option, MAKE prompts you for the partition name. If you do not enter the -DISK__TYPE option, MAKE displays a list of disk types and prompts you for the disk type.

```
CP>SYSCLR
DPM006: Central Processor System initialization completed.
      26 May 89 09:15:14 Friday
CP> BOOT 10314
DPM007: System Booting, please wait.
[CPBOOT Rev. 10.2 Copyright (c) 1987, Prime Computer, Inc.]
[BOOT Rev. 22.1 Copyright (c) 1989, Prime Computer, Inc.]

RUN FILE TREENAME=CMDNCO>MAKE.SAVE

BOOTING FROM 001062 CMDNCO>MAKE.SAVE

**** MAKE <Rev. 22.1> **** Copyright (c) 1988, Prime Computer, Inc.
Enter command line options: -DISK 1060 -DT MODEL_4711 -FORMAT -SPLIT 4992
-NEWDISK -INTELLIGENT_CONTROLLER
Partition name? SYSA29
Cannot MAKE a -IC mode disk running standalone!

DPM400: CPU halted at 036354: 103775
      26 May 89 09:17:34
CP>
```

Note that you cannot use the -IC option when you run MAKE standalone. If you attempt to create a disk in Dynamic Badspot Handling (-IC) mode when running MAKE standalone, MAKE informs you that you cannot and aborts. You must then start over. When running standalone, MAKE also prompts for the type of record allocation scheme you want.

The reason MAKE cannot create an -IC mode partition and must prompt for sectoring information is that an intelligent disk controller cannot be downloaded until PRIMOS is booted; thus MAKE cannot determine the type of controller in your system and set the mode and record allocation properly.

You can specify the `-ALL_CONTROLLER` option when running MAKE standalone; it is unnecessary, however, because MAKE defaults to Nondynamic Badspot Handling (`-AC`) mode when running standalone.

The example below shows the correct procedure for running MAKE standalone.

```
CP>SYSCLR
DPM006: Central Processor System initialization completed.
      26 May 89 09:18:14 Friday
CP> BOOT 10314
DPM007: System Booting, please wait.
[CPBOOT Rev. 10.2 Copyright (c) 1987, Prime Computer, Inc.]
[BOOT Rev. 22.1 Copyright (c) 1989, Prime Computer, Inc.]

RUN FILE TREENAME=CMDNCO>MAKE.SAVE

BOOTING FROM 001062 CMDNCO>MAKE.SAVE

**** MAKE <Rev. 22.1> **** Copyright (c) 1988, Prime Computer, Inc.
Enter command line options: -DISK 1060 -DT MODEL_4711 -FORMAT -SPLIT -NEWDSK
Partition name? SYSA29

Total decimal number of records available:      28560
Decimal records for paging? 4992

Which file sectoring scheme would you like?
Type "0" (Reverse Sector)
   or "1" (Forward Sector)
> 1
Making      4 head partition SYSA29
Tracks:                1020
Sectors per track:      7
Partition is using Forward Sectoring
Partition is in -All_Controller Mode

File system records:      23568
Paging records:          4992

Disk DBS file change! All other partitions on this disk
will be corrupted if you continue.
Do you wish to continue? YES

Beginning format.
Format completed.

Checking for badspots.

Done checking for badspots.
Disk created.

DPM400: CPU halted at 036354: 103775
      26 May 89 09:23:24 Friday

CP>
```

The CPU halts when MAKE has finished. For further information on the Rev. 22.1 bootstrap procedure, see the handbook for your CPU.

Booting MAKE.SAVE From Magnetic Tape

The following example illustrates booting MAKE from magnetic tape to create a system. In the example, the COMDEV is the first four surfaces of an 80MB disk pack and the PAGDEV is the last surface of the same disk pack.

Note

See your CPU handbook for an explanation of the BOOT command and its options.

First, the Operator boots CMDNCO>MAKE.SAVE from tape drive 1 on controller 0 to create and format the command device, or COMDEV. The pathname of MAKE must be entered exactly as it was saved by MAGSAV. For clarity in the example, the -PARTITION option is not entered on the command line. MAKE then prompts for the partition name. It takes a few minutes to format the partition and a few more minutes per surface in the partition to check for badspots. When MAKE finishes, the CPU halts.

```
CP> SYSCLR
DPM006: Central Processor system initialization completed.
      26 May 89 09:08:45 Friday
CP> BOOT 10205
DPM007: System booting, please wait.
[CPBOOT Rev. 10.2 Copyright (c) 1987, Prime Computer, Inc.]
[BOOT Rev. 22.1 Copyright (c) 1989, Prime Computer, Inc.]

RUN FILE TREENAME=CMDNCO>MAKE.SAVE

BOOTING FROM MT1      CMDNCO>MAKE.SAVE

**** MAKE <Rev. 22.1> **** Copyright (c) 1988, Prime Computer, Inc.
Enter command line options: -DISK 1060 -DT SMD -AC -BAUD 1200 -FMT
Partition name? BI0101
Which file sectoring scheme would you like?
Type "0" (Reverse Sector)
      or "1" (Forward Sector)
> 1
Making      4 head partition BI0101
Tracks              823
Sectors per track:      9
Partition is using Forward Sectoring
Partition is in -All_Controller Mode
Partition size in decimal records:      29628

Unable to get badspots from device: 001060
Continuing with make. (MAKE)

Beginning format.
Format completed.

Checking for bad spots.

Done checking for bad spots.
Disk created.

DPM400: CPU halted at 025175: 103775
      26 May 89 09:27:36 Friday
CP>
```

Note that you cannot use the -IC option when booting MAKE. In the example the -AC option was used although it is unnecessary as MAKE defaults to Nondynamic Badspot Handling (-AC) mode.

Next the Operator again boots MAKE.SAVE to create the paging device. MAKE writes the file MFD>BOOT on the disk as part of its tasks. The CPU again halts when MAKE finishes.

```
CP> SYSCLR
DPM006: Central Processor system initialization completed.
      26 May 89 09:28:52 Friday
CP> BOOT 10205
DPM007: System booting, please wait.

[CPBOOT Rev. 10.2 Copyright (c) 1987, Prime Computer, Inc.]
[BOOT Rev. 22.1 Copyright (c) 1989, Prime Computer, Inc.]

RUN FILE TREENAME=CMDNCO>MAKE.SAVE

BOOTING FROM MT1    CMDNCO>MAKE.SAVE

**** MAKE <Rev. 22.1> **** Copyright (c) 1988, Prime Computer, Inc.
Enter command line options: -DISK 20061 -DT SMD -SPLIT 7376 -FMT -NEWDSK
Partition name? PAGDEV
Which file sectoring scheme would you like?
Type "0" (Reverse Sector)
  or "1" (Forward Sector)
> 1
Making    1 head partition PAGDEV
Tracks                823
Sectors per track:    9
Partition is using Forward Sectoring
Partition is in -All_Controller Mode

File system records:      31
Paging records:          7376

Beginning format.
Format completed.

Checking for bad spots.

Done checking for bad spots.
Disk created.

DPM400: CPU halted at 025175: 103775
      26 May 89 09:34:43 Friday

CP>
```

See your CPU handbook for further information and examples of booting PRIMOS and MAKE and other utilities from magnetic tape.

REPAIRING PARTITIONS – FIX_DISK

This chapter discusses the FIX_DISK command, the PRIMOS utility for repairing disk partitions. The chapter presents an itemized list of the uses of FIX_DISK, discusses all of the FIX_DISK command-line options, and shows examples of running FIX_DISK.

You should use FIX_DISK under PRIMOS to repair disks of any revision. Rev. 22.1 FIX_DISK can repair Rev. 20.0 and later disk partitions, which have hashed directories, as well as pre-Rev. 20.0 disk partitions, which do not have hashed directories.

Note

It is recommended that you delete any pre-Rev. 22.1 versions of FIX_DISK. This is to avoid the possibility of mistakenly invoking a pre-Rev. 22.1 version and thereby getting unexpected results. The Rev. 22.1 version of FIX_DISK resides in CMDNC0 as FIX_DISK.SAVE. When you invoke FIX_DISK, a copyright line with the revision stamp is displayed.

WHAT IS FIX_DISK?

FIX_DISK is an Operator command that

- Reads every physical record that is in use on a disk or partition, including records in files, directories, and segment directories
- Checks the quota information on partitions
- Checks that the information in each record header is consistent with the directory that contains the record
- Checks the DSKRAT file for discrepancies
- Checks ACLs
- Checks file system pointers

When any error is identified on a partition, an appropriate error message is displayed. See Appendix B, FIX_DISK Messages, for an explanation of all FIX_DISK messages.

An important feature of `FIX_DISK` is its repair facility. `FIX_DISK` can

- Repair mismatched pointers
- Correct quota information
- Either replace defective records with new empty records or truncate files containing defective records
- Delete defective files
- Rebuild a defective `DSKRAT` file

`FIX_DISK` has other uses as well. It can

- Add new badspot information
- Convert disks made as pre-Rev. 21.0 partitions to Rev. 19.0, Rev. 20.0, or Rev. 21.0 partitions
- Convert disks made as Rev. 22.0 partitions to Rev. 22.1 standard partitions
- Change the maximum and minimum CAM file extent sizes set on Rev. 22.0 and subsequent revision partitions
- Remap equivalence blocks created by `COPY_DISK` or by `PHYRST`
- Check a partition to see whether it needs repair
- Rapidly repair robust partitions containing CAM files
- Determine if a partition was shut down properly and, if it was not, warn you that you should run `FIX_DISK`
- Enable dynamic badspot handling so that partitions may be mirrored
- Display the dynamic badspot (DBS) file
- Change the method of file record allocation

HOW DOES `FIX_DISK` WORK?

`FIX_DISK` performs these tasks, as described in the following paragraphs, when you command it to repair partitions by using the `-FIX` option.

- Locates `DSKRAT` errors
- Locates quota errors
- Handles the `BADSPT` file or the `DBS` file and the `RMA`
- Converts pre-Rev. 21.0 partitions to Rev. 21.0, pre-Rev. 20.0 partitions to Rev. 20.0, or pre-Rev. 19.0 partitions to Rev. 19.0
- Converts Rev. 22.0 partitions to Rev. 22.1 standard partitions

Locating DSKRAT Errors

In order to read each physical record currently in use, FIX_DISK must traverse the entire logical file structure of a partition. While traversing, FIX_DISK creates its own **Record Availability Table (RAT)**, which is updated and checked against the existing Disk Record Availability Table (DSKRAT) for each record. If FIX_DISK notes a discrepancy in comparing its own RAT to the DSKRAT, an error message is displayed at the supervisor terminal.

Locating Quota Errors

On Rev. 19.0 and subsequent revision partitions (on which the quota information is maintained), FIX_DISK compares the count of records used for directories, subdirectories, and files against the actual number of records within the directory. An error message is generated if there is an error in the directory's count of records that have been used.

Converting a Pre-Rev. 21.0 Partition

FIX_DISK can convert a pre-Rev. 21.0 partition

- From Rev. 18.0 to either Rev. 19.0, Rev. 20.0, or Rev. 21.0
- From Rev. 19.0 to Rev. 20.0 or Rev. 21.0
- From Rev. 20.0 to Rev. 21.0.

The options that accomplish these conversions are -CONVERT_19, -CONVERT_20 and -CONVERT_21.

To convert a partition, FIX_DISK does these three things:

- Initializes the quota information
- Upgrades the current badspot file to the proper revision format
- Creates the proper revision stamp in the DSKRAT

When you use FIX_DISK to convert a pre-Rev. 20.0 partition to Rev. 20.0 or to Rev. 21.0, existing directories do not become hashed. Only ACL directories created *after* the conversion will be hashed directories. The MFD cannot be converted to a hashed directory by FIX_DISK. For these and other reasons, you should use MAKE to convert pre-Rev. 20.0 partitions to Rev. 21.0 and you must use MAKE to convert pre-Rev. 22.0 partitions to Rev. 22.1.

Converting to Rev. 21.0 also involves selecting Dynamic Badspot Handling (-IC) mode or Nondynamic Badspot Handling (-AC) mode, changing the DSKRAT to Rev. 21.0 format, and updating the revision stamp. However, remember that, since the MFD cannot be converted to a hashed directory by FIX_DISK and other directories do not become hashed without further steps, it is recommended that you use MAKE and convert pre-Rev. 22.0 partitions to Rev. 22.1, following the steps discussed in Chapter 5, Formatting Disks — MAKE.

Since Rev. 21.0 partitions have the same format as Rev. 22.1 standard partitions, it is not necessary to convert Rev. 21.0 partitions to Rev. 22.1 format unless you wish to use robust partitions, unless you wish to have unlimited CAM file extents, or unless you wish to bring the DSKRAT header information and the revision stamp on the Rev. 21.0 partition to Rev. 22.1. If you want to use robust partitions, you must use Rev. 22.1 MAKE to convert partitions. (See Chapter 7 for information on robust partitions.)

Handling the BADSPT File or the DBS File and the RMA

All badspots encountered are added either to the static badspot file (BADSPT) or to the dynamic badspot file (DBS), if they exist. If no badspot file exists, a new badspot file is created if there is room on the partition. If FIX_DISK encounters an equivalence section in a partition's badspot file, FIX_DISK maps bad records to their equivalence records and then sets the file system pointers to those records. When the file system structure has been completely traversed, the equivalence section is deleted.

Whether you select Dynamic Badspot Handling (-IC) mode or Nondynamic Badspot Handling (-AC) mode, FIX_DISK must activate or deactivate the DBS file and the RMA. In Nondynamic Badspot Handling (-AC) mode, FIX_DISK must also allow PRIMOS access to the static badspot file, BADSPT, if it exists.

WHEN SHOULD YOU USE FIX_DISK?

You should run FIX_DISK in these situations:

- As a routine procedure during a system backup
- If you suspect that the file structure is damaged or that the quota system is damaged (indicated by a warning message from PRIMOS)
- If a problem exists with attaching to a directory or using a file (for example, a database file)
- If a message from COPY_DISK or PHYRST indicates that an equivalence block has been created
- After a power failure that shuts down the system
- When requested to do so by PRIMOS
- When you wish to select either Dynamic Badspot Handling (-IC) mode or Nondynamic Badspot Handling (-AC) mode
- When you wish to change the maximum and minimum CAM file extent sizes
- When you wish to convert a Rev. 22.0 partition to a Rev. 22.1 standard partition and use unlimited CAM file extents

WARNING

If you suspect a problem with the disk drive, disk pack, or disk controller, you should *not* run FIX_DISK with the -FIX option until the hardware is checked. A hardware problem could cause FIX_DISK to erroneously delete files and directories.

Table 6-1 presents recommendations as to when you should run FIX_DISK. See the discussion in the section FIX_DISK and Robust Partitions in Chapter 7 for more detail and explanations concerning these recommendations.

TABLE 6-1. Recommendations for Using FIX_DISK

<i>Type of Halt</i>	<i>Standard Partition</i>	<i>Robust Partition</i>
Forced Shutdown (Successful)	No FIX_DISK unless quotas incorrect; then use fast FIX_DISK	No FIX_DISK unless quotas incorrect; then use fast FIX_DISK
Forced Shutdown (Unsuccessful - No Locate flush)	Full FIX_DISK	Fast FIX_DISK
Trapped Halt	Full FIX_DISK	Fast FIX_DISK
ECCU and Immediate IDC	Full FIX_DISK	Fast FIX_DISK
ECCU and Immediate NDC	Full FIX_DISK	Fast FIX_DISK or Full FIX_DISK See Note in Chapter 7

Why Run FIX_DISK as Part of System Backups?

System backups are an important aspect of protecting the data on your system against accidental loss. It is equally important that all data are backed up correctly. By running FIX_DISK before backing up a partition, you ensure the integrity of the backup procedure itself. Several aspects of the backup procedure may depend on the integrity of the partition being backed up.

When badspot handling has taken place on a partition after you back up the partition to disk by using COPY_DISK, running FIX_DISK on the backup partition reduces the amount of time it takes to recover from a loss when you use the backup partition.

How Does PRIMOS Warn You to Run FIX_DISK?

Whenever a partition is added to the system, PRIMOS checks to see if its file structure or quota information has been damaged. When PRIMOS starts up the partition, it displays the message

```
Starting up revision 22.1 partition "NAME".
```

If it finds that the file structure or quota information may be damaged, it displays the message

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

Generally, disks are added during system cold start. Therefore, you should look at the output generated by the system during each cold start to determine if you should run FIX_DISK. You should look at the output whenever you add a disk at other times for the same reason.

If you attempt to add a robust partition that was not properly shut down because of a system halt and, thus, has a damaged file system, PRIMOS displays this message:

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

You should run fast FIX_DISK in this case. However, if it necessary to access the data immediately, you can use the -FORCE option of ADDISK. See Chapter 7 for additional details.

How Will You Know Whether There Are Problems With a Disk?

Occasionally, users may receive the following error messages, which indicate problems with a disk:

```
Pointer mismatch found. (not the same as POINTER_FAULT$)
```

```
The directory is damaged.
```

```
Directory too large.
```

```
Bad DAM file.
```

```
Bad truncate of segment directory.
```

```
Segment directory error.
```

```
The file is too long.
```

```
Too many subdirectory levels.
```

The following error message at the supervisor terminal also indicates a disk problem:

```
Disk format does not support this revision of PRIMOS.
```

In addition, messages indicating disk read-errors and write-errors may appear at the supervisor terminal or in the system event log file. For example:

```

*** Message from product LOG DISK Rev 1. generated by GEORGE on ENPUB2
(Severity information, logged at 29 NOV 87 14:27:08
DISK READ ERROR DEVICE NUMBER = 003460 (OCT)
(4005 CTRLR 0 UNIT 0)
CRA = 00000010356 (OCT) RCRA = 00000000000 (OCT)
CYLINDER = 34 HEAD = 5 RECORD = 5 (9 SECTORS PER TRACK)
STATUS (ACCUM) = 120011 (OCT) STATUS (LAST) = 120011 (OCT) RETRIES = 10 (UNCORRECTED)

```

When such error messages appear, it may be desirable to run FIX_DISK on the disk on which the error occurred. However, the error message The directory is damaged may appear when a user references several different directories or, particularly, different partitions on the same disk drive. In this case, have a representative from your Customer Support Center check the drive, disk, and controller before you run FIX_DISK. This check ensures that the problem is not in the hardware.

WHAT TO DO BEFORE RUNNING FIX_DISK

FIX_DISK is normally run as part of the backup procedure. When this is the case, follow the steps described in the *Data Backup and Recovery Guide* and use FIX_DISK by following the procedures in that book.

Sometimes, FIX_DISK is run to repair a defective file structure. In this case, follow the procedures described in Chapter 4 to shut down the partition, warning users that the partition you will repair is to be shut down. Then use the disk repair procedure, which is illustrated in Figure 6-1 and explained in the next section.

THE PROCEDURE FOR RUNNING FIX_DISK

If the partition you are repairing is the command device, follow the left-hand path in Figure 6-1: log out all users, stop the Login server by using the STOP_LSR command, and stop DSM by using the STOP_DSM command. (See the discussion of the -COMDEV option in item 7, Repair or Use the Command Device, for details.) Then specify the FIX_DISK -COMDEV option and PRIMOS automatically performs all the procedures illustrated on the right-hand path in Figure 6-1 from shutting the disk down to starting it up again. However, it may be desirable to cold start the system when you repair the command device and FIX_DISK finishes so that all system servers are automatically restarted and any other servers and phantoms are started by inclusion of the appropriate commands in your PRIMOS.COMI file.

If you are not repairing the command device, perform the procedures shown in the right-hand path of Figure 6-1 at the supervisor terminal: shut down and assign the partition; run FIX_DISK; unassign and startup the partition. Note that you can invoke the SHUTDN, DISKS, and ADDISK commands only from the supervisor terminal but you can run FIX_DISK at a user terminal.

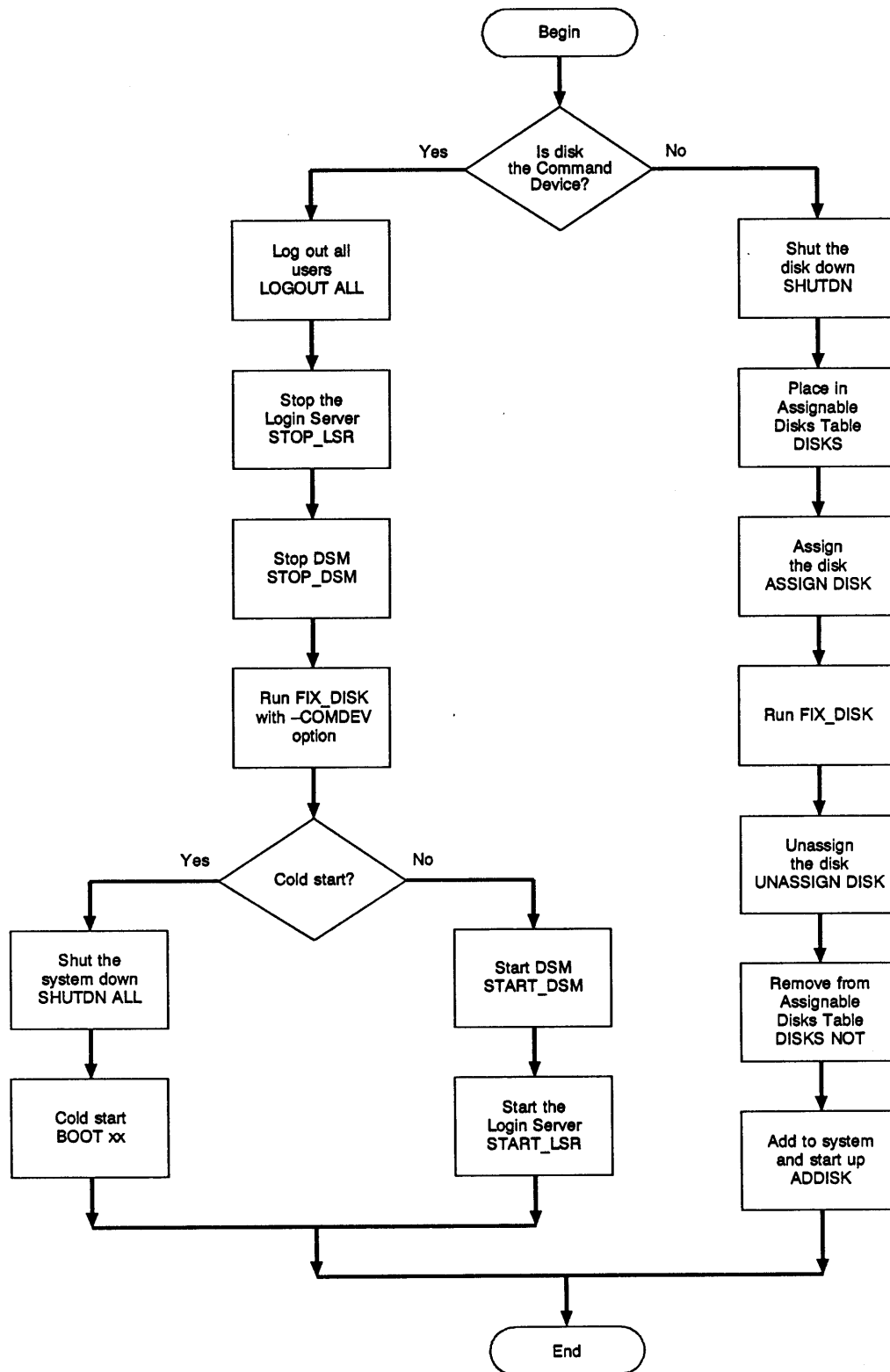


FIGURE 6-1. Disk Repair Using *FIX_DISK*

OPTIONS AVAILABLE FOR RUNNING FIX_DISK

FIX_DISK resides in CMDNCO as CMDNCO>FIX_DISK.SAVE. You invoke FIX_DISK with this command format:

FIX_DISK -DISK *pdev* [*options*]

pdev is the physical device number of the partition you are repairing. See Chapter 3 for directions on constructing physical device numbers. If you are going to repair a partition that is presently running, use the STATUS DISKS command to list the *pdevs* of all your partitions and to determine the *pdev* of the partition you want to repair.

Ten categories of command-line options are used with FIX_DISK:

- Options that specify repair tasks and revision conversions
- Options that select capabilities of disk controllers
- An option that checks to see that the system was properly shut down
- An option that specifies the command device as the target
- Options for use with pre-Rev. 22.1 formats
- An option that displays the dynamic badspot (DBS) file
- Options that modify terminal display
- Options that set the size of CAM file extents
- Options that change the method of file record allocation
- An option that gives you online help

Deciding Which Options to Use

The command-line options are discussed in detail on the following pages in the order shown below. Study these twenty items and write down the options that you wish to use. You *must* specify either -DISK *pdev* or -DISK *pdev* -COMMAND_DEVICE on the command line. The FIX_DISK options are summarized in Appendix F of this book for your convenience.

1. Specify the physical device number (*pdev*) of the partition. (-DISK)
2. Specify the extent of the repair operation. (-FIX, -DUFE, -SUFE, -UFD_COMPRESSION, -NUMBER_OF_RETRIES)
3. Add new badspots to the badspot file. (-ADD_BADSPOT)
4. Select either Nondynamic Badspot Handling or Dynamic Badspot Handling mode. (-ALL_CONTROLLER, -INTELLIGENT_CONTROLLER)
5. Truncate deeply nested directories. (-AUTO_TRUNCATION, -MAX_NESTED_LEVEL)

6. Check to see if the partition needs to be repaired. (**-CHECK**)
7. Repair or use the command device. (**-COMDEV**)
8. Convert partitions created under older revisions forward to Rev. 19.0 from pre-Rev. 19.0, to Rev. 20.0 from pre-Rev. 20.0, and to Rev. 21.0 from pre-Rev. 21.0. (**-CONVERT_19**, **-CONVERT_20**, **-CONVERT_21**)
9. Convert Rev. 22.0 partitions to Rev. 22.1 standard partitions. (**-CONVERT_22.1**)
10. Specify the physical disk type. (**-DISK_TYPE**)
11. Display the DBS file. (**-DUMP_DBS**)
12. Rapidly repair a robust partition. (**-FAST**)
13. Get online help with a display of **FIX_DISK** options. (**-HELP**)
14. Repair a defective or missing DSKRAT. (**-INTERACTIVE**)
15. Specify the amount of terminal output. (**-LEVEL**, **-LIST_FILE**)
16. List badspots from the static badspot file, BADSPT. (**-LIST_BADSPOTS**)
17. Set the maximum and minimum extent sizes for CAM files. (**-MAX_EXTENT_SIZE**, **-MIN_EXTENT_SIZE**)
18. Do not convert a pre-Rev. 19.0 partition. (**-NO_QUOTA**)
19. Specify the method of file record allocation. (**-OVERRIDE_DEFAULT_INTERLEAVE**, **-RESTORE_DEFAULT_INTERLEAVE**)
20. Specify the disposition of files containing badspots. (**-TRUNCATE**)

After studying these twenty items, invoke **FIX_DISK** with the list of options you wish to use. **FIX_DISK** performs its functions and then returns to PRIMOS. As **FIX_DISK** executes, it displays messages describing its progress and any problems it encounters.

1. Specify the Physical Device Number: You must use **-DISK** to specify the physical device number of the partition on which **FIX_DISK** is to operate. (See Chapter 3 for information on determining physical device numbers or use the **STATUS DISKS** command if the partition is presently running.) Follow **-DISK** with the physical device number. **-DISK** and the physical device number *must* be specified on the command line in this format:

```
OK, FIX_DISK -DISK 61260
```

Unless you specify **-COMMAND_DEVICE** (**-COMDEV**, as explained in item 7), **FIX_DISK** responds as if you have already shut down and assigned the partition you are repairing. If you have not assigned the partition, **FIX_DISK** displays a message to that effect and aborts:

```
DISK pdev NOT ASSIGNED  
ERI
```

If you do not include the *pdev*, **FIX_DISK** aborts with this message:

```
Bad Physical Device Number. (cl_par)  
FIX_DISK aborted  
ERI
```

Caution

Do not forget to include `-DISK` *preceding* the pdev on the command line. If the `-DISK` option is inadvertently omitted or entered after the pdev, unpredictable behavior occurs. If this happens, type `CONTROL-P` to stop the program. If you have assigned only one disk (the one to be repaired) to your terminal, there is little chance that any harm has occurred. (The chances are greater if you have assigned two or more disks.) Issue the command `RELEASE_LEVEL -ALL (RLS -ALL)`, then invoke `FIX_DISK` again. Include `-DISK` on the command line in the correct place, followed immediately by the pdev.

It is not necessary to use `-DISK` with the `-HELP` option if you only want help.

2. Specify the Extent of the Repair Operation: If you do not want `FIX_DISK` to attempt repairs, do not specify the `-FIX` option. `FIX_DISK` then reports inconsistencies, but does not attempt to rectify them. Messages from `FIX_DISK` concerning bad file pointers, record addresses, and file structures indicate that files or directories may be deleted or truncated if you do specify `-FIX`.

Caution

Do not use the `-FIX` option if you suspect that the disk drive itself is faulty. First run `FIX_DISK` without using the `-FIX` option. If no unrecoverable disk read-errors or write-errors are reported at the supervisor terminal, run `FIX_DISK` again, using the `-FIX` option. If `FIX_DISK` encounters unrecoverable disk read-errors or write-errors and you used the `-FIX` option, `FIX_DISK` may delete or truncate files.

If you do want `FIX_DISK` to make repairs, you must include the `-FIX` option on the command line. You probably want to include the `-UFD_COMPRESSION` option (abbreviation `-CMPR`) and the option `-DUFE` (delete unknown file entries) or `-SUFE` (save unknown file entries) as well. For example:

```
OK, FIX_DISK -DISK 61260 -FIX -UFD_COMPRESSION -SUFE
```

This command line tells `FIX_DISK` to

- Make all needed repairs (`-FIX`)
- Compress unused space in directory records (`-UFD_COMPRESSION`)
- Save all unrecognizable file entries (`-SUFE`)

You cannot use both `-SUFE` and `-DUFE` on the same command line; if you do, `FIX_DISK` aborts with this error message:

```
Both -DUFE and -SUFE can not be specified at the same time.
```

Caution

If you run a version of `FIX_DISK` older than the revision of the partition you are repairing, use the `-SUFE` option because there may be new file types on the more recent revision of the file system that are unrecognizable to the older version of `FIX_DISK`. The `-SUFE` option results in saving any unrecognizable file entry. If you do not use the `-SUFE` option, the default is `-DUFE` (delete unknown file entries). Be sure that you are running Rev. 22.1 `FIX_DISK`.

Note that the default is `-DUFE` (delete unknown file entries). Be sure that your partition is recently backed up before running `FIX_DISK` and that you are using Rev. 22.1 `FIX_DISK`.

You can specify the number of times that `FIX_DISK` is to attempt to read a record with the `-NUMBER_OF_RETRIES` option (abbreviation `-NUMRTY`). `FIX_DISK` normally makes nine physical adjustments to attempt to read a record and does this two times; that is, the default is two retries. You can change the default with the `-NUMRTY` option. A practical maximum may be four retries. With an intelligent disk controller, `FIX_DISK` displays one error message for each retry. With a nonintelligent disk controller, `FIX_DISK` displays nine error messages with each retry.

You must specify the `-FIX` option if you want `FIX_DISK` to make changes to the partition. An error message is displayed and `FIX_DISK` aborts if you do not use `-FIX` with these options:

<code>-ADD_BADSPOT</code>	<code>-INTELLIGENT_CONTROLLER</code>
<code>-CONVERT_19</code>	<code>-MAX_EXTENT_SIZE</code>
<code>-CONVERT_20</code>	<code>-MIN_EXTENT_SIZE</code>
<code>-CONVERT_21</code>	<code>-OVERRIDE_DEFAULT_INTERLEAVE</code>
<code>-CONVERT_22.1</code>	<code>-RESTORE_DEFAULT_INTERLEAVE</code>
<code>-INTERACTIVE</code>	<code>-UFD_COMPRESSION</code>
<code>-ALL_CONTROLLER</code>	

If you do not specify `-FIX` with the other options, `FIX_DISK` reports any errors encountered but does not attempt to repair the errors.

3. Add New Badspots to the Badspot File: You can use the `-ADD_BADSPOT` option (abbreviation `-ADBADS`) to add new badspots to the static badspot file (`BADSPT`) if the partition is in `-AC` mode or to the dynamic badspot (`DBS`) file if the partition is in `-IC` mode in a command line like this:

```
FIX_DISK -DISK pdev -FIX -ADD_BADSPOT record_1 . . . record_16
```

Enter record numbers (*record_1 . . .*) in octal. A maximum of 16 badspots may be entered on one command line with the `-ADD_BADSPOT` option. If you attempt to enter more than 16 badspots on one command line or if you enter a number that is not octal, an error message is displayed and `FIX_DISK` aborts. The `-FIX` option must be specified with the `-ADBADS` option.

You must use the -ADBADS option to add uncorrectable errors that occur on paging partitions.

The badspot file is updated by FIX_DISK after you enter badspots with the -ADBADS option. When you cold start the system or when you add the partition to the system with the ADDISK command, the DSKRAT is changed appropriately so that PRIMOS does not use the records marked as badspots in the case of an -AC mode partition or PRIMOS is directed to use RMA records in the case of an -IC mode partition.

4. Set the Partition's Disk Controller Mode: In order to have dynamic badspot handling or mirroring on Rev. 21.0 and later partitions, the disk drive on which the partition's physical disk is located must be associated with an intelligent disk controller and the partition must be in Dynamic Badspot Handling (-IC) mode.

Note

For a complete discussion of dynamic badspot handling, see Chapter 8. For a discussion of mirroring and the mirroring commands, see Chapter 9.

To switch the controller mode of a Rev. 21.0 or later partition, use the options **-INTELLIGENT_CONTROLLER** (abbreviation **-IC**) and **-ALL_CONTROLLER** (abbreviation **-AC**). For example, to switch the disk controller mode of a Rev. 22.1 partition from Nondynamic Badspot Handling (-AC) to Dynamic Badspot Handling (-IC) so that the partition is capable of being mirrored, use a command line like this:

```
OK, FIX_DISK -DISK 7462 -DT MODEL_4845 -FIX -INTELLIGENT_CONTROLLER
```

In order to switch the controller mode from -AC to -IC, the physical disk must be associated with a downloaded intelligent disk controller. In order to switch the controller mode of any partition, the first partition of that physical disk must be in the mode to which you wish to switch the other partitions. If it is not, an error message is displayed and FIX_DISK aborts:

```
OK, FIX_DISK -DISK 7462 -DT 600MB -FIX -IC
[FIX_DISK Rev. 22.1 Copyright (c) 1988, Prime Computer, Inc.]
Date: 06/28/89. Time: 15:32.
Partition name is SWTPEA

This is a revision 22.1 partition.
First partition must be in same mode as the conversion on this partition
FIX_DISK aborted
ERI
```

To switch the disk controller mode of a Rev. 21.0 or later partition to allow the partition to be used with any disk controller, use the **-ALL_CONTROLLER** option. The partition can then be used on a disk drive associated with any type of disk controller but it cannot be mirrored unless it is associated with an intelligent disk controller and is in Dynamic Badspot Handling (-IC) mode. Use a command line like this to select Nondynamic Badspot Handling (-AC) mode:

```
OK, FIX_DISK -DISK 141020 -FIX -ALL_CONTROLLER
```


When you use the `FIX_DISK -IC` option to switch a Rev. 22.1 partition to Dynamic Badspot Handling (-IC) mode, the first partition on that physical disk must be in -IC mode and the first partition must be assigned to you. This is because remapped records must be retrieved by `FIX_DISK` from the first partition without interference from another process. The only way to do this is to have the first partition shut down and assigned. If you are switching or using the command device, use the `-COMDEV` option (see item 7).

Note

If the first partition is the command device (COMDEV), use the `-COMDEV pdev` option to have `FIX_DISK` shut it down and assign it to you. (Remember to stop and restart DSM, the Login server, and any other phantoms.) In addition, if you are switching a partition other than the first partition, the first partition must be in the same mode to which you are switching the other partition. Also, you cannot switch from Nondynamic Badspot Handling (-AC) mode to Dynamic Badspot Handling (-IC) mode unless the disk is associated with a downloaded intelligent disk controller.

`FIX_DISK` leaves a Rev. 22.1 partition in its current mode unless you specifically instruct `FIX_DISK` to select either -AC mode or -IC mode.

5. Truncate Deeply Nested Directories: `FIX_DISK` may encounter directories that are nested very deeply (more than 99 levels deep). Normally when this happens, `FIX_DISK` aborts with the following error message, allowing you to consult with the owner of the offending directory tree and take appropriate action.

Ufd nesting exceeds maximum specified.

However, if you want `FIX_DISK` to automatically delete such directories at the 99th level, include the `-AUTO_TRUNCATION` option (abbreviation `-AT`) on the command line, as follows:

OK, `FIX_DISK -DISK 61260 -FIX -AUTO_TRUNCATION`

In addition, you may change the allowable maximum number of nested levels by including the `-MAX_NESTED_LEVEL n` option (abbreviation `-MAX`) on the command line. *n* is a decimal number from 0 to 3855, inclusive. If `FIX_DISK` encounters a directory nesting greater than the decimal *n* value you specify, `FIX_DISK` aborts, or, if `-AUTO_TRUNCATION` was specified, `FIX_DISK` deletes the directories that exceed the maximum nesting level. For example:

OK, `FIX_DISK -DISK 61260 -FIX -MAX_NESTED_LEVEL 20`

Here, the specified maximum nested level is 20 and `FIX_DISK` aborts if directories are nested deeper than that level and you did not include the `-AUTO_TRUNCATION` option. If you also include the `-AUTO_TRUNCATION` option, `FIX_DISK` deletes directories nested deeper than that level. If no maximum level *n* is specified with the `-MAX_NESTED_LEVEL` option or if you do not use the `-MAX_NESTED_LEVEL` option, the maximum level defaults to 99. If you specify an invalid number for *n* (that is, outside the range of 0 to 3855), this error message is displayed:

Invalid max nested level, set to max of 3855

6. Check to See if the Partition Needs to Be Repaired: The **-CHECK** option allows you to determine if the partition has been shut down improperly (for example, due to a system crash) or to determine if the partition has been written on by PRIMOS II (DOS). The **-CHECK** option causes **FIX_DISK** to examine two bits in the **DSKRAT**. One of these bits is set to 0 when the partition is added to the system by the **ADDISK** command and is set to 1 when the partition is properly shut down with the **SHUTDN** command. The other bit is set to 0 if the partition is written on by DOS. If **FIX_DISK** finds that either or both of these bits are 0, a message is displayed warning you to run **FIX_DISK**. For example:

```
OK, FIX_DISK -DISK 44063 -CHECK
[FIX_DISK Rev. 22.1 Copyright (c) 1988, Prime Computer, Inc.]
Date: 03/24/89. Time: 14:44.
Partition name is B00KS
```

This is a revision 21 partition.

Partition not shutdown properly during the previous session; please run fix_disk!

If you suspect that the partition was not shut down properly, you can specify the **-FIX** option with the **-CHECK** option to make the necessary repairs.

7. Repair or Use the Command Device (COMDEV): If the specified disk is the command device (logical device 0 as indicated by a **STATUS DISKS** command), you *must* include the **-COMMAND_DEVICE** option (abbreviation **-COMDEV**) on the command line, as follows:

```
OK, FIX_DISK -DISK 1060 -COMMAND_DEVICE
```

Including this option instructs **FIX_DISK** to automatically perform the steps shown in the right-hand path of Figure 6-1: shut down the command device, assign it, repair it, unassign it, and start it up again. In addition, **FIX_DISK** restores the event-logging state on pre-Rev. 21.0 systems and reestablishes any priority ACL that may have been set on the command device. **FIX_DISK** does this because shutting down the command device disables all event logging and removes any priority ACL set on the command device.

For Rev. 21.0 and later systems, DSM does the logging; therefore, you must stop DSM before running **FIX_DISK** on the command device or on any partition that contains the top-level directory **DSM*** or on any partition that contains the DSM logging directories. Restart DSM when **FIX_DISK** finishes to restart the event logging mechanism. While **FIX_DISK** is operating on the command device or on the partition containing **DSM*** or the logging directories, normal event logging does not take place but event messages appear at the supervisor terminal. If **DSM*** is on a partition other than the command device, messages are logged in the undelivered log file **DSM*>LOGS>UMH>UNDELIVERED.LOG** when **FIX_DISK** operates on the command device.

When using the **-COMMAND_DEVICE** option, you *must* run **FIX_DISK** from the supervisor terminal. In addition, you should log out all users and phantoms before running **FIX_DISK** because they rely on the availability of the command device.

Before using `FIX_DISK` with the `-COMDEV` option, stop the Login server by using the `STOP_LSR` command and stop DSM by using the `STOP_DSM` command. Also stop any other system servers and phantoms that depend on the availability of the command device. When the command device or any partition is shut down, all files on that partition are closed. In the case of the command device, this includes the run files for EPFs. Thus, when a process attempts to access memory, it fails because the EPF is unmapped.

Note

An error can occur while you are running `FIX_DISK` on the command device causing `FIX_DISK` to stop. If this happens, the command device is shut down. Recover by adding the command device and re-initializing your command environment by using the `ICE` command. Then restart `FIX_DISK` by reentering your command line.

Running `FIX_DISK` on the command device with the `-COMDEV` option causes `FIX_DISK` to shut it down. This causes the Login server, the DSM servers, and any other phantoms to abort. You must therefore start up the Login server and DSM after running `FIX_DISK`. You must also start other subsystems that were shut down.

When `FIX_DISK` finishes and the partition restarts, issue the `START_LSR` command to restart the Login server and issue the `START_DSM` command to restart DSM and event logging. Also restart any other system phantoms. However, if you shut down the system and then cold start when `FIX_DISK` finishes, the Login server and other system servers are automatically restarted and DSM is also automatically restarted if you have the `START_DSM` command in your `PRIMOS.COMI` file. Cold starting the system is recommended after running `FIX_DISK` on the command device.

Note

You must use the `-COMDEV` option followed by the `pdev` of the command device not only when you repair the command device but also under the following conditions:

- You are converting a partition to Dynamic Badspot Handling (`-IC`) mode, either from a pre-Rev. 21.0 format or by switching a partition from Nondynamic Badspot Handling (`-AC`) mode *and*
- The partition you are converting, or switching, is not the first partition (the partition containing the first surface (surface 0) of the physical disk) *but*
- The first partition is the command device (`COMDEV - ldev 0`)

This is illustrated in the example below.

Assume you are switching modes of the second partition (`pdev = 32060`) of a disk and the first partition is the `COMDEV` (`pdev = 1460`). Use a command line like this:

```
OK, FIX_DISK -DISK 32060 -FIX -IC -COMDEV 1460
```

Using the `-COMDEV` option causes the command device to be shut down and assigned to you and prevents access to the file system on the command device by other user processes while `FIX_DISK` is making the conversion. Use of the `-COMDEV` option also allows `FIX_DISK` necessary access to the DBS file and the RMA. When `FIX_DISK` finishes, you must restart system servers and phantoms as noted above.

8. Convert the Revision of a Partition: You cannot use FIX_DISK to convert partitions to Rev. 22.0 or from pre-Rev. 22.0 to Rev. 22.1; you must use MAKE to convert existing pre-Rev. 22.0 partitions to Rev. 22.1. (See Converting Partitions to Rev. 22.1 in Chapter 5 and the discussion earlier in this chapter under Converting a Pre-Rev. 21.0 Partition.) You can convert Rev. 22.0 partitions to Rev. 22.1 standard partitions with FIX_DISK; see item 9.

To convert a pre-Rev. 21.0 partition to Rev. 21.0, use the command-line option **-CONVERT_21**. You must use the **-FIX** option with any of the convert options (**-CONVERT_19**, **-CONVERT_20**, **-CONVERT_21**). If you use more than one of the three convert options on the same command line, FIX_DISK displays an error message and aborts.

Caution

In order to have FIX_DISK convert a partition to Rev. 21.0 format, the first partition (the partition that contains surface 0) must be converted before any other partition on that physical disk can be converted. This is necessary because the DBS file and the RMA must be on the first partition and FIX_DISK must have access to them in order to repair other partitions on the disk.

To convert partitions other than the first partition to Rev. 21.0, you must shut down and assign the first partition of that physical disk to yourself. You must shut the first partition down and assign it because FIX_DISK must have access to the DBS file and the RMA in order to convert the other partitions and those files are on the first partition. To convert to Rev. 21.0, use a command line like this:

```
OK, FIX_DISK -DISK 2062 -FIX -CONVERT_21 -DISK_TYPE MODEL_4711
```

The pdev used here (2062) indicates that this is the first partition so it is already shut down and assigned to you in order to run FIX_DISK. Note that you must specify the **-DISK_TYPE** option with the **-CONVERT_21** option. (See item 10.) You should also specify the controller mode of the partition, either Dynamic Badspot Handling (**-IC**) mode or Nondynamic Badspot Handling (**-AC**) mode. (See item 4.)

If you do not specify **-IC** or **-AC**, FIX_DISK determines the type of disk controller the disk is associated with (intelligent or nonintelligent) and sets the mode (**-IC** or **-AC**) accordingly.

To convert a disk made as a pre-Rev. 20.0 partition to a Rev. 20.0 partition, include the **-CONVERT_20** and **-FIX** options on the command line, as follows:

```
OK, FIX_DISK -DISK 1060 -FIX -CONVERT_20
```

If a BADSPT file exists, it is converted to the Rev. 20.0 format. All quota information is initialized, warning and error messages related to quotas are not displayed, and a new revision stamp is created. Existing directories are not hashed, however. (See the discussion on conversions earlier in this chapter.) The **-FIX** option must be used with the **-CONVERT_20** option.

To convert a pre-Rev. 19.0 partition to a Rev. 19.0 partition, use the **-CONVERT_19** and **-FIX** options. You cannot use the **-CONVERT_21**, **-CONVERT_20**, and **-CONVERT_19** options on the same command line.

Note

It is recommended that you use **MAKE** to convert pre-Rev. 20.0 partitions to Rev. 20.0 or to Rev. 21.0 and pre-Rev. 22.0 partitions to Rev. 22.0 or to Rev. 22.1. This is because of directory and file attributes introduced starting at Rev. 20.0. You must use **MAKE** to convert any partition to a Rev. 22.1 robust partition. See Chapter 5 for details.

9. Convert a Rev. 22.0 Partition to Rev. 22.1: You can convert a Rev. 22.0 partition to a Rev. 22.1 standard partition. A Rev. 22.1 partition can contain CAM files with an unlimited number of extents. The option **-CONVERT_22.1** allows you to convert a Rev. 22.0 partition to Rev. 22.1 by updating the revision stamp in the **DSKRAT** of the partition. Earlier revisions than Rev. 22.0 cannot be converted to Rev. 22.1 with **FIX_DISK**; you must use **MAKE** to convert earlier revision partitions to Rev. 22.1 by creating them as Rev. 22.1 partitions. In addition, if you want to create Rev. 22.1 robust partitions, you must have the **MAKE_ROBUST** utility (see Chapter 7).

You must use the **-FIX** option with **-CONVERT_22.1** to convert the partition. It is not necessary to convert the first partition first but it is good practice to do so and to convert all partitions on a physical disk to the same revision. Also it is not necessary to have the first partition assigned to you or to use the **-DISK_TYPE** option because all the conversion does is update the revision stamp in the partition's **DSKRAT**.

Prior to Rev. 22.1, a CAM file could have a maximum of 340 extents, which should be more than sufficient for a well-organized disk partition. However, this limit of 340 extents can be reached on fragmented partitions and with large databases. (See Chapter 7 for a discussion of fragmentation of partitions.) When the limit is reached, the operation that is extending the file is stopped and the file must be copied or the partition remade to reduce the number of extents in the file.

A CAM file performs best when it has a minimal number of extents and the size of each extent is as large as possible. For example, a file of a single extent of five physical records performs better than a file of five extents containing one physical record each. Thus, it is possible for an already poorly allocated CAM file to become even less optimal by increasing the number of extents allowable in the file. You should monitor large CAM files with the **LEM** command. If the number of extents is growing large and the **LCB** command indicates sufficient contiguous space is available to reduce the number of extents, you should copy the file with the **COPY** command, delete the original file, and change the name of the copy back to the original filename. If sufficient contiguous space is not available on the partition, you can save the contents of the partition to magnetic tape with a logical backup utility (such as **MAGSAV**), recreate the partition by using **MAKE**, and restore the files with a complimentary logical utility (such as **MAGRST**).

10. Specify the Physical Disk Type: You must use the **-DISK_TYPE** option (abbreviation **-DT**) with the **-CONVERT_21** option. This is necessary because **FIX_DISK** must create the DBS file and the RMA on the first partition. The amount of disk space **FIX_DISK** uses for these files depends on the disk type. (See Table 5-2 in Chapter 5.)

If you do not specify the **-DISK_TYPE** option with **-CONVERT_21**, **FIX_DISK** displays this message and aborts:

For a Rev 21 conversion, the **-disk_type** model **MUST** be specified.

If you specify **-DISK_TYPE** but do not specify the type or specify an invalid type, **FIX_DISK** displays a list of valid types and aborts. For example:

```
OK, FIX_DISK -DISK 102062 -FIX -CONVERT_21 -DISK_TYPE
[FIX_DISK Rev. 22.1 Copyright (c) 1988, Prime Computer, Inc.]
Date:03/24/89. Time: 14:48.
Invalid disk model name; Valid disk types are:
```

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

```
FIX_DISK aborted
ERI
```

Select a valid type from the left column above and reenter the command line, specifying the disk type.

11. Display the DBS File: You may wish to display the dynamic badspot (DBS) file and save the display in a COMO file. This may be useful if the DBS file is somehow damaged. You can then use **FIX_DISK** or **MAKE** to manually reenter the badspots.

To display the DBS file, use the **FIX_DISK** option **-DUMP_DBS** (abbreviation **-DDBS**). The format of the display is shown here; a detailed display is shown in the section Examples of Running **FIX_DISK**, later in this chapter.

```
OK, FIX_DISK -DISK pdev -DUMP_DBS
```

```
DBS file version version_number.
DBS file has number_of_records records.
Number of badspots = number_of_badspots.
File last modified by Primos/controller.
```

Operator's Guide to File System Maintenance

Badspot			Remap Record		
Cyl,	Head,	Sector	Cyl,	Head,	Sector
-----			-----		
a,	b,	c	r,	s,	t
d,	e,	f	u,	v,	w

pdev in the command line is the *pdev* of the first partition on which the DBS file is located. If the file has not been modified, the line File last modified by is not displayed. The fields in the other lines are filled in appropriately. The letters a, b, c, and so on represent cylinder, head, and sector numbers of the badspots and their remapped records.

In order to display the DBS file of a physical disk, you must have the first partition of that disk assigned to you. If the first partition is the command device, use the `-COMDEV` *pdev* option in a command line like the following, where *pdev* is the *pdev* of the command device, in this case 460:

```
OK, FIX_DISK -DUMP_DBS -COMDEV 460
```

Use of this option will shut down the command device and assign it to you as explained under the discussion of the `-COMDEV` option in item 7. Remember to stop and start DSM, the Login server, and any other phantoms when you use the `-COMDEV` option.

You can have the DBS file displayed while you are repairing some other partition on the disk. The DBS file is displayed when `FIX_DISK` finishes the repair operation. To display the DBS file without specifying the `-FIX` option, use the *pdev* of the first partition where the DBS file resides. If the disk is in Nondynamic Badspot Handling (`-AC`) mode, the DBS file is displayed but it may not be up-to-date; it is updated in Dynamic Badspot Handling (`-IC`) mode.

12. Rapidly Repair a Partition: If the partition you wish to repair with `FIX_DISK` is a robust partition (described in Chapter 7), you can use the `-FAST` option to speed up processing by `FIX_DISK`.

Running `FIX_DISK` with the `-FAST` and `-FIX` options is referred to as *fast* `FIX_DISK` and running `FIX_DISK` with the `-FIX` option but without the `-FAST` option is referred to as *full* `FIX_DISK`.

The purpose of the `-FAST` option is to save time in repairing a robust partition that has experienced a system halt and was thus improperly shut down. Specifying the `-FAST` option tells `FIX_DISK` to check directory entries, including CAM file extent maps, available space (the `DSKRAT`), and the quota system on robust partitions. When you specify the `-FAST` option, `FIX_DISK` uses the extent maps to determine which records are in use in order to rebuild this information if necessary. Since `FIX_DISK` does not read each record header in this case, it is much faster than a full `FIX_DISK`. Thus, if it is more important to access data as soon as possible after a halt and have the partition available, use fast `FIX_DISK`. You may use full `FIX_DISK` to ensure the integrity of all record headers at a more convenient time.

You can use fast FIX_DISK on a standard partition on which the quota system may be damaged but no other problems exist; fast FIX_DISK checks and repairs the quota system in this case. For example, after a successful forced shutdown halt of a standard partition, you may get a message from ADDISK when you restart the system indicating that the quota system may be incorrect. If the standard partition was improperly shut down and you use the -FAST option, FIX_DISK ignores the -FAST option and full FIX_DISK is run.

You can use the -FAST option without the -FIX option to see if it is necessary to repair the partition. However, in order to repair a partition, you must use the -FIX option as shown here:

```
OK, FIX_DISK -DISK 100461 -FIX -FAST
```

Caution

Since using the -FAST option results in FIX_DISK checking only the file structures noted above, the integrity of all recordheaders in these files is not checked. Thus, some potential problems may not be detected unless you run full FIX_DISK; that is, without the -FAST option.

It is recommended that you always run full FIX_DISK (without the -FAST option) on standard (nonrobust) partitions with the one exception related to the quota system noted above. See Table 6-1 or Table for complete recommendations on running Fast FIX_DISK.

13. Get Online Help With a Display of FIX_DISK Options: To see a display of FIX_DISK options with a brief explanation of each option, use this command line:

```
OK, FIX_DISK -HELP
[FIX_DISK Rev. 22.1 Copyright (c) 1988, Prime Computer, Inc.]
Date: 05/24/89. Time: 14:29.
USAGE: FIX_DISK -DISK pdev [options]
OPTIONS:
```

The options and a synopsis of each is then displayed. You may use this command without assigning any disks and without the -DISK option and you may use it from any terminal if you want only help.

A summary of the FIX_DISK command-line options appears in Appendix F, Summary of Command Syntax for MAKE and FIX_DISK, for your convenience and in the *Operator's Guide to System Commands*.

14. Repair a Defective or Missing DSKRAT: If you have already attempted to run FIX_DISK, it may have displayed one of the following error messages and aborted:

The file structure of DSKRAT is bad.

The number of heads is different.
It should be YY is XX

The physical record size is different.
It should be YY is XX

The DSKRAT header has the wrong length.
It should be *YY* is *XX*

The partition cannot be handled by this version of FIX_DISK.

If one of these messages appears, rerun FIX_DISK and specify the options **-INTERACTIVE** (abbreviation **-INT**) and **-FIX** on the command line, as follows.

```
OK, FIX_DISK -DISK 61260 -FIX -INTERACTIVE
```

These options cause FIX_DISK to ask you questions (which you can answer either YES or NO) when it discovers a defective or missing DSKRAT, rather than to abort. The answers you supply enable FIX_DISK to construct a correct DSKRAT. The **-FIX** option must be used with the **-INTERACTIVE** option. See Example of Reconstructing the DSKRAT in the examples section of this chapter.

15. Specify the Amount of Terminal Output: Normally, while FIX_DISK is running, it displays the name of the top-level directory being processed, as follows:

```
BEGIN CLEOPATRA
```

Then, when it finishes that top-level directory, it displays

```
END CLEOPATRA  recs
```

Here, *recs* is the total number of 2048-byte records in use in the directory.

FIX_DISK starts the output with BEGIN MFD and ends the output with END MFD *recs*, because all top-level directories have the MFD as their parent directory. Because listing all of the top-level directories on the disk may take time, you may wish to disable this feature. To disable it, include the **-LEVEL** option on the command line with a decimal argument of 0, as follows:

```
OK, FIX_DISK -DISK 61260 -FIX -LEVEL 0
```

When you use this option as shown, FIX_DISK displays the BEGIN MFD and END MFD messages, but does not display any other BEGIN or END messages.

If you do not specify **-LEVEL** on the command line or if you specify **-LEVEL** without specifying a numerical value, the default is **-LEVEL 1**. Level one means to output only top-level directories in the MFD via BEGIN and END messages. Specifying a level value higher than 1 causes BEGIN and END messages to be displayed for top-level directories, their subdirectories, and so on, to the specified level. (To assist you in understanding the directory structure on a disk, FIX_DISK indents the BEGIN and END messages by two spaces for each level. (See the examples at the end of this chapter.)

Note

Do not confuse the **-LEVEL** option with the **-MAX_NESTED_LEVEL** option. **-LEVEL** affects how much information is displayed at your terminal, but does not otherwise affect FIX_DISK. **-MAX_NESTED_LEVEL**, followed by a decimal number, specifies the maximum allowable nested level of directories (see item 5).

If you wish FIX_DISK to display all the filenames of the files in all of the directories to the level you specify or to the default level, include the **-LIST_FILE** option (abbreviation

-LF) on the command line. The filenames are included under the names of their parent directories. (See the examples at the end of this chapter.)

16. List Badspots: Normally, the only information FIX_DISK prints concerning badspots is the number of lost records they represent. If you would like to see more information on badspots on an -AC mode partition or a partition that was formerly an -AC mode partition and that has a BADSPT file, include the **-LIST_BADSPOTS** option (abbreviation **-LB**) on the command line, as follows:

```
OK, FIX_DISK -DISK 1060 -LIST_BADSPOTS
```

FIX_DISK then lists all known badspots from the BADSPT file, followed by a listing of any equivalence blocks in a display before the BEGIN MFD message. For each known badspot, the track, head, and sector are listed. For each equivalence block, the record numbers of both the badspot and the remapped (alternate) record are listed. In addition, the track, head, and sector numbers are also listed for the badspots and their equivalence blocks. For example:

```
OK, FIX_DISK -DISK 4422 -LIST_BADSPOTS
[FIX_DISK Rev. 22.1 Copyright (c) 1988, Prime Computer, Inc.]
Date: 06/01/89. Time: 16:48.
Partition name is TEMP1
```

```
This is a revision 22 partition.
Disk is in -All_Controller mode with Reverse Allocation
The BADSPT file has 38 entries.
```

```
Track = 39 Head = 11 Sector = 5
Track = 79 Head = 6 Sector = 5
Track = 146 Head = 9 Sector = 7
Track = 160 Head = 23 Sector = 8
.      .      .
.      .      .
.      .      .
```

For disks having many badspots, such as the large FMDs, listing of all of the badspots may take a considerable amount of time, particularly at a low-speed supervisor terminal.

17. Set or Change the Maximum and Minimum Extent Sizes for CAM Files: At Rev. 22.0, you can set the maximum and minimum extent sizes for CAM files. An extent size is the number of contiguous 2048-byte records in one extent of the CAM file. (See Chapter 7 for a discussion of setting extent sizes.) FIX_DISK has two options to allow you to change or set the extent sizes: **-MAX_EXTENT_SIZE** (abbreviation **-MAXSIZ**) and **-MIN_EXTENT_SIZE** (abbreviation **-MINSIZ**).

You can initially set these sizes when you create the partition with MAKE by using similar MAKE options. If you do not use those MAKE options, PRIMOS uses the default number of records shown in Table 6-2 to set the extent sizes, depending on whether the partition is a robust partition or a standard partition.

TABLE 6-2. Rev. 22.1 Default Maximum and Minimum Extent Sizes

	<i>Robust Partition</i>	<i>Standard Partition</i>
<i>Maximum</i>	256	32
<i>Minimum</i>	64	16

If you decide to change these extent sizes after the partition is created, use the `FIX_DISK` options in this format:

```
FIX_DISK -DISK pdev [ -MIN_EXTENT_SIZE ] size -FIX
                    [ -MAX_EXTENT_SIZE ]
```

size is a decimal number representing the minimum or maximum extent size (number of contiguous records in an extent) to be used by the algorithm that PRIMOS uses to extend CAM files. The minimum size cannot be zero (0) and cannot be greater than the maximum. The maximum size can be any value up to 32767.

18. Do Not Convert a Pre-Rev. 19.0 Partition: To run `FIX_DISK` on a partition made as a pre-Rev. 19.0 partition without converting it, include the `-NO_QUOTA` option (abbreviation `-NQ`) on the command line:

```
OK, FIX_DISK -DISK 1060 -FIX -NO_QUOTA
```

Disabling of quota checking is necessary because pre-Rev. 19.0 partitions have no quota information, and error messages are displayed if `FIX_DISK` is run on a pre-Rev. 19.0 partition without the `-NO_QUOTA` option.

19. Set the Record Allocation Direction and Interleave Factor: The recommended method of file record allocation for Rev. 20.0 and later standard partitions depends on the combination of CPU and disk controller in your system. The interleave factor is 3 with forward sectoring for all pre-Rev. 20.0 partitions and for Rev. 22.1 robust partitions. (See Chapter 10 for an explanation of sectoring and interleave factors.) Reverse sectoring with an interleave factors of 1 is supported starting at PRIMOS Rev. 20.0.

Table 10-2 in Chapter 10 presents the recommended record allocation schemes for standard partitions based on the combination of the type of disk controller and the type of CPU in your system. Basically, the method is forward sectoring with an interleave factor of 3 for systems with a nonintelligent disk controller and a CPU in the 9950 class, and reverse sectoring with an interleave factor of 1 for all other combinations of CPU and disk controller. `MAKE` and `FIX_DISK` can determine the CPU and controller combination and set the allocation method accordingly.

You can change these recommended methods on standard partitions. To set the interleave factor to 3 with forward sectoring on a standard partition, use the `-OVERRIDE_DEFAULT_INTERLEAVE` option (abbreviation `-ODI`) as in this command line:

OK, FIX_DISK -DISK 11061 -FIX -OVERRIDE_DEFAULT_INTERLEAVE

Use the **-RESTORE_DEFAULT_INTERLEAVE** option (abbreviation **-RDI**) in a command line like this to set the interleave factor to 1 with reverse sectoring:

OK, FIX_DISK -DISK 11061 -FIX -RESTORE_DEFAULT_INTERLEAVE

You cannot use **-ODI** or **-RDI** on a robust partition; if you do, **FIX_DISK** displays the message You may not change the default interleaving on a robust partition. and ignores the option. The interleave factor remains 3 with forward sectoring on the robust partition.

20. Specify the Disposition of Files Containing Badspots: When **FIX_DISK** finds a record that it cannot read, it creates a null record (a record filled with zeros) on a good portion of the disk and appends the remaining records of the original file to the null record. If more than one sequential record is missing and the file is a SAM file, then **FIX_DISK** creates two null records and displays an error message to indicate that the correct number of records is not known. If you prefer to have **FIX_DISK** truncate a file when it encounters a badspot in any file or when it encounters an uninitialized record in a CAM file on a robust partition, include the **-TRUNCATE** option (abbreviation **-TRU**) as follows:

OK, FIX_DISK -DISK 61260 -FIX -TRUNCATE

When a file is truncated, the part of the file that is located at and beyond the file pointer is eliminated from the file. If the file pointer is positioned at the beginning of the file, all of the information in the file is removed, but the filename remains in the file directory. If you intend to truncate a file, you should have a recent backup copy of that file to restore for the file owner.

WHAT TO DO AFTER RUNNING FIX_DISK

After **FIX_DISK** has finished, you should get the repaired partition back into service by adding, or starting up, the partition. To do this, follow the steps shown on the right-hand side of Figure 6-1. If you are using **FIX_DISK** as part of a disk backup, continue with the system preparation procedure as described in the *Data Backup and Recovery Guide* after you run **FIX_DISK**.

Note

If you ran **FIX_DISK** on the command device by using the **-COMDEV** option and the supervisor terminal origin directory is a password directory, it is advisable to cold start the system. Until the system has been cold started, the supervisor terminal has no origin directory. You can use the **ORIGIN -SET** command to attach the supervisor terminal process to **CMDNC0** or to its origin directory if it is different from **CMDNC0**. For example, to set the supervisor terminal process origin directory to **CMDNC0**, use the command **ORIGIN -SET CMDNC0**.

EXAMPLES OF RUNNING FIX_DISK

This section presents some examples of running FIX_DISK to point out uses and purposes of various options and to demonstrate typical displays produced by FIX_DISK. In cases where only the command line is shown, it is assumed that the disk has been shut down, placed in the Assignable Disks Table, and assigned to the user from the supervisor terminal, as shown in Figure 6-1.

Example of Normal FIX_DISK Display

In the example below, the partition BEEBLE is specified by its pdev (22660), 0 levels of directories are to be displayed, directories are to be compressed, and necessary disk modifications are to be made.

```
OK, SHUTDOWN 22660
OK, DISKS 22660
OK, ASSIGN DISK 22660
OK, FIX_DISK -DISK 22660 -LEVEL 0 -UFD_COMPRESSION -FIX
```

```
[FIX_DISK Rev. 22.1 Copyright (c) 1988, Prime Computer, Inc.]
Date: 04/05/89. Time: 02:37.
Partition name is BEEBLE
```

```
This is a revision 22 partition.
Disk is in -Intelligent_Controller mode with Reverse Allocation
BEGIN MFD
END MFD 29404

74070 records in partition
29404 records used
    0 records lost
44666 records left
    0 records compressed
DSKRAT OK
FIX_DISK finished
OK, UNASSIGN DISK 22660
OK, DISKS NOT 22660
OK, ADDISK 22660
Starting up revision 22 partition "BEEBLE".
OK,
```

FIX_DISK identifies the partition name; determines the PRIMOS revision, the mode (-IC) and the method of record allocation (reverse) of the partition; summarizes crucial data concerning record usage; checks the DSKRAT and finds it to be proper; and signals the completion of its operation. No errors are indicated. No compression of directories takes place. The line records lost refers to record loss caused by badspots; it does not indicate file truncation or record compression.

Example of Error Handling by FIX_DISK

It is quite possible that FIX_DISK may find problems on the partition it examines. Such problems are indicated by error messages. For a complete listing of FIX_DISK error messages, see Appendix B.

The next two examples illustrate how FIX_DISK handles errors. In the first example, 0 levels of directories are printed.

```
OK, FIX_DISK -DISK 1060 -LEVEL 0 -UFD_COMPRESSION -FIX
[FIX_DISK Rev. 22.1 Copyright (c) 1988, Prime Computer, Inc.]
Date: 04/05/89. Time: 03:42.
Partition name is ZAPHOD

This is a revision 22 partition.
Disk is in -Intelligent_Controller mode with Reverse Allocation

BEGIN MFD
ACL at word 513 does not point at a file or access category!
ACL is deleted!
ACL at word 4044 does not point at a file or access category!
ACL is deleted!
Access Category BOOKS.ACAT does not reference an ACL!
Access Category is deleted!
File CMDNCO does not reference an ACL or Access Category!
Changed to default ACL pointer

File CHAPTER does not reference an ACL or Access Category!
Changed to default ACL pointer
File LATE does not reference an ACL or Access Category!
Changed to default ACL pointer
File PAYOFF does not reference an ACL or Access Category!
Changed to default ACL pointer
MFD
The Directory Used count is bad. It should be 67 instead of 71.
MFD>PAYOFF
END MFD 29404

74070 records in partition
29404 records used
  0 records lost
44666 records left
  1 records compressed
DSKRAT UPDATED!
FIX_DISK finished
OK,
```

In the above example, partition 1060 is identified as ZAPHOD; two ACLs in the MFD are deleted; the access category BOOKS.ACAT is deleted; four files are changed to the default ACL; the Directory Used count in PAYOFF is found to be inconsistent and is corrected; one record is freed by compression of directories; the DSKRAT is updated; and operation is returned to PRIMOS.

Example of Conversion and Displaying Directories

In the following example, FIX_DISK explores the entire disk and prints to file system level 9 (the limit specified by -LEVEL). The use of the option -CONVERT_21 indicates that this partition is a pre-Rev. 21.0 partition and FIX_DISK is to convert it to a Rev. 21.0 partition. The first partition on this disk is assigned to the user process because FIX_DISK must have access to the DBS file and the RMA. In order to do the conversion, you must specify the -DISK_TYPE option. FIX_DISK needs to know the disk type in order to properly size the DBS file and the RMA. Since the method of record allocation is not specified with either the -ODI or the -RDI options, and the partition originally is a Rev. 20.0 partition with forward record allocation and an interleave factor of 3, it retains that method of record allocation.

```
OK, ASSIGN DISK 1660
OK, ASSIGN DISK 61660
OK, FIX_DISK -DISK 30462 -LEVEL 9 -CONVERT_21 -CMPR -FIX
[FIX_DISK Rev. 22.1 Copyright (c) 1988, Prime Computer, Inc.]
Date: 05/10/89. Time: 16:25.
For a Rev 21 conversion, the -disk_type model MUST be specified.
FIX_DISK aborted
ERI FIX_DISK -DISK 30462 -LEVEL 9 -CONVERT_21 -CMPR -FIX -DT SMD
[FIX_DISK Rev. 22.1 Copyright (c) 1988, Prime Computer, Inc.]
Date: 05/10/89. Time: 16:26.
Partition name is UNICRN
```

```
This is a revision 21 partition.
Disk is in -All_Controller mode with Reverse Allocation
The quota system may be incorrect.
The BADSPT file has 89 entries.
```

```
BEGIN MFD
  BEGIN CMDNCO
    BEGIN RUN
      END RUN 8
    BEGIN SYS.SIGNOFF.SEG
      END SYS.SIGNOFF.SEG 14
    END CMDNCO 3536
  BEGIN OEDPUS
```

```
The father pointer is bad.
It should be 5332 is 5335
Bad record address = 53340 BRA = 5340 Father = 5332 Type = 0
File is deleted!
MFD>OEDPUS>ATE>0
```

```
Zero record added to file.
MFD>OEDPUS>ATE
```

```
The Directory Used count is bad. It should be 14 instead of 147.
MFD>OEDPUS
```

```
END OEDPUS 85

BEGIN OSPREY
END OSPREY 1001
BEGIN COEPH
END COEPH 8907
BEGIN CLNNOS
```

```

BEGIN HYBRIS
  BEGIN CTHSYS
    END CTHSYS 16
  END HYBRIS 99
END CLNNOS 650
END MFD 13527

59724 records in partition
13527 records used
   6 records lost
46191 records left
   1 records compressed
DSKRAT UPDATED!
FIX_DISK finished
OK,

```

In the above example, FIX_DISK tries to examine the MFD and all directories. Terminal output is set to level 9. FIX_DISK finds an error in the top-level directory OEDPUS and corrects it by adding a record of zeros to the file; the correction is reported; a record is freed by compressing a directory, thus removing empty space on the partition; record information is displayed. The DSKRAT is updated, indicating that the disk was converted to a Rev. 21.0 partition.

Example of Using the -COMDEV Option

On a single-partition system, running FIX_DISK with the -COMMAND_DEVICE (-COMDEV) option results in output similar to the following. (Before issuing the FIX_DISK command line, use the STOP_DSM command to stop DSM and use the STOP_LSR command to stop the Login server. For information on these commands, see the *Operator's Guide to System Commands*.)

```

OK, MAXUSR 0
OK, STOP_DSM
[STOP_DSM Rev. 22.1 Copyright (c) 1989, Prime Computer, Inc.]
[2 Apr 89 15:48:54 Sunday]
DSM shutdown in progress.
Warning: SYSTEM_MANAGER terminated. Event logging will not take place!
OK, STOP_LSR
Really? YES
Phantom 97: Normal logout at 15:49
Time used: 241h 26m connect, 00m 25s CPU, 00m 30s I/O
OK, FIX_DISK -DISK 460 -UFD_COMPRESSION -FIX -COMMAND_DEVICE -DUFE -LIST_BADSPOTS
[FIX_DISK Rev. 22.1 Copyright (c) 1988, Prime Computer, Inc.]
Date: 04/02/89. Time: 15:52.
Partition name is TOPDOG

This is a revision 22.1 partition.
Disk is in -All_Controller mode with Reverse Allocation
Disk not properly shutdown. Full fix_disk will be run.
The quota system may be incorrect.
The badspot section has 2 entries.

Track = 0 Head = 6 Sector = 7
Track = 0 Head = 6 Sector = 0

```


Operator's Guide to File System Maintenance

```
BEGIN MFD
ACL at word 733 does not point at a file or access category!
ACL is deleted!
File SPOOLQ does not reference an ACL or Access Category!
Changed to default ACL pointer
```

```
BEGIN SAD
END SAD 19
BEGIN CMDNCO
END CMDNCO 2337
BEGIN DOS
END DOS 21
BEGIN TOOLS
END TOOLS 69
BEGIN SYSCOM
END SYSCOM 164
BEGIN LIBRARIES*
END LIBRARIES* 1297
BEGIN HELP*
END HELP* 279
BEGIN BACKUP*
END BACKUP* 707
BEGIN SYSOVL
END SYSOVL 151
BEGIN BATCHQ
END BATCHQ 78
BEGIN PRIMENET*
END PRIMENET* 67
BEGIN SYSTEM
END SYSTEM 1083
BEGIN EMACS*
END EMACS* 1275
BEGIN LOGREC*
END LOGREC* 74
BEGIN DOWN_LINE_LOAD*
END DOWN_LINE_LOAD* 180
BEGIN SEGRUN*
END SEGRUN* 252
BEGIN DIRECV
END DIRECV 33
BEGIN SPOOLQ
END SPOOLQ 130
END MFD 8216
```

```
14814 records in partition
8216 records used
0 records lost
6598 records left
0 records compressed
```

DSKRAT OK

FIX_DISK finished

Starting up revision 22 partition "TOPDOG".

OK, START_LSR

OK, START_DSM

[START_DSM Rev. 22.1 Copyright (c) 1988, Prime Computer, Inc.]

[2 Apr 89 16:28:45 Sunday]

DSM initialization started.

OK,

*** DSMSR (user 99 on ENPUB) at 16:29

DSM is now in a steady state.

When you use the -COMDEV option, the partition is automatically shut down and assigned at the start and automatically unassigned and restarted at the conclusion of FIX_DISK. The partition is identified as a Rev. 22.1 partition; the existing badspots are listed; the directories are listed to the default level with the total records used in each; the disposition of all records is displayed at the end; and the DSKRAT is found to be in order. When FIX_DISK finishes, the partition automatically restarts because the -COMDEV option is used.

The START_LSR command is issued next to restart the Login server and the START_DSM command to start DSM and the event logging process. Any other system servers should also be started. If you could start the system when FIX_DISK finishes, the Login server and other system servers are automatically restarted and DSM is automatically restarted if the START_DSM command is in the PRIMOS.COMI file.

Example of Running FIX_DISK Without -FIX

In the following example, FIX_DISK is run with the -LIST_BADSPOTS option and without the -FIX option, to list the badspots and any errors on the partition. FIX_DISK is then run with the -ADD_BADSPOT and -FIX options, to add a new badspot and repair the partition.

```
OK, SHUTDN DISK 1066
OK, DISKS 1066
OK, ASSIGN DISK 1066
OK, FIX_DISK -DISK 1066 -LIST_BADSPOTS
[FIX_DISK Rev. 22.1 Copyright (c) 1988, Prime Computer, Inc.]
Date: 04/05/89. Time: 06:58.
Partition name is GOLD
```

```
This is a revision 22 partition.
Disk is in -All_Controller mode with Forward Allocation
The badspot section has 1 entries.
```

```
Track = 290 Head = 2 Sector = 3 Bad record = '24335.
```

```
BEGIN MFD
  BEGIN CMDNCO
    END CMDNCO 1234
  BEGIN DOS
    END DOS 112
END MFD 1353
```

```
29628 records in partition
1353 records used
1 records lost
28274 records left
0 records compressed
DSKRAT MISMATCH!
FIX_DISK finished
```

Operator's Guide to File System Maintenance

```
OK, FIX_DISK -DISK 1066 -ADD_BADSPOT 22222 -FIX
[FIX_DISK Rev. 22.1 Copyright (c) 1988, Prime Computer, Inc.]
Date: 04/05/89. Time: 07:04.
Partition name is GOLD
```

This is a revision 22 partition.
Disk is in -All_Controller mode with Forward Allocation
The badspot section has 1 entries.

```
BEGIN MFD
  BEGIN CMDNCO
  END  CMDNCO  1234
  BEGIN DOS
  END  DOS  112
Processing add badspot request for '22222'
Added to BADSPT file!
END  MFD  1353
```

```
29628 records in partition
  1353 records used
    2 records lost
28273 records left
  0 records compressed
DSKRAT UPDATED!
FIX_DISK finished
```

```
OK, UNASSIGN DISK 1066
OK, DISKS NOT 1066
OK, ADDISK 1066
OK,
```

In the preceding example, `FIX_DISK` first lists the badspot 24335₈, which already exists in the file `BADSPT`. It checks the `DSKRAT` file and finds that it does not agree with the record allocation information generated by `FIX_DISK`. `FIX_DISK` is then invoked again, with the `-FIX` and `-ADD_BADSPOT` options specified. On the second pass, the badspot information and the `DSKRAT` file are updated.

Example of Record Truncation

`FIX_DISK` is invoked with the `-TRUNCATE` and `-NUMBER_OF_RETRIES` options in this example. `FIX_DISK` truncates any bad records, but tries four times (rather than the default of two) to read the bad records. A problem is found in a file, the file is truncated, and the `DSKRAT` is updated.

```
OK, ASSIGN DISK 1066
OK, FIX_DISK -DISK 1066 -TRUNCATE -NUMBER_OF_RETRIES 4 -FIX
[FIX_DISK Rev. 22.1 Copyright (c) 1988, Prime Computer, Inc.]
Date: 04/05/89. Time: 08:12.
Partition name is GOLD
```

This is a revision 22 partition.
Disk is in -Intelligent_Controller mode with Forward Allocation
The badspot section has 2 entries.

```

BEGIN MFD
2 files point to the same record!
bad record address = 24335
BRA = 24044   Father = 1
File truncated.
MFD>J
The Directory Used count is bad.  It should be 11367 instead of 12101.
MFD
END   MFD  11367

14814 records in partition
11367 records used
    1 records lost
3446 records left
    0 records compressed
DSKRAT UPDATED!
FIX_DISK finished
OK,

```

Example of Reconstructing the DSKRAT

The following example demonstrates the use of the -INTERACTIVE option to repair a damaged DSKRAT. First the user runs FIX_DISK, attempts to display the DBS file and finds out that the partition is corrupted. The next step is to run FIX_DISK with the -FIX and -INTERACTIVE options to repair the DSKRAT. FIX_DISK prompts for information necessary to reconstruct the DSKRAT, for the correct disk type, and for the correct revision for the partition. (See Appendix B, FIX_DISK Messages, for a complete description of the prompts.) FIX_DISK displays error messages as it processes the file system and corrects the errors.

```

OK, FIX_DISK -DISK 462 -DUMP_ DBS
[FIX_DISK Rev. 22.1 Copyright (c) 1988, Prime Computer, Inc.]
Date: 02/09/89.  Time: 08:54.
Partition name is RASCAL

```

```

The DSKRAT header has wrong length.  (RAT_CK)
The number of heads is different.
The physical record size is different.  (RAT_CK)
This is a revision 18 partition.
FIX_DISK aborted
ER! FIX_DISK -DISK 462 -FIX -INTERACTIVE
[FIX_DISK Rev. 22.1 Copyright (c) 1988, Prime Computer, Inc.]
Date: 02/09/89.  Time: 08:55.
Partition name is RASCAL

```

```

The DSKRAT header has wrong length.  (RAT_CK)
The number of heads is different.
The physical record size is different.  (RAT_CK)
This is a revision 18 partition.
Fix DSKRAT header? YES
Physical Disk? 462

```

Operator's Guide to File System Maintenance

Valid disk model names:

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

Please enter model name of disk... SMD

Split Disk? NO

DISK FILE-RECORDS PAGING-RECORDS (DECIMAL)

462 14814 0

Parameter OK? YES

Is this a rev 22.1 partition? NO

Is this a rev 22 partition? YES

Is this a robust partition? NO

Forward Sectoring ? NO

Reverse Sectoring (default)? Return

Please answer "YES" or "NO"? YES

Intelligent controller mode (default)? YES

Partition not shutdown properly during the previous session; please run fix_disk!

The BADSPT file has 11 entries.

BEGIN MFD

Bad file type: special bit not set!

The Beginning Record Address (BRA) pointer (0) is bad, it points to a record that belongs to another file.

System file is bad - ignored!

The current record address (CRA) is bad.

It should be 2 is 6366.

Bad record address = 2 BRA = 2 Father = 1 Type = 0

MFD>BOOT

BEGIN FALCON

END FALCON 57

BEGIN RAVEN

END RAVEN 60

BEGIN PETREL

END PETREL 404

BEGIN GANNET

END GANNET 4699

Unable to move record, rewriting to attempt fix it up.

END MFD 5224

14814 records in partition

5224 records used

1 records lost

9590 records left

0 records compressed

DSKRAT UPDATED!

FIX_DISK finished

OK,

Example of Checking for Proper Shutdown of a Partition

In this example, you run FIX_DISK on the command device with the -CHECK option. First, stop DSM and the Login server. You should also shut down any other servers or phantoms that may need to use the command device. You must also use the -COMDEV option, which automatically shuts down and assigns the partition before FIX_DISK is run and then unassigns and starts the partition when FIX_DISK finishes. Messages concerning the quota system and the necessity of running FIX_DISK are displayed, but the partition is not repaired because the -FIX option is not used.

Run FIX_DISK again with the -FIX option to repair the command device. When FIX_DISK finishes, DSM, the Login server, and any other phantoms must be started or the system may be shut down and then cold started.

```
OK, FIX_DISK -DISK 1260 -COMDEV -CHECK
[FIX_DISK Rev. 22.1 Copyright (c) 1988, Prime Computer, Inc.]
Date: 04/01/89. Time: 18:03.
Partition name is TOPDOG
```

```
This is a revision 22.1 partition.
Disk is in -Intelligent_Controller mode with Forward Allocation
The quota system may be incorrect.
FIX_DISK finished.
Starting up revision 22.1 partition "TOPDOG"
(Quota system may be incorrect; run FIX_DISK.)
```

```
OK, FIX_DISK -DISK 1260 -COMDEV -FIX
[FIX_DISK Rev. 22.1 Copyright (c) 1988, Prime Computer, Inc.]
Date: 04/01/89. Time: 18:30.
Partition name is TOPDOG
```

```
This is a revision 22.1 partition.
Disk is in -All_Controller mode with Forward Allocation
The quota system may be incorrect.
The badspot section has 2 entries.
```

```
BEGIN MFD
  BEGIN CMDNCO
  END CMDNCO 2354
  BEGIN PRIRUN
  END PRIRUN 1659
  BEGIN BACKUP*
  END BACKUP* 379
  BEGIN DOWN_LINE_LOAD*
  END DOWN_LINE_LOAD* 100
END MFD 4453
```

```
29628 records in partition
 4453 records used
   2 records lost
25173 records left
   0 records compressed
DSKRAT OK
FIX_DISK finished
Starting up revision 22 partition "TOPDOG"
OK,
```

Example of Changing Controller Modes

In this example, the second partition of a physical disk is switched from Nondynamic Badspot Handling (-AC) mode to Dynamic Badspot Handling (-IC) mode by using the -INTELLIGENT_CONTROLLER option.

```
OK, DISKS 10460
OK, ASSIGN DISK 10460
OK, FIX_DISK -DISK 10460 -FIX -LEVEL 0 -IC
[FIX_DISK Rev. 22.1 Copyright (c) 1988, Prime Computer, Inc.]
Date: 04/01/89. Time: 18:35.
Partition name is TEST2
```

The first partition needs to be assigned for a -AC/IC conversion

FIX_DISK aborted

ERI ASSIGN DISK 460

```
OK, FIX_DISK -DISK 10460 -FIX -LEVEL 0 -IC
```

[FIX_DISK Rev. 22.1 Copyright (c) 1988, Prime Computer, Inc.]

Date: 04/01/89. Time: 18:38.

Partition name is TEST2

This is a revision 22.1 partition.

First partition must be in the same mode as the conversion on this partition

FIX_DISK aborted

```
ERI FIX_DISK -DISK 460 -FIX -LEVEL 0 -IC
```

[FIX_DISK Rev. 22.1 Copyright (c) 1988, Prime Computer, Inc.]

Date: 04/01/89. Time: 18:40.

Partition name is SYSA29

This is a revision 22.1 partition

Disk is in -Intelligent_Controller mode with Reverse Allocation

The DBS file has 0 entries

BEGIN MFD

END MFD 1119

14814 records in partition

1119 records used

0 records lost

13695 records left

0 records compressed

DSKRAT UPDATED!

FIX_DISK finished

```
OK, FIX_DISK -DISK 10460 -FIX -LEVEL 0 -IC
```

[FIX_DISK Rev. 22.1 Copyright (c) 1988, Prime Computer, Inc.]

Date: 04/01/89. Time: 18:42.

Partition name is TEST2

This is a revision 22.1 partition

Disk is in -Intelligent_Controller mode with Reverse Allocation

The DBS file has 0 entries

BEGIN MFD

END MFD 246

```

14814 records in partition
  246 records used
    0 records lost
14568 records left
    0 records compressed
DSKRAT UPDATED!
FIX_DISK finished
OK, UNASSIGN DISK 460
OK, UNASSIGN DISK 10460
OK, DISKS NOT 460 10460
OK, ADDISK 460 10460
OK,

```

First the disk is assigned to the user and FIX_DISK is invoked. The first partition on this physical disk was not assigned to the user and FIX_DISK aborts. The first partition must be available to FIX_DISK in order to have the DBS file available. The first partition is then assigned and FIX_DISK is invoked again. This time the mode switching cannot take place because the first partition must be in the same mode that the user is switching the second partition to. The user then switches the mode of the first partition and then that of the second partition. When FIX_DISK finishes, the user unassigns the two partitions and adds them to the system. These partitions can be added, or started, on this system because they are associated with an intelligent disk controller.

Example of the DBS File Display

The following example shows the display of the DBS file when the -DUMP_DBS option is used. In this case, the DBS file is on the first partition of a file system disk that is not the command device; therefore it is not necessary to use the -COMDEV option. The display is generated during the normal course of repairing the partition. the first partition must be assigned to you or FIX_DISK displays the message Cannot access RAT on disk 0 at the end of its normal display.

```

OK, ASSIGN DISK (2460 12460)
OK, FIX_DISK -DISK 12460 -DUMP_DBS -FIX -LEVEL 0
[FIX_DISK Rev. 22.1 Copyright (c) 1988, Prime Computer, Inc.]
Date: 04/03/89. Time: 18:42.
Partition name is TPLAB

```

```

This is a revision 22.1 partition
Disk is in -All_Controller mode with Forward Allocation
The DBS file has 5 entries.

```

```

The BADSPT file has 5 entries.

```

```

BEGIN MFD
END MFD 2345

```

```

74070 records in partition
 2345 records used
   18 records lost
71707 records left
    0 records compressed
DSKRAT UPDATED!

```


Operator's Guide to File System Maintenance

```
DBS file version 1
DBS file has 1 record
Number of badspots = 5
File last modified by PRIMOS.
```

Badspot Cyl, Head, Sector	Remap Record Cyl, Head, Sector
713, 7, 0	0, 0, 8
778, 1, 4	0, 1, 8
806, 6, 7	0, 1, 1
807, 1, 1	0, 1, 2
810, 1, 3	0, 1, 3

```
FIX_DISK finished
OK,
```

The user specifies the `-LEVEL 0` option to suppress display of directories under the MFD. Use of the `-FIX` option causes the record allocation information in the DSKRAT to be corrected. The DBS file is listed when `FIX_DISK` finishes repair tasks; the file contains information for five badspots. It was last updated by PRIMOS since the disk is associated with a nonintelligent disk controller. The badspots and their remapped records are listed by cylinder, head, and sector addresses.

INVOKING FIX_DISK FROM MAGNETIC TAPE

If the command device or the `CMDNC0` directory becomes damaged so that the `FIX_DISK` utility is inaccessible, you can invoke `FIX_DISK` from magnetic tape using the `MTRESUME` command. Remember to assign the tape drive first. `MTRESUME` is fully documented in the *Operator's Guide to System Commands*.

When you invoke `FIX_DISK` from magnetic tape, use the `MTRESUME` option `-COMMAND_LINE_OPTIONS` (abbreviation `-CMDOPT`) to specify to `FIX_DISK` the details of the repair. For example, to repair a damaged command device while deleting all unknown file entries and compressing directories, use a command line like this:

```
OK, MTRESUME MT1 FIX_DISK.SAVE -CMDOPT -DISK 1060 -COMDEV -FIX -SUFE -CMPR
```

The output then appears as follows:

```
This is a revision 20 MAGSAV tape.
Date:      04-04-88
Revision:  20
Reel:      1
Name:      REPAIR
```

```
[FIX_DISK Rev. 22.1 Copyright (c) 1988, Prime Computer, Inc.]
Date: 06/05/89. Time: 02:37.
```

```
. . .
. . .
. . .
```

In this case, FIX_DISK.SAVE is run from magnetic tape. After the dialog from the tape portion of the procedure, the FIX_DISK messages appear as they normally do. Before running FIX_DISK in this manner, you must shut down and assign the partition as usual.

Note

When you boot PRIMOS from tape and then invoke FIX_DISK with the MTRESUME command, you cannot repair a disk in dynamic badspot handling (-IC) mode. This is because intelligent disk controllers are downloaded from disk when PRIMOS is booted from disk. Therefore you must use the -AC option if you are repairing an -IC mode partition. You can convert back to -IC mode after you repair the partition and reboot PRIMOS from disk.

If you use MTRESUME and FIX_DISK in the above manner and you boot PRIMOS from tape to do this, you must also use the PRIMOS SETIME command to set the date and time before invoking FIX_DISK if you have a CPU that does not have a diagnostic processor or that has a VCP III or earlier virtual control panel.

Example of Invoking FIX_DISK.SAVE From Tape

FIX_DISK may be invoked from magnetic tape by using the MTRESUME command and specifying the pathname of FIX_DISK as it was saved by MAGSAV on the tape. In this example, FIX_DISK is invoked from magnetic tape on tape drive MTO while PRIMOS is running. FIX_DISK is instructed to repair the partition, compress directories, delete unknown file entries, and display directories down to the tenth level and list all files in those directories.

```
OK, MTRESUME MTO FIX_DISK.SAVE -CMDOPT -DISK 30662 -FIX -DUFE -CMPR -LEVEL 10 -LIST_FILE
```

```
This is a revision 20 MAGSAV tape.
```

```
Date:      04-04-88
```

```
Revision: 22
```

```
Reel:      1
```

```
Name:      REPAIR
```

```
[*FIX_DISK Rev. 22.1 Copyright (c) 1988, Prime Computer, Inc.]
```

```
Date: 06/05/89. Time: 16:49.
```

```
Partition name is SWTPEA
```

```
This is a revision 22 partition.
```

```
Disk is in -Intelligent_Controller mode with Forward Allocation
```

```
BEGIN MFD
```

```
    SWTPEA
```

```
    BOOT
```

```
    DAISY
```

```
    MAYFLOWER
```

```
BEGIN FLOWER
```

```
    DAISY
```

```
BEGIN IRISES
```

```
    PETALS
```

```
BEGIN ROSES
```

```
    PISTILS
```

```
BEGIN PEONY
```

```
    SEPALS
```

```
BEGIN TULIP
```

```
    YELLOW
```

Operator's Guide to File System Maintenance

```
MAGENTA
FUSCIA
  BEGIN CFOILS
    YELLOW
  END CFOILS 340
  END TULIP 1680
  END PEONY 3361
  END ROSES 5384
  END IRISES 7740
  END FLOWER 10517
END MFD 10592
```

```
14814 records in partition
10592 records used
  0 records lost
4222 records left
  0 records compressed
DSKRAT OK
FIX_DISK finished
OK,
```

The FIX_DISK display lists all of the files in directories down to the subdirectory CFOILS and lists the records used in each directory.

PART III
SPECIAL TOPICS

ROBUST PARTITIONS

This chapter discusses the following topics:

- Understanding robust partitions and what they provide
- Understanding the -FAST option of FIX_DISK
- Understanding the robust partition file system
- Evaluating the use of robust partitions
- Creating robust partitions
- Administrating robust partitions
- Using MAKE, FIX_DISK, ADDISK, and mirroring in conjunction with robust partitions

Before you decide to convert your disk partitions to the robust format, please take a few minutes to read this chapter carefully. Then answer the questions in the section Evaluating the Use of Robust Partitions and fill out the Robust Partitions Evaluation Form.

WHAT IS A ROBUST PARTITION?

A **robust partition** is a new type of disk partition at Rev. 22.1. Robust partitions reduce the time that it takes to recover from a system halt. All files and segment directory subfiles on a robust partition are physically stored as CAM files. The CAM file structure allows the -FAST option of FIX_DISK to quickly check the extent map and verify the physical structure of the CAM file. This same capability is not available on a **standard** (non-robust) partition. (See Chapter 1 and Figures 1-2, 1-3, and 1-4 for descriptions of file types.)

Another major advantage of robust partitions is that PRIMOS advises you whenever the result of a system halt requires you to run FIX_DISK on a partition. PRIMOS cannot require you to run FIX_DISK on a standard partition after a system halt nor can FIX_DISK indicate when it should be run except in the case of an incorrect quota system.

What Robust Partitions Can Provide

Robust partitions offer several advantages that can significantly reduce the length of time that is required for you to resume normal operations after a system halt. Some of these advantages derive from the robust partition structure. A few of the advantages are based upon the inherent characteristics of CAM files. The purpose of this subsection is to explain the nature of the advantages that robust partitions offer.

Advantages: Advantages of using robust partitions include

- System availability is improved because some halts do not require `FIX_DISK` to be run and others require only fast `FIX_DISK` (`FIX_DISK -FAST`) in place of full `FIX_DISK`.
- PRIMOS tells you whether or not you must run `FIX_DISK` on the partition when you use the `ADDISK` command. This saves you the time of running `FIX_DISK` unnecessarily.
- Robust partitions can improve upon your ability to resume operations after some system halts.
- File deletions and truncations are faster since it is necessary to read only extent maps rather than every data record.
- Writing out full records using the PRIMOS subroutine `PRWFSS` is 50% faster.
- Robust partitions offer a faster record access mechanism for some environments.
- Robust partitions offer the most advantage when you have large files or segment directories with large subfiles.

Because the design of robust partitions specifically improves the ability to recover from a system halt, the disk format is less likely to suffer from some types of directory corruption that can occur on a standard partition. Because of the file system structure implemented on a robust partition, fast `FIX_DISK` can verify the integrity of the user directories. This can greatly reduce the length of time that is required to run `FIX_DISK`. As a result, you can quickly check the directory structure.

Logical File Types: Robust partitions introduce a new concept called **logical file typing**. All files stored on a robust partition are physically stored as CAM files. Although you might open a file with a logical file type of SAM, PRIMOS physically creates the file as a CAM file. This is transparent to all higher levels of software and allows you to move existing applications to a robust partition without modification. This logical-to-physical mapping also allows PRIMOS to more tightly control the file structure on a robust partition, without changing the logical appearance of that file structure.

Because every file and every segment directory subfile on a robust partition is physically stored as a CAM file, there is less likelihood that a file will be damaged by a corrupt record header chain. Since CAM file data records are not chained through the record headers, corruption of a data record header does not cause the remainder of the file to be lost. Also, the extent map mechanism means that fast `FIX_DISK` is able to detect file structure corruption very quickly by checking the extent map.

Record Errors: The introduction of robust partitions offers a new method of responding to a corrupted data record. On a standard disk partition, a pointer mismatch (e\$ptrm) error occurs if the record header chaining is corrupt. This error is fatal to the application and can be corrected only by running FIX_DISK. This same error can occur on a robust partition, but PRIMOS reports it as an uninitialized block (e\$zero) and re-initializes the data record header, filling the data record with nulls. Although, there is now a null data record, the file can still be accessed without requiring you to run FIX_DISK to correct the error. If the application detects this error, it can take its own corrective action, which may include a data-management rollback procedure to correct the data integrity of the database. (Prime DBMS, Prime® ORACLE®, MIDASPLUS™, and PRISAM™ all treat the uninitialized block as a fatal error; the application fails and returns to PRIMOS.)

Record Access: Robust partitions also offer a faster record access mechanism for some environments. Typically, a large CAM file provides faster data access than a large DAM file. This is noticeable when you have multiple users accessing the same file simultaneously and when the file is larger than 512 disk records (1 megabyte). This faster access can be an advantage if your application does not already use CAM files.

File Deletion: Deleting a large file is always significantly faster on a robust partition than on a standard partition. Two files cannot claim the same data record on a robust partition. On a standard partition, PRIMOS must verify that all of the records within the file actually belong to the file. Verification is not necessary on a robust partition.

Restrictions on the Use of Robust Partitions

There are a few restrictions on when you can use a robust partition.

Shutdowns: Because the ADDISK command checks a robust partition, you must run FIX_DISK if the partition was not cleanly shut down. This can be inconvenient if you do not regularly run FIX_DISK after every system halt. Forcing you to run FIX_DISK in this case, however, provides better assurance of file structure integrity.

In order to reduce the time necessary to recover from a system halt, you need to use the -FAST option of FIX_DISK (fast FIX_DISK). Fast FIX_DISK checks the directory structure and CAM file extent maps only. Please refer to the section Understanding the Concept of Recoverability and then read through the section Understanding the -FAST Option of FIX_DISK, below.

Bootng: The boot procedure can only access files stored as SAM files. All files on a robust partition are stored as CAM files. The PRIMOS boot procedure cannot access any file stored on a robust partition. This means that you should not convert your command partition to a robust partition. This also means that you cannot use a robust partition as an alternate boot device.

Disk Space Required: Sometimes a robust partition requires more disk space than a standard partition to store the same amount of data. A SAM file on a standard partition contains only data records. When you move the SAM file to a robust partition, the file requires an additional record for the extent map. This means that a file that was stored as a single-record SAM file on a standard partition becomes a two-record CAM file on a robust partition. You must allow enough additional space for the conversion.

The amount of additional space required depends on the file type. ACLs and ACATs do not require additional space. Each SAM file requires one additional disk record for an extent map. DAM files might not require any additional disk space. CAM files do not require any additional disk space. Remember however, that CAM files allocate data records in blocks called extents. There are occasions when PRIMOS appends unused data records to the end of a CAM file. These records occupy additional disk space. Generally, you can minimize all of these considerations by placing only large database files on a robust partition.

The size of a segment directory is not significant to the discussion about robust partitions. The segment directory structure is itself not changed. Size considerations instead focus on the size of the individual subfiles within a segment directory. To determine the size of subfiles, see the discussion of segment directories in the section Robust Partition File System Objects.

Directories: The directory structure itself is changed on a robust partition. A Rev. 22.1 standard partition uses a hashed directory structure. A robust partition uses a linear directory structure. Entries should consist of only a small number of large files in each directory on the robust partition to maintain the directory search time.

Sectoring: Robust partitions do not support reverse sectoring. Whenever you convert a partition to the robust format, sectoring is automatically set to forward.

Accessing Rev. 22.1 Disks: Rev. 22.1 disks are a new format. To locally access either standard or robust Rev. 22.1 partitions, you must be running Rev. 22.1 PRIMOS. You can access Rev. 22.1 partitions remotely on a network, however, such as through PRIMENET. This means that you should not reboot your local system to an earlier version of PRIMOS. Insure that all of the PRIMOS upgrade has been successfully completed before you begin the conversion to robust partitions.

Null Records: Finally, understand that PRIMOS can insert a null-filled data record into your database as a result of a system halt. This rare event would cause a fatal error on a standard partition.

Understanding the Concept of Recoverability

You should understand one essential concept before deciding whether or not to use robust partitions. Robust partitions improve upon **recoverability**, or your ability to resume operations after a system halt. Similar to `FIX_DISK`, robust partitions do not offer any protection against disk corruption; they offer only an improvement in your ability to detect disk corruption. This is one reason why it is important to use robust partitions only for files that an application-level data verification routine can properly check.

In many cases, you can find a degree of data integrity corruption by running full `FIX_DISK`. This is not the reason for running `FIX_DISK`; `FIX_DISK` was designed to check file system integrity and does not check data integrity. Nevertheless, many locations rely on `FIX_DISK` to indicate whether the data integrity of a file has been compromised. This appears to work on a standard partition because full `FIX_DISK` detects corrupted data record headers. The assumption is made that if the data record headers are not corrupt, the data records are probably not corrupt either. Some of this ability to detect data corruption is lost when fast `FIX_DISK` (`FIX_DISK -FAST`) is used on a robust partition because fast `FIX_DISK` will not read any data record headers and therefore cannot verify the validity of the data record headers. Used properly, fast `FIX_DISK` offers the advantage of rapidly repairing your partitions but this can only be an advantage when you have an alternative process in place to verify data record integrity.

Robust partitions offer help in minimizing the inconvenience caused by a hardware failure, which can cause data loss. Table 7-1, Recommendations for Using `FIX_DISK`, later in this chapter, summarizes types of system halts and the necessary action to properly respond to those halts. These recommendations are applicable to systems using either standard or robust partitions. From an examination of Table 7-1, you can see that robust partitions offer the advantage of effectively utilizing the `-FAST` option of `FIX_DISK` (fast `FIX_DISK`) for those system halts that are trapped and processed through the PRIMOS slow-halt mechanism.

UNDERSTANDING THE `-FAST` OPTION OF `FIX_DISK`

The `-FAST` option of `FIX_DISK` (fast `FIX_DISK`) allows the System Operator to quickly verify the integrity of the file structure. `FIX_DISK` does not provide any check on the integrity of the data contained within the files. Only a utility that understands the data management application can verify the data within a file.

This section explains the functionality of fast `FIX_DISK` on a robust partition and then briefly compares the functionality when you run fast `FIX_DISK` on a standard partition.

Both robust partitions and standard (nonrobust) partitions support the `-FAST` option. The `-FAST` option is less useful, however, on a standard partition because it can be used only if the partition was cleanly shut down and then only repairs the quota system.

FIX_DISK Action: FIX_DISK acts identically on the file system directory structure on both standard and robust partitions whether you enable the -FAST option or not. FIX_DISK checks the entire directory structure and verifies the integrity of every directory and segment directory entry. Use of fast FIX_DISK, however, limits the degree of verification on files within directories.

Use of fast FIX_DISK also limits the degree of verification of subfiles within a segment directory. This is an important technical detail. A segment directory is a special type of directory structure that contains a set of subfiles. All of the data is contained within the subfiles. Like any directory, there is a directory header that contains all of the information about the contents of that directory. A segment directory can contain many subfiles. Both full and fast FIX_DISK verify every directory header and every segment directory header. Use of the -FAST option allows FIX_DISK to provide directory structure verification more quickly.

When fast FIX_DISK completes without finding any mismatches, it has checked that the directory structure is intact and that the correct number of disk records have been allocated for the data files. You cannot be sure, however, that the data records actually have the correct data within them. To verify the data record content, you must run a verification routine of a data management package on any data management files.

Full FIX_DISK provides one additional level of verification that fast FIX_DISK does not provide. Full FIX_DISK reads every data record header within every file. Full FIX_DISK then verifies that the record header is properly initialized. Do not, however, rely on FIX_DISK as an indicator of the integrity of the data in a disk record.

Full and Fast FIX_DISK Comparison: To better understand the benefits robust partitions offer, we must distinguish between CAM file functionality and robust partition functionality. Full FIX_DISK processes a CAM file identically whether it is on a robust partition or on a standard partition. The operation of fast FIX_DISK depends whether the CAM file is on a standard partition or on a robust partition. On a standard partition, fast FIX_DISK verifies the last two data records within every CAM file extent. On a robust partition, fast FIX_DISK verifies only the extent map.

On a robust partition, all files are automatically stored as CAM files. Through the logical file typing mechanism, the physical file type is transparent to all higher levels of software. It is the physical typing, however, that is important to FIX_DISK.

In order for FIX_DISK to know which disk records a physical SAM file on a standard partition uses, FIX_DISK must check every record because SAM files do not have an index or an extent map. When FIX_DISK encounters a SAM file, it must read a record header, find the pointer to the following record, and then repeat the process. Thus, both full FIX_DISK and fast FIX_DISK must read through the entire SAM file. PRIMOS physically stores all SAM files as CAM files on a robust partition and, thus, FIX_DISK needs to check only the extent map.

In conclusion, the -FAST option is available on both standard and robust partitions. Fast FIX_DISK verifies the full directory structure on both standard and robust partitions. You can run fast FIX_DISK on a standard partition only when the partition has been cleanly shut down. If you need to run FIX_DISK on a regular basis, robust partitions can reduce the time required.

UNDERSTANDING THE ROBUST PARTITION FILE SYSTEM

The robustness of a robust partition is transparent to nearly all software. Robust partitions introduce a new concept called *logical file typing*. In previous revisions of PRIMOS there were three types of physical files. A file could be a physical SAM file, a physical DAM file, or a physical CAM file. This physical typing determines exactly how the file is strung together to make it an entity. Robust partitions separate the physical file structure from the logical, or application-level, file structure. (See Chapter 1 for a description of the physical file types.)

Every file that is created on a robust partition is physically organized as a CAM file. This means that every file on a robust partition has an extent map that tells PRIMOS where the actual data records are stored. All of this is transparent to higher levels of software. LD, for example, reports the existence of SAM, DAM, and CAM files on a robust partition. If your application opens a SAM file on a robust partition, it appears to be a SAM file. This is the logical file type and it determines which application-level operations are possible. Underneath the application, however, PRIMOS converts the operations into the proper steps to access the correct data record in the physical CAM file that actually exists.

How Do You Know if a Partition Is Robust?

Because of logical file typing, a robust partition can be completely transparent to the system users. There are three commands that you, as the System Operator, can use to determine which partitions are robust.

You can determine if a local partition is robust by looking at the status information for the partition with the STATUS DISKS command or with the LIST_DISKS command. The display includes an indication that a partition is robust, as shown in the following example.

OK, STATUS DISKS

Disk	Ldev	Pdev	System	Robust
GROUP1	0	4461		
OATIOS	1	4260		
BIGDSK	2	4463		Robust

The LD command is another useful tool for determining whether a partition is robust. Unlike the STATUS DISKS and LIST_DISKS commands, the LD command reports robustness on both local and remote disks. The following example shows information that LD reports on a robust partition.

OK, ATTACH <BIGDSK>MFD

OK, LD

<BIGDSK>MFD (LUR access) Robust

4 Files.

BADSPT	BOOT	BIGDSK	DYNBSP
--------	------	--------	--------

12 Directories

BATCH	CMDNCO	CRUSHES	DOS
DSA.ENM	FED	JANE	LEPRUN
MFD	BULL_BOARD	R1040A	WORD*

OK,

Robust Partition File System Objects

The file system objects supported on a robust partition consist of the following subset of the objects that are supported on a standard PRIMOS partition:

- CAM files (with logical SAM, DAM, and CAM file typing)
- Segment directories
- Linear user directories
- EPFs
- Access Control Lists (ACLs)
- Access Categories (ACATs)

This subset represents all of the PRIMOS file system objects that are found on a standard disk partition except for the physical SAM and DAM files. Robust partitions support logical SAM and DAM files only as physical CAM files. Also, robust partitions support only the linear directory structure. Robust partitions do not support hashed directories. This subset of entry types supported by robust partitions is best for use with a few large files.

CAM Files: CAM files are extent-based files. An extent is a variable number of contiguous 2048-byte disk records. (See Figure 1-4 and the description of CAM files in Chapter 1.) The addresses of the CAM file extents are indexed in the extent maps that exist for each CAM file. Each extent map for a CAM file can index a maximum of 340 extents. The use of CAM files requires that additional memory be allocated for extent maps because an extent map table resides in memory for each open CAM file. The amount of memory required varies since the number of extent maps and, thus, the size of the extent map table is determined by the number of extents in the file.

A Rev. 22.1 standard or robust partition can contain CAM files with a large number (16,381) of extents. Prior to Rev. 22.1, a CAM file could have a maximum of 340 extents and, thus, only one extent map. The previous limit of 340 extents can be reached on

fragmented partitions and with large databases. (See the subsection Fragmentation of Partitions later in this chapter for a discussion of fragmentation.) When the limit is reached, the operation that is extending the file stops and you must either copy the file or remake the partition to reduce the number of extents in the file. With unlimited CAM file extents available at Rev. 22.1, it is unlikely that CAM files will reach the limit, but fragmentation can occur if there are many extents in a file.

File System Objects: Other than the special files created by MAKE, CAM files are the only non-directory file system objects on a robust partition. When you first create the partition, MAKE builds the special files BOOT, DSKRAT, DYNBSP, and a badspot file. A standard partition supports non-directory file system objects of the types SAM, DAM, or CAM. A robust partition supports only the CAM physical type of non-directory file system objects. Because SEGSAM and SEGDM entries are actually a type of directory structure, these structures are also supported on a robust partition. All subfiles within a segment directory, however, are created as CAM physical file types.

Whenever a user creates or copies a non-directory file system object on a robust partition, PRIMOS automatically organizes the file physically as a CAM file. This physical typing is distinct from the logical typing. The logical file type remains that which was set by the subroutine or application that creates the file. The logical file type can be SAM, DAM, or CAM but the physical file type will always be CAM on a robust partition.

The physical file typing is transparent to the software that accesses the file. The file continues to behave as if it were a file of the logical type. Even EPFs, which are DAM files on standard partitions, are physically organized as CAM files on robust partitions. If you check the file type through a subroutine call or by using LD -DETAIL, the file type listed is always the logical file type. Robust partitions are thus able to support all three non-directory logical file types even though all of the files are physically CAM files.

Copying Files: You can copy CAM files to robust partitions, both locally and remotely. You can copy CAM files *from* robust partitions, both locally and remotely, as long as the partition you are copying the CAM file to is Rev. 20.0 format (or later). You can copy CAM files to pre-Rev. 20.0-format disks as long as you use the -SAM or -DAM conversion options when you invoke the COPY command. This same conversion process is also available through the COPY command when copying a CAM file to a Rev. 20.0 or later standard partition.

CAM File Management: PRIMOS subroutines allow management of CAM files that have a CAM logical file type. See the *Subroutines Reference II: File System* for discussions of these subroutines:

- CF\$EXT to move a CAM file's physical end-of-file
- CF\$REM to return information about a CAM file's physical layout on disk
- CF\$SME to set a CAM file's extent length value

Segment Directories: Segment directories exist on both standard and robust partitions at Rev. 22.1. Both partition types support both SEGSAAM and SEGDAAM directory types. Segment directories are used by a number of data management applications and products, such as Prime INFORMATION™.

On a robust partition, the subfiles in a segment directory (either SEGSAAM or SEGDAAM) are always physically organized as CAM files. The segment directory, however, retains the logical type of either SEGSAAM or SEGDAAM. Thus, it is not necessary to change existing applications employing segment directories for use on a robust partition.

It is important to know the size of the individual subfiles within a segment directory before you move the segment directory onto a robust partition. You can use either LD or SIZE to determine the size of the subfiles within a segment directory. An example of determining the size of the subfiles is shown below.

```
OK, LD ROB.P>MIDASPLUS>DEMO>@@ -SIZE
```

```
<SOFTWR>ROB.P>MIDASPLUS>DEMO (ALL access)
```

```
5 Files.
```

```
name                size  type  rbf
```

```
-----
0                    1   sam
1                    2   sam
11                   2   sam
185                  4   dam
```

```
OK, SIZE ROB.P>MIDASPLUS>DEMO>@@
```

```
1 record in sam file "ROB.P>MIDASPLUS>DEMO>0" (336 halfwords)
2 records in sam file "ROB.P>MIDASPLUS>DEMO>1" (1088 halfwords)
2 records in sam file "ROB.P>MIDASPLUS>DEMO>11" (1088 halfwords)
4 records in dam file "ROB.P>MIDASPLUS>DEMO>185" (3640 halfwords)
OK,
```

If these files are moved to a robust partition, the SAM files will each use one more record for extent maps but the DAM file will not need any more records. There will then be 8 records in SAM files and 4 records in the DAM file on the robust partition.

Directories: All directories on robust partitions are linear (that is, sequential) as opposed to hashed. Directories on Rev. 22.1 standard partitions are hashed directories. Access to hashed directories is faster than access to linear directories. Thus, users who have directories with many files should consider the change in directory access speed before moving to robust partitions.

Linear directories on robust partitions are just like ACL directories on standard partitions, except that entries are stored in a linear rather than hashed fashion. Robust partition linear directories support all the same attributes supported by standard partition ACL directories, including the DTA (date/time accessed) and DTC (date/time created) attributes.

Performance Characteristics of the Robust File System

The robust partition format does not directly affect the performance of any applications. There are three indirect effects, however, that might alter the performance characteristics of an application accessing data on a robust partition.

- Robust partitions do not support hashed directory structures. Therefore, performance can degrade if the directory contains a large number of files. See the subsection Robust Partition File System Objects, above.
- Robust partitions do not support reverse sectoring. Thus, there are specific situations when this can cause performance to degrade. See the subsection Restrictions on the Use of Robust Partitions, above.
- All other performance characteristics of robust partitions are extensively tied to the use of CAM files.

Some of the performance characteristics are

- CAM files offer improved performance when used in the proper situations. Typically, a CAM file offers better performance than a SAM or DAM file when the file is larger than one megabyte (512 disk records). This is a very generalized statement and refers specifically to the lowest level file. Within a segment directory, each individual subfile should be larger than one megabyte.
- Generally CAM files perform better than SAM or DAM files when a large number of users simultaneously access random records in the file. Simultaneous access is normally how users access a database file. This is generally not how a CAD application accesses a file, however.
- A single user reading a CAM file sequentially might notice slower performance than that available from a SAM or DAM file on a standard partition.
- If your application can call PRWF\$\$ to write 2048-byte disk records, a CAM file is faster.
- File truncation and file deletion are always faster on a robust partition because the robust structure allows the data records to be released without being unthreaded, or verified. Therefore, the DSKRAT is updated but the individual data records do not need to be read.
- Appending a large block of data can be faster on a CAM file. Because each extent is pre-allocated, a large append operation needs to extend the file only once. This benefit is visible, however, only if your application writes the entire block of data in one operation.

Prime Data Management Packages on a Robust Partition

This section discusses the effect of robust partitions on Prime data management packages.

DBMS: Most file structure issues that affect other applications do not affect DBMS. DBMS uses pre-allocated data files that do not tend to grow or shrink except during special administrative operations. Thus, there is less likelihood that the types of errors that fast `FIX_DISK` can detect will corrupt a DBMS file. Additionally, DBMS already takes advantage of the performance improvements available from CAM files.

If it is important that you run `FIX_DISK` after every system halt, robust partitions can save you time.

Prime INFORMATION: For Prime INFORMATION files, the dynamically hashed file type is best suited for robust partitions. Dynamically hashed Prime INFORMATION files are structured as segment directories with a few relatively large subfiles. (Refer to the previous subsection, Restrictions on the Use of Robust Partitions, to determine the size of the subfiles within a segment directory.)

You can use all other types of Prime INFORMATION files on a robust partition. Remember, however, that every SAM file and every SAM subfile requires one additional disk record for the extent map. If you have Prime INFORMATION statically hashed files, which are SAM files, the sizes of those files will double on a robust partition.

It is necessary to check the data integrity of the Prime INFORMATION files after running fast `FIX_DISK`. You can verify the data contained within the Prime INFORMATION file using the `FIX.FILE` command `VERIFY` option. `FIX_DISK` checks only file system structure.

MIDASPLUS: Before you move a MIDASPLUS segment directory to a robust partition, check the size of the subfiles. (Refer to the previous subsection, Restrictions on the Use of Robust Partitions, for information on determining the size of the subfiles within a segment directory.) If the subfiles are less than a few hundred disk records, you might see a performance degradation if you convert to robust partitions. If the subfiles are larger, however, you might find a performance improvement when you convert to robust partitions.

One distinguishing feature of MIDASPLUS is the index block splitting process. If you have had problems with corruption of index subfiles after a system halt, robust partitions might reduce the number of these problems because CAM files requires fewer updates to add a new record to the end of the file.

Remember to check the data integrity of the MIDASPLUS files after running fast `FIX_DISK`. `FIX_DISK` checks only the file system structure. (Please review the section Understanding the `-FAST` Option of `FIX_DISK`, above.) Use the `CREATK` command, selecting the `COUNT` option, to quickly check the data integrity of a MIDASPLUS subfile.

Prime ORACLE: Most of the file structure issues that affect other applications do not affect Prime ORACLE. Prime ORACLE uses pre-allocated data files that do not tend to grow or shrink except during special administrative operations. Thus, there is less likelihood that the types of errors that fast FIX_DISK can detect will corrupt a Prime ORACLE file. After considering these factors, base your decision on whether you need to run fast FIX_DISK after every system halt. If you do, robust partitions can save you time.

PRISAM: PRISAM INDEXED and PRISAM SEQUENTIAL files are good candidates for use on a robust partition. These files tend to expand. On a standard disk partition, the process of adding an additional record to the end of the file can be left incomplete if there is a system halt. Because CAM files require fewer steps to add a record, there is less likelihood that the operation will be incomplete.

PRISAM RELATIVE files are pre-allocated. Most of the file structure issues that affect other applications do not affect pre-allocated PRISAM files. Pre-allocated data files do not tend to grow or shrink except during special administrative operations. This means there is less likelihood that the types of errors that fast FIX_DISK can detect will corrupt a pre-allocated file.

PRISAM already takes advantage of the performance improvements available from CAM files on a standard disk partition. It is unlikely that robust partitions will noticeably impact the performance of PRISAM.

After considering these factors, base your decision on whether you need to rapidly run FIX_DISK after every system halt. If you do, robust partitions can save you time.

Non-Prime Data Management Files

If you have a data management package purchased from a company other than Prime, check with that company before converting to robust partitions. If this is not possible, try to use the previous section as a guide. Look for the important indicators. Consider these items:

- Is the file pre-allocated? Generally, pre-allocated files do not derive as much benefit from robust partitions.
- Are the subfiles large? Larger (more than 512 records) files and subfiles can improve in performance on a robust partition.
- Are there problems with corruption caused by system halts? Although robust partitions cannot prevent this corruption, robust partitions allow you to find the problem more quickly.
- Does the application have a tool to verify the integrity of the data within the file? This is an important tool that is useful on all partition types. Be aware that full FIX_DISK might have been used as a substitute for running a proper database verification routine. If this is the case, then you must change your procedures when you use robust partitions. (See the discussion in Understanding the -FAST option of FIX_DISK.)

Other Files

Other files are all of those files that are not specifically tied to a data management package. These other files might include text files, mail files, spooler files, source code, runfiles, and system configuration files. In general, these files are not appropriate for a robust partition. However, you can use the following guidelines to help in determining whether or not these files should reside on a robust partition.

- Is the file large? Larger (more than 512 records) files and subfiles can improve in performance when converted from SAM or DAM to CAM by being copied to a robust partition. Smaller files can show a performance degradation.
- SAM files gain one extra disk record when converted to CAM files. This means that a one-record SAM file occupies two records when placed on a robust partition. Generally, you do not want to use the robust partition for spooler files, batch jobs, FTS queues, or other applications that have a large number of small files.
- Are there problems with corruption caused by system halts? Although robust partitions cannot prevent this corruption, robust partitions allow you to find the problem more quickly.
- Do you understand that the tradeoff offered by using the -FAST option of FIX_DISK means that the data records are not checked? Data management packages generally have a special tool to perform this integrity check. File system objects like mailboxes, source code, and system configuration files might be left in a corrupted state for a long time before a check is made to insure their integrity. You must decide if fast FIX_DISK is appropriate for these files.

EVALUATING THE USE OF ROBUST PARTITIONS

This section poses some questions about your file system and your use of FIX_DISK. If you answer no to any of the first four questions, robust partitions are not appropriate for your system. If you answer yes to all four questions, please continue.

1. Do you want to consistently use the -FAST option with FIX_DISK to improve the MTTR (mean-time-to-recover) for your system? (Please read through the section Understanding the -FAST Option of FIX_DISK before answering this question.)
2. Do you run FIX_DISK on at least one disk partition after most system halts?
3. Do you want to know when it is not necessary to run FIX_DISK after a system halt?
4. Are you willing to run FIX_DISK on a robust partition every time the ADDISK command indicates that it is required?

Robust partitions are most beneficial when you properly prepare to use them. There are some disk partitions you will not want to convert to the robust format. Use the following questions and the Robust Partitions Evaluation Form on the next page to help you decide which disk partitions are best suited to the robust format.

Date: _____

Yes/No Do you want to use `FIX_DISK -FAST` regularly?

7-15

5. Which disk partitions do you consistently run FIX_DISK on after a system halt? Make a list of those partitions on the Robust Partitions Evaluation Form. List the physical device number (pdev) and the name of the partition. You require both the pdev and the partition name when you convert a partition to the robust format.
6. Fill in the appropriate column to indicate whether or not you intend to boot from a disk partition. Since you cannot boot from a robust partition, you cannot convert some partitions to robust partitions. This includes your command partition and might include other partitions if you frequently need to boot from a partition that is not the normal command partition.
7. Indicate the percent full that the AVAIL command reports for each partition. Some free space must exist on a disk partition in order to successfully complete the conversion process. The amount of additional space that is required depends on the size and type of files that are on the existing partition. For this information, review the paragraph Disk Space Required within the earlier section Restrictions on the Use of Robust Partitions.
8. List the names of any data management packages that use this disk partition. A data management package can be a development environment supplied by Prime, such as Prime INFORMATION and Prime ORACLE, or a data management package might be an application that your company developed. If you are unsure of the answer to this question, read the section Prime Data Management Applications on a Robust Partition, above.
9. Can the data management package verify the integrity of the data records on the robust partition? A software package with an integral transaction-recovery feature probably meets this need. Prime supplies this feature with DBMS, PRISAM, and Prime ORACLE. Other packages, such as Prime INFORMATION and MIDASPLUS, have functions available that you can use to verify the integrity of the data records.

Indicate on the evaluation form whether or not you use the data verification functionality offered with the data management package.

10. Decide whether or not you want to convert this disk partition into a robust partition. This partition is a good candidate for conversion into a robust partition if all of the following statements are true:
 - You have answered yes to the first four questions, above.
 - You do not intend to boot from this partition.
 - This partition has sufficient space.
 - Any data management package that may use this partition can effectively make use of the features offered by robust partitions.
11. Fill in the last column on the evaluation form to indicate the appropriateness of this disk partition for conversion into the robust format.
12. If you have decided that you want to use robust partitions, contact your Prime Support Analyst (call the Prime 800 number) and discuss the Robust Partitions Evaluation Form and any other details of your use of robust partitions.

CREATING ROBUST PARTITIONS

This section describes how to convert existing partitions to robust partitions and how to create new robust partitions.

Converting Existing Partitions

You can use the `FIX_DISK` option `-CONVERT_22.1` with the `-FIX` option to convert existing Rev. 22.0 partitions to Rev. 22.1 standard partitions. This results in updating the revision stamp in the `DSKRAT` of the partition and allows you to create CAM files with an unlimited number of extents. You cannot, however, convert the resulting Rev. 22.1 standard partition to a robust partition.

In order to convert existing Rev. 22.0 partitions to Rev. 22.1 *robust* partitions, you must use `MAKE` followed by the `MAKE_ROBUST` utility (discussed below) as well as available logical backup utilities such as `MAGSAV` and `MAGRST`, or `COPY`.

To convert existing pre-Rev. 22.0 partitions to Rev. 22.1 standard partitions, you also must use `MAKE`. For example, use this procedure to convert a partition to a Rev. 22.1 standard partition.

1. Use `MAGSAV` or another logical tape utility to save the data from the existing partition to tape.
2. Create the Rev. 22.1 standard partition by using `MAKE`.
3. Use `MAGRST` or another logical tape utility to restore the data to the newly created partition if you do not intend to convert this partition to a robust partition.

Alternatively, you may create a Rev. 22.1 standard partition, convert it to a robust partition and copy data to it from another partition using the `COPY` command or restore the data by using `MAGRST`. The file system organizes all files copied to the robust partition as CAM files but preserves their logical file type. All file directories on the robust partition will be linear directories.

You may mirror Rev. 22.1 standard and robust partitions in the normal manner as discussed at the end of this chapter.

The process for converting any partition to a Rev. 22.1 standard or robust partition is summarized here.

<i>Existing partition</i>	<i>Rev. 22.1 partition</i>	<i>Use</i>
pre-Rev. 22.0	standard	<code>MAKE</code>
pre-Rev. 22.0	robust	<code>MAKE</code> followed by <code>MAKE_ROBUST</code>
Rev. 22.0	standard	<code>FIX_DISK</code> (cannot convert to robust)
Rev. 22.0	standard	<code>MAKE</code>
Rev. 22.0	robust	<code>MAKE</code> followed by <code>MAKE_ROBUST</code>

Creating New Robust Partitions

MAKE_ROBUST is the utility that is necessary to create robust partitions. MAKE_ROBUST converts Rev. 22.1 standard partitions, which you create with MAKE, to robust partitions. The partition that you want to convert to a robust partition must be a Rev. 22.1 standard partition that has not been altered in any way. It must contain the standard file system objects that MAKE creates on the partition, as described in the first section of this chapter. The standard partition must also include the ACLs that are placed on the MFD by MAKE and that, by default, are in effect for all the file system objects on the partition. MAKE_ROBUST rejects any partition that is not a Rev. 22.1 standard partition or one that has been altered in any way.

To obtain MAKE_ROBUST, see item 12 in a previous section Evaluating the Use of Robust Partitions.

Follow this procedure to create a Rev. 22.1 robust partition:

1. Run MAKE to create a Rev. 22.1 standard partition.
2. Run MAKE_ROBUST to convert the standard partition to a robust partition.

MAKE_ROBUST does not replace MAKE; rather MAKE_ROBUST augments MAKE. MAKE_ROBUST does not perform any of the functions of MAKE, such as creating file system objects or doing badspot checking. MAKE_ROBUST simply alters portions of the file system structure to tell PRIMOS that the partition is robust and enforces forward sectoring as the method of file record allocation. You can also specify the robust partition minimum and maximum extent sizes with MAKE_ROBUST. If you do not specify these values, MAKE_ROBUST sets them to the default values (64 for the minimum and 256 for the maximum) whether or not you previously set them with MAKE.

MAKE_ROBUST Syntax

The format of the MAKE_ROBUST command is

MAKE_ROBUST *pdev* [*options*]

The arguments and options are as follows.

- | | |
|----------------------------------|--|
| <i>pdev</i> | Tells MAKE_ROBUST the physical device number of the partition that you are converting to a robust partition. You may use <i>pdev</i> by itself or you may use -DISK <i>pdev</i> as in the MAKE command line. |
| -HELP | Displays the command line usage and a brief description of each of the options. |
| -MIN_EXTENT_SIZE [<i>size</i>] | |
| -MINSIZ | Specifies the minimum extent <i>size</i> for CAM files on this Rev. 22.1 robust partition. If you do not specify the minimum extent size, MAKE defaults to 64 records for minimum extent sizes on robust partitions. |

-MAX_EXTENT_SIZE [*size*]

-MAXSIZ Specifies the maximum extent *size* for CAM files on this Rev. 22.1 robust partition. If you do not specify the maximum extent size, MAKE defaults to 256 records for maximum extent sizes on robust partitions.

MAKE_ROBUST Messages

Special messages indicate that MAKE_ROBUST is unable to convert the partition or is unable to interpret the command line. Appendix E explains these messages.

ADMINISTRATION OF THE FILES ON A ROBUST PARTITION

This section discusses the allocation of CAM file records on a robust partition and the possible fragmentation of robust partitions.

File Allocation on Robust Partitions

The default allocation algorithm for CAM files on all partitions is designed to reduce disk fragmentation and wasted space in files. (See the section Fragmentation of Partitions later in this chapter.) The algorithm uses the minimum and maximum extent sizes specified for the partition. You can specify the minimum and maximum extent sizes using MAKE and FIX_DISK options. The default extent sizes for Rev. 22.1 robust partitions are 64 records for the minimum and 256 records for the maximum. The default extent sizes for Rev. 22.0 and later standard partitions are 16 records for the minimum and 32 records for the maximum. The minimum and maximum extent sizes cannot be set equal to zero or equal to each other. The maximum value for extent size is 32,767 records.

You can set the extent sizes on any Rev. 22.0 and later partition. You cannot, however, set extent sizes on pre-Rev. 22.0 partitions. The default minimum and maximum extent sizes for CAM files on pre-Rev. 22.0 partitions are 16 records and 32 records, respectively. These defaults are small enough to result in keeping the amount of time required to extend a CAM file to a practical minimum (since all the records on pre-Rev. 22.0 partitions must be initialized) and still allow files to grow larger on a pre-Rev. 22.0 partition. Initialization consists of clearing out any header information that may be in the records, such as pointers to other records if these records were previously used in a file. CAM file records on Rev. 22.1 robust partitions are not initialized when they are allocated.

The Allocation Algorithm: The allocation algorithm allocates blocks, or groups, of records in graduated sizes based on the size of the file. Allocation requests equal the current file size until the maximum extent size is reached; as a result, the file size doubles at each extension. After the maximum size is reached, a block of records equal to the maximum extent size is allocated. This means that the size of a file generally grows in the following increments (using the default maximum extent size of 256 records for a robust

partition): 1, 2, 4, 8, 16 . . . 256, 512, 768, 1024 When the file is larger than the maximum extent size (256 in this case), it grows in increments of maximum extent size records.

In the default algorithm, whenever a file is extended, any free contiguous space at the end of the file is added to the file, even if it is less space than the requested number of records or less than the minimum extent size. If there is not sufficient free contiguous space at the end of the file, additional extents are allocated until the requested size is reached. If the initial request is greater than the minimum extent size, all the additional extents are at least the minimum extent size.

A user may optionally specify an extent size for individual files by using the COPY -MXL option or by using the CF\$SME system subroutine in an application program. This may be done with a large file, for example, so that fewer records are allocated on the next extension, thus saving disk space and reducing fragmentation. If the extent size is specified in this way, the default allocation algorithm is not used. Instead, whenever a file is extended, any contiguous space at the end of the file is used and an additional extent is added so that the total new space in the file is equal to the extent size specified.

Minimum Extent Size: The minimum extent size is used when there is not enough free contiguous space for PRIMOS to either double the file size or to allocate the maximum extent size when extending a file. This helps to prevent disk fragmentation and helps to keep the number of extents in a file to a minimum by ensuring that the file is not extended by using a small number of records. A CAM file performs best when it has a minimum number of extents and when the size of each extent is as large as possible. For example, if a file is to be extended by 50 records and the minimum extent size is one record, the extension could occur in 50 one-record extents. If the minimum extent size is 25 records, the file is extended in a maximum of 2 extents. Under certain circumstances, the default allocation algorithm, minimum extent size, and extent lengths are ignored. These circumstances are discussed in the later section Fragmentation of Partitions.

When PRIMOS extends a file and the number of free contiguous records available is not equal to either the current file size or to the maximum extent size, PRIMOS uses the largest block of free contiguous records that is equal to or larger than the minimum extent size. If no such block exists, the user sees the message *Insufficient free contiguous blocks* and PRIMOS cannot extend the file. In this case, files must be deleted in order to free the necessary space. In addition, the disk may be too fragmented so that you must reduce the fragmentation, as discussed in the section Fragmentation of Partitions later in this chapter.

Allocation Example: As an example of how the allocation algorithm works, assume there is a file that currently has a block of 512 records in one extent and is followed by a block of 50 records of available space. (See Figure 7-1.) Also assume that the default minimum and maximum extent size limits of 64 and 256 records are in effect and that the partition has additional unused space of 800 contiguous records.

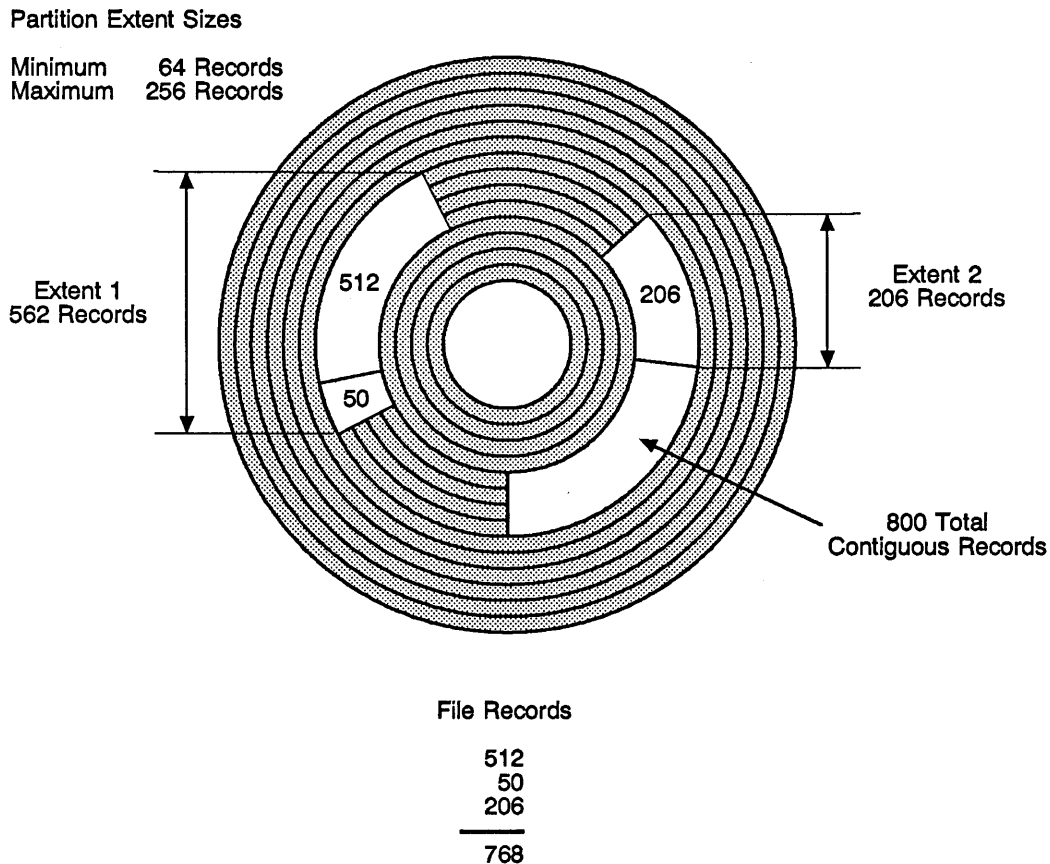


FIGURE 7-1. Allocation Example

When additional records are needed for the file, the default algorithm requests the default maximum extent size, 256 records, since the file is currently larger than 256 records. The 50 records of free space at the end of the file are allocated for the file and the current extent is expanded to include these records. To satisfy the original request, 206 records must still be allocated. Since there is sufficient space (the 800 contiguous records), the 206 records are allocated from the 800 contiguous records in one more extent. The result is that the file now has 768 records in two extents. If there had been no available extent of at least 64 records (the minimum extent size), no additional extents would have been allocated; the file size would be only 562 records (512 + 50) in one extent.

Extent Size Defaults: The defaults for minimum and maximum extent size were chosen to provide a user with an environment that allows both large files and small files to exist and not use excessive space. You can change the default minimum and maximum extent sizes to adjust to specific situations. For example, if all the files on a partition are extremely large (say 10,000 records each), the maximum extent size can be made larger to reflect the anticipated size of files. If fragmentation of files is a concern, the minimum extent size can be increased. You change the default minimum and maximum extent sizes on an existing partition by using `FIX_DISK`.

Setting Extent Sizes

To ensure the optimum performance of robust partitions and CAM files, you need to consider several issues. When you use large files, you should set the minimum extent size to accommodate extension of files in larger pieces. If the extent size is small, allocation occurs more frequently, requiring more extents for the file and potentially leading to severe fragmentation of the partition.

To further increase performance, applications should write to the end of the file in increments of a record (2048 bytes). Applications should also buffer records internally and then write them as a whole, since the logical end-of-file is then updated less often. Thus, extents for large CAM files should be multiple records, which means that a file may have unused records because the physical end-of-file is beyond the logical end-of-file. (The SIZE command shows this difference; see the example in Chapter 5 in the section Example of Creating a Robust Partition.)

An application can also set an extent size for individual CAM files on Rev. 20.0, Rev. 21.0, and Rev. 22.0 partitions. This size is used as the allocation size whenever the application extends a file. You can also set extent sizes on Rev. 22.1 standard and robust partitions; this is effective for all file extensions done by PRIMOS when individual file extent size is not set. However, if an application explicitly requests (through CF\$EXT) that a number of records be allocated for a file, the file extent size that is set for the partition is not used.

Fragmentation of Partitions

Robust partitions may become fragmented after extensive use (for example, if users create, extend, and delete files frequently). That is, the free, or unused, blocks of contiguous records remaining may be small and randomly placed on the partition. In this case, if a user attempts to add to a file, there may not be enough contiguous space to do so even though the remaining space appears to contain enough records. If this happens, PRIMOS returns an error stating that there are insufficient free contiguous records or, in certain circumstances, PRIMOS adds the records to the file and displays a warning message at the supervisor terminal stating that the partition is highly fragmented.

To help you determine the degree to which the partition is fragmented, use the LIST_CONTIGUOUS_BLOCKS command (abbreviation LCB). The format of this command is

```
LIST_CONTIGUOUS_BLOCKS [ { partition_name } ] [ options ]
LCB                    [ { -LDEV ldev } ]
```

Use the following arguments and options.

partition_name
-LDEV *ldev*

Use the name of any added partition as shown in the listing produced by the STATUS DISKS command or use -LDEV to enter the partition's logical device number. *ldev* is the *decimal*

logical device number. (You can use the PRIMOS command TYPE and the function [OCTAL *number*] to convert the octal *ldev* to decimal. For example, if the *ldev*₈ is 17 and you enter TYPE [OCTAL 17], you get 15 as the decimal result.) If you do not give a *partition_name* or a logical device number (*ldev*), the MFD of the current attach point is used.

-DISPLAY *number*
-DSP

Display the sizes of the decimal *number* of 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 you can specify is from 6 through 1024. If you specify a number outside of this range, LCB uses the default number of blocks.

-FREE *size*

Specify the minimum *size*, in decimal number of records, of the largest blocks of free contiguous space that you want displayed. The default minimum size is 32 records and the range of *size* that you can specify is from 0 through 9999 records. If you specify a size outside of this range, LCB uses the default size.

-HELP
-H

Displays the usage of the command and presents a brief description of the options.

The following information about the partition is displayed by default by the LIST_CONTIGUOUS_BLOCKS command:

- Sizes of the 20 largest blocks, or fragments, of free contiguous space larger than 32 records
- Total number of fragments of free contiguous blocks of records
- Minimum and maximum extent sizes
- Total number of records in the partition
- Total number of free records (available space)
- Percentage full (records used, total minus free, divided by total records)

You can change both the number of blocks and the minimum size of those blocks of free contiguous space to be displayed by using the -DISPLAY and -FREE options.

The following example shows the output displayed by the LCB command.

```
OK, LCB INTEG
[LCB Rev. 22.1 Copyright (c) 1988 Prime Computer, Inc.]
```

```
Partition: INTEG
```

Largest 20 blocks of free	2973	335
contiguous space larger	2211	282
than 32 records.	2035	280
	1976	195
	1894	191
	1854	146
	1588	109
	599	90
	401	86
	335	50

Number of fragments :	99	Total records :	238896
Minimum extent size :	50	Free records :	18319
Maximum extent size :	200	Percentage full :	92.33%

Note that the display of the LCB command allows you to determine the present maximum and minimum extent sizes for a partition.

Criteria for Determining Fragmentation: Use the information below to establish criteria for determining whether your partitions are highly fragmented:

- The number and size of free contiguous space larger than 32 records
- The number and size of fragments as deduced from the display
- The operations you are doing in the file system, such as creating and extending large files or small files or compiling programs

For example, from the previous LCB display, a user could create a 10,000-record file on the INTEG partition, using the first five blocks of free contiguous space even though the partition is presently 92% full. There would then still be 94 fragments of free contiguous space and 11 of them are larger than 100 records each.

A different situation exists in the example of the following partition.

```

OK, LCB FOXTST
[LCB Rev. 22.1 Copyright (c) 1988 Prime Computer, Inc.]
+-----+
| Partition: FOXTST |
+-----+
| 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 : 86.89% |
+-----+
OK,

```

The 20 largest blocks of free contiguous space are all less than 200 records, there are 3958 fragments, and the partition is about 87% full. If a user creates a 2000-record file, it will use more than the existing 20 blocks of free contiguous space. The disk is highly fragmented and is a candidate for running the procedure discussed in the following section to compact it.

Under certain circumstances, PRIMOS displays the following message at the supervisor terminal when a partition is highly fragmented:

```
*** From PRIMOS: Partition name has little free contiguous space
```

This message means that there are no blocks of free contiguous space larger than 30 records on the partition. Under these unusual conditions, PRIMOS grants requests for new CAM files or new CAM extents by using a secondary allocation algorithm but ignores any specified minimum number of records for the new extent. At least one and at most 30 records are allocated, depending on the individual request.

This behavior continues as long as the partition remains in the highly fragmented state. While in this state, the error Insufficient free contiguous blocks is not generated by PRIMOS in response to any request for a new CAM file or for a new CAM extent for an existing file. However, if you do not relieve the condition, the partition may run out of space altogether and the error Disk full is generated.

To relieve this condition, see the following section What to Do if Partitions Are Fragmented. Once the condition is relieved, the first time a new CAM file or CAM extent is created, PRIMOS detects that the condition has been relieved. When this occurs, PRIMOS displays the following message at the supervisor terminal:

```
*** From PRIMOS: Partition name has regained free contiguous space
```

At this point, PRIMOS reverts to using the standard allocation algorithm, described earlier in this chapter.

What to Do if Partitions Are Fragmented: If your partition is highly fragmented, you should do the following:

1. Save the data on the partition to tape or to another disk by using a logical backup utility, such as MAGSAV or COPY.
2. Run MAKE on the fragmented partition with any necessary options.
3. If the partition is to be a robust partition, use the MAKE_ROBUST utility to convert to robust partition format.
4. Restore the data to the newly created partition by using a logical backup utility complementary to the one used in step 1.

If you have recently run full FIX_DISK on the fragmented partition so that you are confident that all badspots have been detected, you may use the MAKE -BADLEV 0 option to reduce the time it takes MAKE to recreate the partition. Do *not* use the -NEW_DISK option as you do not want to make a new badspot file and lose the existing badspot file. (See Chapter 5, Formatting Disks — MAKE.)

FIX_DISK AND ROBUST PARTITIONS

Typically, system halts involving robust partitions require that you run FIX_DISK, generally fast FIX_DISK (FIX_DISK with the -FAST option). You must run FIX_DISK after halts that involve machine checks, power failures, and other critical failures. Halts that result in a clean shutdown during halt processing do not require you to run FIX_DISK. Approximately 25% of halts are in this latter category. (See Table 7-1.)

When to Run FIX_DISK

When you add a robust partition, PRIMOS notifies you whether it is necessary to run FIX_DISK. If you use ADDISK and the robust partition is not corrupted, the partition is added. If the robust partition is corrupted, it is not added and a message is returned requesting that FIX_DISK be run on the partition:

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

The partition is then not accessible. (For exceptions, see the following section ADDISK and Robust Partitions.)

If the halt results in an unrecoverable read or write error on the partition and it is necessary to immediately correct this situation, you should run full FIX_DISK to find the disk record that is in error and add it to the badspot file. Do not run fast FIX_DISK in this situation unless the disk is associated with an intelligent controller and is in -IC mode. Fast FIX_DISK detects the error on robust partitions only if the error is in directory entries or CAM file extent maps.

Once `FIX_DISK` finishes, the partition is robust again and can be added for both reading and writing. You should run fast `FIX_DISK` periodically to remove directory entries and data records that are not in use. Use full `FIX_DISK` to detect and remove badspots from partitions. If an unrecoverable error occurs, you can use either fast or full `FIX_DISK` to add the badspot to the badspot file.

Whenever a partition is improperly shut down, as in a system halt, the possibility of errors occurring on the partition exists. The only way to correct these errors is to run `FIX_DISK`. On Rev. 22.1 robust partitions, you can use fast `FIX_DISK` to recover directory information and repair errors.

Summary of Recommendations for Running `FIX_DISK`

Table 7-1 summarizes recommendations for running full `FIX_DISK` (`FIX_DISK -FIX`) or fast `FIX_DISK` (`FIX_DISK -FIX -FAST`) for both standard and robust partitions. Halts correspond to the types discussed in your CPU handbook in the chapter on handling halts and hangs. If you cold start after a halt, PRIMOS reminds you to run `FIX_DISK` if the disk needs repairing. If you warm start and the warm start is successful but the disk needs repair, you are not reminded, but you should follow the recommendations in Table 7-1.

The terms in the left column of Table 7-1 have these meanings:

- A successful Forced Shutdown means that the system locate buffers were flushed. An unsuccessful Forced Shutdown means that the system locate buffers were not flushed. The message at your supervisor terminal indicates whether the Forced Shutdown was successful or not.
- IDC refers to downloaded intelligent disk controllers; that is, disk controllers that are operating as intelligent disk controllers. (A message appears on the supervisor terminal at cold start when an intelligent disk controller is downloaded.)
- NDC refers to nonintelligent disk controllers (such as Model 4005 disk controllers) and intelligent disk controllers that are not downloaded and are thus operating as nonintelligent controllers.

The major difference between the ECCU and Immediate halts on machines with an intelligent disk controller (IDC) and those with a nonintelligent disk controller (NDC) is that the IDC successfully writes the record to the disk while the NDC may not successfully write the record out to disk and you may have an unrecoverable error.

The types of halts are defined in your CPU handbook.

Caution

An unrecoverable error that is not detectable by fast FIX_DISK could exist in a file in cases of disk controller failure, some processor failures, or errors in the file system software. However, since these occurrences are rare, you may use fast FIX_DISK (as summarized in Table 7-1) on a robust partition to improve recovery time. You can run full FIX_DISK at a convenient time.

TABLE 7-1. Recommendations for Using FIX_DISK

<i>Type of Halt</i>	<i>Standard Partition</i>	<i>Robust Partition</i>
Forced Shutdown (Successful)	No FIX_DISK unless quotas incorrect; then use fast FIX_DISK	No FIX_DISK unless quotas incorrect; then use fast FIX_DISK
Forced Shutdown (Unsuccessful)	Full FIX_DISK	Fast FIX_DISK
Trapped	Full FIX_DISK	Fast FIX_DISK
ECCU and Immediate (IDC)	Full FIX_DISK	Fast FIX_DISK
ECCU and Immediate (NDC)	Full FIX_DISK	Fast FIX_DISK or Full FIX_DISK See Note below

Note

With ECCU and Immediate halts and a nonintelligent disk controller (NDC), it is possible to have a corrupted data record on the disk if the NDC does not successfully write the record to disk. Fast FIX_DISK does not detect the corrupted data record on a robust partition if it is not one of the directory or extent map records. Full FIX_DISK detects the corrupted record and attempts to fix the record. If the record cannot be repaired and restored, full FIX_DISK either replaces the record with all zeroes or, if you specify the -TRUNCATE option, truncates the file at the bad record.

If you use fast FIX_DISK and the corrupted record is a data record, a subsequent attempt to access the bad record (which is not fixed by fast FIX_DISK) results in an unrecoverable read error. If this happens, your application software may handle the error; if it does not, you should run either fast or full FIX_DISK with the -ADD_BADSPOT option to add the record as a badspot and then recover the record from backup.

FIX_DISK Options

FIX_DISK has a new option, -CONVERT_22.1, that allows you to convert a Rev. 22.0 standard partition to a Rev. 22.1 standard partition capable of supporting CAM files with unlimited numbers of extents as discussed earlier in this chapter under Converting Existing Partitions. If you convert a partition to a Rev. 22.1 standard partition with FIX_DISK, you cannot then convert it to a robust partition.

FIX_DISK also has two options to support CAM files on robust partitions: -MIN_EXTENT_SIZE (abbreviation -MINSIZ) and -MAX_EXTENT_SIZE (abbreviation -MAXSIZ). These options set or change the extent sizes for Rev. 22.1 robust and standard partitions in the same manner as described for MAKE. Thus you can change the maximum and minimum extent sizes when you run FIX_DISK.

ADDISK AND ROBUST PARTITIONS

If you attempt to add a robust partition that was not properly shut down because of a system halt and, thus, has a damaged file system, PRIMOS displays this message:

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

If it is necessary to immediately access the data on a robust partition that has a damaged file system, you can do so with the new ADDISK -FORCE option. The -FORCE option forces a robust partition to be added even though the file system on the partition is no longer consistent. When a partition is added in this way, it is write-protected so that the data is available but no corruption can occur; that is, the PROTECT argument of ADDISK is implicitly invoked with -FORCE in this case. Adding a partition in this way is not considered normal operational procedure. When you do add the partition in this way, PRIMOS displays this message:

```
*** Proper shutdown of Robust Partition diskname did not take place.  
*** The Partition will be write protected.
```

A partition added with the -FORCE option is write-protected because the consequences of forcing a robust partition to be added without running FIX_DISK on it are potentially serious. Errors such as doubly allocated records, which result in pointer mismatch errors, are not detectable on a robust partition during file system operations. This type of error could result in a large loss of data if a user updates files having this error after the partition is forcibly added. The only way to resolve this error is to run FIX_DISK on the robust partition before it is added to the system. After FIX_DISK is run, you can add the partition normally.

MIRRORING ROBUST PARTITIONS

Robust partitions may be mirrored. However, keep in mind the following results of mirroring partitions.

- If the primary partition is a standard partition, the secondary partition is also a standard partition after the copy server finishes.
- If the primary partition is a robust partition, the secondary partition is also a robust partition after the copy server finishes.

See Chapter 9, Disk Mirroring, for complete information on disk mirroring.

DYNAMIC BADSPOT HANDLING

Generally, all physical disks have badspots. Previously, **static badspots** (badspots that exist on a disk when it is first partitioned) either have been found by MAKE during disk partitioning, entered by the Operator when running MAKE or FIX_DISK, or read by MAKE from a vendor flaw map. **Dynamically occurring badspots** (those static badspots that are marginally defective and missed by MAKE or those badspots that are actually developing due to disk media degradation) have been handled by FIX_DISK only when a partition is shut down for repair, not while the partition is in operation.

At Rev. 21.0 and later, an intelligent disk controller (that is, a disk controller that is microprocessor-based) can handle both static badspots and dynamically occurring badspots. This process of **dynamic badspot handling** allows PRIMOS to access an apparently error-free partition. Dynamic badspot handling can only be done by intelligent disk controllers.

This chapter discusses the following subjects in relation to dynamic badspot handling:

- Requirements
- Disk error handling
- Rev. 22.1 disks
- Physical copying of disks
- Operator commands

REQUIREMENTS FOR DYNAMIC BADSPOT HANDLING

To activate dynamic badspot handling

- You must have an appropriate revision intelligent disk controller in the system.
- You must be using Rev. 21.0 or later PRIMOS.
- The intelligent disk controller must be in **Dynamic Badspot Handling (-IC) mode**.

Dynamic badspot handling is not supported with a nonintelligent disk controller or with an intelligent disk controller in Nondynamic Badspot Handling (-AC) mode. The conversion of a partition to Dynamic Badspot Handling (-IC) mode from Nondynamic Badspot Handling (-AC) mode can be done only with an intelligent disk controller. Conversions of partitions from Dynamic Badspot Handling (-IC) mode to Nondynamic Badspot Handling (-AC) mode can be done with any disk controller.

All pre-Rev. 22.1 disks can be used on a Rev. 22.1 system. All disks partitioned by Rev. 22.1 MAKE or converted to Rev. 22.1 format by FIX_DISK contain the appropriate files for dynamic badspot handling. At Rev. 21.0 and later, an intelligent disk controller in Dynamic Badspot Handling (-IC) mode can handle all badspots by using the dynamic badspot (DBS) files.

The DBS files are inactive on Rev. 22.1 disks associated with nonintelligent disk controllers or with intelligent disk controllers in Nondynamic Badspot Handling (-AC) mode. FIX_DISK marks the badspots as being in use so that PRIMOS does not attempt to write to them.

Use FIX_DISK to convert between the two modes, Dynamic Badspot Handling (-IC) and Nondynamic Badspot Handling (-AC), of Rev. 22.1 disks. Although -IC mode disks cannot be handled on a nonintelligent disk controller, an -AC mode (nonintelligent disk controller compatible) disk can be used normally on the intelligent disk controller. However, it cannot be mirrored while in -AC mode nor does dynamic badspot handling occur on it. (See Chapter 9, Disk Mirroring.)

On either type of disk controller (intelligent or nonintelligent), use MAKE and FIX_DISK as shown in Table 8-1 to get the Rev. 22.1 partition in the mode that you want, either Dynamic Badspot Handling (-IC) or Nondynamic Badspot Handling (-AC).

TABLE 8-1. Using MAKE and FIX_DISK for -IC or -AC Mode

<i>Partition</i>	<i>Mode Desired</i>	<i>Utility</i>	<i>Remarks</i>
New	-IC	MAKE	Must have intelligent controller
New	-AC	MAKE	
Existing	-AC to -IC	FIX_DISK	Must have intelligent controller
Existing	-IC to -AC	FIX_DISK	

Note

You cannot add a Dynamic Badspot Handling (-IC) mode partition to a file system if that partition is on a disk drive associated with a nonintelligent disk controller.

DISK ERRORS

Errors that occur when a partition is in operation; that is, dynamically occurring badspots, are of two basic classes: **write errors** and **read errors**. Write errors, which are detected at the time of the write, prevent the data from being written to the disk. This type of error is usually due to a corrupted or marginal physical record header.

Read errors are of two types: **correctable** (or recoverable) and **uncorrectable**. PRIMOS handles the correctable type by attempting to read the record a specified number of times until the data is successfully read. If the read is not successful, PRIMOS attempts error correction. If either of these methods is successful, PRIMOS retrieves the correct data and does nothing about the error. The next time this record is accessed, PRIMOS may encounter an uncorrectable error. PRIMOS handles the uncorrectable type by simply displaying an error message and not retrieving the data.

Thus, dynamic badspot handling is intended to address these two cases:

- Write errors
- Correctable read-errors

Error Handling

Intelligent disk controllers handle disk errors caused by badspots on the disk in the following ways.

Write Errors and Correctable Read-errors: In the case of a write error, the intelligent disk controller remaps the data to an unused record and writes the data to that record. The controller also links the bad record to the remapped record.

In the case of the correctable read-error, the intelligent disk controller determines whether a record needs to be remapped either by retrying the read operation a specified number of times or by determining that error correction must be done. The controller then remaps the data to an unused record and writes the corrected data into the unused record. The controller then links the bad record to the remapped record.

Static Badspots: Original, or static, badspots (supplied by a vendor flaw map on the disk, supplied by the badspot file, or entered manually with MAKE or FIX_DISK) are also handled by the intelligent disk controller. The controller remaps these badspots using the same technique as for dynamically occurring badspots. The controller gets these original badspots from the dynamic badspot (DBS) file on the first partition of the disk. The DBS file contains the addresses of all known badspots on the entire physical disk. It also contains the addresses of the records in the **remapped area (RMA)** that are available for remapping badspots. The RMA is a reserved area of the physical disk designed to contain file records that are originally addressed to a badspot. The RMA also contains records that are available for remapping dynamically occurring badspots.

Uncorrectable Read-errors: Uncorrectable read-errors that occur dynamically cannot be handled by the controller. The data for the record containing an uncorrectable read-error is lost and the controller cannot recover it. If the controller were to remap the record immediately, the controller could access the new record but the data would be incorrect. Thus, an uncorrectable read-error is left for FIX_DISK to repair either when it normally scans the file system or when you use the -ADD_BADSPOT option. You must use the -ADD_BADSPOT option if the uncorrectable error occurs on a paging partition.

Although the controller cannot handle an uncorrectable read-error, it is probable that the first time the badspot is encountered, it produces a correctable read-error. In this case the controller remaps it. Thus, very few uncorrectable read-errors will be encountered.

Summary: To summarize, the purpose of dynamic badspot handling is

- To provide intelligent disk controllers with enough information to effectively handle static badspots and dynamically occurring badspots
- To provide a mechanism to convert Rev. 22.1 disks to Dynamic Badspot Handling (-IC) mode and to reverse this process
- To provide an apparently error-free disk media to PRIMOS necessary for disk mirroring

REV. 22.1 DISKS AND DYNAMIC BADSPOT HANDLING

Rev. 22.1 disks contain two files related to dynamic badspot handling: the dynamic badspot (DBS) file and the remapped area (RMA). These files are located on the first partition of a physical disk (the partition that contains the first surface, surface 0, of the physical disk). These files are not visible and are not displayed in a listing of the MFD produced by an LD command.

There is also no badspot file named BADSPT on a disk with dynamic badspot handling unless the disk is or was a Rev. 22.1 disk in Nondynamic Badspot Handling (-AC) mode and badspots exist on it. There is a file named DYNBSP in the MFD of the first partition of all Rev. 22.1 disks in either mode and whether or not badspots exist on the disk. This file is used only for security of the DBS file. It provides access checking of the DBS file by a specific ACL and provides a lock to control writing to the DBS file.

The addresses of known badspots on a physical disk are stored in the DBS file by MAKE or added to the DBS file by FIX_DISK. Any badspots encountered subsequently (dynamic badspots) are added to the DBS file by the disk controller or by FIX_DISK if the error is unrecoverable. The disk controller determines when to add a badspot to the DBS file based on the number of reattempted read operations and whether error correction is necessary.

If the DBS file becomes 80% or more full, the disk contains an unacceptable number of physical defects and a head crash is probably imminent. You are notified of a possible hardware problem by the following message from ADDISK when you add a partition on the disk to the system:

```
(Dynamic badspot remapping area is at least 80 percent full.  
Disk may have a hardware problem. Please have it checked.)
```

If you receive this message, contact your Customer Support Center.

If the DBS file becomes full, the RMA is also full and no new badspots can be added, although existing mapped badspots are still handled normally.

NEED FOR CONVERSION TO REV. 22.1 DISKS

It is necessary to convert partitions to Rev. 21.0 or later in Dynamic Badspot Handling (-IC) mode to enable disk mirroring (see Chapter 9, Disk Mirroring). To convert pre-Rev. 22.0 partitions to Rev. 22.1, you must use MAKE. Rev. 22.0 partitions can be converted to Rev. 22.1 with either MAKE or FIX_DISK.

Prior to Rev. 21.0, when badspots were marked in use in the DSKRAT, two partitions could not appear identical to PRIMOS, unless they each had no badspots or unless they each had badspots in the same locations. (Both of these events are highly unlikely.) At Rev. 21.0 and beyond, with dynamic badspot handling, the disk controller makes entire

physical disks appear error-free to PRIMOS. The intelligent controller does this by redirecting all attempted accesses to a badspot to a remapped record. Remapping is implemented on pre-Rev. 21.0 partitions when you use the `-CONVERT_21` option of `FIX_DISK` to convert them to Rev. 21.0. `FIX_DISK` modifies the record header to indicate that the record is remapped.

Nonintelligent disk controllers cannot read the modified record headers. To prevent a nonintelligent disk controller from attempting to access badspots, `FIX_DISK` marks all badspots listed in the DBS file as being in use in the `DSKRAT`. Thus, the fact that the nonintelligent controller cannot read the modified header does not matter.

`FIX_DISK` can convert a Rev. 21.0 or later partition created in Dynamic Badspot Handling (-IC) mode to make it compatible with nonintelligent disk controllers. `FIX_DISK` can also restore Rev. 21.0 and later partitions, which are compatible with nonintelligent disk controllers, to Dynamic Badspot Handling (-IC) mode.

Repartitioning Rev. 21.0 or Later Partitions

If you want to create pre-Rev. 21.0 partitions on a disk that has been partitioned as a Rev. 21.0 or later disk in Dynamic Badspot Handling (-IC) mode, you must use the `-FORMAT` option of `MAKE` to remodify the record headers. This is necessary because of differences in the way intelligent disk controllers handle badspots. Thus, you must use the `-FORMAT` option of `MAKE` in these circumstances:

- The partition is either Rev. 21.0 or later format and is in Dynamic Badspot Handling (-IC) mode and you are recreating it either as a pre-Rev. 21.0 partition or as a new Rev. 21.0 or later partition.
- The disk is new or has not been used on a Prime system before.

When you repartition a Rev. 21.0 or later first partition, you are prompted to be sure that you want to use `MAKE` on existing partitions. This is because, when you change the geometry of the first partition, the DBS file and the remapped area are changed and this affects all partitions on the disk. (See the following section discussing the first partition.) If you specify the `-FORMAT` option at this time, the prompt appears as follows:

```
Disk DBS file change! All other partitions on this disk
will be corrupted if you continue.
Do you wish to continue?
```

If you do not specify the `-FORMAT` option, `MAKE` defaults to the `-FORMAT` option in this case and prompts as follows:

```
Disk DBS file change! All other partitions on this disk
will be corrupted if you continue.
MAKE will enable -FORMAT to change the DBS file.
Do you wish to continue?
```

It is necessary to use the `-FORMAT` option in this case because the record headers contain remapping information and this information must be changed to reflect the new circumstances.

In addition, if for some reason you plan to use a version of MAKE that is at a revision earlier than the revision of the partitions you are recreating (for example, Rev. 21.0 MAKE on Rev. 22.1 partitions) to do the partitioning (not recommended), be sure the partitions you are recreating are in Nondynamic Badspot Handling (-AC) mode so that MAKE can find the earlier revision badspot file, BADSPT. If the partitions are not in Nondynamic Badspot Handling (-AC) mode, you must first run FIX_DISK with the -ALL_CONTROLLER and -FIX options to convert to Nondynamic Badspot Handling mode (-AC).

Reconstructing a Damaged Dynamic Badspot (DBS) File

If the DBS file somehow becomes inaccessible (very unlikely), you will have to do one of the following:

- Reenter the badspots manually
- Use MAKE to reconstruct the DBS file
- Use MAKE to reconstruct the DBS file from a vendor flaw map written on the disk (for disks that have this feature)

The disk continues to function on an intelligent controller if the DBS file is inaccessible, but if a new badspot occurs, the disk will stop functioning, or, if you want to convert to Nondynamic Badspot Handling (-AC) mode, you will be unable to.

Caution

All partitions on a physical disk must be of one mode, either Dynamic Badspot Handling (-IC) or Nondynamic Badspot Handling (-AC). Furthermore, it is recommended that all partitions on a physical disk be of the same revision of PRIMOS.

The First Partition of a Rev. 22.1 Physical Disk

The first partition (the partition containing surface 0) of Rev. 21.0 and later physical disk defines how all the partitions on the disk are to be treated by the disk controller. If a Rev. 21.0 or later disk with a first partition in Dynamic Badspot Handling (-IC) mode has a partition on it that is Rev. 20.0 or pre-Rev. 20.0, that partition automatically has dynamic badspot handling occurring on it.

The Rev. 20.0 or pre-Rev. 20.0 partition can continue to operate with an intelligent controller, but if pre-Rev. 21.0 utilities (such as MAKE, FIX_DISK, or COPY_DISK) are used on it, the utilities will fail. In addition, the Rev. 20.0 or pre-Rev. 20.0 partition will not function on a disk drive with a nonintelligent controller. Therefore, all partitions on a disk must be in the same mode: either Dynamic Badspot Handling (-IC) or Nondynamic Badspot Handling (-AC). It is also recommended that all partitions on a disk be of the same revision.

WARNING

Do not use incompatible versions of MAKE on the first partition of a Rev. 21.0 or later disk (that is, use Rev. 22.1 MAKE on Rev. 22.1 partitions). In addition, do not use MAKE with the -FORMAT option on an existing first partition of a Rev. 21.0 or later disk. If you do, the entire physical disk (all partitions on the disk) may be corrupted because MAKE overwrites the DBS file in these cases. Thus, the data for records that initially were to be written to badspots on any partition of the physical disk, and that were thus written to the RMA, are lost. Be sure all the data on a disk is backed up before using MAKE.

In addition, do not change the number of surfaces in the first partition of the physical disk. If you do, the basic pdev of the first partition is changed and all other partitions on that disk lose any data that was remapped to the RMA.

Changing the Geometry of the First Partition: The first partition contains the DBS file and the RMA for all badspots on the entire physical disk. If the DBS file and the RMA are damaged, any files having remapped records lose those records, since the pointers to those records are lost. In addition, the DBS file and the RMA are allocated as contiguous records on the first partition and the allocation depends on the size, or geometry, of the first partition. If the size is changed, the allocation of the DBS file and of the RMA changes. In order to change the geometry, or the basic pdev, of the first partition, follow this procedure:

1. Back up all files to magnetic tape or to disk.
2. Convert all of the partitions on the disk to Nondynamic Badspot Handling (-AC) mode using the `FIX_DISK -ALL_CONTROLLER` option.
3. Run MAKE to recreate the first partition with the new geometry and in Dynamic Badspot Handling mode.
4. Convert the other partitions on the disk to Dynamic Badspot Handling (-IC) mode by using the `-INTELLIGENT_CONTROLLER` option of `FIX_DISK`.

The DBS file and the RMA use a small amount of disk space. This space is unavailable to the file system in both modes, -IC and -AC. In Nondynamic Badspot Handling (-AC) mode, slightly more disk space is used since any data stored in the RMA must be moved to the partition holding the file associated with that data. The data is moved by `FIX_DISK` when the conversion is made.

Handling Badspots in the RMA: Badspots that are physically in the RMA are also listed in the DBS, which provides a record of them and prevents access to them. Badspots that occur dynamically in the RMA are handled in the same way; the original badspot (which was initially remapped to the RMA) is now remapped to a new remapping record, and the old remapping record (which is now a badspot) is mapped to itself to prevent any future access to it. Therefore, even if several records in the RMA degrade, no loss of performance results because the controller does not have to go indirectly to several records. It goes instead directly to the most recent remapping record.

Access by MAKE and FIX_DISK to the First Partition: MAKE and FIX_DISK require access to the first partition of a physical disk when they are working on other partitions of that disk because remapped records must be retrieved from the first partition without interference from the file system. The only way to do this is to have the first partition shut down and assigned. Thus, you must assign the first partition on the physical disk to yourself when you use FIX_DISK to perform these tasks:

- Convert pre-Rev. 21.0 partitions to Rev. 21.0 format
- Convert a partition from Dynamic Badspot Handling (-IC) mode to Nondynamic Badspot Handling (-AC) mode

In addition to assigning the first partition, the disk controller must be an intelligent controller when you use FIX_DISK to do the following:

- Convert a pre-Rev. 21.0 partition to a Rev. 21.0 partition in Dynamic Badspot Handling (-IC) mode
- Convert a Rev. 21.0 or later partition from Nondynamic Badspot Handling (-AC) mode to Dynamic Badspot Handling (-IC) mode

Note

If the first partition is the command device (COMDEV), use the FIX_DISK -COMDEV option to shut down and assign the first partition. (Remember to stop the Login server and DSM first.)

USING COPY_DISK, PHYSAV, AND PHYRST

Pre-Rev. 21.0 versions of the utilities COPY_DISK, PHYSAV, and PHYRST do not recognize the DBS and RMA files that exist on Rev. 21.0 or later revision partitions. Thus, these utilities copy the entire disk, overwriting the DBS file of the target disk and causing the RMA to contain meaningless data.

Caution

Use only Rev. 22.1 versions of COPY_DISK, PHYSAV, and PHYRST to copy files to and from Rev. 22.1 partitions. In general, use only versions of these utilities that are compatible with the revision of the partition you are dealing with. Rev. 21.0 and later versions of these utilities do not copy the DBS file and RMA to the target partition and, thus, do not corrupt the DBS and RMA files of the target partition.

Valid Data Transfers

In addition, the COPY_DISK, PHYSAV, and PHYRST utilities make data transfers between valid combinations of disk controllers and disk modes. Only these data transfers can be made:

- *From* a Dynamic Badspot Handling (-IC) mode partition on an intelligent disk controller *to* a Nondynamic Badspot Handling (-AC) mode partition on either type of disk controller
- *From* any Nondynamic Badspot Handling (-AC) mode partition *to* a second, third, or higher Dynamic Badspot Handling (-IC) mode partition (a partition starting at other than surface 0)
- *From* any Nondynamic Badspot Handling (-AC) mode partition *to* a first Dynamic Badspot Handling (-IC) mode partition (a partition starting at surface 0)

If you copy from a non-first partition to a first partition, be sure the first partition has enough space for the data *plus* the DBS file and RMA, which are already on the first partition. The first partition must also already be a Rev. 21.0 or later partition of the correct geometry (that is, having the same number of surfaces as the partition being copied).

Invalid Data Transfers

You *cannot* make the following data transfers:

- *From* a Rev. 22.1 partition *to* a first partition that is not already a Rev. 22.1 partition of the correct geometry (having the same number of surfaces as the partition being copied from)
- *From* a partition in Dynamic Badspot Handling (-IC) mode connected to a nonintelligent disk controller *to* any partition
- *From* a non-first partition *to* a first partition

If you attempt the latter copy using COPY_DISK or PHYRST, you will see the error message Cannot copy [restore] to a partition with a larger dynamic badspot remap area and the copy utility will abort.

EFFECT OF DYNAMIC BADSPOT HANDLING ON OPERATOR COMMANDS

The three operator commands, ADDISK, MAKE, and FIX_DISK, are affected by Dynamic Badspot Handling as discussed in the following paragraphs.

ADDISK and Dynamic Badspot Handling

If you attempt to use ADDISK to add a Dynamic Badspot Handling (-IC) mode partition from a disk drive with a nonintelligent disk controller, an error message is displayed stating that you must run FIX_DISK to convert the disk in order to allow access to it. The partition is not added to the system. For example:

```
OK, ADDISK 40463
*** Disk 40463 - A dynamic badspot handling disk cannot be
started on this controller. Run FIX_DISK to convert to
an -All_Controller format disk.
ERI
```

If you add a Dynamic Badspot Handling (-IC) mode partition to the system using ADDISK and the RMA has reached 80% or more of capacity, a warning message is displayed. This message warns you that an unusual number of dynamic badspots have occurred and that the disk may be approaching a head crash or some other hardware problem and should be checked. The disk is added, however. For example:

```
OK, ADDISK 103024
Dynamic badspot remapping area is at least 80 percent full.
Disk may have a hardware problem. Please have it checked.
Starting up Revision 21 partition "EAGLE"
OK,
```

If you receive this message, contact your Customer Support Center.

MAKE and Dynamic Badspot Handling

You use MAKE to create partitions having dynamic badspot handling. If you do not specify the mode, the default mode is based on the type of controller that the disk you are partitioning is associated with: either a nonintelligent disk controller, in which case the default mode is Nondynamic Badspot Handling (-AC) or an intelligent disk controller, in which case the default mode is Dynamic Badspot Handling (-IC). Use Nondynamic Badspot Handling (-AC) mode only for removable disks (80MB and 300MB SMDs), provided that you do not want to mirror them. Use the default in all other cases so that the controller capability is used.

When you use MAKE, you must create the first partition of a physical disk (the partition that contains surface 0) before you create any other partition on that disk. You create the first partition of the physical disk first because the DBS file and the RMA must be on the first partition and these files are used for badspot handling on all other partitions of the disk. Prior to Rev. 21.0, it made no difference in what order you partitioned disks except if you wished to keep redundant badspot files. However, it is always good practice to start with the first partition and proceed in order and it is mandatory that you do this with partitions created for dynamic badspot handling.

The format of the MAKE command to create partitions having dynamic badspot handling is

```
MAKE -DISK pdev -DISK_TYPE type -INTELLIGENT_CONTROLLER
```


You must use the `-DISK_TYPE` option to specify to `MAKE` the type of disk you are creating. `MAKE` needs to know the type of physical disk to get the disk geometry information in order to build the correct size DBS file and RMA. You may also use the `-FORMAT` and `-NEWDSK` options. (See the previous section `Creating pre-Rev. 21.0 Partitions`.) If you do not specify the `-FORMAT` option, `MAKE` will use this option by default to modify the record headers and display the message `MAKE` will enable `-FORMAT` to change the DBS file. If you do not specify the `-NEWDSK` option, `MAKE` displays the following message and continues.

```
Unable to get badspots from device: pdev
Continuing with make. (MAKE)
```

To create a Rev. 21.0 or a Rev. 22.0 partition, use the `-DSKREV` option with either `-AC` or `-IC`. If you use the `-AC` and `-IC` mode switches with a `-DSKREV` parameter less than 21, an error message is displayed and `MAKE` aborts. For example:

```
OK, MAKE -DISK 42022 -DSKREV 20 -IC
**** MAKE <Rev. 21.0> **** Copyright (c) 1986 Prime Computer, Inc.
-AC/-IC mode not available with a DISK_REVISION earlier than 21.
ERI
```

If you do not specify the `-DSKREV` option with either `-IC` or `-AC`, `MAKE` creates a Rev. 22.1 partition.

FIX_DISK and Dynamic Badspot Handling

`FIX_DISK` leaves a partition in its current mode unless you specifically instruct `FIX_DISK` to convert to either Nondynamic Badspot Handling (`-AC`) mode or to Dynamic Badspot Handling (`-IC`) mode. You can also have `FIX_DISK` display a summary of the DBS file.

Caution

In order to have `FIX_DISK` convert a pre-Rev. 21.0 partition to Rev. 21.0 format, you must convert the first partition (the partition that contains surface 0) before you convert any other partition on that physical disk. This is necessary because the DBS file and the RMA must be on the first partition and `FIX_DISK` must have access to them to repair other partitions on the disk.

`FIX_DISK` does the following with respect to dynamic badspot handling:

- Converts a pre-Rev. 21.0 partition to a Rev. 21.0 partition having either Dynamic Badspot Handling (`-IC`) mode or Nondynamic Badspot Handling (`-AC`) mode
- Switches a Rev. 21.0 or later Dynamic Badspot Handling (`-IC`) mode partition to a Nondynamic Badspot Handling (`-AC`) mode partition
- Switches a Rev. 21.0 or later Nondynamic Badspot Handling (`-AC`) mode partition to a Dynamic Badspot Handling (`-IC`) mode partition

Note

To convert a partition to Dynamic Badspot Handling (-IC) mode, the physical disk must be associated with an intelligent disk controller.

You can use either MAKE or FIX_DISK to convert a partition from Rev. 20.0 to Rev. 21.0 format. You can use only MAKE to create a partition as a Rev. 22.0 format partition because there is no FIX_DISK option to convert partitions to Rev. 22.0. You can convert a Rev. 22.0 partition to a Rev. 22.1 standard partition with FIX_DISK. In order to convert any partition to a robust partition, you must use MAKE to create a Rev. 22.1 standard partition first. See Chapter 7 for details on robust partitions.

For simple conversions involving only a change in controller mode and involving no disk geometry, or basic pdev, changes, FIX_DISK is fastest to use.

To convert a partition from Rev. 19.0 or earlier to Rev. 21.0 format with FIX_DISK, it is necessary to convert to Rev. 20.0 first, because of hashed directories, and then convert to Rev. 21.0 format. See Chapter 5 for a discussion of converting pre-Rev. 21.0 revision partitions to Rev. 21.0 or later partitions.

If you are converting a partition from pre-Rev. 21.0 to Rev. 21.0 with the -CONVERT_21 option and you do not specify the controller mode with either -AC or -IC, FIX_DISK determines the type of disk controller that is associated with the disk drive and sets the partition controller mode accordingly. Rev. 21.0 format partitions are compatible with Rev. 22.0 and Rev. 22.1 standard partitions.

FIX_DISK Conversion to Nondynamic Badspot Handling (-AC) Mode: Use FIX_DISK to convert a partition that is in Dynamic Badspot Handling (-IC) mode to a partition in Nondynamic Badspot Handling (-AC) mode. To make this conversion, FIX_DISK must relocate file records, which were remapped to the RMA in Dynamic Badspot Handling mode, to the partition containing the parent file.

Note

To convert a partition, you must use the ASSIGN command to assign to yourself the first partition (the partition containing surface 0) of the physical disk and the partition to be converted. If the first partition is the command device, use the -COMDEV option to shut down and assign it.

To convert to Nondynamic Badspot Handling (-AC) mode, use the FIX_DISK format

FIX_DISK -DISK pdev -FIX -ALL_CONTROLLER

FIX_DISK Conversion to Dynamic Badspot Handling (-IC) Mode: To convert a Rev. 22.1 Nondynamic Badspot Handling (-AC) mode partition to a format that allows full utilization of dynamic badspot handling and mirroring on an intelligent disk controller, use FIX_DISK to convert and allow access to the badspots defined in the DBS file. To convert a partition to Dynamic Badspot Handling (-IC) mode, the physical disk must be associated with an intelligent disk controller.

To convert to Dynamic Badspot Handling (-IC) mode, use the `FIX_DISK` format

`FIX_DISK -DISK pdev -FIX -INTELLIGENT_CONTROLLER`

Note

You must use the `-COMDEV` option followed by the `pdev` of the command device under the following conditions:

- You are converting a partition to Dynamic Badspot Handling (-IC) mode, either from a pre-Rev. 21.0 format or by switching a partition from Nondynamic Badspot Handling (-AC) mode and
- The partition you are converting, or switching, is not the first partition (the partition containing the first surface of the physical disk) but
- The first partition is the command device (`COMDEV - ldev 0`)

You should stop the Login server and DSM before using the `-COMDEV` option.

For example, if you are switching a Rev. 22.1 partition from -AC mode to -IC mode and the partition is the second partition on a physical disk that contains the command device as logical device 0 (`ldev 0`), specify the `-COMDEV` option in a command line like this to tell `FIX_DISK` where to find the command device with the DBS file.

OK, `FIX_DISK -DISK 71020 -FIX -IC -COMDEV 3420`

`FIX_DISK` Display of DBS File: You can display the dynamic badspot (DBS) file and save the display in a COMO file. This can be useful if the DBS file is somehow damaged because you can then use `FIX_DISK` or `MAKE` to manually reenter the badspots.

To display the DBS file, use the `FIX_DISK` option `-DUMP_DBS` (abbreviation `-DDBS`) in this command format

`FIX_DISK -DISK pdev -DUMP_DBS`

In order to display the DBS file of a disk, you must have the first partition of that disk assigned to you. You can display the DBS file without specifying the `-FIX` option as in the example above; in this case, use the `pdev` of the first partition.

You can also have the DBS file displayed while you are repairing some other partition on the physical disk. It is displayed when `FIX_DISK` finishes the repair operation. If the disk is in Nondynamic Badspot Handling (-AC) mode, the DBS file is displayed but it may not be up-to-date; it is updated in Dynamic Badspot Handling (-IC) mode.

After `FIX_DISK` has finished repairing files or updating the badspot file, the DBS file is displayed in the following tabular format.

DBS file version *version_number*.
 DBS file has *number_of_records* records.
 Number of badspots = *number_of_badspots*.
 File last modified by *controller/Primos*. /* No Line if no changes

Badspot	Remap Record
Cyl, Head, Sector	Cyl, Head, Sector
(badspot_1)	(remap_1)
(badspot_2)	(remap_2)
.	.
.	.
.	.

MESSAGES ASSOCIATED WITH DYNAMIC BADSPOT HANDLING

When the disk controller remaps a record, a disk error message is displayed at the supervisor terminal. (See Appendix D for the message format.) At the end of the message is the comment Bad record remapped. indicating that the controller has remapped the record.

If you specify the `FIX_DISK -INTELLIGENT_CONTROLLER` option and you are repairing the first partition, the following message is displayed:

Activating DBS file

Otherwise, this message is displayed:

Deactivating DBS file

Appendices A, B, and C list other messages associated with dynamic badspot handling.

DISK MIRRORING

This chapter discusses **disk mirroring**, the use of two logically equivalent partitions with identical data written to them.

For many applications, any circumstance that causes processing of data to stop is undesirable. In the case of a disk failure, recovering from the failure requires downtime. The application must be stopped, the disk must be repaired or replaced, and recovery procedures must be run to restore the system to its original state.

THE PURPOSE OF MIRRORING

The purpose of disk mirroring is to increase system availability by making it possible to process with pairs of logical disks. These logical disks are equivalent such that if one fails, the other is an exact duplicate and is available for use. The transition to the use of the duplicate disk is automatic.

Disk mirroring allows PRIMOS to

- Mirror partitions on different disk drive units (which thus have different disk drive unit numbers) of the same disk controller
- Mirror partitions on disk drive units that have the same disk drive unit numbers but are on different disk controllers
- Mirror partitions on different disk drive units of different disk controllers
- Continue processing on one partition if the other fails
- Copy a partition as a background process while the partition pair is being mirrored (a catch-up copy)

When you mirror partitions, all records written to a partition, called the **primary partition**, are also written to another partition, called the **secondary partition**. Thus, all write operations are duplicated.

Reading of records is not duplicated. Reading is split so that the records in the first half of the partition are read from the primary partition and the records in the second half are read from the secondary partition. This process reduces the average time it takes to read a record (compared to reading all records from one of the partitions) because the average seek time is reduced.

MIRRORING REQUIREMENTS

The requirements for disk mirroring are as follows:

- Both the primary partition and the secondary partition must be in disk drives associated with downloaded intelligent disk controllers that are capable of dynamic badspot handling (the IDC1).

Note

If the primary partition and the secondary partition are on different disk controllers, the controller is eliminated as a common point of failure. In addition, performance improves when you are mirroring partitions on different disk controllers.

- The two partitions must be Rev. 22.0 or Rev. 21.0 partitions.
- The two partitions must be in Dynamic Badspot Handling (-IC) mode so that dynamic badspot handling can take place on them.
- The two partitions must be on the same model disk; that is, they must be on the same physical disk types.
- The two partitions must be identical with respect to size (number of surfaces) and position (starting surface number) on the physical disks. (They thus will have identical basic pdevs before the pdev is modified for disk drive unit number and disk controller address.)
- A maximum of 128 partitions can be mirrored at one time; that is, there can be a maximum of 64 pairs of mirrored partitions.
- Assigned partitions cannot be mirrored.
- It is not possible to mirror both the paging portion and the file system portion of a split partition. Generally this means that only the paging portion can be mirrored because you start the paging mirror at system startup by a configuration directive. In addition, if the paging portions of two partitions are mirrored, it is *not* possible to add the file system portion of either partition with the ADDISK command.
- One or more of the following directives must be in the configuration file. (See the section Configuration Directives for Mirroring below.)

```
MIRROR
COMDVM pdev
PAGINM pdev1 [. . . pdev8]
```

- You can mirror robust partitions; however, the type of partition that results (either standard or robust) depends on what the primary partition is. See Mirroring and Robust Partitions in Chapter 7.

Since the catch-up copy facility in the mirroring process makes a physical copy of the primary partition that you want to mirror to the secondary partition, the resulting secondary partition becomes the same revision (either Rev. 21.0, Rev. 22.0, or Rev. 22.1) and the same type of partition (standard or robust) as the primary partition.

Performance

If you mirror one partition of a physical disk, you should mirror every partition on that physical disk for best performance. In addition, configure each of the two partitions of a mirrored pair on different disk controllers, if possible. This provides better reliability and performance because if mirrored partitions, and thus their physical disks, are associated with a single controller, the controller can be a single failure point for both partitions.

Caution

You can mirror only some of the logical partitions on a physical disk. However, doing this will have a negative performance impact if there is much activity on the nonmirrored partition(s). It is thus strongly recommended that you mirror all the partitions on a physical disk if you plan to mirror any partitions on that physical disk.

CONFIGURATION DIRECTIVES FOR MIRRORING

You can activate mirroring or initiate mirroring of the command device (COMDEV - ldev 0) or the paging devices at system startup by the use of directives in the configuration file. If you want to mirror any partitions, at least one of the following directives must be in the configuration file:

```
MIRROR
COMDVM pdev
PAGINM pdev1 [. . . pdev8]
```

Using MIRROR

Use MIRROR if neither the command device (COMDEV) nor any of the paging devices is to be mirrored at system startup but you want to be able to mirror partitions after startup. If the MIRROR directive is in the configuration file, you can then mirror any partitions after system startup.

If either or both of the directives COMDVM and PAGINM are in the configuration file, mirroring is turned on at cold start for the command device (COMDEV) in the case of the

COMDVM directive and for the paging devices in the case of the PAGINM directive. In addition, these directives have the same effect as the MIRROR directive; that is, file system partitions can be mirrored after system startup.

Using COMDVM

Use COMDVM to initiate mirroring of the command device (COMDEV) at system startup. The COMDVM directive must follow the COMDEV directive in the configuration file. For example:

```
COMDEV 2420
COMDVM 2520
```

You specify the pdev on which to mirror the COMDEV with the COMDVM directive. If the specified pdev cannot be used for any reason, a message is displayed and the COMDEV starts without mirroring. You can start mirroring of the COMDEV after system startup by using the MIRROR_ON command (described later in this chapter).

Using PAGINM

The PAGINM directive must follow the PAGING directive to mirror paging devices at system startup. It is not possible to set up mirroring in conjunction with the pre-Rev. 21.0 PAGDEV or ALTDEV directives.

Use the PAGINM directive to specify the pdevs (*pdev1* . . . *pdev8*) on which to mirror the corresponding pdevs specified with the PAGING directive. If a pdev used in the PAGING directive does not have a mirror, a 0 must be used in the corresponding position in the PAGINM directive. For example:

```
PAGING 1020 100421 71023 100463
PAGINM 1060      0 71065      0
```

In this example, the second (100421) and fourth (100463) paging partitions are not mirrored, whereas device 1020 is mirrored on an identical partition on disk drive 0 associated with disk controller 1 at address 26₈ (1060), and device 71023 is mirrored by an identical partition on disk drive 2 associated with disk controller 1 at address 26₈ (71065).

If a mirrored paging partition specified by *pdev* cannot be used for any reason, a message is displayed and the associated paging partition starts without a mirror. You can still start a mirror for the paging partitions by using the MIRROR_ON command after the system has started; a catch-up copy then starts.

CATCH-UP COPY

The **catch-up copy** facility is a mechanism for making a *physical* copy of the primary partition on the secondary partition while mirroring is in effect. The catch-up copy facility allows a mirrored pair to be set up after a primary partition has begun operation. In this case, you are prompted to be sure you want the catch-up copy to start. This facility also allows two partitions that are started together but are not identical to be updated so that they are identical.

WARNING

The copy server copies *from* the primary partition *to* the secondary partition. Be sure that you know which is the primary and which is the secondary partition when you use the MIRROR_ON command. You could destroy current data if you use the command incorrectly.

If your primary partition is not up-to-date (it is the older of the two mirrored partitions), change the positions of the pdevs in the MIRROR_ON command so that the secondary is now the primary. If you start up the mirrored pair with the older partition as the primary, the copy server copies the older data to the secondary. The more recent data of the secondary is thus destroyed. It is especially important to change the positions of the pdevs in this case if the MIRROR_ON command is in your PRIMOS.COM1 file and you cold start your system.

If two partitions to be mirrored are started simultaneously and the partitions have different shutdown stamps indicating that the partitions were not shut down at the same time and, thus, are not identical, the catch-up copy facility starts automatically. However, paging partitions are not copied when they are started simultaneously at cold start. In all cases, the shutdown stamp is set at shutdown and cleared at startup.

The catch-up copy is accomplished by a server while the partitions are on line. The server reads a record from the primary partition, writes that record to the secondary partition, and then advances to the next record on the primary partition. The server activity is prevented from interfering with reading and writing by PRIMOS on the mirrored partitions. Until the catch-up copy is complete, records are read by PRIMOS from the primary partition only. If the catch-up copy facility fails for any reason (for example, the copy server logs out), the mirror is disabled.

In place of using the catch-up copy facility, you can use Rev. 22.0 COPY_DISK or PHYSAV and PHYRST to make the two partitions physically identical. However, in this case, the partitions must remain shut down until the copy is complete.

Startup of the Catch-up Copy Facility

The copy server phantom performs the catch-up copy. The copy server is started by the following sequence of actions:

1. The MIRROR_ON command is issued at the supervisor terminal or a configuration directive (such as COMDVM) that establishes a mirror is read at cold start.
2. The shutdown stamps from the DSKRAT of each partition in the mirrored pair are compared if the partitions are started simultaneously. (No shutdown check is made on paging partitions.) If the shutdown stamps are identical or if the two partitions are paging partitions being mirrored at cold start, there is no need to do a catch-up copy and mirroring starts.

If the shutdown stamps are not identical or if the two partitions are not started simultaneously, step 3 is carried out.

3. You are queried to be sure that mirroring is desired. This is necessary because *the catch-up copy overwrites whatever data is on the secondary partition* with data from the primary partition.
4. The copy server process starts, updating the secondary partition of the mirrored pair.
5. A message is displayed upon successful completion of the copy. If the copy fails, a message is displayed indicating this failure and the mirror is disabled (breaks).

EFFECT OF MIRRORING ON OPERATOR COMMANDS

Mirroring affects the STATUS command and the processing of the PRIMOS.COMI file and results in messages from the ADDISK and SHUTDN commands as discussed in the following paragraphs.

The STATUS DISKS Display

As shown in the example below, when you issue the STATUS DISKS command, a table of mirroring information is displayed if mirroring is in effect for any partitions on your system. The column headings having to do with mirroring are displayed whether or not mirroring is activated.

The pdev of the primary partition is displayed in the Pdev field of the STATUS DISKS display unless a problem on the primary partition has caused an automatic switch to the secondary partition. In that case, the pdev of the secondary partition is displayed in the Pdev field. The status of each mirrored partition, either Active or Inactive, is displayed in the State field indicating which partition is being used. "Active, copying" means that a catch-up copy is in progress and "Active, copy needed" means that a catch-up copy is necessary but the catch-up copy process is not running yet.

If you issue the STATUS ALL command at the supervisor terminal, mirroring information is displayed for paging partitions; otherwise it is not. The format in either case is the same as in this example:

Disk	Ldev	Pdev	System	Robust	Mirror		
					Primary	Secondary	State
ADMINS	0	4420			4420	4622	Active
PURCHS	1	122420			122420	122462	Active, copying
HUMRES	2	23463			23423	23463	Inactive, primary off
ACCTGS	3	14321			14321	14327	Inactive, secondary off
PAYROL	4	14121					
MAINT	5	120421			120421	120721	Active, copy needed
ENGRG1	6		ENG				
BLDGS	7		ENG				
GRNDS	10		ENG				

Using the ADDISK Command

You use the ADDISK command to start partitions and to make file system partitions known to the system. ADDISK checks to be sure that a partition to be added is not currently in use in a mirrored pair. If the partition is in use in a mirror, the disk is not added and a message is displayed at the supervisor terminal:

PDEV *pdev* conflicts with assigned or paging or mirrored device. (addisk)

If you attempt a warm start on your system and it is successful, you may encounter an error subsequent to the warm start when adding or mirroring disks that are associated with an intelligent disk controller. You may see the following message or a similar message from the MIRROR_ON command.

Could not OPEN PDEV *pdev* due to controller or device errors. (addisk)

If this message does appear, wait until the following message appears at the supervisor terminal.

DLL and init ICOP complete (cntrlr_address) - (disk_init)

The intelligent disk controller is now downloaded again and you can now add or mirror the disks that you previously attempted to add or mirror.

This problem occurs only on disk drives associated with intelligent disk controllers and only on those disk drives where all the logical disks (partitions) on a physical disk were shut down prior to the warm start.

Using the SHUTDN Command

You use the SHUTDN command to take a partition out of the file system. After you issue the SHUTDN command, PRIMOS checks to be sure that the partition to be shut down is not currently in use in a mirrored pair. If the partition is being mirrored, the partition is not shut down and the following message is displayed at the supervisor terminal:

Must use MIRROR_OFF to shut down mirrored PDEV *pdev*. (shutdn)

In this case, use the `MIRROR_OFF` command (described later in this chapter) to shut down the mirrored partition. Note, however, that the `SHUTDN ALL` command shuts down mirrored partitions as part of the system shutdown process.

The PRIMOS.COMI File

If you initiate mirroring of the command device at startup with the `COMDVM` configuration directive or initiate mirroring of other file system partitions at startup by placing the `MIRROR_ON` command in the `PRIMOS.COMI` file, and the mirrored partitions are not identical such that a catch-up copy must start, you are queried with the following message to be sure that you want to continue:

A catch-up copy of primary device *pdev1* to secondary device *pdev2* will be started.
Are you sure you want to continue?

The `PRIMOS.COMI` file stops and waits for your response. After you enter your response, `PRIMOS.COMI` continues with the startup process.

Note

Although the `PRIMOS.COMI` file stops at this point, messages from the startup of other system servers and phantoms may cause the above prompt to scroll off the screen of a video display terminal. If you have the `MIRROR_ON` command in your `PRIMOS.COMI` file and your system appears to be hung, enter a return. If the `PRIMOS.COMI` file did stop for this reason, it will now continue. Mirroring does not start but you can manually start the mirroring process with the `MIRROR_ON` command later. In the case of mirroring the command device with the `COMDVM` directive, the prompt is issued before processing of the `PRIMOS.COMI` file.

PRIMOS MIRRORING COMMANDS

You can activate disk mirroring at system startup by using configuration directives or, while the system is running, by using the `MIRROR_ON` command. To turn off mirroring, use the `MIRROR_OFF` command. You can issue the `MIRROR_ON` and `MIRROR_OFF` commands, which are described in this section, only from the supervisor terminal.

Caution

Do not use the `MIRROR_ON` or the `MIRROR_OFF` command while in the `RESUS` environment because they may request terminal input. See the *DSM User's Guide* for details.

The MIRROR_ON Command

To start the mirroring process, use the following command format. The options are described below.

Syntax:

```
MIRROR_ON pdev1 pdev2 [ { -HELP
MON                                     } ]
```

Options:***pdev1 pdev2***

Specifies the physical device numbers of the two partitions that are to be mirrored. The geometry of the two partitions must be the same; that is, they must have the same basic pdev. *pdev1* and *pdev2* cannot be assigned partitions; if either is assigned, an error message is displayed. pdevs are not needed with the -HELP option. The requirements for mirroring partitions are described under Mirroring Requirements, earlier in this chapter.

pdev2 must not be known to the system, that is, it must not be a currently added file system partition, an assigned partition, or a paging partition, and it must not be mirrored currently. A message is displayed at the supervisor terminal if *pdev2* is part of a currently mirrored pair.

pdev1 can be unknown (in which case it is assumed to be a file system partition) or it can be a known (added) file system or paging partition that is not currently being mirrored.

-HELP
-H

Displays command usage similar to the above syntax. For an online explanation of the options, type HELP MIRROR_ON.

-PRIORITY_SELECT
-PRISEL

Allows you to forcibly mirror a local dual-ported disk, which was added to another system, after the other system has halted. A dual-ported disk is physically connected to two CPUs through the dual-porting option package but can be accessed by only one CPU at a time. If the CPU having the logical connection to the dual-ported disk halts, the other CPU can then access the disk. This provides for greater file system availability. Do not use this option if the other system is running.

WARNING

Do not attempt to priority select a dual-ported disk if the other system has control of the disk drive. If you do, you may corrupt the data on the disk. You should only priority select a disk if your system has control of the dual-ported disk or if the other system halted while it had control of the dual-ported disk.

The MIRROR_ON command is valid only if one or more of the following directives is in the configuration file when the system is started:

```
MIRROR
COMDVM pdev
PAGINM pdev1 [. . . pdev8]
```

The following prompt is issued when a request is made to mirror a pair of partitions that are not identical, based either on their shutdown stamps or on not being started or stopped simultaneously.

A catch-up copy of primary device *pdev1* to secondary device *pdev2* will be started.
Are you sure you want to continue?

If you want the mirroring process to start and a catch-up copy to proceed as indicated, answer YES but be aware that the primary is always copied to the secondary. (See the warning in the Catch-up Copy section). If you start mirroring by using a COMINPUT file with the MIRROR_ON command, the COMI file stops and awaits your response from the supervisor terminal. The COMI file resumes processing after you respond.

Note

While PRIMOS is awaiting your response, users cannot login and the system may appear to be hung. When you use the MIRROR_ON command, you should promptly answer the above prompt.

The maximum number of partitions that can be mirrored is 128. If you attempt to start more than 64 mirrored pairs of partitions, an error message is displayed at the supervisor terminal.

The MIRROR_OFF Command

To stop the mirroring process, use the following command format.

Syntax:

```
MIRROR_OFF pdev1 pdev2 { -SHUT_BOTH [-FORCE]
MOFF                  { -SHUT_PRIMARY
                      { -SHUT_SECONDARY
                      { -HELP
```

You must specify one, and only one, of the four options. If the copy server is active (that is, a catch-up copy of the primary partition to the secondary partition is in progress), you cannot use the -SHUT_PRIMARY option but you can use either the -SHUT_SECONDARY or the -SHUT_BOTH option. If you use either the -SHUT_SECONDARY option or the -SHUT_BOTH option, PRIMOS attempts to log out the copy server. If PRIMOS cannot log out the copy server, an error message is displayed. You must then log out the copy server and reissue the MIRROR_OFF command.

Options:

pdev1 pdev2

Specifies the physical device numbers of the two partitions that are presently mirrored and that you want to shut down as a mirrored pair. Not needed with the -HELP option.

pdev1 and *pdev2* must be a mirrored pair; if they are not, an error message is displayed at the supervisor terminal. *pdev1* must be the primary partition and *pdev2* must be the secondary

partition; if they are not, an error message is displayed at the supervisor terminal. Use the STATUS DISKS command or the LIST_DISKS -LOCAL command to list the pdevs of all currently mirrored partitions.

-SHUT_BOTH
-SB

Turns off mirroring and shuts down both partitions, giving them identical shutdown stamps. If mirroring was already disabled, an error message is displayed at the supervisor terminal and the two partitions do not get identical shutdown stamps.

This option is invalid with paging partitions. Paging partitions can be shut down only at system shutdown with the SHUTDN ALL command. If paging partitions are specified, an error message is displayed.

-FORCE
-F

Operates like the -FORCE option of the SHUTDN command. -FORCE makes it possible to shut down a partition even if problems prevent it from being read. Use only with MIRROR_OFF -SHUT_BOTH.

When you use the -FORCE option and disk errors occur, the shutdown stamps may not be updated; thus, the next attempt to mirror the pair requires a catch-up copy. The display of disk error messages immediately after MIRROR_OFF -SHUT_BOTH -FORCE indicates that the shutdown stamps are not identical on both partitions. The partition for which the error message occurs is now the older partition and the other is the most up-to-date and should now be used as the primary partition.

-SHUT_PRIMARY
-SP

Turns off mirroring, if it was on, and shuts down the primary partition. A subsequent attempt to mirror this primary partition and secondary partition pair requires a catch-up copy because the shutdown stamps cannot match since the two partitions were not shut down at the same time.

Caution

The primary partition is now the older of the two and a catch-up copy results in the primary being copied to the secondary. Thus, make the secondary partition the primary partition in a subsequent MIRROR_ON command.

The -SHUT_PRIMARY option is valid only if mirroring is on or if the primary partition is inactive; otherwise an error message is displayed at the supervisor terminal. The secondary partition is now indistinguishable from an ordinary file system or paging partition. However, the primary partition is left in an indeterminate state because a user may have been writing to a file and the file is not saved. Using the -SHUT_PRIMARY option in this way is the same as using the SHUTDN command

to shut down a file system partition without warning and logging out users so that they can close all files. If you use this option, also use normal shutdown procedures.

If read requests or write requests are pending on the primary partition after five seconds, an error message is displayed. This should rarely occur. If it does, this command option can be reissued until it works.

-SHUT_SECONDARY
-SS

Turns off mirroring, if it was on, and shuts down the secondary partition. A subsequent attempt to mirror this primary partition and this secondary partition requires a catch-up copy because the shutdown stamps cannot match since the partitions were not shut down at the same time. This option is valid only if the mirror is on or if the secondary partition is inactive; otherwise an error message is displayed. The primary partition is now indistinguishable from an ordinary file system or paging partition. However, the secondary partition is left in an indeterminate state because a user may have been writing to a file and the file is not saved. Using the -SHUT_SECONDARY option in this way is the same as using the SHUTDN command to shut down a file system partition without warning and logging out users so that they can close all files. If you use this option, also use normal shutdown procedures.

If read requests or write requests are pending on the secondary partition after five seconds, an error message is displayed. This should rarely occur. If it does, this command option can be reissued until it works.

-HELP
-H

Displays command usage similar to the above syntax. For an online explanation of the options, type HELP MIRROR_OFF.

When you use the -SHUT_PRIMARY or the -SHUT_SECONDARY option, PRIMOS displays the following message and prompt to be sure that you intend to break the mirror.

The mirror of primary device *pdev1* and secondary device *pdev2* will be broken.
Are you sure you want to continue?

While PRIMOS is awaiting your response, users cannot login and the system may appear to be hung. When you use the MIRROR_OFF command, you should promptly answer the above prompt.

ERRORS

If one of the two mirrored partitions fails, all records are read from the remaining good mirrored partition (the **survivor**). Similarly, records are written only to the survivor. This assumes that the mirrored partitions are identical and a catch-up copy is not in progress at the time of the failure.

The failure of one of the mirrored partitions and disabling of the mirroring process is referred to as **breaking the mirror**. Users are not aware of the transition. However, the STATUS DISKS display and messages displayed at the supervisor terminal indicate that the transition occurred. Failures on the remaining partition are handled like failures on any unmirrored partition.

As noted previously, mirroring can be done only on disks connected to intelligent disk controllers that are capable of dynamic badspot handling. This means that PRIMOS does not encounter correctable read-errors, correctable write-errors, or uncorrectable write-errors.

If PRIMOS does encounter one of the above error conditions (such as during the initial stage of a cold start or briefly after a warm start, before the controller is downloaded), PRIMOS does the following:

- Handles correctable read-errors and write-errors but does not break the mirror
- Handles uncorrectable write-errors and breaks the mirror
- Handles uncorrectable read-errors, but breaks the mirror if uncorrectable read-errors occur on both partitions of the mirrored pair

The following failures break the mirror:

- An attempt to write to a write-protected disk
- An uncorrectable write-error
- An uncorrectable read-error while PRIMOS is attempting to access a record from both partitions of the mirrored pair
- Any failure of the copy server that performs the catch-up copy

Very few errors should occur on mirrored partitions because dynamic badspot handling is active on these partitions. Other situations that require recovery action, such as a system halt, occur infrequently. In general, then, the recovery procedures outlined below should rarely be necessary.

Error Recovery for Write-Errors

If an uncorrectable write-error occurs, the mirror is disabled, or broken, and the system continues to operate on the good partition. The uncorrectable write-error occurs only under the following conditions. Solutions are described in the following paragraphs.

- The write-protect switch is set.
- The controller is in Nondynamic Badspot Handling (-AC) mode.
- The DBS/RMA of the partition is full.
- Physical problems, such as loose cables, exist.

What to Do if the Write-protect Switch Is Set: Perform these steps for every partition on the physical disk where a write-error occurred.

1. Turn off the write-protect switch.
2. Use `MIRROR_OFF` to take the partition having the error out of the mirror.
3. Use `MIRROR_ON` to restart the mirror. A catch-up copy will start.

What to Do if the Controller Is in Nondynamic Badspot Handling (-AC) Mode: The controller should be in Nondynamic Badspot Handling (-AC) mode only during the initial stage of a cold start and for less than a minute after a warm start. You can determine that the write error occurred while the controller was in Nondynamic Badspot Handling mode by using the `DISPLAY_LOG` command on the system event log file. If this is the case, perform these steps for every partition on the physical disk where a write-error occurred.

1. Take the damaged partition out of the mirror with the `MIRROR_OFF` command.
2. Restart the mirror with the `MIRROR_ON` command. A catch-up copy will start. The record where the write-error occurred is remapped by the controller in Dynamic Badspot Handling (-IC) mode.

What to Do if the DBS/RMA of the Partition Is Full: The following message probably was displayed when the system was started and the partition was added to the system:

Dynamic badspot remapping area is at least 80 percent full.

This message indicates that the observed uncorrectable write-error occurs because the area where the dynamic badspots are remapped is nearly full. Verify that this area is full by invoking `FIX_DISK` with the `-DUMP_DBS` option on the first partition. A full DBS/RMA area indicates unacceptable quality of the physical disk and you should not use it. Install a replacement disk. You can reestablish the mirror by using the `MIRROR_ON` command.

Note

Whereas a removable disk pack (SMD) can be replaced easily, replacing an FMD requires shutting down all the disk drives on the same controller. Strategic use of multiple disk controllers in conjunction with mirroring makes it possible to shut down all the drives on one controller and replace a drive without shutting down the system. Thus, be sure that mirrored partitions are on different disk controllers if possible.

What to Do if Physical Problems Exist: If an uncorrectable write-error occurs and the disk controller is in Dynamic Badspot Handling (-IC) mode and the remapped area is not full, the problem is probably physical. The occurrence of several uncorrectable write-errors on more than one partition further suggests a physical problem. If this is the case, follow this procedure:

1. Take the damaged partition out of the mirror with the MIRROR_OFF command.
2. Correct the problem.
3. Restart the mirror with the MIRROR_ON command. A catch-up copy will start.

Perform these steps for every partition on the physical disk where a write-error occurred.

Error Recovery for Read-Errors

Uncorrectable read-errors are not handled by the dynamic badspot mechanism. However, since uncorrectable read-errors generally start out as correctable read-errors and since dynamic badspot handling remaps correctable read-errors, very few uncorrectable read-errors should be encountered. If PRIMOS encounters an uncorrectable read-error on one of the mirrored partitions, PRIMOS retrieves the data from the other mirrored partition and attempts to write the data to the first partition. This process causes the bad record to be added to the badspot file and the data is remapped to another location. The mirror is not broken unless the same record is unreadable on both partitions of the mirrored pair. In the latter case, use the following procedure.

What to Do if the Mirror Breaks With an Uncorrectable Read-error for the Same Record on Both Partitions:

If the mirror should break while the partition pair has an uncorrectable read-error on both partitions on the same record (a very unlikely event), do the following:

1. Use the MIRROR_OFF command with the -SHUT_BOTH option to stop the mirror and use the SHUTDN *pdev* command to shut down both partitions so that no new data can be written to the primary partition.
2. Run FIX_DISK on one of the partitions. Use the -ADD_BADSPOT option to add the bad record to the badspot file. Keep a list of any files that were damaged as reported by FIX_DISK.
3. Run FIX_DISK on the other partition of the pair. Again use the -ADD_BADSPOT option to add the bad record to the badspot file. Keep a list of any files that were damaged on that partition as reported by FIX_DISK. The damaged files are the same files on both partitions.
4. Recover the damaged files from a recent backup to magnetic tape or other media to the primary partition. Inform the owners of the damaged files of the backup.
5. Use the MIRROR_ON command to restart the mirror; a backup copy will start.

Note

If any records were written to files after the mirror broke, the data from the secondary partition will be stale, or not up-to-date, since writing occurs only on the primary partition after the mirror is turned off. The secondary partition is made up-to-date when the mirror is restarted with the `MIRROR_ON` command.

Catch-up Copy Failure

Any uncorrectable read-error or write-error encountered by the catch-up copy process causes the mirror to break. If the error occurs while *writing* a record to the secondary, recover by following the above procedures for write-errors. If the error occurs while *reading* a record from the primary, recover by running `FIX_DISK` on the primary partition and use the `-ADD_BADSPOT` option to add the bad record to the badspot file. In both cases, use the `MIRROR_OFF` command before using the recovery procedures. The mirror can then be restarted with the `MIRROR_ON` command and the catch-up copy starts.

If the mirror breaks due to some other problem or condition encountered by the copy server (for example, the copy server was logged out at the supervisor terminal), the mirror can be restarted with the `MIRROR_ON` command and the catch-up copy starts.

System Halts

If, during a system halt, PRIMOS is able shut down in an orderly manner, mirrored partitions will have the same shutdown stamp and can be started without any special recovery action. If PRIMOS is not able to shut down in an orderly manner, the shutdown stamps will not match and a catch-up copy will start when the mirror is turned on, even if no errors are revealed when `FIX_DISK` is run on the primary partition.

MIRRORING MESSAGES

Informational and error messages associated with the mirroring process and the mirroring commands are presented and discussed in Appendix C.

RECORD ALLOCATION AND SECTORING

PRIMOS uses two methods for allocating disk sectors, or records, for files: forward sectoring and reverse sectoring. This chapter describes these methods.

PRIMOS RECORD ALLOCATION

The PRIMOS disk record allocation algorithm maintains a logical ordering of disk records. The algorithm uses bits stored in the DSKRAT to represent the entire set of records on a partition. Records are grouped in logical sets of 16 for allocation.

The actual physical location of these records on the disk is dependent on the physical configuration of the disk, such as the number of surfaces in the partition, the number of cylinders, or tracks, on the disk and the number of sectors in a track. Records within the group of 16 are allocated to one or more tracks. When a given track is filled, the track on the next surface is used. The entire cylinder is allocated in this manner. When this cylinder is fully allocated, the next cylinder is used.

Records are allocated in ascending order. The records are interleaved with a preset interleave factor. The **interleave factor** is the number of sectors that are skipped before reading or writing the next record. The interleave factor allows PRIMOS enough time to process the record just read and then to issue a request for the next sequential record to be read before the sector where the next record is located comes under the read head (the read head is on sector).

Forward Sectoring

In **forward sectoring**, PRIMOS allocates a physical sector for a record, then skips two sectors and allocates the next sector. This skipping is referred to as the **interleave factor** and is used to account for the time it takes the disk controller to write one record and return to the disk to write the next record. By that time the next sector under the disk read/write head should be three ahead of the last sector. For example, with forward sectoring, if the first record is written to physical sector 5 (see A in Figure 10-1), the second record is written to physical sector 8.

Forward sectoring records on disk is suited to nonintelligent disk controllers. The nonintelligent disk controller is not capable of buffering records. A different scheme is useful for intelligent disk controllers that can buffer records that pass under the read head before the read head arrives on sector for the desired record. For intelligent disk controllers, records are allocated to a file so as to maximize the number of records of the file lying on the same physical track that passes under the read head before the desired record. These file records are logically contiguous to the desired record.

Reverse Sectoring

Reverse sectoring is an alternative method used by PRIMOS to allocate records for files and is designed to take advantage of an intelligent controller's ability to **buffer**, or cache (temporarily store) records. In this scheme, the interleave factor is based on the maximum number of sectors on a track of the disk's surface minus 1. For example, if a track contains 8 records, the reverse sector interleave factor for that disk is 8 minus 1, or 7.

For example, as shown in Figure 10-1, a disk with 8 sectors per track is to be allocated in reverse order. The disk is spinning in a counterclockwise direction as you look at it. If the first sector allocated is physical sector 5 as shown in A of Figure 10-1, the next sector to be allocated will be sector 5 plus the interleave factor of 1 less than the number of sectors per track, or sector 5 plus 7 sectors, which is sector 4 as shown in B of Figure 10-1. After sector 4, the next sector to be allocated is sector 4 plus 7 sectors, or sector 3. This scheme then continues until all available sectors on a disk are allocated.

If you follow this allocation scheme, you will note that sectors are allocated in a "counting down" fashion: that is, 5, 4, 3, 2, 1, and so on. Thus it appears that they are allocated in reverse order: hence the name. Also, since the records are allocated contiguously, the interleave factor appears to be 1. The reason for this allocation scheme is to make logical file records contiguous when they are written so that, when the intelligent disk controller gets a record (say the logical record that is stored in physical sector 8), it will also get and buffer, or cache, the records stored in physical sectors 1 through 7 as they pass under the read head. If these records logically follow the file record stored in physical sector 8, they are available for processing and the controller will not have to go to the disk to get them, thus saving processing time.

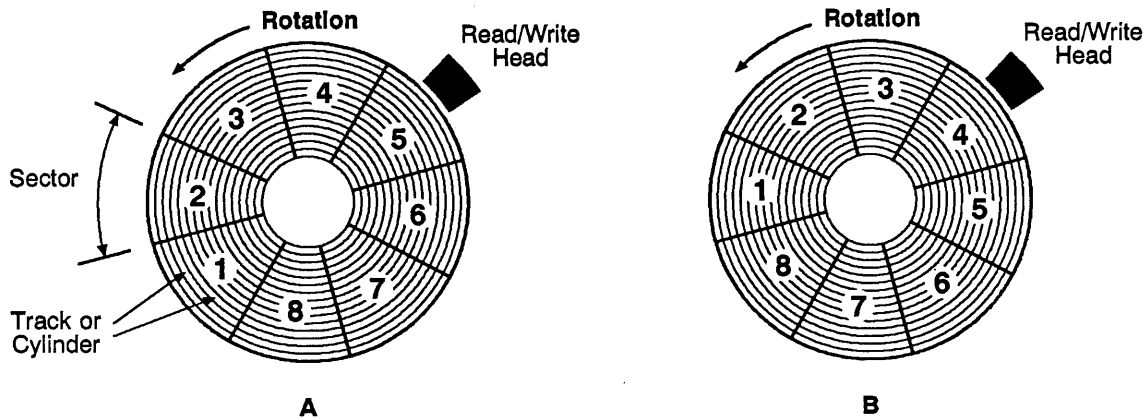


FIGURE 10-1. Record Allocation

ALLOCATION ORDER

The order in which disk records are allocated for a file is structured to optimize retrieval of records by intelligent disk controllers, that is, disk controllers that are microprocessor-based and are capable of buffering data. Records are allocated to maximize the number of sequential records that are in the disk controller's memory. When you want to read a record, the controller reads the records that logically follow the one you want before getting to your record. These other records in the controller's memory are then available to be used and do not have to be read again from disk.

Thus, the way in which disk records are laid out on disk takes advantage of disk controllers that can read ahead and buffer records. Disk I/O performance is improved on disks using intelligent disk controllers by taking full advantage of the controller's buffering capabilities. However, disks using nonintelligent disk controllers do not suffer any performance degradation brought about by optimizing the allocation of disk records for intelligent controllers.

Note

It is recommended that you always use reverse sectoring with CPUs not in the 9950 class (CPUs in the 9950 class consist of the 4000 and 6000 series CPUs, the 2850, the 2950, the 9755, and those with model numbers numerically equal to or larger than 9950, such as 9955) and if the majority of your standalone applications do little actual processing of the retrieved records.

Record Allocation Dependencies

The allocation of records by PRIMOS is as follows:

- If the partition is pre-Rev. 20.0, allocate records in forward order using an interleave factor of 3.
- If the partition is a Rev. 20.0 or later standard partition and the file type is CAM, allocate records in forward order using an interleave factor of 3.
- If you are using pre-Rev. 21.0 PRIMOS, allocate records in forward order using an interleave factor of 3.
- If the partition is a Rev. 22.1 robust partition, allocate records in forward order using an interleave factor of 3.
- If the partition is a standard (nonrobust) partition created with Rev. 22.1 MAKE or is converted to Rev. 21.0 format with FIX_DISK, allocate records using reverse sectoring and an interleave factor of 1.

These dependencies are summarized in Table 10-1.

TABLE 10-1. PRIMOS Record Allocation

<i>Dependency</i>	<i>Allocation Order</i>	<i>Interleave Factor</i>
pre-Rev. 20.0 partition	Forward	3
Rev. 20.0 and later standard partitions and CAM files	Forward	3
Using pre-Rev. 21.0 PRIMOS	Forward	3
Rev. 22.1 robust partition	Forward	3
Create as or convert to Rev. 20.0 or later standard partition	See Table 10-2	

Note

If you are creating robust partitions (by using the MAKE_ROBUST utility), the method of file record allocation is set by MAKE_ROBUST to forward. No reverse sectoring takes place on robust partitions. If you use either the -RDI or the -ODI options of FIX_DISK on robust partitions, FIX_DISK informs you that these options cannot be used in this case.

Allocation Direction for Standard Partitions

At PRIMOS Rev. 21.0 and later, the default record allocation direction and the default disk record interleave factor after you create a standard (nonrobust) partition with MAKE or convert a pre-Rev. 21.0 partition to Rev. 21.0 format with FIX_DISK, depend on the disk controller and CPU combination (see Table 10-2). Reverse sectoring with an interleave factor of 1 optimizes the sequential retrieval of records on an intelligent disk controller. You or your System Administrator may change the method of record allocation and the interleave factor on standard partitions by using the MAKE and FIX_DISK options -RESTORE_DEFAULT_INTERLEAVE (-RDI) or -OVERRIDE_DEFAULT_INTERLEAVE (-ODI).

PRIMOS determines the combination of disk controller type and CPU type and does the following for standard (nonrobust) partitions:

- For a CPU in the 9950 class and an intelligent disk controller, use reverse sectoring with an interleave factor of 1. CPUs in the 9950 class consist of the 4000 and 6000 series CPUs, the 2850, the 2950, the 9755, and those with model numbers numerically equal to or larger than 9950, such as 9955.
- For a CPU in the 9950 class and a nonintelligent disk controller, use forward sectoring with an interleave factor of 3.
- For all other CPUs and any disk controller, use reverse sectoring with an interleave factor of 1.

These conditions are summarized in Table 10-2.

TABLE 10-2. PRIMOS Default Record Allocation

<i>Controller</i>	<i>CPU</i>	<i>Allocation Direction/ Interleave Factor</i>
Intelligent	Less than 9950 class	Reverse/1
Nonintelligent	Less than 9950 class	Reverse/1
Intelligent	9950 class	Reverse/1
Nonintelligent	9950 class	Forward/3

When you create Rev. 20.0 or later standard partitions by using MAKE or convert partitions to Rev. 20.0 or Rev. 21.0 by using FIX_DISK, those utilities determine the CPU and disk controller combination. They then set a bit in the DSKRAT header to indicate the record allocation direction according to the above recommendations. This direction then becomes the record allocation direction for that combination. Thus, depending on the controller and CPU combination, record allocation could be either forward or reverse.

Note

FIX_DISK sets the record allocation bit only when you use the -CONVERT_21 option; it does not set the record allocation bit at any other time.

SAM and DAM File Management

Disk records for both SAM and DAM files may be allocated in either forward order using an interleave factor of 3 or in reverse order using an interleave factor of 1 on Rev. 20.0 or later standard partitions.

Setting the Interleave Factor

You do not have to convert partitions to Rev. 22.1 in order to use Rev. 22.1 PRIMOS or to benefit from performance gains of allocating disk records in reverse order. Both SAM and DAM files on a Rev. 20.0 or later standard partition may be allocated in reverse order with an interleave factor of 1. Only Rev. 21.0 and later PRIMOS recognize reverse allocation. However, a file may have records allocated in either order or in both orders and you may use Rev. 20.0 PRIMOS to read records from that file. PRIMOS gets a record by its address, which is determined after record allocation.

To set the interleave factor and the direction of sectoring on a pre-Rev. 22.1 partition, you can do one of the following:

- Run Rev. 22.1 MAKE to create the partition as a Rev. 20.0 or later partition.
- Run Rev. 22.1 FIX_DISK to convert a pre-Rev. 21.0 partition to Rev. 21.0 or a pre-Rev. 20.0 partition to Rev. 20.0.
- Indicate the allocation direction and the interleave factor on a Rev. 20.0 or later partition by using the MAKE or the FIX_DISK options -ODI or -RDI.

Performing one of the above procedures allows you to benefit from the allocation of disk records in reverse order with an interleave factor of 1.

You must use Rev. 21.0 or later PRIMOS on a partition in which the record allocation is reverse order with an interleave factor of 1 in order to realize performance improvement.

Which Method of Record Allocation Should You Use?

Certain standalone applications, which do little processing of the data being read from or written to disk, perform better on CPUs in the 9950 class with nonintelligent disk controllers if the disk has forward record allocation with an interleave factor of 3 rather than reverse record allocation with an interleave factor of 1. Examples of such applications are MAGSAV, MAGRST, BRMS, COPY, and FUTIL (assuming they are run standalone).

Some of the applications noted, such as MAGSAV and BRMS, attain an interleave factor of 3. Thus some systems, which back up large databases to tape using BRMS or MAGSAV, need a method of changing the interleave factor while running PRIMOS with nonintelligent controllers and a 9950 class CPU.

You can change the interleave factor on particular Rev. 20.0 and later standard (nonrobust) partitions either when you create the partition by using MAKE or when you repair it by using FIX_DISK.

It is not necessary to convert Rev. 20.0 and later partitions to Rev. 22.1 partitions in order to change the disk record interleave factor. To change the disk record interleave factor, use the MAKE or FIX_DISK options -OVERRIDE_DEFAULT_INTERLEAVE (-ODI) and -RESTORE_DEFAULT_INTERLEAVE (-RDI), available at Rev. 22.1.

If you convert partitions to Rev. 22.1, you cannot use a pre-Rev. 22.1 version of PRIMOS with the Rev. 22.1 partition.

Note

You cannot locally add a Rev. 22.1 partition to a system that is running a pre-Rev. 22.1 version of PRIMOS because of the changes in the header of the Rev. 22.1 DSKRAT. However, you may add a Rev. 22.1 partition remotely, using PRIMENET, to a system running pre-Rev. 22.1 PRIMOS.

APPENDICES

MAKE MESSAGES

Special messages indicate that MAKE is unable to create the disk according to the options specified on the command line or according to other input. This appendix lists the messages from MAKE in alphabetical order and presents a brief explanation of each message. If a variable, such as a number, is the first word in the message, the second word is used for alphabetizing. Variable names in messages are italicized, for example, *pdev*.

For a discussion of MAKE and its command-line options, see Chapter 5, Formatting Disks — MAKE. For a summary of the FIX_DISK messages, see Appendix B, FIX_DISK Messages.

-AC/-IC modes not available with a DISK_REVISION earlier than 21.

You can use the -ALL_CONTROLLER and -INTELLIGENT_CONTROLLER options only on Rev. 21.0 and later partitions.

An option appeared more than once (MAKE)

An option was repeated on the MAKE command line. MAKE aborts. Retype the MAKE command line being careful not to repeat options.

Bad disk revision: *token*

Bad disk revision, must be either 18, 19, 20, 21, 22, or 22.1.

Disk revision?

You have selected a revision number (*token*) other than 18, 19, 20, 21, 22, or 22.1 for the -DISK_REVISION option. Select revision 18, 19, 20, 21, 22, or 22.1 and enter the correct revision at the prompt. (You can use a decimal point and zero with any of the revisions, for example, 21.0.)

Badspot file is not a SAM file!

It has been ignored. (MAKE)

The badspot file on device *pdev* specified by the -COPY_BADSPOTS_BY_DEVICE option is not the correct file type. MAKE ignores this option and does not further attempt to copy from the badspot file on device *pdev*.

BADSPT file is full! Badspot not added to BADSPT file.

No more badspot locations can be added to the badspot file. The location of the particular badspot is displayed. The disk contains more badspots than allowed by PRIMOS. Replace the disk.

Both -IC and -AC modes cannot be specified!

You can specify only one of the controller mode options on a command line. MAKE aborts. Reenter the command line with only one of the options.

Cannot access first partition of disk for Rev 21 DBS file check!

MAKE could not read the DSKRAT on the first partition and aborts. Either you did not create the first partition first or there is some other problem with the first partition. Restart MAKE and create the first partition.

Cannot access the first partition. No badspot info. available!

You did not assign the first partition, so MAKE cannot get any badspot information from the DBS file on that partition. MAKE continues.

Cannot add badspot! (ADBSP\$)

MAKE cannot add a badspot, displays this message, and aborts. This is an unexpected error and may indicate a hardware problem, such as a bad disk controller or a corrupted download file. It may also indicate incorrect usage of the MAKE options and arguments, such as an incorrect disk type with the -DT option. Reenter your command line, being sure that you have entered the correct options and arguments. If the problem persists, call your Customer Support Center.

Cannot add records <=20 to bad spot file. MAKE

A badspot has been found within records 0₈ through 20₈ (decimal 0 through 16). Records 0 through 16, which contain the bootstrap file, badspot file, DSKRAT, and MFD, must not contain any badspots. A disk with flaws in these locations is not usable by PRIMOS. MAKE aborts.

You may also get this message if you use the wrong disk type keyword with the -DISK_TYPE option. This message will also be followed by disk-error messages in this case. To recover, rerun MAKE being sure that you have specified the correct disk type from the table in Chapter 5.

A possible cause of this error is that the disk has not yet been formatted. If you did not specify -FORMAT on the command line, reenter the command line and include the -FORMAT option. If the error occurs again, replace the disk.

Cannot handle badspots on records <=20. MAKE

You entered a record number (or track/head/sector combination that represented a record number) that was 20₈ or less (decimal 16). Records 0 through 16 contain the bootstrap and badspot file and must not contain any badspots. A disk with flaws in these locations is not usable by PRIMOS. MAKE aborts.

Cannot MAKE a -IC mode disk running standalone!

Because the intelligent disk controller code cannot be downloaded until PRIMOS is running, you cannot create an -IC partition when running MAKE standalone. MAKE aborts. Create the partition in -AC mode. You can convert to -IC later with FIX_DISK.

Cannot MAKE a -IC mode disk on a non-intelligent controller!

In order to use the -INTELLIGENT_CONTROLLER option to create a partition in -IC mode, the disk controller must be an intelligent disk controller (IDC1).

Cannot read 1st DBS file record.

MAKE encounters an error condition in attempting to read the DBS file. This is an unexpected error and may indicate a hardware problem, such as a bad disk controller or a corrupted download file. It may also indicate incorrect usage of the MAKE options and arguments, such as an incorrect disk type with the -DT option. Reenter your command line, being sure that you have entered the correct options and arguments. If the problem persists, call your Customer Support Center.

Cannot read DBS file. Cannot add badspots!

Continuing with make. (MAKE)

MAKE encounters an error condition in attempting to read the DBS file. You did not use the -NEW_DISK option when making a new disk and MAKE looks for an existing badspot file in that case. MAKE creates a new badspot file.

Cannot write DBS file.

MAKE encounters an error condition while attempting to write the DBS file, displays this message, and aborts. This is an unexpected error and may indicate a hardware problem, such as a bad disk controller or a corrupted download file. It may also indicate incorrect usage of the MAKE options and arguments, such as an incorrect disk type with the -DT option. Reenter your command line, being sure that you have entered the correct options and arguments. If the problem persists, call your Customer Support Center.

Copy badspots from partition?

You did not specify a partition name when you used the -CPYNAM option. Enter the partition name at the prompt.

Copy badspots from physical device?

You did not specify a physical device number (pdev) when you used the -CPYDEV option. Enter the pdev at the prompt.

-DISK_REVISION and -PRE_REV19 cannot both be specified.

-PRE_REV19 is an option used on Rev. 19.0 systems and is obsolete at Rev. 20.0. Use -DISK_REVISION 18 instead.

Disk DBS file change! All other partitions on this disk
will be corrupted if you continue
Do you wish to continue?

Disk DBS file change! All other partitions on this disk
will be corrupted if you continue
MAKE will enable -FORMAT to change the DBS file.
Do you wish to continue?

You are attempting to recreate the existing first partition of a Rev. 21.0 or later disk and the disk geometry information specified in the pdev you have given will result in changing the geometry of the first partition. As a result, all partitions on this disk will be corrupted since the DBS and the RMA, which are on the first partition, contain information for all partitions on this physical disk.

If you specify the -FORMAT option, you see the first message above. If you do not specify the -FORMAT option, you see the second message and MAKE defaults to the -FORMAT option. It is necessary to use the -FORMAT option when changing the first

partition of the disk because the record headers must be changed to reflect new circumstances and -FORMAT accomplishes this.

If you answer YES in either case, MAKE continues and you will destroy any badspot information and remapped records on the other partitions on this disk. If you answer NO, MAKE aborts.

Disk type is incompatible with this disk revision.

The disk revision you have specified with the -DISK_REVISION option and the disk type specified with the -DISK_TYPE option are incompatible. Only the SMDs, CMDs, Model 4475, and the 68MB, 158MB, 160MB, and 675MB FMDs can contain pre-Rev. 20.0 partitions. All other disks can contain only Rev. 20.0 and later partitions. MAKE prompts again for the disk revision.

Disk type must be one of the following:

Disk type?

Either you entered an incorrect argument or you entered no argument with the -DISK_TYPE option. Enter the correct disk storage device type following the -DT option or at the prompt from the list displayed.

Disk revision?

You did not specify the disk revision with the -DSKREV option. See the Bad disk revision message.

DISK WT ER *device_nn PRIMOS record_nn status-word*

A disk write-error has occurred. The offending record is added to the badspot file, and MAKE prints a message documenting this fact. Status 177776 indicates that a DISK-NOT-READY status has been detected. The program waits for the disk to become ready, then retries the write.

A possible cause of this error is that the disk has not yet been formatted. If you did not specify -FORMAT on the command line, reenter the command line and include the -FORMAT option.

Another possibility is that the disk is write-protected. If this is the case, set the disk drive to permit writing and reenter the MAKE command line.

DSKRAT pointer mismatch MAKE

While MAKE is attempting to read a record and set that record address in the DSKRAT, MAKE finds that the record does not match values that MAKE compares it to. MAKE displays this error message and aborts. This is an unexpected error and may indicate a hardware problem, such as a bad disk controller or a corrupted download file. It may also indicate incorrect usage of the MAKE options and arguments, such as an incorrect disk type with the -DT option. Reenter your command line, being sure that you have entered the correct options and arguments. If the problem persists, call your Customer Support Center.

Enter command line options:

This message appears when you boot MAKE.SAVE. Enter the command-line options that you wish to use.

Error during Format! MAKE

While formatting a new disk (as indicated by the previous message Beginning format.), MAKE encounters an error, displays this message and aborts. This is an unexpected error and may indicate a hardware problem, such as a bad disk controller or a corrupted download file. It may also indicate incorrect usage of the MAKE options and arguments, such as an incorrect disk type with the -DT option. Reenter your command line, being sure that you have entered the correct options and arguments. If the problem persists, call your Customer Support Center.

Extent size has to be greater than 0.

Please input new values for MINIMUM & MAXIMUM extent size.

Either you did not enter a value for one of the CAM file extent sizes or you entered a 0. Reenter proper values at the appropriate prompt, MINIMUM_EXTENT_SIZE? or MAXIMUM_EXTENT_SIZE?.

File not found

This error may occur if you did not include the .SAVE suffix at the RUNFILE TREENAME= prompt when booting CMDNCO>MAKE.SAVE.

First partition of disk not Rev 21 or later. Cannot build other rev 21 or later partitions

In order to create Rev. 21.0 or later partitions on a disk, the first partition of that disk must be Rev. 21.0 or later. You must also create the first partition before other partitions on the disk if you are creating Rev. 21.0 or later partitions.

Flaw map is bad. Continuing with MAKE.

MAKE is unable to read the flaw map from any surface on the disk. MAKE continues processing using the default level of badspot checking.

Flaw map unsupported. Continuing with MAKE.

MAKE was unable to read any records from the area of the disk set aside to contain the flaw map, regardless of whether the records were flaw map records. MAKE continues processing using the default level of badspot checking.

Since a flaw map is written on each surface of the Model 4735 and 4845 disks, it is possible that all of the areas, or tracks, containing the flaw map information in a particular partition may be unreadable. However, valid flaw maps may exist on surfaces outside the partition that is being created. Thus, if this message is displayed, rerun MAKE and use the -COPY_BADSPOTS_BY_DEVICE option (-CPYDEV *pdev*). This allows MAKE to read the badspot file from a partition with a valid flaw map.

Flaw map is unreadable. Continuing with MAKE.

MAKE was able to read records from the area of the disk set aside to contain the flaw map, but could not find or could not read all the records that form the flaw map on any one disk surface. MAKE continues processing using the default level of badspot checking.

Since a flaw map is written on each surface of the disk that support flaw maps, it is possible that all of the areas, or tracks, containing the flaw map information in a particular partition may be unreadable. However, valid flaw maps may exist on surfaces outside the partition that is being created. Thus, if this message is displayed, rerun MAKE and use the -COPY_BADSPOTS_BY_DEVICE option (-CPYDEV *pdev*). This allows MAKE to read the badspot file from a partition with a valid flaw map.

Insufficient size for minimum file system. Leave at least *nn* records.

The partition you are creating is a Rev. 21.0 or later partition. It is a split paging partition and is the first partition of the physical disk and, thus, will have the DBS file and the RMA written to it. You have not specified enough room on this partition for the DBS file and the RMA; MAKE aborts. Use the -SPLIT option and specify enough file system records to include the DBS file and the RMA. (See Table 5-5 in Chapter 5.)

Invalid baud rate, must be either 110, 300, 1200, or 9600.

Either you entered an incorrect baud rate or you entered no baud rate with the -BAUD option. Enter the appropriate baud rate with the -BAUD option or at the Baud rate? prompt.

Invalid number of paging records!

The number of paging records that you have specified for the paging portion of a split disk is invalid, probably because you specified more records than are on the partition. MAKE again displays the total number of records available and prompts for the number of records you want to set aside for paging. Enter a decimal number less than the total number of records available. This number should be evenly divisible by 16. Use the MOD function to determine the remainder of dividing by 16. For example, using the functions TYPE [MOD 22221 16] gives a remainder of 13. If you subtract 13 from 22221, the result is 22208, which is evenly divisible by 16. If you subtract another 16 from this sum, you have 22192 records for paging and 29 records for the file system on a three-surface SMD.

Invalid partition name.

You used an invalid partition name either when initially naming the partition or with the -COPY_BADSPOTS_BY_NAME option. The partition name may contain a maximum of six characters. The first character must not be a digit or dash (-); the name can contain only alphabetic characters, digits, and the special characters _ # \$ & * - . /.

Invalid physical device number.

You entered an invalid physical device number. The request for a physical device number is repeated. Physical device numbers are octal numbers and can contain only the digits 0 through 7. See Chapter 3 for a complete guide to the construction of physical device numbers for disks, or partitions.

Invalid, retype last entry.

You entered an invalid track/head/sector number while badspots were being entered.

Level of bad spot checking? (default = *d*)

You used the -BADSPOT_LEVEL option and did not specify the level of badspot checking. Either specify a number between 0 and 4, inclusive, or press and MAKE uses the default *d*.

nnnnn lost records

One or more badspots affected the current partition. The number of lost records is displayed when MAKE finishes.

MAKE will enable -FORMAT to change the DBS file.

You are using MAKE on an existing Rev. 21.0 or later first partition in -IC mode and are using all surfaces on the disk thus changing the geometry of the partition. This results in changing the DBS file. MAKE uses the --FORMAT option to reset the physical record headers.

MAXIMUM_EXTENT_SIZE is only available starting at Rev. 22.

You used the -MAXSIZ option in conjunction with the -DISK_REVISION option having an argument less than 22.

*max/min*_EXTENT_SIZE has to be a decimal number.

You entered something other than a decimal number to indicate the maximum or the minimum extent size. Reenter the number at the appropriate prompt (MAXIMUM_EXTENT_SIZE? or MINIMUM_EXTENT_SIZE?).

Minimum_extent_size cannot be greater than Maximum_extent_size.

Please input new value.

Either you entered a value for minimum extent size that was larger than that of the maximum extent size or you did not enter a maximum extent size value. Reenter the values at the appropriate prompts.

MINIMUM_EXTENT_SIZE is only available starting at Rev. 22.

You used the -MINSIZ option in conjunction with the -DISK_REVISION option having an argument less than 22.

Missing physical device number. For options type: MAKE -HELP (MAKE)

You did not include the -DISK option with or without the physical device number. MAKE aborts. Reenter the complete MAKE command line including the -DISK option. If you do not include the pdev, MAKE prompts you for it.

Must specify a -AC/-IC mode for a revision 21 disk!

When running MAKE standalone, you must specify -ALL_CONTROLLER because MAKE cannot determine the type of controller in your system or download an intelligent disk controller until PRIMOS is running.

-NO_INIT option cannot be used in conjunction with the -INIT option.

You used both the -INIT and -NO_INIT options on the command line. Restart MAKE and use only one of these options.

No badspot file on partition: *name* Ignoring copy_badspots_by_name option. (MAKE)

MAKE did not find a badspot file on the partition specified by -CPYNAM. MAKE continues without this information.

-ODI/-RDI modes not available with a DISK_REVISION earlier than 21.

You attempted to set the method of record allocation and interleave factor on a partition that is being created at a revision earlier than Rev. 21.0. MAKE aborts. Reenter the command line without either one of these options.

-OVERRIDE_DEFAULT_INTERLEAVE option cannot be used in conjunction with the RESTORE_DEFAULT_INTERLEAVE option

You cannot use both the -ODI and -RDI options on the same command line.

Partition is in All_Controller Mode

Either you specified the `-ALL_CONTROLLER` option or you did not specify the controller mode for this Rev. 21.0 partition by using either the `-INTELLIGENT_CONTROLLER` or the `-ALL_CONTROLLER` option. In the latter case, MAKE determines the controller mode by the disk type and the software and hardware revisions of the disk controller and sets the mode accordingly.

Partition is in Intelligent_Controller Mode

Either you specified the `-INTELLIGENT_CONTROLLER` option or you did not specify the controller mode of this Rev. 21.0 partition by using either the `-INTELLIGENT_CONTROLLER` or the `-ALL_CONTROLLER` option. In the latter case, MAKE determines the controller mode by the disk type and the software and hardware revisions of the disk controller and sets the mode accordingly.

Partition is using *direction* sectoring

Either you used one of the `-ODI` or `-RDI` options or MAKE determined the file allocation method based on the type of controller and CPU in your system. *direction* is either Reverse or Forward.

Partition name?

You did not specify the partition name with the `-PARTITION` option. Enter the name here.

Partition not found: *name*

Ignoring `copy_badspots_by_name` option. (MAKE)

MAKE did not find the partition specified by `-CPYNAM` from which to copy the badspot file. MAKE continues without this information.

Physical device?

You did not specify the physical device number (pdev) with the `-DISK` option. Enter the pdev here.

Processing flaw map.

Processing of flaw map completed.

These messages are displayed in the normal course of creating a disk partition when you use the `-DISK_TYPE` option with either the `MODEL_4735` or the `MODEL_4845` argument and you do not use either the `-BADLEV` option or the `-NO_FLAW_MAP` option.

Read error, record = *record-number*; track = *track*,
head = *head*, sector = *sector*

A read attempt is unsuccessful while MAKE is testing the partition for badspots. MAKE transforms the octal record number to decimal head and track values and to sector if the partition is Rev. 19.0 or later. MAKE then displays the location of the badspot and modifies the badspot file (BADSPT) and the DSKRAT appropriately.

Sanity check of DBS file failed. DBS file corrupted!

Continuing with make. (MAKE)

MAKE checks to see that the DBS file contains records that are actually on the disk being created. If they are not, this message is displayed. You also get this message if you do not create the first partition of a new disk first or if you do not use the `-NEWDSK` option when creating a new Rev. 21.0 or later partition over an existing Rev. 21.0 or later partition.

TOO BIG

You attempted to use MAKE while operating under PRIMOS II. MAKE cannot run under PRIMOS II (DOS) at Rev. 20.0 and subsequently.

Too many heads on this partition! Please verify the PDEV you are using.

The pdev that you specified with the -DISK option is incorrect for the disk type you are using. See Chapter 3 for instructions on determining pdevs.

Total decimal number of records available: *rec*

Decimal records for paging?

You have used the -SPLIT option to create a paging partition. MAKE displays the decimal number of records available and prompts for the number of paging records. The decimal number of paging records should be divisible by 16 and you should leave a specified amount of file system records, in particular for a split first partition. See the discussion under -SPLIT in Chapter 5.

Unable to get badspots from device: *pdev*

Continuing with make. (MAKE)

MAKE was unable to read the badspot file on the specified partition *pdev* used with the -CPYDEV option. MAKE ignores the option in this case and continues, creating a badspot file if necessary.

This message also appears when you change the geometry of the first partition and when you are creating a new partition over an old partition and do not use the -NEWDSK option to create a new badspot file. MAKE continues and creates a new badspot file.

Unable to get DBS file from device: *pdev*

Continuing with make. (MAKE)

This message appears when you use the -CPYDEV option on an -IC partition and MAKE cannot find the DBS file on the first partition. MAKE ignores the option in this case and continues.

Unable to read DSKRAT MAKE

MAKE encounters an error condition while attempting to read the DSKRAT, displays this message, and aborts. This is an unexpected error and may indicate a hardware problem, such as a bad disk controller or a corrupted download file. It may also indicate incorrect usage of the MAKE options and arguments, such as an incorrect disk type with the -DT option. Reenter your command line, being sure that you have entered the correct options and arguments. If the problem persists, call your Customer Support Center.

Unable to read existing DSKRAT on disk *pdev*.

MAKE encountered an error in attempting to read the DSKRAT on this partition. If this is a new partition, there is no DSKRAT readable by MAKE and you should use the -NEW_DISK option to avoid this message. MAKE continues processing.

Unable to write DSKRAT MAKE

MAKE encounters an error condition while attempting to write the DSKRAT, displays this message, and aborts. This is an unexpected error and may indicate a hardware problem, such as a bad disk controller or a corrupted download file. It may also indicate incorrect usage of the MAKE options and arguments, such as an incorrect disk type with the -DT option. Reenter your command line, being sure that you have entered the correct options and arguments. If the problem persists, call your Customer Support Center.

Unrecognizable TOKEN: *token*

For options type: MAKE -HELP (MAKE)

You have entered characters that MAKE cannot recognize. MAKE aborts. You may have entered an invalid command-line option or omitted the dash (-) preceding the option. Reenter the command line or enter MAKE -HELP to see a list of options.

Unrecovered error

A write operation was not successful after 10 tries. MAKE aborts and returns you to PRIMOS command level.

A possible cause of this error is that the disk has not yet been formatted. If you did not specify -FORMAT on the command line, reenter the command line and include the -FORMAT option.

Which file sectoring scheme would you like?

Type "0" (Reverse Sector)

or "1" (Forward Sector)

>

When you run MAKE standalone, you are prompted for the type of file sectoring and, thus, the interleave factor. See the discussion in Chapters 5 and 10. Enter either a 0 or a 1 at the > prompt. If the partition is robust, only forward sectoring can take place and MAKE defaults to forward sectoring.

Write error, record = *record-number*; track = *track*,

head = *head*, sector = *sector*

A write attempt is unsuccessful while MAKE is testing the partition for badspots. MAKE transforms the octal record number to decimal head and track values and to sector if the partition is Rev. 19.0 or later. MAKE then displays the location of the badspot and modifies the badspot file (BADSP) and the DSKRAT appropriately.

FIX_DISK MESSAGES

This appendix contains error messages generated by the `FIX_DISK` command. Many of the messages displayed by `FIX_DISK` are informational, that is, they tell you of any errors encountered. Many of these messages are generated due to failure of other software and are not generated or caused by users.

If the repair operation results in the truncation or deletion of files or records, you can restore these files or records from backup copies. Thus, it is important to have a good backup procedure. (See the *Data Backup and Recovery Guide*.)

The messages below are in alphabetical order. If the article *The* or a variable, such as a number, is the first word in the message, the second word is used for alphabetizing. Exceptions to this are the first three messages, which have *1st*, *2nd*, and *3rd* as the second words. An explanation follows each message. Variable names and numerical output in the messages are italicized — for example, *category_name*. The name of the module issuing the message may appear in parentheses following the message.

For a discussion of `FIX_DISK` and its command-line options, see Chapter 6, Repairing Partitions — `FIX_DISK`. For a summary of the `FIX_DISK` command-line options, see Appendix F, Summary of Command Syntax for `MAKE` and `FIX_DISK`.

If you specify the `-FIX` option on the command line, `FIX_DISK` repairs any correctable errors or inconsistencies. If you do not specify the `-FIX` option, `FIX_DISK` reports any errors but does not correct them.

FIX_DISK ERROR MESSAGES

The 1st file entry of the MFD is not DSKRAT

FIX_DISK checks that the first entry in the MFD is DSKRAT. If this entry is missing and the -FIX and -INTERACTIVE options were not specified on the command line, FIX_DISK aborts.

The 2nd file entry of the MFD file is not MFD

FIX_DISK checks that the second entry in the MFD is MFD. If this entry is missing, FIX_DISK aborts. Remake the partition by using MAKE.

The 3rd file entry of the MFD is not BOOT.

FIX_DISK checks that the third entry in the MFD is BOOT. If this entry is missing, FIX_DISK aborts. Remake the partition by using MAKE.

Access Category *category_name* does not reference an ACL!

The ACL pointer of an access category does not point to a valid ACL. If you specify the -FIX option, the access category is deleted, and all objects that it protects revert to default protection.

Access Category *category_name* is not pointed at by ACL it points to!

The ACL pointer of an access category points to an ACL that doesn't point back to it. If you specify the -FIX option, the access category is deleted, and all objects that it protects revert to default protection.

Access Category *category_name* points outside the directory!

The ACL pointer of an access category points outside the directory. If you specify the -FIX option, the access category is deleted, and all objects that it protects revert to default protection.

ACL at word XX does not point at a file or access category!

The owner pointer of an ACL does not point to a file or access category. If you specify the -FIX option, the ACL is deleted.

ACL at word XX is not pointed at by object it points to!

The owner pointer of an ACL points to an object that does not point back to it. If you specify the -FIX option, the ACL is deleted.

ACL at word XX points outside the directory!

The owner pointer of the ACL is pointing to something in a different directory. If you specify the -FIX option, the ACL is deleted.

Activating DBS file

This message indicates a conversion to -IC mode or a conversion to a Rev. 21.0 first partition in -IC mode; the DBS file is activated.

Added to BADSPT file!

The record you added with the -ADD_BADSPOT option is added.

Added to the DBS file!

The record you added with the -ADD_BADSPOT option is added to the DBS file.

Already exists in BADSPT file!

The BADSPT file already contains information on the bad record that was found, even though that bad record is also in use. FIX_DISK continues.

Attempt to re-run FIX_DISK to add this badspot.

For a discussion of this message, see the message FIX_DISK is unable to read badspot

Back pointer in free list at XX is YY, it should be ZZ.

The backward pointer in the free list does not point back to the previous free entry. The file system maintains a list of free, or vacant, directory entries. These are entries that have been deleted from the directory and thus freed for use. To save space in the partition, you can remove these free entries by using the -UFD_COMPRESSION (-CMPR) option to compress the directory. This message indicates that the directory is damaged. Use the -FIX option to repair the directory.

The backward pointer is bad.

It should be YY instead of XX!

Bad record address = XX BRA = YY Father = ZZ Type = N

The backward pointer of a record does not point back to the previous record of a file. In the case of the first record of a file, the backward pointer is not zero. If you specify the -FIX option, the backward pointer is fixed to point to the previous record when the BRA word of this record matches the first record address of this file. If the BRA word of this record does not match the first record address of the file, the file is truncated. You can restore the file by using a backup procedure.

The backward pointer of the next record does not point back to the pre-remap record!

The backward pointer of a record does not point back to the previous record, which has been remapped to a location that does not contain a badspot. In the case of the first record of a file, the backward pointer is not zero. If you specify the -FIX option, the backward pointer is fixed to point to the previous record when the BRA word of this record matches the first record address of this file. If the BRA word of this record does not match the first record address of the file, the file is truncated. You can restore the file by using a backup procedure.

Bad data count XX in segdir YY!

The segment directory contains a different number of records than the record header indicates it should contain. If you specify the -FIX option, the header is fixed.

Bad entry length at record XX, word YY.

A record has been corrupted and does not have a valid length. Use the -FIX option to repair the record.

Bad equivalence section!

The equivalence section of the badspot file is not in the correct format and is ignored. Other messages, such as Single zero record added... will be displayed. Use the -FIX option to delete the equivalence section.

Bad file type: illegal 1st 8 bits!

It should be XX instead of YY.

Bad record address = XX BRA = YY Father = ZZ Type = N

A correctable error is found in the header of a directory file or a SAM segment directory file indicating an incorrect file type. FIX_DISK corrects the error.

Bad file type: special bit not set!

Bad record address = XX BRA = YY Father = ZZ Type = N

The special bit in the BOOT, MFD, or DSKRAT files has not been set. If you specify the -FIX and -INTERACTIVE options, the DSKRAT file is rebuilt. Otherwise you may have to create the partition again by using MAKE.

Bad forward pointer (correctable).

Contiguous Access Method (CAM) files are in use on this pre-Rev. 22.0 partition. Use the -FIX option to correct the error.

Bad forward pointer (uncorrectable).

Contiguous Access Method (CAM) files are in use on this pre-Rev. 22.0 partition. This error is uncorrectable; consequently, if you use the -FIX option, you may receive a message indicating that the file has been truncated or deleted. In either case, restore the file by using a backup procedure.

Bad header in BADSPT file, ignored!

The header of the BADSPT file is incorrect. FIX_DISK continues, but badspot handling is disabled.

Bad length (XX) on index block, entry deleted.

A table of records in a directory is kept for access to the directory. The index of the table is corrupted. The table affects how fast the entries are accessed. This error condition means that access will be slower if you use the -FIX option because this option causes deletion of the index block. Delete this directory and restore it by using a backup procedure.

Bad "minimum extent length" value (correctable).

Contiguous Access Method (CAM) files are in use on this pre-Rev. 22.1 partition. The value of the extent length, or number of records in a CAM file, is less than ten. Use the -FIX option to correct the error.

Bad number of extents value (correctable).

Contiguous Access Method (CAM) files are in use on this pre-Rev. 22.0 partition. The number of blocks of contiguous records found in the file header is either greater than the maximum allowed or it is less than zero. Use the -FIX option to correct the error.

Bad Physical Device Number (cl_par)

The physical device number that is specified in the command line is incorrect. Reenter the command line and use a correct pdev.

Bad record address: *XX*

BRA = *YY* Father = *ZZ* Type = *file_type*

Record address *XX* in the record header is bad. The Beginning Record Address (BRA), father pointer (FRA), and file type are also displayed. If you specify the -FIX option, the address is corrected, if possible. Otherwise the file is truncated. You can restore the file by using a backup procedure.

BADSPT: Track = *nn* Head = *nn* Sector = *nn* Record = '*XX*'

You specified the -LIST_BADSPOTS option and the badspots are listed.

The BADSPT file cannot be read, ignored!

BADSPT file was not found in the current partition. Badspot handling is disabled.

The BADSPT file is bad, ignored!

The BADSPT file that is found by FIX_DISK is bad. Badspot handling is disabled.

The Beginning Record Address (BRA) pointer is bad.

It should be *XX* is *YY*.

Bad record address = *XX* BRA = *YY* Father = *ZZ* Type = *N*

The Beginning Record Address (BRA) pointer of the records within the file (except the first record) should point to the first record of the file. If you specify the -FIX option, the BRA pointer is fixed.

The Beginning Record Address (BRA) pointer *XX* is bad, it points to a record that belongs to another file.

Bad record address = *XX* BRA = *YY* Father = *ZZ* Type = *N*

In checking the integrity of a record header, FIX_DISK finds that the Beginning Record Address (BRA) pointer points outside the current file. If you specify the -FIX option, the file is truncated. You can restore the file by using a backup procedure.

Both -AC and -IC conversion modes can not be specified at the same time.

You cannot specify both of the controller mode selection options (-ALL_CONTROLLER or -INTELLIGENT_CONTROLLER) on the same command line. Reenter the command line with the one selection you want.

Both -OVERRIDE_DEFAULT_INTERLEAVE and RESTORE_DEFAULT_INTERLEAVE can not be specified at the same time.

When selecting the type of record allocation, you have used both selection options on the command line. Only one of these options is permitted. Reenter the command line with the one selection option you want to use.

Both -DUFE and -SUFU can not be specified at the same time.

Since the -DUFE and -SUFU options have opposite effects, they cannot both be specified on the same command line. Reenter the command line specifying one or the other. If you specify neither, -DUFE (delete unknown file entries) is the default.

CAM file(s) in DBS/RMA area prevent conversion to rev 21 disk.

Please save & delete these files, and restore following the conversion.

Just before this message appears, you see the message The following CAM file . . . and that message contains the pathname of the offending files. When converting a partition to Rev. 21.0 with -CONVERT_21, a CAM file, *pathname*, is found in the area of the first partition where it is necessary for FIX_DISK to create a DBS file and the

RMA. `FIX_DISK` aborts. If you want to convert this partition to Rev. 21.0, save the offending files to tape, delete them from the disk, convert the disk, and then restore the CAM files you saved and deleted.

Cannot access RAT on disk *pdev*.

In trying to read the DBS file, `FIX_DISK` could not find the `DSKRAT` on the first partition, *pdev*. Be sure that the first partition is added to the system or is assigned to you at the supervisor terminal.

Cannot add badspot to DBS file.

Controller code: *DD*

The DBS file may be full, the first partition may not be added, or the disk controller may have encountered a problem (the disk may be write-protected). `FIX_DISK` aborts. The code returned by the disk controller is also displayed (*DD*). If the first partition is added and the disk drive is not write-protected, you may have a hardware problem; call a representative from your Customer Support Center.

Cannot convert a revision *nn* disk to revision *mm* disk.

You cannot revert the revision of a partition using `FIX_DISK`; that is you cannot, for example, convert a Rev. 21.0 partition to a Rev. 20.0 partition and you cannot convert any revision partition to Rev. 22.0 with `FIX_DISK`. You can convert from an older revision forward to a more recent revision, up to Rev. 21.0, with the `-CONVERT_19`, `-CONVERT_20`, and `-CONVERT_21` options.

Cannot convert a revision *nn* disk to revision 21 disk.

The only partitions that can be converted to Rev. 21.0 are pre-Rev. 21.0 partitions. You have attempted to convert a Rev. 21.0 or later partition to Rev. 21.0.

Cannot convert a revision *nn* disk to revision 22.1 disk.

The only revision partition that you can convert to Rev. 22.1 with `FIX_DISK` is a Rev. 22.0 partition. To convert an earlier revision partition to Rev. 22.1, you must use `MAKE`.

Cannot determine controller type (`RCINF$`)

The controller type cannot be determined; a nonintelligent controller is therefore assumed.

Cannot find comdev in disk list!

`-COMDEV` was specified as an option on the command line, but the partition specified is not added to the system. Use `STATUS DISKS` to verify the *pdev* of the `COMDEV` (logical device 0) and reenter the command line with the correct *pdev*.

Cannot `FIX_DISK` a `-IC` mode disk on a non-intelligent controller!

You cannot run `FIX_DISK` on a Rev. 22.0 partition in Dynamic Badspot Handling (`-IC`) mode that is not associated with an intelligent disk controller except to switch controller modes. You can use only the `-FIX` option and either the `-IC` option or the `-AC` option.

Cannot process `BADSPOT` for records less than `XX`!

A certain number of records is reserved at the beginning of a partition for system information. The size of this area depends on the size of the partition. Badspots cannot occur in this region. This message indicates that a badspot falls in this reserved region. If this region contains an uncorrectable badspot, `PRIMOS` cannot use the partition.

Cannot read DBS file!

Either the first partition is not assigned to you or the DBS file is corrupted. In the case of a corrupted DBS file, FIX_DISK will reconstruct it if you are presently running FIX_DISK on the first partition. If you are not presently running FIX_DISK on the first partition, do so and FIX_DISK will reconstruct the DBS file.

Cannot read record XX of DBS file. Will rebuild DBS and RMA area.

A bad record is encountered while FIX_DISK is reading the DBS file. FIX_DISK attempts to reconstruct the DBS file and the RMA if this is the first partition. If it is not the first partition, run FIX_DISK on the first partition.

Cannot update DBS file!

In trying to update the DBS file, FIX_DISK could not read the file. Be sure that the first partition is added to the system or is assigned to you at the supervisor terminal.

Cannot use -IC/-AC options on a Pre-Revision 21 disk!

You can change the controller mode and use dynamic badspot handling and mirroring only on Rev. 21.0 and later partitions.

Cannot write DBS file!

The first partition may not be available so that FIX_DISK can write to the DBS file. Be sure that the first partition is assigned to you and that the disk drive where the first partition is located is not write-protected. It is also possible that there is a badspot in the area where the DBS file and the RMA must go.

Cannot write updated DBS file!

The first partition may not be available so that FIX_DISK can write to the DBS file. Be sure that the first partition is assigned to you and that the disk drive where the first partition is located is not write-protected.

Changed to default ACL pointer.

Every file must point either to the default ACL, to a specific ACL, or to an access category. Every access category must point to a specific ACL. Every ACL must point back either to a file or to an access category. If a file or an access category does not meet these criteria and you specify the -FIX option, that file's ACL pointer is set to the default.

Command Line Error. (get_arg)

For more information type "fix_disk -help"

You have made an error on the command line. Reenter the command line and use correct syntax or use the -HELP option of FIX_DISK to get help.

The current record address (CRA) is bad.

It should be XX is YY.

Bad record address = XX BRA = YY Father = ZZ Type = N

The Current Record Address (CRA) pointer of this record does not match the current record. If you specify the -FIX option, the CRA is corrected.

The DAM index is too long to represent the DAM file!

The DAM index is too long to represent the data records!

There are fewer data records in a DAM file than the DAM file's index indicates. If you specify the -FIX option, the index is truncated. You can restore the file by using a backup procedure.

The data count in the first record of the UFD is too small.

The information needed by PRIMOS to access this file or record does not exist. Use the -FIX option to correct the data count.

Date and time not set. Type START or REN to re-enter.

FIX_DISK must have the date and time set in PRIMOS in order to run. This is usually not a problem because the date and time are set when you ordinarily boot PRIMOS either by PRIMOS reading the diagnostic processor clock or by explicitly setting the date and time with the PRIMOS SETIME command at the end of the boot and configuration processes. If you use the MTRESUME command, however, to repair a damaged command device or for some other reason and, to do so, you boot PRIMOS from tape on an older machine, you must use the PRIMOS SETIME command to set the date and time before invoking FIX_DISK.

DBS file is corrupted. Run FIX_DISK on partition *pdev*.

This is not the first partition, but the DBS file version or the DBS file header or the DBS file entry size on the first partition is corrupted. Run FIX_DISK on the first partition *pdev*.

DBS file is corrupted. Will rebuild DBS and RMA area.

This is the first partition and the DBS file version or the DBS file header or the DBS file entry size is corrupted. FIX_DISK attempts to reconstruct the DBS file and the RMA.

The DBS file cannot be read, ignored!

The DBS file was not found in the current partition. See the previous message.

Deactivating DBS file

This message indicates a conversion to -AC mode or a conversion to a Rev. 21.0 first partition in -AC mode; the DBS file is deactivated.

Directory is longer than 64K!

The maximum size of a directory is 64K halfwords. If a directory exceeds this limit, truncation occurs if you specify the -FIX option. The directory information is probably corrupted and not really longer than 64K words. You can restore the directory from a backup.

The Directory Used count is bad. It should be YY instead of XX.

The directory-used count for this directory does not match the directory-used count that is calculated by FIX_DISK. (The directory file itself and all the files in it are counted.) If you specify the -FIX option, the directory-used count is corrected.

Disk is full! *filename* file entry cannot be created.

There is insufficient space on the partition for FIX_DISK to write the directory entry for the BADSPT file or the DBS file that was just created. FIX_DISK continues, but badspot handling is disabled. Make room on the partition by deleting some files and then rerun FIX_DISK.

Disk is full! *filename* file is incomplete.

There is insufficient space on the partition for FIX_DISK to write more information into the BADSPT file or the DBS file. The appropriate badspot file then cannot contain

any new badspot information. Make room on the partition by deleting some files and then rerun FIX_DISK.

Disk is full! *filename* file cannot be created.

There is insufficient space on the partition for FIX_DISK to write the BADSPT file or the DBS file. The appropriate badspot file then does not exist. FIX_DISK continues, but badspot handling is disabled. Make room on the partition by deleting some files and then rerun FIX_DISK.

Disk is full! Cannot allocate another record! (fix_disk)

There is insufficient space on the partition for FIX_DISK to allocate another record for badspot processing. Make room on the partition by deleting some files and then rerun FIX_DISK.

Disk is in *-mode_Controller* mode with *dir* Allocation

Disk is in *dir* allocation mode.

If the partition is a Rev. 21.0 or later partition, FIX_DISK tells you the controller mode (ALL or INTELLIGENT) and the type of record allocation (Forward or Reverse); otherwise only the type of record allocation is displayed.

Disk not properly shutdown. Full fix disk will be run.

FIX_DISK determines from a bit in the RAT that the disk was not shut down properly and the partition is a standard partition. Full FIX_DISK will be run on this partition even if you specified the -FAST option because fast FIX_DISK may not fully correct the situation on a standard partition. See Table 7-1 for recommendations concerning running FIX_DISK.

Disk Read/Write Error. Record = XX

Track = YY Head = ZZ Sector = SS.

An error occurred during the reading or writing of record XX. If you specify the -FIX and -TRUNCATE options, the file is truncated on read errors and this badspot record is added to the BADSPT file. If you do not specify the -TRUNCATE option, a zero record is added.

Disk record address 'XX is illegal!

A record points to address XX (octal), which is outside the current partition. If you specify the -FIX option, the address is corrected.

The DSKRAT header has wrong length. (RAT_CK).

The file structure of DSKRAT is bad. (RAT_CK).

The file system is larger than the disk. (RAT_CK)

The length of UFD header is incorrect.

The number of heads is different. It should be YY is XX.

The physical record size is different. (RAT_CK).

You may see one or more of the above five error messages. They mean that the information contained in the DSKRAT header does not correspond to the information computed from the physical device number. Either the pdev is incorrect or the DSKRAT header contains incorrect information. You may not have created the partition by using MAKE or you may need to remake the partition.

If you do not specify the -INTERACTIVE and -FIX options, FIX_DISK aborts. Otherwise, FIX_DISK prompts

Fix DSKRAT header?

A NO response causes FIX_DISK to abort. A YES response initiates a dialog that results in fixing the DSKRAT. FIX_DISK first prompts

Physical Disk?

Enter the physical device number (pdev) of the partition containing the DSKRAT to be fixed. FIX_DISK computes the number of records in the partition from the pdev and the model of disk by displaying a list of models and then prompting for the model:

Valid disk model names:

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

Please enter model name of disk...

If you enter an invalid model, FIX_DISK reprompts for a model name. FIX_DISK then prompts

Split disk?

If part of the partition is to be used for paging, then enter YES; otherwise, enter NO. If you answer YES, FIX_DISK then prompts

Paging Records (Decimal)?

Enter the number of records to be used for paging. This number should be divisible by 16. If you specify too many paging records, you will not leave enough room for system overhead, such as the DBS file if this is a first partition of a Rev. 21.0 or later disk (see Table 5-5 in Chapter 5), or a minimum amount (3 records) at other disk revisions, FIX_DISK displays the following message and reprompts for the number of paging records:

Too many paging records.

After you enter a reasonable number for paging records, FIX_DISK then displays the partition number, file records, and paging records in the form

DISK	FILE-RECORDS	PAGING-RECORDS (DECIMAL).
XX	YY	ZZ

and prompts

Parameters OK?

If the numbers are incorrect, enter NO. FIX_DISK attempts to recompute the parameters. If the numbers are correct, enter YES. You then either see the message This must be a rev 20, 21, 22. or 22.1 partition. based on features this partition supports (such as variable geometry) and are prompted

```
Is this a rev 22.1 partition?
Is this a rev 22 partition?
Is this a rev 21 partition?
Is this a rev 20 partition?
```

or you are prompted

```
Is this a rev 19 partition?
```

Answer these questions appropriately depending on whether you are repairing a Rev. 22.1, a Rev. 22.0, a Rev. 21.0, a Rev. 20.0, or a Rev. 19.0 partition. FIX_DISK then repairs the DSKRAT from the information you have supplied. If this is a Rev. 22.0 or later partition, FIX_DISK then prompts to determine if this is a robust partition:

```
Is this a robust partition?
```

If this is a Rev. 22.1, a Rev. 22.0, a Rev. 21.0, or a Rev. 20.0 standard partition or you are converting to either Rev. 21.0 or Rev. 20.0, you are then prompted for the record allocation scheme you want, in this manner:

```
Forward Sectoring ?
Reverse Sectoring ?
```

These two lines are displayed one at a time and one will have the word (default) after it, depending on the CPU and disk controller in your system (see Table 10-2). If this is a Rev. 22.1 robust partition, you are not prompted for the record allocation scheme; it is set to forward allocation.

After you answer the record allocation prompt, FIX_DISK prompts for the disk controller mode in a similar manner:

```
Intelligent controller mode ?
All controller mode ?
```

The default mode is indicated by the word (default).

DSKRAT MISMATCH!

The record allocation information in the DSKRAT disagrees with the record allocation information generated by FIX_DISK during processing. You did not specify -FIX so FIX_DISK does not correct the DSKRAT file. If the DSKRAT is all right, the message DSKRAT OK is displayed.

DSKRAT UPDATED!

The record allocation information in the DSKRAT disagrees with the record allocation information generated by FIX_DISK during processing, and you specified the -FIX option. FIX_DISK corrects the DSKRAT file. If the DSKRAT is all right, the message DSKRAT OK is displayed.

Entry at word 'XX at hash index 'XX is in a circular list.

The hash links are chained in a circular fashion. If you specify the -FIX option, the hash chain is terminated at the previous entry to correct the circularity.

Entry at word *XX* is at hash index *YY*, it should be at *ZZ*.

`FIX_DISK` computed the hash index for an entry in the directory and found that the entry is not where the index indicates. Use the `-FIX` option to correct the entry.

Entry at word *XX*, hash index *YY* is not valid.

Removed from hash table.

The entry found by `FIX_DISK` is not a hashed entry but may have pointed to an entry that does not have a name, such as an ACL or a deleted or free entry. Use the `-FIX` option to repair the entry.

Entry at word *XX* not in free list.

A free, or vacant, entry is found that is not in the list of free entries. The file system maintains a list of free, or vacant, directory entries. These are entries that have been deleted from the directory and thus are freed for use. To save space in the partition, you can remove these free entries by using the `-UFD_COMPRESSION (-CMPR)` option to compress the directory. This message indicates that the directory is damaged. Use the `-FIX` option to repair the directory and to put the entry in the free list.

Entry at word *XX* not in hash table.

The entry for a file is not in the directory's hash table. Use the `-FIX` option to put it in the hash table.

Entry at word *XX* was already in hash table.

An entry is found in a directory's hash table more than once. Use the `-FIX` option to remove duplicate entries, leaving only one entry for this file.

This entry is deleted.

If an error is encountered while `FIX_DISK` is checking the badspot file, the entry for that badspot is deleted from `BADSPT` if you specify `-FIX`.

EOF occurs in the middle of entry at record '*YY*', word *XX*

A directory ends in the middle of the last directory entry. If you specify the `-FIX` option, the entry is deleted.

Equivalence Section Begins.

The equivalence section has *dd* entries.

`FIX_DISK` is processing the equivalence section of the badspot file. If processing is successful, the number of entries, *dd*, is displayed. Otherwise, another error message is displayed.

Error reading first record of subsequent extent (uncorrectable).

An uncorrectable error is encountered in checking a CAM file and attempting to read an extent subsequent to the first extent in the file. `FIX_DISK` truncates the file beginning at the bad extent. You can restore the file from the most recent backup.

Error occurs during validation.

`FIX_DISK` was unable to read the MFD record on a file system partition. You may need to remake the partition.

Extent map full, unable to remap badspot record.

This message refers to Contiguous Access Method (CAM) files. The index, or extent map, can contain 340 entries for pre-Rev. 22.1 partitions or 16381 entries for a Rev. 22.1 or later partition. The extent map is full and thus no room is available to remap a badspot because a new extent cannot be created. If you specify the -FIX option, the record is truncated from the file. You can restore the directory from a backup.

Extent map is invalid.

The Beginning Record Address (BRA) pointer (XX) is bad, it points to a record that belongs to another file.

This error refers to a CAM file on a robust partition. The BRA does not match the entry for the file in the directory. If you specify the -FIX option, FIX_DISK deletes the file. You can recover the file from a backup copy.

Fatal error during remapping: record not found in dam index

A record is found to be bad, but FIX_DISK is unable to remap it to a new location. The record is deleted.

Fatal error has occurred in the file structure of a DAM Segment Directory!

FIX_DISK is unable to fix the pointers in a DAM segment directory. The directory is truncated or deleted. You can restore the directory from a backup.

The father pointer is bad.

It should be XX is YY.

Bad record address = XX BRA = YY Father = ZZ Type = N

The Beginning Record Address (BRA) pointer of the first record of a file does not point to the BRA of the directory or SEGDIR in which this file is entered (its father). If you specify the -FIX option, the file is deleted. You can restore the directory from a backup.

The *filename* file cannot be read, ignored!

While FIX_DISK is processing existing badspots, FIX_DISK encounters an error while attempting to read either the DBS file or the BADSPT file. The badspot information is lost. If you have a list of the existing badspot, such as a copy of the -DUMP_DBS display, you can reenter the badspots by using the -ADDBADS option.

File *filename* does not reference an ACL or Access Category!

The ACL pointer of a file does not point to a valid ACL or access category. If you specify the -FIX option, the file reverts to default protection.

File entry in password ufd has non-zero acl pointer!

FIX_DISK has found a file in a password directory that has an ACL set on it. If you specify the -FIX option, the pointer is corrected.

File *filename* is not pointed at by ACL it points to!

The ACL pointer of a file does not point to a valid ACL or access category. If you specify -FIX, the file reverts to default protection.

File *filename* points outside the directory!

The ACL pointer of a file points to something in a different directory. If you specify the -FIX option, the ACL pointer is changed to the default type.

The file structure of DSKRAT is bad. (RAT_CK)

The DSKRAT file contains bad record pointers or contains inconsistent information. If you specify both the -INTERACTIVE and the -FIX options, you are prompted

Fix it?

If you answer YES, FIX_DISK attempts to reconstruct the DSKRAT file structure. Otherwise, FIX_DISK aborts. (See the message The DSKRAT header has wrong length. (RAT_CK).)

The file system is larger than the disk. (RAT_CK)

The partition size, as computed from the information in the RAT built by FIX_DISK, is larger than the size specified by the pdev you entered with the -DISK option on the command line. If you specify the -INTERACTIVE and -FIX options, you are prompted Fix DSKRAT header? If your response is YES, you are further prompted for information to repair the DSKRAT. If your response is NO or if you did not specify the -INTERACTIVE and -FIX options, FIX_DISK aborts. Use the -FIX option with the -INTERACTIVE option to repair the DSKRAT. (See the message The DSKRAT header has wrong length. (RAT_CK).)

file type mismatch!

it should be 0 instead of XX.

An error is found in the DAM index level; FIX_DISK corrects the error.

File type mismatch!

It should be XX instead of YY.

FRA = MMMM BRA = NNNN.

File type mismatch!

It should be XX instead of YY.

FRA = MMMM BRA = NNNN.

Bad record address = XX BRA = YY Father = ZZ Type = N

The file type in the first record of the file does not match the file type in the directory entry. If you specify the -FIX option, the file is deleted. You can restore the file by using a backup procedure.

First partition (pdev) is not a Revision 21 or higher partition.

Must convert the first partition to Revision 21 before any other partitions

The partition pdev is the first partition on this physical disk and must be converted before any other partitions on this disk, because the DBS file and the RMA for the entire physical disk must be on the first partition. If you intend to convert partitions, convert the first partition (pdev) first.

First partition must be in the same mode as the conversion on this partition

In converting a partition, that is not the first partition, you have specified either -INTELLIGENT_CONTROLLER or -ALL_CONTROLLER mode and the first partition of this physical disk is not in the same mode. Either select the same mode for this partition as for the first partition or change the mode of the first partition.

The first partition needs to be assigned for this conversion

To select the controller mode of a Rev. 22.0 partition that is not the first partition or to convert to a Rev. 21.0-format partition, you must assign the first partition so that FIX_DISK has access to the DBS.

FIX_DISK is unable to read badspot 'XX' to remap the record.

In order to remap a data record, FIX_DISK must read the record first. In this case, FIX_DISK encounters an uncorrectable read-error and cannot read the data. If this is a Rev. 21.0 conversion, the message *Resetting badspot file!* is also displayed. If you used the -FAST option, FIX_DISK advises you to run full FIX_DISK (without -FAST) and add the badspot. If you did not initially use -FAST, FIX_DISK advises you to rerun full FIX_DISK and add the badspot. The data in the record that has become a badspot is lost in these cases; restore the file from backups.

Fix DSKRAT header?

You have specified the -INT and -FIX options. See the discussion under The DSKRAT header has the wrong length.

-FIX option must be specified if -CMPR, -CONVERT_19, -CONVERT_20, -CONVERT_21, -INTELLIGENT_CONTROLLER, -ALL_CONTROLLER, ADD_BADSPOT, -OVERRIDE_DEFAULT_INTERLEAVE, -RESTORE_DEFAULT_INTERLEAVE -MIN_EXTENT_SIZE or -MAX_EXTENT_SIZE option is selected.

In order to compress directories, convert partitions to another revision, select controller modes, add badspot entries to BADSPT, select record allocation mode, or change the maximum or minimum extent size of CAM files, you must specify the -FIX option. If you do not, FIX_DISK aborts.

For a Rev 21 conversion, the -disk_type model MUST be specified.

When you use the -CONVERT_21 option to convert a partition to Rev. 21.0 format, you must also use the -DISK_TYPE option. FIX_DISK builds the DBS file and the RMA on a Rev. 21.0-format partition and must know the type of disk, because the size of the DBS file and RMA depend on the disk geometry.

The following CAM file is in the DBS/RMA area and will prevent conversion of this disk to revision 21.

pathname

After this message appears, you see the following message.

CAM file(s) in DBS/RMA area prevent conversion to rev 21 disk.

Please save & delete these files, and restore following the conversion.

When converting a partition to Rev. 21.0 with -CONVERT_21, a CAM file, *pathname*, is found in the area of the first partition where it is necessary for FIX_DISK to create a DBS file and the RMA. FIX_DISK aborts. If you want to convert this partition to Rev. 21.0, save the offending files to tape, delete them from the disk, convert the disk, and then restore the CAM files you saved and deleted.

The forward pointer XX is bad, it is not in the range of the current partition.

Bad record address = XX BRA = YY Father = ZZ Type = N

The address that the forward pointer points to is not between zero and the maximum record address of this partition. If you specify the -FIX option, the file is truncated. You can restore the file by using a backup procedure.

The forward pointer XX is bad, it points to a record that belongs to another file.

Bad record address = XX BRA = YY Father = ZZ Type = N

The record that the forward pointer points to belongs to another file. This error may occur if the current DSKRAT is bad or if the BADSPT file was changed since the last time FIX_DISK was run. If you specify the -FIX option, the file is truncated. You can restore the file by using a backup procedure.

The forward pointer of the previous record does not point to the pre-remap record!

The forward pointer of the previous record points to another file after having been remapped. This error may occur if the current DSKRAT is bad or if the BADSPT file was changed since the last time FIX_DISK was run. If you specify the -FIX option, the file is truncated. You can restore the file by using a backup procedure.

The forward pointer of the top level index record of DAM file is not zero.

The top-level index must be only one record long; therefore, the forward pointer of this record must be zero. If you specify the -FIX option, the pointer is set to zero.

Free list at word XX contains a non free entry.

The directory entry found in the free list is an entry that is in use. The file system maintains a list of free, or vacant, directory entries. These are entries that have been deleted from the directory and thus freed for use. To save space in the partition, you can remove these free entries by using the -UFD_COMPRESSION (-CMPR) option to compress the directory. Use the -FIX option to remove that entry from the free list.

Free list at word XX contains a too small free entry.

An invalid entry is found in the free list. The file system maintains a list of free, or vacant, directory entries. These are entries that have been deleted from the directory and thus freed for use. To save space in the partition, you can remove these free entries by using the -UFD_COMPRESSION (-CMPR) option to compress the directory. Use the -FIX option to remove that entry from the free list.

Free list is circular at word XX.

An entry is found in the free list that points to the entry that points back to it. The file system maintains a list of free, or vacant, directory entries. These are entries that have been deleted from the directory and thus freed for use. To save space in the partition, you can remove these free entries by using the -UFD_COMPRESSION (-CMPR) option to compress the directory. Use the -FIX option to correct the pointers.

Handling soft disk error recovery for 'XX

FIX_DISK has found a correctable error at record number XX (octal) and will add it to the badspot file.

The hash table size is incorrect.

The size of the hash table for a directory is either less than the minimum or greater than the maximum. Use the -FIX option to correct the inconsistency.

Inconsistent entry. Record = XX, Word = YY
Entry is deleted!

The information in an entry within a directory is inconsistent and cannot be reconciled. If you specify the -FIX and -DUFE options, the entry of this file is deleted. You can restore the file by using a backup procedure.

Incorrect hash function version number; is XX, should be YY.

The algorithm used to compute entries for the hash table is invalid. Other messages will be displayed concerning entries in the hash table. Use the -FIX option to correct the hashing function.

Incorrect logical end of file (correctable).

A CAM file contains more records than are indicated in the extent map of the file. Use the -FIX option to fix the inconsistency.

Index entry NN is bad, should be XX is YY.

The list of records at NN contains a record address that is not part of this directory. Use the -FIX option to correct the entry.

Index entry NN is bad, should be 0 is XX.

The list of records at NN contains a record address that is not part of this directory. Use the -FIX option to correct the entry.

The index level of this DAM file is incorrect.
It should be YY instead of XX!

The index level word of this record is incorrect. It should be zero for SAM files or one less than the previous level for DAM files. If you specify the -FIX option, the index level word is fixed.

The index level of this record is incorrect.
It should be XX is YY.

Bad record address = XX BRA = YY Father = ZZ Type = N

A file that is not a DAM file is found to have an index level other than 0. It is fixed by making the index level 0.

The index level of the top index record of a DAM file is incorrect.
It is XX, it should be > 0 & < 4.

The index level of the top index record of a DAM file is incorrect. It should be between 0 and 4. If you specify the -FIX option, the index record is corrected.

The index of this DAM file is too short. The DAM index may be rebuilt!

A DAM file contains more data records than its index indicates. If you specify the -FIX option, the index is fixed if the extra index words will fit into the index record; if the words will not fit, the next message is displayed.

The index of this DAM file is too short to represent the data records.

This message is displayed if the condition noted in the previous message is not reparable. A DAM file contains more data records than its index indicates. If you specify the -FIX option, the inconsistency between the index and the number of data records is resolved, and the file is truncated. You can restore the file by using a backup procedure.

Insufficient disk space available for conversion. Please free up *dd* records on partition

When you use the `-CONVERT_21` option on the first partition, `FIX_DISK` must move records from the DBS area and create the DYNBSP file. Before doing this, `FIX_DISK` must ensure that there is enough space to hold the DBS file, the RMA records, and the DYNBSP file. `FIX_DISK` finds that there is not enough space and informs you that you must delete *dd* records from the partition.

Internal error - UFD overflow!

The number of files and the length of their filenames are greater than the amount of space allotted for them in the `FIX_DISK` database. This error is unlikely to occur unless you have a partition with an extremely large number of small files with long (32-character) filenames and many deeply nested directories.

Invalid disk model name; Valid disk types are:

Either you did not enter a valid disk type with the `-DISK_TYPE` option or you did not enter any disk type. `FIX_DISK` displays a list of types; choose the correct one from the list and reenter the command line. (The list is shown in the discussion of the message The DSKRAT header has the wrong length.)

Invalid max nested level, set to max of 3855.

The maximum level of nested directories specified with the `-MAX_NESTED_LEVEL` option is invalid and out of the range of 0 through 3855. It is set to 3855.

The length of UFD header is incorrect.

See the message The DSKRAT header has wrong length. (`RAT_CK`).

The MFD is full, unable to add *filename* file.

The partition has insufficient space to enable `FIX_DISK` to write or add to either the BADSPT file or the DYNBSP file. `FIX_DISK` continues, but badspot handling is disabled. Clean up the partition by having users delete unnecessary files and directories.

MINIMUM_EXTENT_SIZE cannot be greater than MAXIMUM_EXTENT_SIZE. (`rat_ck`)

Current MINIMUM and MAXIMUM extent values are: *min*, *max*

You specified the minimum extent size of a CAM file to be larger than the maximum extent size. The current maximum and minimum extent size are displayed and `FIX_DISK` aborts. If you want to change the maximum and minimum extent sizes, reenter the command line with proper values.

MINIMUM_EXTENT_SIZE cannot be greater than MAXIMUM_EXTENT_SIZE. (`cl_par`)

Please respecify MINIMUM and/or MAXIMUM extent values.

You specified the CAM file minimum extent size to be larger than the maximum extent size. Reenter the *size* values on a new command line.

Missing badspot number. (`cl_par`).

You specified the `-ADD_BADSPOTS` option, but did not enter any badspot records on the command line.

Missing maximum extent size (`cl_par`)

Missing minimum extent size (`cl_par`)

You specified either the `-MAXSIZ` or the `-MINSIZ` option but did not specify the corresponding CAM file extent *size*. Reenter the command line, specifying the appropriate *size*.

Moving DBS file system records.

In Nondynamic Badspot Handling (-AC) mode, any data stored in the RMA must be moved to the partition containing the file associated with that data.

Multiple CONVERT_ options can not be specified at the same time.

Only one of the -CONVERT_19, -CONVERT_20, or -CONVERT_21 options can be specified on a command line.

The next extent map record address of the previous map does not point to the pre-remap record!

The address of the next extent map in the previous extent map does not point to either the beginning record address or the remapped beginning record address of the next extent map. If you specified the -FIX option, the file is truncated. You can restore the file from a recent backup.

The next index does not match the forward pointer of the current data record!

The pointers of the index section and the data section do not agree. If you specify the -FIX option, the following actions are taken: the backward pointer of the record that is pointed to by the DAM index and the backward pointer of the record that is pointed to by the forward pointer of the current data record are examined; the record whose backward pointer points to the previous data record is chosen; if neither backward pointer points to the previous record or both backward pointers point to the previous record, the file is truncated.

Non zero data count(s) past logical eof (correctable).

FIX_DISK has found an indication of more records in a file going beyond the logical end-of-file (EOF). If you specify the -FIX option, the data counts of all records beyond the logical EOF are set to zero.

The number of heads is different. It should be YY is XX.

See the message The DSKRAT header has wrong length. (RAT_CK).

Paging Records (Decimal)?

You are interactively repairing the DSKRAT. See the discussion under The DSKRAT header has the wrong length.

Parent of a file is not a UFD or SEGDIR

The pointer to a directory or to a segment directory identifies a file header. FIX_DISK aborts.

The partition cannot be handled by this version of Fix_disk.

FIX_DISK found a discrepancy in the DSKRAT header or in the pdev indicating that the partition on which you are running FIX_DISK and the version of FIX_DISK you are using are incompatible. FIX_DISK aborts unless you specify the -FIX and the -INTERACTIVE options. If you do specify the -FIX and the -INTERACTIVE options, you are prompted

Do you wish to change the disk revision?

A NO response causes FIX_DISK to abort. A YES response initiates the dialog discussed under the message The DSKRAT header has wrong length.

Partition not shutdown properly during the previous session;
please run fix_disk!

The partition was not shut down under PRIMOS with the SHUTDN command; for example, the system halted, the disk drive was spun down, or the partition was written on by PRIMOS II (DOS). This message results if you run FIX_DISK with the -CHECK option; it is the only message you will see indicating problems when you use the -CHECK option. To check for other problems, run FIX_DISK but do not specify either -CHECK or -FIX. Use -FIX to correct any problems.

Do you wish to change the disk revision?

A NO response causes FIX_DISK to abort. A YES response initiates the dialog discussed under the message The DSKRAT header has wrong length.

Physical Device Number is missing. (CL_PAR)

The physical device number is not specified in the command line with the -DISK option. FIX_DISK aborts. Reenter the command line and specify the correct pdev.

Please answer "YES" or "NO"?

The previous prompt requires a yes or no answer. Acceptable answers, in either uppercase or lowercase, are YES, YE, Y, NO, N.

The physical record size is different. (RAT_CK).

See the message The DSKRAT header has wrong length. (RAT_CK).

Pre Rev 22 disks may not be robust.

In checking the DSKRAT, FIX_DISK found the partition to be labeled as robust but of a disk revision prior to Rev. 22.0. FIX_DISK changes the DSKRAT to indicate that the partition is not robust.

The prev. extent map record address of the next extent map record does not point back to the pre-remap record!

The record address of the previous extent map does not point to either the beginning record address or the remapped beginning record address for the next extent. If you specified the -FIX option, the file is truncated. You can restore the file from a backup.

Processing add badspot request for XX

You used the -ADD_BADSPOT option and the badspot (record address XX) is being remapped to the RMA.

The Quota system may be incorrect.

This message is issued if the partition was changed under PRIMOS II or if the partition was not shut down correctly. Because PRIMOS II does not support quotas, some directories on this partition may contain incorrect quota information.

Record address XX is out of range YY.

The next record address XX is out of range of the current record address YY in a segment directory. If you specify the -FIX option, the file is deleted. You can restore the file by using a backup procedure.

The record: *XX* has not been initialized
The record will be zeroed out.

The record: *XX* has not been initialized
The file will be truncated.

During verification of the integrity of the header of a record on a robust partition, the beginning record address for the record is found to be incorrect or the record's identification is found not to be unique. If you used the *-TRUNCATE* option, the file is truncated; otherwise, the record is replaced with zeros.

REMAP: Track = *NN* Head = *NN* Sector = *NN* Record = '*XX*'
You specified the *-LIST_BADSPOTS* option and the badspots are listed.

Re-run *FIX_DISK* without *-FAST* option to add this badspot.
See the message *FIX_DISK* is unable to read badspot '*XX*' to remap the record.

Resetting badspot file!
See the message *FIX_DISK* is unable to read badspot '*XX*' to remap the record.

Segdir has more than 64K entries!
The maximum size of a segment directory (*SEGDIR*) is 64K halfwords. If a *SEGDIR* exceeds this limit, it is truncated if you specify the *-FIX* option. You can restore the directory by using a backup procedure.

Single zero record added starting at word *XX*.
chars length

A zero record has been created in place of an unreadable record in a SAM file.

Single zero record added starting at word *XX*.
index length

A zero record has been created in place of an unreadable record in a DAM file and the index and index length are displayed.

Split disk?
You are interactively repairing the *DSKRAT*. See the discussion under The *DSKRAT* header is the wrong length.

Structure error for a DAM segdir. (*SEGDR_CHK*)
The header information for a DAM segment directory is inconsistent with the file information. If you specify the *-FIX* option, the inconsistency is corrected.

System console command only (*fix_disk*).
You must be at the supervisor terminal to use the *FIX_DISK -COMDEV* command.

System file is bad - ignored!
An error that would normally cause deletion of a file has been found in one of the MFD's special files (*BOOT*, *MFD*, *DSKRAT*). The file will not be deleted. To correct this condition, you may need to remake the partition by using *MAKE*.

This is a remapping record of record address '*XX*'.
A read error occurred while *FIX_DISK* was attempting to remap a record. The record is not remapped, but *FIX_DISK* continues.

This is a revision *rev* robust partition.

This is a revision *rev* partition.

FIX_DISK determines the type (robust or standard) and the revision of the partition that is being repaired.

Too many paging records.

You are interactively repairing the DSKRAT. See the discussion under The DSKRAT header is the wrong length.

The Tree Used count is bad. It should be YY instead of XX

The tree-used count of this directory does not match the tree-used count that is calculated by FIX_DISK. If you specify the -FIX option, the tree-used count is fixed.

Treename is not available.

FIX_DISK found an error on its second pass in this record. Information on what file the record belongs to is missing. This error should occur only if the disk is failing.

2 files point to the same record!

bad record address = XX

BRA = YY Father = ZZ

2 files point to the same record!

Bad record address = XX BRA = YY Father = ZZ Type = N

Two files each have a record with the same address; that is, they point to the same record. If you specify the -FIX option, all entries for the second and subsequent files pointing to that record are truncated. You can restore the files by using a backup procedure.

The UFD header is missing.

The header of the MFD is missing or the wrong type. If you specify the -FIX option, the directory file is deleted. Try to repair the header by using the -INTERACTIVE and -FIX options.

The UFD header length is incorrect.

The length of the directory header is wrong. If you specify -FIX, it is corrected.

The UFD header of the MFD is missing.

In checking to see if the partition is a valid PRIMOS file system, FIX_DISK does not find the MFD header entry. FIX_DISK aborts. Reformat the partition by using MAKE.

Ufd nesting exceeds maximum specified.

FIX_DISK cannot follow the directory tree because directories are nested to more than *n* levels deep (default *n* = 99). FIX_DISK aborts unless you specify the -AUTO_TRUNCATION option, in which case directories that are nested too deeply are truncated. You can restore the directory from a backup.

UFD truncated.

If an error occurs in a directory while FIX_DISK is checking the directory (the name of which is displayed in the BEGIN message), the directory is truncated if you specify the -FIX option. You can restore the directory from a backup.

Unable to add_badspot 'XX'. (fix_disk)

FIX_DISK is unable to remap a badspot to another record because the partition is full. Clean up the partition and rerun FIX_DISK.

Unable to add zero record(s).

FIX_DISK found an unreadable record in a file, but is unable to add a record of zeros because the partition is full. FIX_DISK aborts.

Unable to add zero record(s). (fix_disk)

FIX_DISK found an unreadable record in a SAM file, but is unable to add a record of zeros because the partition is full. The file is truncated. You can restore the file by using a backup procedure.

Unable to correctly handle password origin directories.

Origin was: *origin_path*

Please use the -set option to the ORIGIN command to set the origin of the console. (fix_disk)

This message is displayed only when the -COMDEV option is used and only when the supervisor terminal origin directory is a password (not an ACL) directory. Use the command ORIGIN -SET *origin_path* to set the origin directory to *origin_path*. The supervisor terminal origin directory is normally CMDNC0.

Unable to create a new extent map.

Just previous to this message, FIX_DISK displays the message Disk is full!; thus, FIX_DISK cannot create the new extent map that is necessary in order to remap a badspot. If you specify -FIX, the file is truncated. You can restore the file from a backup.

Unable to move record, rewriting to attempt to fix it up.

FIX_DISK has encountered a badspot in one of the first 16 records of the partition and this record cannot be moved. The first 16 records on a partition cannot be defective. FIX_DISK attempts to write to the record again and if the write attempt is unsuccessful, FIX_DISK aborts. If FIX_DISK aborts, call a representative from your Customer Support Center.

Unable to read extent map.

FIX_DISK is unable to find the extent map of a CAM file possibly because of a badspot. If you specify the -FIX option, the file is deleted. You can restore the file by using a backup procedure.

Unable to read next extent map.

FIX_DISK cannot read a subsequent extent map after the first extent map, possibly because of a badspot. If you specified the -FIX option, the file is truncated at this point. You can restore the file from a recent backup.

Unable to register geometry information with PRIMOS. (RAT_CK)

You are using the wrong version of FIX_DISK. Rev. 20 FIX_DISK attempts to determine the number of sectors per track on the disk, since disks may have different geometries at Rev. 20. FIX_DISK aborts.

Unable to reset origin directory at this revision of PRIMOS.

Origin was: *origin_path*

You used the -COMDEV option and an older version of PRIMOS or FIX_DISK is unable to reset the origin. Cold start the system to set the origin directory of the supervisor terminal.

Unable to skip second badspot in this file. (*fix_disk*)

BRA = *XX*, FRA = *YY*.

FIX_DISK has found more than one bad record in one file, and is unable to determine how many bad records have been found. If you specify the -FIX option, the file is truncated. You can restore the file by using a backup procedure.

Unexpected error *NN* from hash_uid. (*fix_disk*)

While attempting to compute a hash index, the hashing algorithm returns an error. The error is ignored. If you specify the -FIX option and this message is displayed, there may be other problems with the disk. Call you Customer Support Center.

Unknown entry type (*XX*). Record = *YY* Word = *ZZ*

FIX_DISK does not recognize the entry type of the current record. If you specify the -FIX option, the file to which the record belongs is truncated. You can restore the file by using a backup procedure.

Unknown file type *XX* encountered.

Record = *YY* Index = *ZZ*

The file type *XX* in the file entry is unknown. It is either an illegal file type or a new file type that is not recognized by this version of FIX_DISK. If you specify the -DUFE and -FIX options, this file entry is deleted. If you omit these options, the file entry is left untouched, and no compression is performed for the directory in which this file entry resides.

Unknown logical file type.

FIX_DISK does not recognize the logical file type. If you specify the -FIX option on the command line, FIX_DISK changes the file type to the default type and issues the message: Changed to default type.

Unknown number of records removed:

Two zero records added starting at word *XX*.

FIX_DISK has found more than one contiguous bad record. Two records of zeros are added and any additional good records follow the zero records.

Unrecoverable read error. (*fix_disk*)

FIX_DISK cannot read the DSKRAT file while trying to reconstruct a truncated SAM file by adding zero records. The SAM file is left truncated, but other errors will probably occur.

Unrecoverable read error. (*VFY_RAT*)

Unrecoverable read error while reading the DSKRAT file. (*RAT_CK*)

Unrecoverable read error while reading DSKRAT. (*repair_rat_file_header*)

These messages may be issued if the DSKRAT file contains any bad record pointers or contains inconsistent information. If you specify both the -INTERACTIVE and -FIX options on the command line, FIX_DISK attempts to reconstruct the DSKRAT file structure. Otherwise, FIX_DISK aborts. In the latter case, rerun FIX_DISK with the -FIX and -INTERACTIVE options.

Unrecoverable read error reading first partition's DSKRAT file (BADSPPT)
 Unrecoverable error reading first partition's DSKRAT file. (RAT_CHK)

Be sure that the first partition is assigned to you. If it is not, it may be damaged; run FIX_DISK on the first partition in that case.

Unrecoverable read error while reading UFD.(update_dir_count)

This is an extremely unlikely error. A CAM file was remapped and FIX_DISK is unable to do the directory quota update. This does not cause a problem and the quota count will be fixed the next time you run FIX_DISK.

Vacant entry on free list but not a main entry at XX.

A list of free, or vacant, entries in a directory is maintained. These are entries that have been deleted and thus freed for use. Adjacent free entries are merged into one entry. In this case, the free list points within a merged entry rather than to the start of the entry. Use the -FIX option to correct the list.

Warning: Illegal reference to record zero ignored.

A pointer has been found pointing to the BOOT file from a record unrelated to the BOOT file. This message indicates corruption of a directory entry. If you specify the -FIX option, the entry is deleted.

WARNING: The 3rd file entry of the MFD file is not BOOT.

The partition may need to be remade!

FIX_DISK checks that the third entry in the MFD is BOOT. If this entry is missing, FIX_DISK does not abort. If you specified the -UFD_COMPRESSION option on the command line, the message Compression has been disabled! is also displayed.

The word count of record XX is bad.

The word count of this record XX is bad.

Bad record address = XX BRA = YY Father = ZZ Type = N

The data halfword count of a record is not appropriate. For every record except the last record, the data halfword count should equal the record data size. The data halfword count of the last record should be between zero and the record data size. If you specify the -FIX option, the halfword count is set to the appropriate value.

Writing DBS file

FIX_DISK is creating or updating a blank DBS file. The intelligent disk controller fills in the badspot locations.

You may not change the default interleaving on a robust partition.

action option ignored.

Do not use either the -RDI or the -ODI options with robust partitions because the allocation direction is always forward with an interleave factor of three on robust partitions. The *action* you specified, either Restore or Override, is ignored and FIX_DISK continues.

You must use the -auto_truncation flag to process this disk

If directory nesting exceeds the maximum specified, and you did not specify the -AUTO_TRUNCATION option, this message is displayed and FIX_DISK aborts. Either specify more levels of nesting with the -MAX_NESTED_LEVEL option or use the -AUTO_TRUNCATION option to truncate the directory tree.

Zero record will be added to file.

FIX_DISK has encountered an unreadable record. A record of zeros will be added to mark the location of the unreadable record.

MIRRORING MESSAGES

This appendix lists the informational and error messages associated with the mirroring process and the mirroring commands. The messages are presented in alphabetical order with a brief explanation of each message. If an article, such as *The* or *This*, is the first word in the message, the second word is used for alphabetizing. Variable names in the messages are italicized, for example, *pdev*.

For a discussion of mirroring and the mirroring commands, see Chapter 9.

A catch-up copy of primary device *pdev1* to secondary device *pdev2* will be started.

Are you sure you want to continue?

You specified the MIRROR_ON command and the two partitions to be mirrored are not identical either based on their shutdown stamps, or because they were not started or stopped simultaneously. If you want to mirror these partitions and have the catch-up copy start, answer YES. Be sure you know which partition has the most recent data. (See the warning in the Catch-up Copy section of Chapter 9.)

Cannot mirror on COMDEV pdev *pdev* - incorrect Rev, mode, or model.

Cannot mirror COMDEV pdev *pdev* - incorrect Rev. or mode.

Cannot mirror on COMDVM pdev *pdev* - incorrect Rev., mode, or model.

Cannot mirror on PAGINM pdev *pdev* - incorrect Rev, mode, or model.

Cannot mirror PAGING pdev *pdev* - incorrect Rev. or mode.

These messages are displayed at cold start. The *pdev* specified in the configuration file with either the COMDVM, the PAGINM, or the PAGING directive is that of a partition that is not a Rev. 22.0 or a Rev. 21.0 partition or that is not in Dynamic Badspot Handling (-IC) mode.

Cannot mirror on PAGINM pdev *pdev* - mirror table full.

More than 128 partitions have been specified as mirrored pairs and another partition cannot be started as a mirror.

Cannot shut down both - devices *pdev1* and *pdev2* are being used for paging. (mirror_off)

You cannot use the MIRROR_OFF -SHUT_BOTH command on paging partitions. One of the paging partitions must be running while the system is running. You can shut down

one of the pair of mirrored paging partitions. The other paging partition can be shut down only at system shutdown with the SHUTDOWN ALL command.

Cannot shut down primary device *pdev1* - catch-up copy
did not complete successfully and secondary device *pdev2*
is not up to date. (mirror_off)

The copy server logged out before completing the catch-up copy to the secondary partition of the mirrored pair. This may have happened either because an uncorrectable read-error on the primary partition occurred or because an uncorrectable write-error occurred or because the catch-up copy server may have been logged out. Look at the output on the supervisor terminal to see if there was an uncorrectable write error on the secondary partition. If so, run FIX_DISK with the -ADD_BADSPOT option on the secondary partition to add that badspot. Then use the MIRROR_ON command to start the mirror and the copy server. If there was no uncorrectable error, assume the copy server was logged out and use the MIRROR_ON command to start it.

Cannot shut down primary device *pdev1* - catch-up copy
is in progress to bring secondary device *pdev2*
up to date. (mirror_off)

You cannot shut down the primary mirrored partition before the catch-up copy to the secondary partition is finished because the secondary partition is not up-to-date and cannot be used until the catch-up copy is complete.

Cannot shut down primary device *pdev1* -
primary is only survivor. (mirror_off)

You can shut down the primary partition of the mirrored pair with the -SHUT_PRIMARY option only if the mirror is on or if the primary partition is damaged and the secondary partition is functioning.

Cannot shut down secondary device *pdev2* -
secondary is only survivor. (mirror_off)

You can shut down the secondary partition of the mirrored pair with the -SHUT_SECONDARY option only if the mirror is on or if the secondary partition is damaged and the primary partition is functioning.

Cannot spawn Copy Server for Ldev *ldev* - no right. (START_COPYSR)

The copy server phantom can be started only from the supervisor terminal by the User 1 process.

Cannot spawn Copy Server for Ldev *ldev*. Unable to attach to *dirname*.
(START_COPYSR)

In attempting to make a catch-up copy for logical device *ldev*, the copy server cannot be started because an attach to the directory *dirname* (usually SYSTEM) was not possible. This may be because the partition or the directory does not exist or because user SYSTEM does not have proper access rights to the directory. Check the ACLs on *dirname*; SYSTEM should have ALL access.

Cannot spawn Copy Server for Ldev *ldev*, status *code* (START_COPYSR)

The copy server cannot be started to make a catch-up copy for logical device *ldev*. The status *code* returned is one of the following: *e\$nr it* (Insufficient access rights.); *e\$bver* (Incorrect version number.); *e\$bpar* (Bad parameter.); *e\$exst* (Already exists.); or *e\$nval* (Validation error.) Take appropriate action based on the status.

Catch-up copy in progress. Copy Server must be logged out before the mirror is turned off. (mirror_off)

A catch-up copy must not be in progress on the pair of partitions for which mirroring is being turned off with the MIRROR_OFF command using either the -SHUT_BOTH or the -SHUT_SECONDARY option. Ordinarily, the MIRROR_OFF command logs out the copy server and you do not see this message. If it is necessary for you to stop the catch-up copy facility, first log out the copy server process (copy_server). Then use the MIRROR_OFF command to stop the mirroring process.

COMDVM must follow COMDEV directive.

COMDVM directive ignored.

The COMDVM directive must follow the COMDEV directive in the configuration file. If you want the command device to be mirrored at cold start, change the configuration file so that it is correct.

Copy Server for Ldev *ldev* aborted by raising of hardware condition at *pointer*
Fault type was *nn*

The copy server aborted due to hardware errors.

Copy Server for Ldev *ldev* completed catch-up copy.

The copy server successfully updated the two mirrored partitions, the primary being logical device *ldev*, so that they are identical.

Copy Server for Ldev *ldev* logging out due to forced logout.

The copy server has been forcibly logged out. To restart the copy server, first use the MIRROR_OFF command and then use the MIRROR_ON command.

Copy Server for Ldev *ldev* logging out due to error or shutdown.

The copy process failed in attempting to update the mirrored pair of partitions, the primary being logical device *ldev*, due either to a disk read-error or write-error or to a shutdown by the MIRROR_OFF command. Use the MIRROR_OFF command and run FIX_DISK on the mirrored partitions. Then you can restart the mirror with the MIRROR_ON command. This will also cause the catch-up copy to start. (See the section Catch-up Copy Failure in Chapter 9.)

Copy Server for Ldev *ldev* logging out - mirror was inactive.

A catch-up copy for mirroring logical device *ldev* started but the mirroring process aborted. This is an unlikely occurrence.

Copy Server for Ldev *ldev* starting catch-up copy.

The catch-up copy for the mirror of logical device *ldev* is starting.

Could not CLOSE PDEV *pdev1* due to controller or device errors. (mirror_on)

Could not CLOSE PDEV *pdev2* due to controller or device errors. (mirror_on)

Could not OPEN PDEV *pdev2* due to controller or device errors. (mirror_on)

A controller or other hardware error occurred when you attempted to start the mirroring process. Check to be sure that the disk drive is powered on, that it is on line, and that it is not write-protected. Similar messages may also appear when you use ADDISK.

If you attempt a warm start on your system and it is successful, you may encounter an error subsequent to the warm start when adding or mirroring disks that are associated with an intelligent disk controller. In this case, you may see the last message above or a similar message from the ADDISK command. If this message does appear, wait until the following message appears at the supervisor terminal.

DLL and init ICOP complete (cntrlr_address) - (disk_init)

The intelligent disk controller is now downloaded again and you can now add or mirror the disks that you previously attempted to add or mirror.

This problem occurs only on disk drives associated with intelligent disk controllers and only on those disk drives where all the logical disks (partitions) on a physical disk were shut down prior to the warm start.

Could not shut down secondary device *pdev2*.

You answered something other than Y, YE, or YES at the prompt asking if you want the catch-up copy to continue (see the first message in this section), causing PRIMOS to attempt a shutdown of the secondary partition. PRIMOS encountered an error and the secondary partition is now not accessible. Use the MIRROR_OFF command with the -SHUT_SECONDARY option. Then run FIX_DISK on the secondary partition.

Device *pdev* is not mirrored. (mirror_off)

The *pdev* specified with the MIRROR_OFF command is not part of a mirrored pair. Check the *pdev* for the partition you want to shut down or remove from mirroring.

*** Disk *pdev* - A dynamic badspotting partition cannot be started on this controller.

Run FIX_DISK to convert to an -All_Controller format partition.

Both the primary partition and the secondary partition must be associated with disk controllers that can perform dynamic badspot handling. You have attempted to start mirroring with a Dynamic Badspot Handling (-IC) mode partition that is on a nonintelligent disk controller. Run FIX_DISK with the -ALL_CONTROLLER option on this partition. This message is also displayed if you attempt to add a Dynamic Badspot Handling (-IC) mode partition on a disk drive associated with a nonintelligent disk controller.

Duplicate *pdev pdev* is ignored by COMDVM directive.

Duplicate PDEV *pdev* is ignored by PAGINM directive.

These messages are displayed at cold start. The same *pdev* is entered more than once as an argument to either the COMDVM directive, the PAGINM directive, or another directive in the configuration file. Mirroring will not take place on either the command device or the paging device indicated. Check the configuration file to see what you intended.

Force option can only be used with Shut_Both option. (mirror_off)

You can use the -FORCE option only with MIRROR_OFF -SHUT_BOTH.

I/O still pending on device *pdev*. Unable to shut down disk.

Please retry. (mirror_off)

Read requests or write requests are pending either on the primary partition five seconds after you issued the -SHUT_PRIMARY option or on the secondary partition five seconds after you issued the -SHUT_SECONDARY option. Reissue the command option.

Invalid COMDVM argument *pdev* is ignored by COMDVM directive.

Invalid PAGINM argument *pdev* is ignored by PAGINM directive.

These messages are displayed at cold start. A *pdev* argument for the COMDVM directive to mirror the command device or for the PAGINM directive to mirror a paging device is an invalid *pdev* and is ignored. If you want to mirror the command device or a paging device at startup, correct the *pdev* in the configuration file.

Maximum number of mirrored partitions already in use. (mirror_on)

No more than 128 partitions (64 pairs) may be set up as mirrored pairs. You cannot start another mirror with the MIRROR_ON command if there are already 64 pairs of mirrored partitions.

The mirror of primary device *pdev1* and secondary device *pdev2* will be broken.

Are you sure you want to continue?

This message applies when you specify the -SHUT_PRIMARY or the -SHUT_SECONDARY option of the MIRROR_OFF command. Answer YES if that is what you intend. If you answer NO, nothing is done and you are returned to PRIMOS command level. If you want to turn the mirror off and want the two partitions to have identical shutdown stamps, use the -SHUT_BOTH option.

Note

While PRIMOS is awaiting your response, users cannot login and the system may appear to be hung. When you use the MIRROR_ON command, you should promptly answer the above prompt.

Mirrored disks out of sync. (mirror_off)

This is a warning. Even though both partitions of the mirrored pair are being shut down, the partitions are known not to be identical because the mirror is already broken or the catch-up copy is not complete. Their shutdown stamps are not updated. Use the MIRROR_ON command to start a catch-up copy and make the two partitions identical.

Mirroring is not enabled. (mirror_on)

Mirroring is not enabled. (mirror_off)

The MIRROR_ON command is valid only if one or more of the MIRROR, COMDVM, or PAGINM directives was in the configuration file at cold start. If mirroring is not enabled, the MIRROR_OFF command is ineffective.

Must specify a Shut option. (mirror_off)

One of the shut options (-SHUT_BOTH, -SHUT_PRIMARY, or -SHUT_SECONDARY) must be specified with the MIRROR_OFF command. Reenter the command with the appropriate option.

Must specify two devices. (mirror_off)

Must specify two devices. (mirror_on)

You must specify two pdevs in the MIRROR commands to start or stop the mirroring process. Reenter the command with the correct pdevs in the proper order.

Must use Mirror_OFF to shut down mirrored disk *pdev*. (shutdn)

You used the SHUTDN pdev command in an attempt to shut down a partition that is part of a mirrored pair of partitions. You must use the MIRROR_OFF command with the -SHUT_BOTH option. However, if you use the SHUTDN ALL command, both partitions of the mirrored pair are shut down but you are not warned if the partitions are not identical.

Only one Shut option is allowed. (mirror_off)

You must specify only one of the shut options (-SHUT_BOTH, -SHUT_PRIMARY, or -SHUT_SECONDARY). You may not specify more than one.

PAGINM args must correspond 1-1 to PAGING args.

PAGINM directive ignored.

This message is displayed at cold start. The PAGINM and the PAGING directives in the configuration file must both have the same number of *pdev* arguments. If you want the paging devices to be mirrored at cold start, change the configuration file so that it is correct.

PAGINM must follow PAGING directive.

PAGINM directive ignored.

This message is displayed at cold start. The PAGINM directive must follow the PAGING directive in the configuration file. If you want the paging devices to be mirrored at cold start, change the configuration file so that it is correct.

PDEV *pdev* conflicts with assigned or paging or mirrored device. (addisk)

PDEV *pdev* conflicts with existing LDEV *ldev*. (addisk)

Using the ADDISK command, you have attempted to start up a partition that may already be part of a mirrored pair of partitions and thus may already be started.

"*pdev*" is not a valid PDEV. (mirror_on)

"*pdev*" is not a valid PDEV. (mirror_off)

The *pdev* you specified in the MIRROR_ON command or the MIRROR_OFF command is not a valid pdev. Check the pdevs of the partitions that you used.

PDEVs are identical. (mirror_on)

The pdevs you specified in the MIRROR_ON command are identical and indicate the same partition. Although the basic pdevs of a mirrored pair must be the same, the pdevs used in the command line must be adjusted for disk drive unit number and disk controller address and thus will be different. Check the pdevs you used.

PDEVs are not on same position on disk. (mirror_on)

Both partitions of the mirrored pair must have the same number of surfaces and the same starting surface number such that the pdevs are similar and differ only by the disk drive unit number and possibly by the controller number.

Primary *pdev1* and secondary *pdev2* are not on same disk model. (mirror_on)

Both partitions of the mirrored pair must be on physical disks of the same model, or type.

Primary device *pdev1* and secondary device *pdev2*
are not a mirrored pair. (mirror_off)

The partitions you specified by the pdevs in the MIRROR_OFF command are not presently mirrored. Check the pdevs you used and reenter the command with the correct pdevs.

Primary device *pdev1* cannot be added. (mirror_on)

You may be trying to add too many partitions to the system; there can be only 237₁₀ (355₈) partitions added to a system. Use STATUS DISKS to see how many are on your system. If there is some other reason for this message, a previous message should indicate the reason.

Primary device *pdev* conflicts with assigned device. (mirror_on).

The primary partition that you are attempting to mirror is an assigned partition and therefore cannot be started with the MIRROR_ON command. Unassign the partition and then start the mirror.

Primary device *pdev1* disk Rev. or mode is not mirrorable. (mirror_on)

pdev1 cannot be mirrored either because it is not a Rev. 22.0 or a Rev. 21.0 partition or it is not in Dynamic Badspot Handling (-IC) mode and dynamic badspot handling cannot thus be enabled.

Primary device *pdev1* is not a file system or paging device. (mirror_off)]

The primary partition, *pdev1*, is neither an added, or started, partition nor is it a paging partition that is being mirrored. Use the STATUS DISKS command at the supervisor terminal to check that the *pdev* you used is on the system as either an added partition or a paging partition.

Primary device *pdev1* is on a controller that does not support mirroring.

(mirror_on)

Secondary device *pdev2* is on a controller that does not support mirroring.

(mirror_on)

In order to be mirrored, a partition must be Rev. 21.0, or subsequent, on a disk drive associated with an intelligent disk controller, and have dynamic badspot handling enabled. Check that the controller for the partition you have specified as *pdev1* or *pdev2* is an intelligent disk controller (IDC) and that you specified a correct *pdev* that itself specified the correct controller address.

Secondary device *pdev2* disk Rev. or mode not mirrorable. (mirror_on)

pdev2 cannot be mirrored either because it is not a Rev. 22.0 partition or it is not in Dynamic Badspot Handling mode and dynamic badspot handling cannot thus be enabled.

Secondary device *pdev2* is in use. (mirror_on)

The partition you have designated as the secondary partition of a mirrored pair is already either added, assigned, or being used for paging. Specify another partition.

Secondary device *pdev2* is not a file system or paging device. (mirror_off)

The secondary partition, *pdev2*, is not an added, or started, partition and is not a paging partition. Use the STATUS DISKS command at the supervisor terminal to check that the *pdev* you used is on the system as either an added partition or a paging partition that is mirrored.

Secondary device *pdev1* is on a controller that does not support mirroring.

(mirror_on)

In order to be mirrored, a partition must be on a disk drive associated with an intelligent disk controller and have dynamic badspot handling enabled. Check the controller for the partition you have specified as *pdev2*.

Server for Ldev *ldev* aborted by raising of *v* condition at *p*

Copy Server for Ldev *ldev* aborted by raising of hardware condition at *p*

Fault type was *nn*

The copy server aborted due to hardware errors.

Starting Copy Server for Ldev *ldev*.

The catch-up copy for a mirrored pair of partitions that is not identical is starting.

System Error: Unable to purge locate. (mir_on_cmd\$)

PRIMOS has failed a consistency check. You should shut the system down, take a tape dump, and reboot. Then run FIX_DISK on all disks.

This is not the copy servers user type.

A process other than the copy server attempted to do the catch-up copy. This is an unlikely error but means that someone has written a routine to access the copy server process.

Unstartable PDEV *pdev* is ignored by PAGINM directive.

This message is displayed at cold start. The partition *pdev* specified with the PAGINM directive in the configuration file cannot be started. Check that the *pdev* in the configuration file is a valid *pdev*.

WARNING: The primary device may have broken the mirror previously.
A catch-up copy may overwrite valid data on the secondary device!

A catch-up copy of primary device *pdev1* to secondary device *pdev2*
will be started.

Are you sure you want to continue?

You either used the MIRROR_ON command or the COMDVM configuration directive in the PRIMOS.COMI file in attempting to start the mirroring process. The mirror of primary partition *pdev1* and secondary partition *pdev2* was broken and the primary partition was improperly shut down. The data on the secondary partition is likely more current. Thus, the secondary partition should be the primary partition. If you wish to change the positions of the *pdevs* in the command or in the configuration directive, answer NO to this prompt.

D

DISK ERRORS

Appendix D describes the meaning of disk-error messages sent to the supervisor terminal. In the messages, all record address, device, and status numbers are in octal. The format and meaning of these messages depend on the type of disk controller that generated the error condition.

There are two general classes of disk controllers for disks supported by Prime (see Table 3-1 for a list of these disks).

- Nonintelligent disk controllers
- Intelligent disk controllers

Intelligent disk controllers can operate in either of two modes, Nondynamic Badspot Handling (-AC) mode or Dynamic Badspot Handling (-IC) mode.

Note

Intelligent disk controllers are programmable and are downline loaded at cold start with Intelligent Channel Order Protocol (ICOP) software. This software enables the intelligent disk controller to operate either in Dynamic Badspot Handling (-IC) mode or in Nondynamic Badspot Handling (-AC) mode. In either mode, the controller acts as an intelligent disk controller. If the controller is not downline loaded with ICOP software, it functions as a nonintelligent disk controller.

NONINTELLIGENT DISK CONTROLLERS

The sections that follow describe error detection and correction on Storage Module Disks (SMDs), Cartridge Module Devices (CMDs), and Fixed-Media Disks (FMDs) associated either with nonintelligent disk controllers or with intelligent disk controllers that have not been downloaded with ICOP software and thus function as nonintelligent disk controllers.

Error Message Format

Under PRIMOS, disk read-errors and disk write-errors on supported disks generate error messages with the following format. The parts of the error message are explained below.

```
*** Message from product LOG DISK Rev 1. generated by USER on SYSTEM
    (Severity information. logged at day month year time)
DISK XXXX ERROR
DEVICE NUMBER = pdev (OCT) (ctrlr CTRLR dd UNIT dd)
CRA = 0000000000 (OCT) RCRA = 0000000000 (OCT)
CYLINDER = dd HEAD = dd RECORD = dd (dd SECTORS PER TRACK)
STATUS (ACCUM) = 000000 (OCT) STATUS (LAST) = 000000 (OCT) RETRIES = dd (err)
```

XXXX	READ for a read error and WRITE for a write error.										
<i>dd</i>	A decimal number.										
000000	An octal number.										
<i>pdev</i>	The octal physical device number of the disk, or partition, on which the error occurred.										
<i>ctrlr</i>	A number designating the disk controller. These numbers are as follows: <table> <tr> <td>4004</td><td>Nonintelligent disk controller</td></tr> <tr> <td>4005</td><td>Nonintelligent disk controller</td></tr> <tr> <td>2047</td><td>2250 CPU disk controller</td></tr> <tr> <td>2382</td><td>2455-type CPU disk/tape controller</td></tr> <tr> <td>10019</td><td>Intelligent disk controller (IDC1)</td></tr> </table>	4004	Nonintelligent disk controller	4005	Nonintelligent disk controller	2047	2250 CPU disk controller	2382	2455-type CPU disk/tape controller	10019	Intelligent disk controller (IDC1)
4004	Nonintelligent disk controller										
4005	Nonintelligent disk controller										
2047	2250 CPU disk controller										
2382	2455-type CPU disk/tape controller										
10019	Intelligent disk controller (IDC1)										
CRA	The desired record number within the partition (one 32-bit word).										
RCRA	The actual record number read (one 32-bit word). This number is displayed only if the CRA does not equal RCRA. On read operations RCRA should match the CRA. On write operations RCRA has no validity.										
STATUS	The reason for the error. Status is indicated by the sum of two or more 16-bit halfwords provided in Table D-1. (<i>ACCUM</i>) is the accumulated bit-wise OR of all the status words obtained at each retry and (<i>LAST</i>) is the last status word obtained.										
RETRIES	The number of times the read or write operation was attempted unsuccessfully before the error message was displayed. Maximum is 10.										
<i>err</i>	The disposition of the error, UNCORRECTED or RECOVERED. If the error is RECOVERED, PRIMOS displays CORRECTED WORD # = 000000 where the octal number is the record address.										

TABLE D-1. *Nonintelligent Controller Status Words*

<i>Status Word</i>	<i>Meaning</i>
177777	Bad record identifier
177776	Device not ready
177773	Controller hung or warm start occurred
177772	Seek failure
100000	Always set
040000	DMX overrun
020000	Write-protect violation
010000	Check error, uncorrectable read on controller 2382
004000	Checksum error
002000	Header check failure
000100	Error encountered on last read but retries performed and data recovered
000010	Disk drive seeking
000004	Illegal seek
000002	Select error
000001	Not available or not ready

Error Correction

The nonintelligent disk controller writes a 32-bit fullword correcting code checksum on each record. An error detection and correction scheme using an Error Correction Code (ECC) is implemented for supported disks.

The code, together with the correction logic in the controller, is capable of detecting any of the following:

- Two error bursts whose combined length does not exceed 22 bits
- A single error burst whose length does not exceed 32 bits
- Any odd number of errors

In addition, the ECC algorithm is capable of correcting any single error burst whose length does not exceed 11 bits. Error correction is attempted only after ten attempts to read a record have failed.

The message printed in the *err* field when error correction was attempted and failed is
UNCORRECTED.

The message printed in the *err* field when error correction succeeds is
RECOVERED

An example of a disk error message follows.

Operator's Guide to File System Maintenance

```
*** Message from product LOG DISK Rev 1. generated by GEORGE on ENPUB2
(Severity information. logged at 29 NOV 87 14:27:08)
DISK WRITE ERROR
DEVICE NUMBER = 003460 (OCT) (4005 CTRLR 0 UNIT 0)
CRA = 00000010356 (OCT) RCRA = 00000000000 (OCT)
CYLINDER = 34 HEAD = 5 RECORD = 5 (9 SECTORS PER TRACK)
STATUS (ACCUM) = 120011 (OCT) STATUS (LAST) = 120011 (OCT) RETRIES = 10 (UNCORRECTED)
```

The error message is interpreted as follows: On physical device 3460, the STATUS (120011) indicates that there is a write-protect violation, the drive was seeking, and the drive is not available; the write-protect switch was on.

Note

If DSM is running on your system, DSM controls the display of all disk-error messages at the supervisor terminal through the DSM CONSOLE log selection. (See the *DSM User's Guide*.) The display is similar to the examples shown in this appendix.

INTELLIGENT DISK CONTROLLERS

The sections that follow describe error detection and correction on Storage Module Disks (SMDs), Cartridge Module Devices (CMDs), and Fixed-Media Disks (FMDs) associated with downloaded intelligent disk controllers in either Dynamic Badspot Handling (-IC) or Nondynamic Badspot Handling (-AC) mode.

Error Detection

Under PRIMOS, disk read-errors and disk write-errors on fully supported disks associated with intelligent disk controllers generate error messages with the following format. The parts of the error message are explained below.

```
*** Message from product LOG DISK Rev 1. generated by USER on SYSTEM
(Severity information. logged at day month year time)
DISK ERROR IN ICOP MODE: OPCODE = 000000 (OCT)
DEVICE NUMBER = pdev (OCT) (ctrlr CTRLR dd UNIT dd)
CRA = 00000000000 (OCT) RCRA = 00000000000 (OCT)
CYLINDER = dd HEAD = dd RECORD = dd (dd SECTORS PER TRACK)
LSW = 000000 000000 (OCT) PSW = 000000 000000 (OCT)
I/O TIME = 000000 (OCT)
(INTERRUPT TYPE = 000000 (OCT) REQUEST ID = 000000 (OCT))
```

dd A decimal number.

000000 An octal number.

OPCODE Either READ, WRITE, OPEN, CLOSE, FORMAT, WRITE VERIFY, ASYNC WRITE, or an octal number code for the current operation. See Table D-2 for a description of this code.

pdev The octal physical device number of the disk, or partition, on which the error occurred.

<i>ctrlr</i>	The model number of the disk controller. These numbers are as follows:
4004	Nonintelligent disk controller
4005	Nonintelligent disk controller
2047	2250 CPU disk controller
2382	2455-type CPU disk/tape controller
10019	Intelligent disk controller (IDC1)
CRA	The desired record number within the partition (one 32-bit word).
RCRA	The actual record number read (one 32-bit word). On read operations RCRA should match the CRA. On non-read operations RCRA has no validity.
CYLINDER HEAD RECORD SECTORS	The location on the disk of the bad record. CYLINDER is the track number, HEAD is the surface number, RECORD is the sector number, and SECTORS PER TRACK refers to the geometry of the physical disk, or the number of records on a track. These values are valid only for operations that are specific to a record. On format, only CYLINDER and HEAD are valid. On open and close operations, none of the values are valid.
LSW	Two 16-bit halfwords indicating the logical status of the controller or the reason for an error. The meanings of the first logical status halfword are shown in Table D-3 and the meanings of the second logical status halfword are shown in Table D-4. LSW 1 is a general description of the problem. You should always look at LSW 2 and the physical status halfwords for additional information.
PSW	Two 16-bit halfwords indicating the physical status of the controller or the reason for an error. The error may be indicated by the sum of the halfwords. The meanings of the first physical status halfword are shown in Table D-5 and the meanings of the second physical status halfword are shown in Table D-6.
I/O TIME	The elapsed time, in milliseconds, during which the read or write operation was attempted unsuccessfully before the error message was displayed.
INTERRUPT TYPE	An octal number indicating operation completed (4) or unrecoverable I/O error detected (5).
REQUEST ID	An octal number representing the address of a queue block and used by a representative from your Customer Support Center for debugging purposes.

The following example shows an ICOP disk-error message.

Operator's Guide to File System Maintenance

*** Message from product LOG DISK Rev 1. generated by WALKER on PRIME1
(Severity information. logged at 29 NOV 87 21:31:41
DISK ERROR IN ICOP MODE: OPCODE = 000014 (OCT)
DEVICE NUMBER = 006166(OCT) (10019 CTRLR 2 UNIT 3)
CRA = 00000023401 (OCT) RCRA = 00000000000 (OCT)
CYLINDER = 29 HEAD = 17 RECORD = 3 (14 SECTORS PER TRACK)
LSW = 101400 000004 (OCT) PSW = 000000 020021 (OCT)
I/O TIME = 000016 (OCT)
(INTERRUPT TYPE = 000005 (OCT) REQUEST ID = 150354 (OCT)

The OPCODE indicates that a write was attempted on pdev 6166 associated with intelligent disk controller 2 and disk drive 3. The address and location of the bad record are indicated. LSW1 indicates command completion was unsuccessful and LSW2 indicates four retries of the write were attempted. PSW2 indicates that the disk is write-protected, thus faulted and not ready. The last retry operation was attempted for 14 (16₈) milliseconds and an unrecoverable I/O error was detected.

TABLE D-2. Command Operation Code (OPCODE)

<i>OPCODE</i> (octal)	<i>Meaning</i>
000000	Response check for controller functionality
000001	Disk seeking
000005	Open spindle; logically activate a disk for use (OPEN)
000006	Close spindle; logically deactivate a disk (CLOSE)
000013	Read a record (READ)
000014	Write a record (WRITE)
000016	Write a record and verify the write (WRITE VERIFY)
000017	Format a disk (FORMAT)
000024	Add a badspot
000144	Asynchronously write a record (ASYNC WRITE)
000145	Entering ICOP mode

Note

The left two octal numbers (4 bits) in the above codes have no meaning and can be ignored.

*TABLE D-3. First Logical Status Halfword
LSW 1 Intelligent Controller*

<i>Logical Status Halfword 1</i>	<i>Meaning</i>
140000	Drive error, information in physical halfwords
110000	Badspot handling invoked, information in other status words
101400	Command completion unsuccessful
100400	Command completed successfully with multiple retries
100200	Illegal parameter passed with command block
100100	Protocol error
100040	Undefined OTA function code
100020	Disk open, operation requested must be done on closed disk
100010	Disk closed, has not been added or assigned
100004	Bad operation code or operation code length
000000	Command completed successfully

TABLE D-4. Second Logical Status Halfword
LSW 2 Intelligent Controller

Logical Status Halfword 2	Meaning
100000	Record has been remapped
040000	ICOP controller in DBS mode on ADDISK or ASSIGN
020000	Requested a remap of an already remapped record
010000	Disk number nonexistent or out-of-range
004000	Attempt to access RMA badspot
002000	Drive error attempting badspot I/O; record not remapped, DBS mode exited
001000	Data error attempting badspot I/O; record not remapped, DBS mode exited
000400	Error occurred reformatting record; bits 6 and 7 (1000 ₈ and 2000 ₈) denote type, DBS mode exited
000200	Error occurred accessing DBS record; bits 6 and 7 (1000 ₈ and 2000 ₈) denote type, DBS mode exited
000XXX	Retry count. On read operations, ranges to 104 ₈ (bits 14 - 16). 10 ₈ through 100 ₈ indicate that the controller is physically adjusting on the track in attempting the read, as follows:
000000	Normal read
000010	Servo plus
000020	Servo plus and data strobe early
000030	Servo plus and data strobe late
000040	Data strobe early
000050	Data strobe late
000060	Servo minus
000070	Servo minus and data strobe early
000100	Servo minus and data strobe late
	1 ₈ through 3 ₈ indicate number of retries. 4 ₈ indicates error correction (ECC) used. The controller sequences through the above nine methods and retries the read up to three times for each method.
	On write operations, 1 ₈ through 4 ₈ indicate number of retries.

*TABLE D-5. First Physical Status Halfword
PSW 1 Intelligent Controller*

<i>Physical Status Halfword 1</i>	<i>Meaning</i>
004000	Parity error on data transfer to or from CPU
000010	Cannot remap when disk is write-protected
000004	Unable to select disk on ADDISK or ASSIGN
000002	Encountered reformatted header (see Note)
000001	Data transfer timeout

Note

If the controller encounters reformatted headers, either the controller went out of DBS mode for this drive or the DBS file does not match reformatted headers on this disk. Look at previous errors in the system event log for LSW2 of 000400 and 000200 as the cause of the present error.

To correct these errors, try shutting down all partitions on the drive where the error occurs and then adding them back to the system. If the problem persists, back up all partitions on the drive and then run MAKE on all the partitions with the -FORMAT and -NEW_DISK options. These errors may also indicate that the physical disk is failing.

TABLE D-6. Second Physical Status Halfword
PSW 2 Intelligent Controller

<i>Physical Status Halfword 2</i>	<i>Meaning</i>
040000	Data transfer overrun
020000	Disk write-protected (see drive manual)
010000	Read check, see LSW 2
004000	Data parity error
002000	Header not found
001000	Underflow on read or write, controller memory bandwidth insufficient
000400	No servo clock detected (on ADDISK or ASSIGN)
000200	No response from disk, issued seek, seek not complete
000100	Dual port; other port has disk (see drive manual)
000040	Sequencer detected error
000020	Disk faulted (see drive manual)
000010	Disk seeking, not on cylinder (see drive manual)
000004	Seek error (see drive manual)
000002	Disk selection invalid (see drive manual)
000001	Disk not ready (see drive manual)

Error Correction

Intelligent disk controllers perform extensive error correction in addition to the error detection and correction scheme of the nonintelligent controller. Multiple attempts are made to read a record after the normal read process fails. If the attempted reads are successful, that is, the error is a recoverable read-error, the data is remapped after it is recovered if the controller is in -IC mode, and the physical record is marked as a badspot by the controller so that the file system can not use that record again. If the error is unrecoverable or the controller is in -AC mode, a message is displayed informing you of the error and the bad record address is included. You should then add that bad record to the badspot file by using `FIX_DISK`.

In the case of an uncorrectable disk write-error, the disk controller in -IC mode remaps the record to a good location on the disk and adds the bad location to the dynamic badspot file.

In addition to the error messages and status words, the controller itself has LEDs that indicate status. The LEDs are in two separate banks: the first bank contains a red and a green LED and the second bank contains four yellow LEDs arranged in the order shown below.

green red yellow-4 yellow-3 yellow-2 yellow-1

When lit, the LEDs have the following meanings:

green	ICOP functioning correctly
red	ICOP error, see yellow-2, -3, or -4 setting
yellow-4	Invalid interrupt occurred
yellow-3	Excessive ECC errors occurred
yellow-2	Communication with PRIMOS failed
yellow-1	ICOP run lamp; flashes two times per second when controller is downloaded

Controller Messages

The following messages are associated with the intelligent disk controller. All but one generally indicate serious problems associated with activating the intelligent disk controller. One message indicates successful operation of the controller.

Could not attach to `DOWN_LINE_LOAD*` - (disk_init)

An attempt to attach to `DOWN_LINE_LOAD*` failed. Check to be sure the top-level directory `DOWN_LINE_LOAD*` exists. The disk controller will operate as a nonintelligent disk controller.

Could not OPEN PDEV *pdev* due to controller or device errors. (assign_disk)

Could not OPEN PDEV *pdev* due to controller or device errors. (addisk)

The disk controller could not access disk *pdev*. Among the reasons are that the disk drive is off line or not ready, or a cable is loose or not connected. If those are not the reasons, the downline load file may not be the correct revision for the controller, or may be in error, or the controller may have malfunctioned.

Disk download file *filename* not found - (disk_init)

The downline load file *filename* was not found in the DOWN_LINE_LOAD* directory. The controller will operate as a nonintelligent disk controller. You should install the downline load file.

Disk I/O error (assign_disk)

An error is encountered during an attempt to read from disk. Check to be sure that the disk drive is on line and ready.

DLL and init ICOP complete *controller_address* - (disk_init)

Controller *controller_address* is successfully downline loaded and initialized as an intelligent controller with Intelligent Channel Order Protocol.

Failure to enter ICOP mode *controller_address* - (disk_init)

Disk controller *controller_address* failed to be initialized as an intelligent controller. This message indicates a hardware problem, for example, a download to controller revision mismatch or a controller memory problem. This message is entered in the system event log and the controller functions as a nonintelligent controller.

I/O Errors while processing a disk DLL file *filename* - (disk_init)

Errors were encountered during a read of the disk controller downline load file *filename*. The downline load file may be corrupted. The controller operates as a nonintelligent disk controller.

To determine the type and revision of your intelligent disk controller downline load file, resume the DLINFO program. For example:

```
OK, R DOWN_LINE_LOAD*>DLINFO
```

```
[DLINFO Rev. 22.0 Copyright (c) Prime Computer, Inc. 1987]
```

```
Usage: DLINFO dl_file
```

```
ERI R DOWN_LINE_LOAD*>DLINFO DOWN_LINE_LOAD*>IDC1.DL
```

```
[DLINFO Rev. 22.0 Copyright (c) Prime Computer, Inc. 1987]
```

```
DL File Type = 0
```

```
Project Name:
```

```
Project Identifier:
```

```
    hardware revision: 9
```

```
    software revision: 105
```

```
    Primos revision : 22.1
```

```
Date: THU, OCT 27 1988
```

```
Time: 13:12:37
```

```
Medium: DMT - with checksum
```

```
Max Packet Size = 2048 bytes
```

```
Total Program Size = 26062 bytes
```

```
Total Number of Packets = 50
```

```
OK,
```

MAKE_ROBUST MESSAGES

Special messages indicate that MAKE_ROBUST is unable to convert the partition according to the options specified on the command line or according to other input. This appendix lists the messages from MAKE_ROBUST in alphabetical order and presents a brief explanation of each message. If *The* or *This* is the first word in the message, the second word is used for alphabetizing. Variable names in messages are italicized, for example, *pdev*.

There are three groups of related messages among those listed here (Read and write errors, Errors associated with changes in the MFD, and Errors associated with MFD entry types). These three groups consist of a single explanation followed by a listing of the error messages.

For a discussion of MAKE_ROBUST and its command-line options, see Chapter 7. For a summary of the MAKE messages, see Appendix A and for a summary of the FIX_DISK messages, see Appendix B.

Command line errors

Bad parameter. Maximum extent size must be between 1 and 32767 (MAKE_ROBUST)

You specified a number for the maximum extent size outside of the range of 1 to 32767. Reenter the command line with the proper value.

Bad parameter. Maximum extent size must be larger than the minimum (MAKE_ROBUST)

Either you entered a value for minimum extent size that was larger than that of the maximum extent size or you did not enter a maximum extent size value. Reenter the values at the appropriate prompts.

Bad parameter. Minimum extent size must be between 1 and 32767 (MAKE_ROBUST)

You specified a number for the minimum extent size outside of the range of 1 to 32767. Reenter the command line with the proper value.

Bad parameter. Reading command line (MAKE_ROBUST)

MAKE_ROBUST encountered an unexpected character or token when parsing your command line. Reenter the command line.

Bad parameter. You must supply a pdev. (MAKE_ROBUST)

You must specify the partition that you want to convert by entering the physical device number (pdev) of the partition either with the -DISKS option or without it.

Bad parameter. You must supply only one pdev (MAKE_ROBUST)

You can convert only one partition at a time. Reenter the command line with only one pdev.

Read and write errors

In the following group of messages, MAKE_ROBUST encountered an error in attempting to read or write a system directory or file. *error_code* is a standard PRIMOS error message such as No room, Device not assigned., Bad number of words., or Bad parameter. Check that the disk is running, the partition is assigned to you, and the disk is not write-protected.

error_code. Reading CMDNCO. (MAKE_ROBUST)

error_code. Reading DOS. (MAKE_ROBUST)

error_code. Reading the CMDNCO directory (MAKE_ROBUST)

error_code. Reading the disk rat (MAKE_ROBUST)

error_code. Reading the DOS directory (MAKE_ROBUST)

error_code. Reading the MFD (MAKE_ROBUST)

error_code. Rewriting the CMDNCO directory (MAKE_ROBUST)

error_code. Rewriting the DOS directory (MAKE_ROBUST)

error_code. Rewriting the MFD record. (MAKE_ROBUST)

error_code. Updating the disk rat. (MAKE_ROBUST)

Partition errors

The partition must be a rev 22.1 standard partition. (MAKE_ROBUST)

The partition you want to convert to a robust partition must be created with Rev. 22.1 MAKE prior to conversion.

The partition is already robust! (MAKE_ROBUST)

This partition was converted previously. MAKE_ROBUST exits.

Errors associated with changes in MFD

In the following group of messages, the partition you want to be converted to a robust partition must be created with Rev. 22.1 MAKE and must not be altered or added to in any way. Use MAKE to recreate the partition and then do the conversion with MAKE_ROBUST. Since you have already created the partition once and it has a badspot file, you can use the -BADLEV 0 option, but be sure that you do not use the -NEWDSK or -FORMAT options.

Sorry. A vacant entry was found in the MFD record.
Partition can't be converted - It has been used! (MAKE_ROBUST)

Sorry. An access category was found.
Partition can't be converted - It has been used! (MAKE_ROBUST)

Sorry. An unexpected non-special file was found: *filename*
Partition can't be converted - It has been used! (MAKE_ROBUST)

Sorry. CMDNCO directory has more than 1 record.
Partition can't be converted - It has been used! (MAKE_ROBUST)

Sorry. CMDNCO must be an ACL directory.
Partition can't be converted - It has been used! (MAKE_ROBUST)

Sorry. CMDNCO not found.
Partition can't be converted - It has been used! (MAKE_ROBUST)

Sorry. DOS directory has more than 1 record.
Partition can't be converted - It has been used! (MAKE_ROBUST)

Sorry. DOS must be an ACL directory.
Partition can't be converted - It has been used! (MAKE_ROBUST)

Sorry. DOS not found.
Partition can't be converted - It has been used! (MAKE_ROBUST)

Sorry. MFD has more than 1 record.
Partition can't be converted - It has been used! (MAKE_ROBUST)

Sorry. Non-special BADSPT found.
Partition can't be converted - It has been used! (MAKE_ROBUST)

Sorry. Non-special BOOT found.
Partition can't be converted - It has been used! (MAKE_ROBUST)

Sorry. Non-special DYNBSP found.
Partition can't be converted - It has been used! (MAKE_ROBUST)

Sorry. The MFD acl has been modified.
Partition can't be converted - It has been used! (MAKE_ROBUST)

Errors associated with MFD entry types

For the following group of messages, in checking the directory headers in the MFD, MAKE_ROBUST found an entry that should not be there or that is not the proper type or did not find an entry that should exist. Use MAKE to recreate the partition and then do the conversion with MAKE_ROBUST. Since you have already created the partition once and it has a badspot file, you can use the -BADLEV 0 option, but be sure that you do not use the -NEWDSK or -FORMAT options.

ERROR: Another directory header found in MFD record.
I can't make sense of this partition! (MAKE_ROBUST)

ERROR: BOOT file not found.
I can't make sense of this partition! (MAKE_ROBUST)

ERROR: Disk rat not found.
I can't make sense of this partition! (MAKE_ROBUST)

ERROR: First entry in MFD record is not a directory header.
I can't make sense of this partition! (MAKE_ROBUST)

ERROR: MFD acl not found.
I can't make sense of this partition! (MAKE_ROBUST)

ERROR: MFD directory index block not found.
I can't make sense of this partition! (MAKE_ROBUST)

ERROR: MFD entry BRA does not point to MFD record.
I can't make sense of this partition! (MAKE_ROBUST)

ERROR: MFD entry not found.
I can't make sense of this partition! (MAKE_ROBUST)

ERROR: Two BADSPT files found.
I can't make sense of this partition! (MAKE_ROBUST)

ERROR: Two BOOT files found.
I can't make sense of this partition! (MAKE_ROBUST)

ERROR: Two CMDNCO files found.
I can't make sense of this partition! (MAKE_ROBUST)

ERROR: Two directory index blocks were found in the MFD record.
I can't make sense of this partition! (MAKE_ROBUST)

ERROR: Two disk rat files found.
I can't make sense of this partition! (MAKE_ROBUST)

ERROR: Two DOS files found.
I can't make sense of this partition! (MAKE_ROBUST)

ERROR: Two DYNBSP files found.
I can't make sense of this partition! (MAKE_ROBUST)

ERROR: Two MFD acl entries found.
I can't make sense of this partition! (MAKE_ROBUST)

ERROR: Two MFD files found.
I can't make sense of this partition! (MAKE_ROBUST)

ERROR: Unexpected end of MFD record found.
I can't make sense of this partition! (MAKE_ROBUST)

ERROR: Unknown entry found in MFD record.
I can't make sense of this partition! (MAKE_ROBUST)

SUMMARY OF COMMAND SYNTAX FOR MAKE AND FIX_DISK

This appendix summarizes the MAKE and FIX_DISK command-line options and is provided for your convenience. For more detailed descriptions, see the text in Chapters 5 and 6. See Chapter 4 for a description of what to do before and after running MAKE and FIX_DISK.

SUMMARY OF MAKE COMMAND SYNTAX

When you boot MAKE from disk or from magnetic tape or when you use MAKE at the PRIMOS command level, the following are the available options you use with MAKE. When you use MAKE at PRIMOS command level, enter the options on this command line:

MAKE -DISK *pdev* [*options*]

For a full explanation of the use of these options and for cautions on their use, see the full text in Chapter 5.

<i>MAKE Option</i>	<i>Action</i>
-DISK [<i>pdev</i>]	Specifies <i>pdev</i> of the partition you are creating. -DISK must be used and must be the first option on the command line. (Not needed with -HELP.) If you do not include <i>pdev</i> , MAKE prompts for it.
-DSK	
-ALL_CONTROLLER	Specifies that the partition should be created to be compatible with any type of disk controller (nonintelligent controller and intelligent controller). The partition is not capable of dynamic badspot handling or mirroring. Use the -AC option when running MAKE standalone or MAKE defaults to -AC.
-AC	

*MAKE Option**Action*

- BADSPOT_LEVEL [n]** Specifies the level of badspot checking that you wish to take place. *n* ranges from 0 through 4. The default for SMDs is 1, for 2450 system FMDs is 2, and for other FMDs is 4. If you do not specify *n* with this option, MAKE prompts for it.
- BADLEV**
- BAUD_RATE [baud]** Specifies the supervisor terminal baud rate if it is to be different from the default baud rate of 300 bps. If you use this option but do not include *baud* on the command line, MAKE prompts for a valid baud rate.
- BAUD**
- COPY_BADSPOTS_BY_DEVICE [pdev]** Specifies the *pdev* of the partition containing the badspot file, BADSPT, that MAKE is to copy. If you do not include *pdev* with this option, MAKE prompts for it.
- CPYDEV**
- COPY_BADSPOTS_BY_NAME [diskname]** Specifies the name of the partition containing the badspot file, BADSPT, that MAKE is to copy. If you do not include *diskname* with this option, MAKE prompts for it.
- CPYNAM**
- DISK_REVISION [rev]** Specifies the revision of the partition that you want MAKE to create. *rev* must be 18, 19, 20, 21, 22, or 22.1. If you do not include *rev* or if you include an invalid *rev*, MAKE prompts for a valid *rev*. If you do not include this option, MAKE creates a Rev. 22.1 partition.
- DSKREV**
- DISK_TYPE [type]** Specifies the particular type of disk to be processed by MAKE. This is a required option, and if you do not specify **-DISK_TYPE** or if you do not include *type*, MAKE lists the valid types and prompts for *type*. Use the correct *type*. Designations and definitions of available *type* are as follows:
- DT**

<i>Type</i>	<i>Meaning</i>
SMD	80 or 300 MB removable
CMD	Cartridge Module Device
68MB	68 MB Fixed Media
158MB	158 MB Fixed Media
160MB	160 MB Fixed Media
600MB	600 MB Fixed Media
MODEL_4475	300 MB Fixed Media
MODEL_4711	60 MB Fixed Media
MODEL_4714	84 MB Fixed Media
MODEL_4715	120 MB Fixed Media
MODEL_4719	258 MB Fixed Media
MODEL_4721	328 MB Fixed Media
MODEL_4735	496 MB Fixed Media
MODEL_4845	770 MB Fixed Media
MODEL_4860	817 MB Fixed Media

*MAKE Option**Action***-FORMAT****-FMT**

Specifies that the disk has never been used on a Prime system and MAKE should initialize the data patterns on the partition. If you want to create pre-Rev. 21.0 partitions on a disk that has previously been partitioned as a Rev. 21.0 or later disk, use the -FORMAT option to remodify the record headers. (See Chapter 5 for details.)

WARNING

Do not use the -FORMAT option on an existing first partition of a physical disk having other Rev. 22.0 or Rev. 21.0 partitions on it.

-HELP

Specifies that you want a help display similar to this listing of MAKE options and arguments.

-INIT

Specifies that MAKE should initialize the file system portion of the partition. Recommended that you do not use this option to save processing time. -NO__INIT is the default.

-INTELLIGENT_CONTROLLER**-IC**

Specifies that a partition should be created to be compatible with intelligent disk controllers capable of dynamic badspot handling and mirroring. Create the first partition of the disk first if you are creating Rev. 22.0 or Rev. 21.0 partitions. You cannot use the -IC option when running MAKE standalone.

-MAP_UNCORR**-UNCORR**

Specifies not to consider records with correctable errors as badspots.

-MAX_EXTENT_SIZE [size]**-MAXSIZ**

Specifies the maximum extent *size* for CAM files at Rev. 22.0 and later. If you do not specify the maximum extent size, MAKE defaults to 32 records for standard partitions.

-MIN_EXTENT_SIZE [size]**-MINSIZ**

Specifies the minimum extent size for CAM files at Rev. 22.0 and later. If you do not specify the minimum extent size, MAKE defaults to 16 records for standard partitions.

-NEW_DISK**-NEWDSK**

Specifies that this is the first time the partition is being created and that MAKE should not look for an existing badspot file but should create a new badspot file if necessary.

-NO_INIT

Specifies that MAKE is not to initialize the file system portion of the partition. This is the default.

<i>MAKE Option</i>	<i>Action</i>
-NO_FLAW_MAP -NOFLMP	Specifies that MAKE should not process flaw maps but should use the default level of badspot checking for the disk type involved. Using the flaw map is the default. (See Table 5-4.)
-OVERRIDE_DEFAULT_INTERLEAVE -ODI	Specifies the method of file record allocation to be forward with an interleave factor of 3. If you do not specify -ODI or -RDI, MAKE sets the method based on the CPU and disk controller in your system. If you run MAKE standalone, MAKE prompts for the method of file record allocation.
-PARTITION [name] -PART	Specifies a valid six-character <i>name</i> for the partition you are creating. If you do not include this option on the command line, MAKE prompts for a name.
-QUERY_BADSPOTS -QBADS	Specifies that you want to enter known badspots manually, either from a list or from a hard-copy flaw map provided by the manufacturer.
-ROBUST -ROB	If you use the -ROBUST option, MAKE informs you that this option is not available and refers you to this book.
-RESTORE_DEFAULT_INTERLEAVE -RDI	Specifies the method of file record allocation to be reverse with an interleave factor of 1. If you do not specify -ODI or -RDI, MAKE sets the method based on the CPU and disk controller in your system. If you run MAKE standalone, MAKE prompts for the method of file record allocation.
-SPLIT [recs]	Specifies that the disk is to be a split partition to be used for paging, with <i>recs</i> records used as paging records and the rest of the records used for file system data. If you omit <i>recs</i> , MAKE displays the number of available records and prompts for the number of paging records. That number should be divisible by 16.

SUMMARY OF FIX_DISK COMMAND SYNTAX

When you use FIX_DISK at PRIMOS command level or when you resume FIX_DISK.SAVE from magnetic tape, the following are the available options you use. When you use FIX_DISK at PRIMOS command level, enter the options on this command line:

FIX_DISK -DISK *pdev* [*options*]

For a full explanation of the use of these options and for cautions on their use, see the full text in Chapter 6.

<i>FIX_DISK Option</i>	<i>Action</i>
------------------------	---------------

-DISK <i>pdev</i>	Specifies the <i>pdev</i> of the partition you are repairing. -DISK must be used and must be the first option on the command line. (Not needed with -HELP.) <i>pdev</i> must be included with -DISK.
--------------------------	--

-ADD_BADSPOT <i>rec_num-1</i> [. . . <i>rec_num-16</i>]	
--	--

-ADBADS	Specifies as many as 16 record numbers to be added to the badspot file. FIX_DISK attempts to map the bad records to new locations. Record numbers (<i>rec_num-1</i> ... <i>rec_num-16</i>) are specified in octal, offset from the beginning of the partition.
----------------	--

-ALL_CONTROLLER -AC	Selects Nondynamic Badspot Handling mode for this Rev. 22.0 partition. Either -AC or -IC is recommended when converting to Rev. 21.0 with the -CONVERT_21 option. You must use the -FIX option with -AC.
--------------------------------	--

-AUTO_TRUNCATION -AT	Truncates or deletes directories nested too deeply in a directory tree. If you omit this option, FIX_DISK aborts if the maximum depth is reached. Maximum depth is set with the -MAX_NESTED_LEVEL option (default is 99). You must use the -FIX option with -AT.
---------------------------------	--

-CHECK	Checks to see whether a partition was shut down properly or whether PRIMOS II wrote on the partition. If the partition was not shut down properly or if PRIMOS II wrote on the partition, FIX_DISK should be run with the -FIX option, and you may wish to include the -FAST option if the partition is robust.
---------------	---

-COMMAND_DEVICE [<i>pdev</i>]	
--------------------------------------	--

-COMDEV	Indicates that FIX_DISK is to operate on the command device; this option shuts the command device down, assigns it, repairs it, unassigns it, and starts it up again.
----------------	---

FIX_DISK Option

Action

-COMDEV (continued)

Use the optional *pdev* if

- You are switching -IC to -AC or -AC to -IC or are converting from pre-Rev. 21.0 and
- The partition being created is not the first partition and
- The first partition of this disk is the command device.
- The first partition is the command device and you want to display the DBS file.

When you use this option, **FIX_DISK** must be invoked from the supervisor terminal. Warn all users, then log out all users, and shut down all servers. You must restart servers when **FIX_DISK** finishes, particularly the Login server and DSM.

-CONVERT_19

Converts a partition to a Rev. 19.0 format partition from an earlier revision. Converts the BADSPT file to Rev. 19.0 format; initializes quota information; disables the display of warning/error messages related to quotas; and creates a new revision stamp. You must use the -FIX option with -CONVERT_19.

-CONVERT_20

Converts a partition to Rev. 20.0 format from an earlier revision. Directories created *after* the conversion are hashed. Existing directories are not hashed; they retain their current formats. In converting from pre-Rev. 20.0, -CONVERT_20 converts the BADSPT file to Rev. 20.0 format; initializes quota information; disables the display of warning/error messages related to quotas; and creates a new revision stamp. The MFD is not hashed. You must use -FIX with -CONVERT_20.

-CONVERT_21

Converts a partition to a Rev. 21.0 format partition from an earlier revision. If a badspot file exists, it is read and the data is written to the DBS file. The DBS file and the RMA are created and initialized on the first partition of the physical disk. The revision stamp is updated. You must use the -FIX and the -DISK_TYPE options with -CONVERT_21.

-CONVERT_22.1

Converts a Rev. 22.0 partition to a Rev. 22.1 standard partition having unlimited CAM file extents by updating the partition's DSKRAT. You must use the -FIX option with -CONVERT_22.1.

-DISK_TYPE [type]
-DT

Specify the *type* of disk to partition. Required with -CONVERT_21 since size of DBS file and RMA depend on disk type.

*FIX_DISK Option**Action***-DUMP_DBS
-DDBS**

Displays the dynamic badspot file. If the partition is in Dynamic Badspot Handling (-IC) mode, the DBS file is up-to-date; if in Nondynamic Badspot Handling (-AC) mode, the DBS file may not be up-to-date. Use with -DISK *pdev* to specify the first partition where the DBS file is located or with -COMDEV *pdev* if the first partition is the command device.

-DUFE

Eliminates all inconsistent file entries or entries of unknown type. -DUFE is the default. If you use -DUFE or do not specify -SUFE, unknown file entries are eliminated, directories containing unknown file entries are compressed, and the DSKRAT is altered to indicate which records are actually in use. Use -SUFE to avoid the accidental deletion of valid file entries caused by running the wrong version of FIX_DISK. You cannot use both -DUFE and -SUFE on the same command line.

-FAST

Rapidly checks the condition of a robust partition or rapidly repairs a robust partition. You must include the -FIX option to repair the partition. See Table 7-1 for recommendations on when to use the -FAST option (fast FIX_DISK). You can also use this option with a standard partition where the only file system inconsistency is a damaged quota system. If there are other problems on the standard partition, FIX_DISK defaults to ignoring the -FAST option and full FIX_DISK is run.

-FIX

Corrects quota information, truncates or deletes defective files, generates a corrected DSKRAT if the current one is bad, and maps the badspot records to the BADSPT file and to the DBS file. If you omit the -FIX option, no disk modifications are performed.

Use -FIX whenever repair or conversion operations are to be performed. However, if you suspect that the disk drive itself is faulty, do not use -FIX. Must be used with these options:

-ADD_BADSPOT	-INTELLIGENT_CONTROLLER
-CONVERT_19	-ALL_CONTROLLER
-CONVERT_20	-ODI
-CONVERT_21	-RDI
-CONVERT_22.1	-UFD_COMPRESSION
-INTERACTIVE	-MAX_EXTENT_SIZE
	-MIN_EXTENT_SIZE

It is important to run FIX_DISK once without using the -FIX option. FIX_DISK then reports inconsistencies, but does not attempt to repair them.

FIX_DISK Option

Action

- HELP** Displays a list of `FIX_DISK` options and brief explanations of their operation. Can be used alone and from any terminal.
- INTELLIGENT_CONTROLLER**
- IC** Selects Dynamic Badspot Handling mode for this Rev. 22.0 partition allowing dynamic badspot handling and mirroring to take place. Either `-IC` or `-AC` is recommended when converting to Rev. 21.0 with the `-CONVERT_21` option. Use `-DISK_TYPE` and `-FIX` with `-IC`.
- INTERACTIVE** Asks questions leading to construction of a consistent `DSKRAT` if the current `DSKRAT` is defective or missing. If you omit `-INT` and the current `DSKRAT` is bad or missing, `FIX_DISK` aborts. You must use the `-FIX` option with `-INTERACTIVE`.
- INT** Use this option if `FIX_DISK` has previously aborted and generated an error message.
- LEVEL [n]** Sets *n* (decimal) as the lowest level in the tree structure to be displayed. When this option is omitted, the default value is level 1, the first level in the MFD.
- LIST_BADSPOTS** Displays octal record numbers of all badspots and, for equivalence blocks, displays record numbers of the badspot and the remapped record. In addition to the record number, the track, head, and sector numbers are also listed for badspots and equivalence blocks.
- LB**
- LIST_FILE** Displays filenames in all directories down to the directory level displayed according to the `-LEVEL n` in effect at the time.
- LF**
- MAX_EXTENT_SIZE size**
- MAXSIZ** Specifies that you wish to change the maximum extent size for CAM files at Rev. 22.0. You must use the `-FIX` option with `-MAXSIZ`.
- MAX_NESTED_LEVEL [n]**
- MAX** Sets *n* (decimal) as the maximum depth to which directories can be nested. The default maximum depth is 99 levels.
- MIN_EXTENT_SIZE size**
- MINSIZ** Specifies that you wish to change the minimum extent size for CAM files at Rev. 22.0. You must use the `-FIX` option with `-MINSIZ`.
- NO_QUOTA** Assumes the partition is not a quota partition; disables quota checking. You must use this option on all pre-Rev. 19.0 disks because these disks cannot be quota disks.
- NQ**

FIX_DISK Option Action

-NUMBER_OF_RETRIES *number*

-NUMRTY Specifies the number of times FIX_DISK is to try to read a failing record. Each retry includes nine different ways of reading a record. The default is two retries.

-OVERRIDE_DEFAULT_INTERLEAVE

-ODI Changes the file record allocation direction to forward and the interleave factor to 3. Valid only with Rev. 20.0, Rev. 21.0, and Rev. 22.0 standard partitions.

-RESTORE_DEFAULT_INTERLEAVE

-RDI Changes the file record allocation direction to reverse and the interleave factor to 1. Valid only with Rev. 20.0, Rev. 21.0, and Rev. 22.0 standard partitions.

-SUFE

Saves all inconsistent file entries or entries of unknown type. If you omit -SUFE, the default is -DUFE and all unknown file entries are eliminated, directories containing unknown file entries are compressed, and the DSKRAT is altered only to indicate which records are actually in use. Use this option to avoid the accidental deletion of valid file entries caused by running the wrong version of FIX_DISK. You cannot use both -DUFE and -SUFE on the same command line.

-TRUNCATE

-TRU

Truncates files when an uncorrectable badspot or an uninitialized record in a CAM file on a robust partition is found. When a file is truncated, the part of the file that is located at or beyond the file pointer is eliminated from the file. If the file pointer is at the beginning of the file, all the information in the file is removed, but the filename remains in the file directory. Normally when FIX_DISK encounters an uncorrectable badspot or an uninitialized record, it creates a null record on a good portion of the partition and appends the remaining records of the original file to the null record.

-UFD_COMPRESSION

-CMPR

Compresses directories by eliminating all entries for file system objects flagged as being deleted. Use of this option results in a decrease in the search time for top-level directories. You must use the -FIX option with -CMPR.

GLOSSARY

The following terms are used in this book and are defined here for your convenience.

access control list (ACL)

A list of users and their access rights to file system objects as produced by the LIST_ACCESS command.

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 (-AC) mode file listing badspots on a partition. There is one MFD>BADSPT file per partition if there are badspots on the partition. This file consists of physical addresses of each badspot on the partition.

BRA

Beginning record address; the address of the beginning of a record.

breaking the mirror

Disabling of the mirroring process between two partitions by shutting down one of the mirrored pair.

buffer

To temporarily store records as an intelligent disk controller does when reading records.

cache

See buffer.

CAM

See Contiguous Access Method (CAM).

catch-up copy

The copy initiated when PRIMOS determines that two partitions that are to be mirrored are not identical either because their date and time of shutdown (DTS) stamps are not the same or because the two partitions were not started at the same time.

COMDEV

The PRIMOS system command device; the logical disk where PRIMOS and related files exist; logical device 0 (ldev 0).

correctable error

A disk data-error that PRIMOS or the disk controller can correct by using an error correction code or by physically adjusting the read head on the surface of the disk.

CMD

Cartridge module device; a type of physical disk for file system storage that has a removable portion and a fixed portion.

Contiguous Access Method (CAM)

A method of allocating and storing records in a file. The file records are stored contiguously in extents. PRIMOS accesses the records by reading an extent map.

cylinder

The intersection of individual tracks on all the surfaces of a disk; for example, if a geometrical solid was passed through track 123 of each surface of a physical disk, it would form a cylinder. *See also disk formatting.*

DAM

See Direct Access Method.

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 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 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 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* Winchester disk; 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.

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 (-IC) 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.

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.

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 Figure 10-1. *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**.

ldev

An octal number between 0 and 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.

logical device number

See ldev.

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.

Nondynamic Badspot Handling (-AC) mode

The state of a disk that allows all disk controllers (intelligent and nonintelligent controllers) to access the disk. The badspot handling process consists of adding badspots to the file BADSPT automatically by MAKE; adding badspots manually by using the appropriate MAKE or FIX_DISK options; copying badspots from other partitions by using the appropriate MAKE or FIX_DISK options; and adding badspots from vendor flaw maps.

nonintelligent disk controller

A disk controller that is incapable of buffering data and that can execute only one command at a time. *See also intelligent disk controllers.*

paging

Moving files that are not currently in use out of main memory to a designated area of disk storage referred to as the paging device. This technique makes it appear as if the system has more memory than it actually does and is the basis for virtual memory.

paging partition

A logical disk where paging records reside; used by PRIMOS for paging as part of the virtual memory space. Paging partitions are always automatically split beginning at Rev. 20. Also referred to as *paging disks*. A Rev. 21.0 or later system can have a maximum of eight paging partitions.

partition

A logical grouping of physical disk surfaces that provides a logical range of disk record addresses. This area is defined by the DSKRAT and by the physical device number (pdev) of the partition.

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 (pdev)

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.*

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.

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.

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.

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.

Segment Directory (SEGDIR)

Contains entries referenced by file numbers from 1 to 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.

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.

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.

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. CMDNC0 is a top-level directory on the command device.

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).

INDEX

INDEX

Numbers

9950 CPU,
class defined, 10-3, 10-5

A

ACLs,
default setting on MFD, 5-2
defined, G-1
MAKE, set by on partition, 5-30
setting or changing on partition, 5-30
ADDISK command, 4-1, 4-5
copying badspots, 5-10
Dynamic Badspot Handling (-IC) mode,
8-11
FIX_DISK, use with, 6-7
force FIX_DISK, 7-3
-FORCE option 7-30
message, full DBS, 8-5
mirroring and, 9-7
modification for robust partitions, 7-2
PROTECT argument, 7-30
renaming partitions, 5-20
robust partitions, 7-30
warm start problem, 9-7
Addresses, table of controller, 3-7
Algorithms,
CAM file record allocation, defined,
7-20
CAM files on robust partition, 7-20
record allocation, 10-1
Allocation,
choosing method of, 10-7
default direction, 10-5
default, table, 10-5
direction for standard partitions, 10-5

PRIMOS records, table of, 10-4
records, order of, 10-3
SAM and DAM files, 10-6
See also Record allocation
Applications, standalone, 10-7
ASSIGN DISK command, 4-5
copying badspots, 5-11
Assignable Disks Table, 1-2, 4-1
defined, G-1

B

Backups,
logical save, 2-4
partition, MAKE procedure, 5-27
physical save, 2-4
running FIX_DISK during, 6-5
Badspots,
adding to BADSPT file, 6-12
checking algorithm, 5-14
checking level, 5-9
copying, 5-10
copying, example of, 5-36
default handling mode, 8-11
defined, G-1
discussion of, 2-2
disk media, 2-1
dynamic handling of, 2-3
dynamic handling option, 5-14
dynamic handling, defined, G-4
dynamic handling, requirements for,
8-2
dynamic handling, summary of purpose
of, 8-4
dynamic, defined, 8-1, G-3
dynamic, discussion of, 2-3
enter by track, head, sector, 5-24

Badspots, (continued)

- entering known, 5-20, 5-23
- entering known, example of, 5-34
- equivalence section, 6-4
- file, 2-2
- file, creating new, 5-11
- file, type of, 6-4
- files, keeping redundant, 5-25
- flaw map, ignoring, 5-17
- flaw maps, use of, 5-16
- flaws, 2-2
- list of, 5-20
- listing, 6-23
- message from COPY_DISK or PHYRST, 2-7
- physical copy and dynamic handling of, 2-5
- recording, 5-23
- remap to RMA, 2-3
- remapping on Model 4719, caution on, 2-4
- RMA, handling, 8-8
- static, defined, 8-1, G-8
- static, discussion of, 2-3
- static, handling by intelligent controller, 2-3, 8-4
- verification level, 5-9
- See also* BADSPT file; Files, BADSPT

BADSPT file,

- adding to, 6-12
- building, 2-2
- creation by MAKE, 5-2
- handling, 6-4
- on dynamic badspot handling partition, 8-5

See also Files, BADSPT; Badspots

Baud rate,

- default, 5-10
- setting on boot disk, 5-10

Bits, meaning of in pdev, 3-12

Block, equivalence, 2-5, 2-6

See also Equivalence blocks

Booting,

- MAKE from disk, 5-37
- MAKE from tape, 5-40
- robust partitions and, 7-3

BRA, defined, G-1

Buffer,

- controller ability, 10-2
- defined, G-2

Bytes, number in record, 3-2

C

Cache,

- controller ability, 10-2
- defined, G-2

CAM files,

- access to, example of, 1-11
- algorithm for record allocation, 7-20
- algorithm for, 7-20
- CF\$SME subroutine, 7-21
- copying of, 7-9
- defined, G-2
- deletion of on robust partition, 7-3
- discussion of, 7-8
- extent map, 1-9
- extent maps, discussion of, 7-8
- extent sizes, default, 5-15, 6-24
- extent sizes, setting, 5-15, 6-23
- extents, 1-9
- fast FIX_DISK on, 7-6
- fragmentation, 1-11
- fragmentation, reducing, 6-18
- maximum extents, 6-18
- minimum and maximum extent sizes, 7-20
- minimum extent size, 7-21
- monitor with LEM, 6-18
- organization, 1-9
- performance of, 6-18, 7-11, 7-21
- simultaneous access of, 7-11
- speed of data access, 7-3
- structure, 1-10
- subroutines for, 7-9
- unlimited extents, 7-8

Cartridge Module Devices,

See CMDs; Disks

Catch-up copy,

- COPY_DISK or PHYSAV, use of, 9-5
- defined, 9-5, G-2
- failure of, 9-16
- startup of, 9-6
- warning on, 9-5

Cautions,

- fast FIX_DISK and some errors, 7-29
- first partition, converting, 8-12
- FIX_DISK, converting partitions, 6-17
- FIX_DISK, pdev and -DISK option, 6-11

Cautions, (continued)

FIX_DISK, use of -FAST option, 6-21
 FIX_DISK, use of -FIX option, 6-11
 FIX_DISK, use of pre-Rev. 22.1
 version and -SUFE option, 6-12
 MAKE, pdev and -DISK option, 5-6
 MAKE, specifying disk type, 5-8
 MAKE, using pre-Rev. 22.1 version,
 5-2
 MIRROR_ON and MIRROR_OFF, use
 in RESUS, 9-8
 mirroring and catch-up copy, 9-11
 mirroring part of physical disk, 9-3
 mode (-AC, -IC) on physical disk, 8-7
 Model 4719 and badspot remapping,
 2-4
 partitions and Rev. 22.1 copy utilities,
 8-9
 CMDs,
 defined, G-2
 disk type, 2-2
 drive unit numbers for, 3-3
 partitioning, 3-18
 pdevs for, 3-14
 pdevs, table of, 3-19
 See also Disks
 Cold starts,
 after FIX_DISK, 6-16
 controller download, 5-14
 COMDEV,
 defined, G-2
 mirroring at startup, 9-3, 9-8
 running FIX_DISK on, 4-4
 COMDVM configuration directive, 9-3
 example of use, 9-4
 Commands,
 ADDISK, 4-1, 4-5
 ADDISK, and copying badspots, 5-10
 ADDISK, and mirroring, 9-7
 ADDISK, and robust partition, 7-30
 ADDISK, message on full DBS, 8-5
 ADDISK, warm start problem, 9-7
 ASSIGN DISK, 4-5
 ASSIGN DISK, and copying badspots,
 5-11
 ASSIGN, 1-2
 COPY -MXL, 7-21
 COPY, files to robust partition, 7-18
 COPY, reducing CAM file extents, 6-18
 DISKS NOT, 4-5

DISKS, 1-2, 4-4
 FIX_DISK, messages from, B-1
 ICE, re-initializing the supervisor
 terminal environment, 6-16
 LCB options, 7-24
 LCB, display of, 7-24
 LCB, format of, 7-23
 LD and robust partitions, 7-7
 LEM, monitoring CAM files, 6-18
 MAKE, messages from, A-1
 MESSAGE, 4-3
 MIRROR_OFF, format, 9-10
 MIRROR_OFF, options, 9-10
 MIRROR_ON in PRIMOS.COMI, 9-8
 MIRROR_ON, configuration directives
 necessary for, 9-9
 MIRROR_ON, format, 9-8
 MIRROR_ON, options, 9-9
 MIRROR_ON, warning on use of, 9-5
 mirroring, 9-8
 mirroring, effect of, 9-6
 MTRESUME, invoking FIX_DISK, 6-38
 pdev, using in, 3-1
 SHUTDOWN, 4-1
 SHUTDOWN, and mirroring, 9-7
 START_DSM, 4-4, 6-16
 START_LSR, 4-4, 6-16
 STATUS DISKS and robust partitions,
 7-7
 STATUS DISKS, and mirroring, 9-6
 STATUS DISKS, determining pdevs, 6-9
 STOP_DSM, 4-4, 6-16
 STOP_LSR, 4-4, 6-16
 UNASSIGN DISK, 4-5
 CONFIG directives,
 required for mirroring, 9-2, 9-3, 9-9
 Controllers,
 disk, defined, 3-3
 disk, mirroring and -AC mode, 9-14
 disk, mirroring performance, 9-3
 download file, D-12
 intelligent, 10-2
 intelligent, and dynamic badspot
 handling, 2-3
 nonintelligent, 10-2
 See also Disk controllers, Intelligent
 disk controllers; Nonintelligent disk
 controllers
 Converting,
 partitions to Rev. 22.1, 5-3

Converting, (continued)

- partitions to Rev. 22.1, procedure for, 7-18
- partitions, with FIX_DISK, discussion 6-3

COPY command,

- copying files to robust partition, 7-18
- MXL option, 7-21

Copy, physical, 2-4

COPY_DISK command,

- badspot handling by, 2-5
- caution on use of, 8-9
- dynamic badspot handling and, 8-9
- mirrored partitions, use of for, 9-5
- running FIX_DISK, 6-5
- valid data transfers, 8-10

CPUs,

- 9950 class, defined, 5-19, 10-3, 10-5
- sectoring recommendations related to, 10-3

Creating disks, 5-1

- defined, 1-2

See also Disks; Partitions, MAKE utility

Cylinder, defined, G-2

D

DAM files, 1-8, 1-9

- defined, G-2
- record allocation, 10-6
- structure, 1-9

Data management,

- use on robust partitions, 7-17

Data,

- valid transfers and physical copy utilities, 8-10
- verifying integrity of, 7-17

DBMS on robust partitions, 7-12

DBS file,

- additions to, 8-5
- contents of, 2-3
- damage to, 8-8
- damaged, reconstructing, 8-7
- defined, G-2
- disk geometry, caution on changing, 5-8
- display of, 6-19, 8-14
- display, example of, 6-37
- full, 8-5

- function of, 8-4
- handling, 6-4
- location of, 8-5
- prompt for change of, 8-6
- sizes of, 5-22
- status of, 8-2

See also Files, DBS

Defaults,

- ACL setting on MFD, 5-2
- badspot handling mode, 5-9, 8-11
- baud rate, 5-10
- extent sizes, 7-20
- extent sizes, discussion of, 7-22
- extent sizes, maximum and minimum, 5-15, 6-24
- interleave factor, 10-5
- record allocation direction, 10-5
- sectoring, 5-19

Devices, defined, 1-1

Directives, configuration, required for mirroring, 9-2, 9-3, 9-9

Directories,

- access to, 1-5
- ACL and hashing, 1-7, 5-2
- compressing entries, 6-11
- created by MAKE, 1-2, 5-2
- discussion of, 1-5
- hashed, 1-7
- hashed, and FIX_DISK conversion, 6-3
- hashed, and MAKE, 5-2
- linear on robust partitions, 7-10
- linear structure, 7-8
- linear, access to, 7-10
- nested, 6-14
- password, and hashing, 1-7
- robust partitions and, 7-10
- segment and FIX_DISK, 7-6
- segment, 1-7
- segment, file types, 7-9
- segment, structure of, 7-6
- top-level, defined, 1-5, G-9
- tree structure, 1-6
- truncating, 6-14

Disk controllers,

- address, 3-3
- address, table of, 3-7
- defined, 3-3
- messages, D-11
- modes, D-1
- See also* Controllers

- Disk drives,
 - defined, 2-2, G-3
 - dual-ported, and mirroring, 9-9
 - dual-ported, warning on priority select, 9-9
 - unit numbers for FMDs, 3-3
 - unit numbers for SMDs and CMDs, 3-3
 - See also* Disks
 - Disk mirroring,
 - See* Mirroring
 - DISKS command, 4-4
 - FIX_DISK, use with, 6-7
 - DISKS NOT command, 4-5
 - Disks, 1-1
 - allocating space, 1-4
 - assignable, 4-1
 - assignable, defined, G-1
 - assigning and unassigning, 4-3
 - assigning and unassigning, procedure for, 4-4
 - badspot checking, default level of, 5-9
 - badspots, 2-1
 - baud rate, 5-10
 - characteristics of, 3-2
 - CMD, 2-2, 3-1
 - creating, 5-1
 - dynamic badspot handling, 2-3
 - error messages, D-1
 - errors, 8-3
 - file system, 4-1
 - file system, defined, G-4
 - FMD, 2-2, 3-1
 - formatting, 5-1
 - fragmentation, preventing, 7-21
 - geometry, 2-1
 - geometry, caution on, 5-8
 - hardware problems, warning on, 6-5
 - logical, 1-1
 - logical, defined, G-6
 - message indicating problems, 6-6
 - pack, 2-2
 - paging, 5-1, 5-21
 - partitioning specific types, 3-14
 - partitioning, 5-1
 - physical, 1-1, 2-1
 - preformatted, 5-12
 - size of, 3-2
 - SMD, 2-2, 3-1
 - static badspot handling, 2-3
 - storage, 1-1
 - types of, 2-1, 3-1
 - types, list of valid, 6-19
 - types, table of, 5-8
 - unit number, table of, 3-7
 - user, 5-1
 - user, defined, G-9
 - Winchester, 2-2
 - See also* Disk drives; CMDs; SMDs; FMDs; MAKE utility; Partitions; Robust Partitions; First partition
 - DLINFO program,
 - checking disk controller file, D-12
 - Downline load,
 - ICOP software, 2-3
 - intelligent controller message, 5-14, 9-7
 - Drives, disk,
 - See* Disk drives
 - DSKRAT, 1-4
 - bits in, shutdown and PRIMOS II, 6-15
 - defined, G-3
 - discussion of, 2-2
 - locating errors in, 6-3
 - marking badspots in, 8-6
 - name of, 1-4, 5-1, 5-20
 - repair of, 6-21
 - See also* Files, DSKRAT
 - DSM subsystem, FIX_DISK on
 - COMDEV, 6-14, 6-15
 - Dual-ported disks,
 - See* Disk drives, dual-ported
 - Dynamic badspot handling,
 - ADDISK, effect on, 8-11
 - commands, effect on operator, 8-10
 - COPY_DISK, PHYSAV, and PHYRST, 8-9
 - FIX_DISK operation and, 8-12
 - FIX_DISK, effect on, 8-12
 - MAKE, effect on, 8-11
 - messages associated with, 8-15
 - mode, defined, G-4
 - See also* Badspots
 - DYNBSP file,
 - creation by MAKE, 5-2
 - purpose of, 8-5
- E**
- EPFs, file type of, 7-9

- Equivalence blocks,
 - creating, 2-5
 - creating, results of, 2-7
 - defined, 2-6, G-4
 - need for, 2-6
 - See also* Badspots
- Equivalence section, badspot file, 6-4
- Error messages,
 - See* Messages
- Errors,
 - conditions that break mirror, 9-13
 - correctable read, handling by intelligent controller, 8-4
 - correctable, defined, G-2
 - correctable, handling by PRIMOS, 8-3
 - correction by intelligent disk controller, D-11
 - disk, 8-3
 - intelligent controller, handling by, 8-4
 - mirroring, handling by PRIMOS, 9-13
 - null-filled records, 7-4
 - pointer mismatch, 7-3
 - read and write messages, 6-6
 - read, 8-3
 - read, recovery from in mirroring, 9-15
 - uncorrectable read, and mirroring, 9-15
 - uncorrectable read, handling by intelligent controller, 8-4
 - uncorrectable read, use of FIX_DISK
 - ADBADS for, 8-4
 - uncorrectable, handling by PRIMOS, 8-3
 - uncorrectable, record numbers of, 6-13
 - uninitialized block, 7-3
 - write, 8-3
 - write, handling by intelligent controller, 8-4
 - write, recovery from in mirroring, 9-14
 - See also* Messages
- Event logging,
 - FIX_DISK on directories, 6-15
- Examples,
 - ADDISK command and mirroring, 9-7
 - CAM file access, 1-11
 - COMDVM, use of, 9-4
 - controller download file, checking, D-12
 - copying badspot files, 5-26
 - FIX_DISK, changing controller modes, 6-36
 - FIX_DISK, checking partition shutdown, 6-35
 - FIX_DISK, converting partitions, 6-28
 - FIX_DISK, DBS file display, 6-37
 - FIX_DISK, DSKRAT reconstruction, 6-33
 - FIX_DISK, error handling, 6-27
 - FIX_DISK, normal display, 6-26
 - FIX_DISK, record truncation, 6-32
 - FIX_DISK, using -COMDEV option, 6-29
 - FIX_DISK, using without -FIX option, 6-31
 - forward sectoring, 10-2
 - LCB display, 7-24
 - MAKE utility, 5-32
 - MAKE utility, booting, 5-38, 5-40
 - MAKE, badspots, copying, 5-36
 - MAKE, badspots, input of known, 5-34
 - MAKE, first partition, remaking, 5-33
 - MAKE, normal display, 5-32
 - MAKE, paging partition, creating, 5-35
 - MTRESUME, using to invoke
 - FIX_DISK from tape, 6-38, 6-39
 - PAGINM, use of, 9-4
 - pdev construction, 3-13
 - pdev determination, 3-9
 - pdev for all drives and controllers, 3-9
 - pdev worksheet, 3-10
 - pdevs for SMDs, 3-11
 - physical copy of partition, 2-5
 - record allocation, 7-21, 7-22, 10-2
 - reverse sectoring, 10-2
 - running FIX_DISK, 6-26
 - SHUTDN command and mirroring, 9-7
 - size of subfiles, 7-10
 - SMDs, partitioning, 3-10
 - STATUS command and mirroring, 9-7
- Extent maps,
 - discussion of, 7-8
- Extents,
 - CAM file, 1-9
 - defined, G-4
 - map, CAM file, 1-9
 - maximum number of, 6-18
 - maximum size, default, 5-15
 - minimum size, 7-21
 - minimum size, default, 5-15
 - sizes, default minimum and maximum, 7-20

Extents, (continued)

- sizes, discussion of default, 7-22
- sizes, setting, 5-15, 6-23, 7-20

F

Fast FIX_DISK,

- See* FIX_DISK utility; FIX_DISK options

Features, of PRIMOS revisions, 5-4

File system,

- concepts, 1-4
- initialization of, warning on, 5-13
- purpose, 1-4
- See also* Files

Files,

- administration of on robust partition, 7-20
- allocation on robust partition, 7-20
- badspot, 2-2
- badspot, copying, 5-25
- badspot, creating new, 5-11
- badspot, redundant, 5-25
- BADSPT, 2-2
- BADSPT, adding to, 6-12
- BADSPT, creation by MAKE, 5-2
- BADSPT, defined, G-1
- BADSPT, handling, 6-4
- BADSPT, on dynamic badspot handling partition, 8-5
- CAM; *See* CAM files
- CF\$SME subroutine for CAM, 7-21
- checking data integrity of INFORMATION files, 7-12
- checking data integrity of MIDASPLUS files, 7-12
- COMI, and mirroring, 9-10
- COMINPUT, to run MAKE, 5-2
- controller download, D-12
- DAM organization, 1-8
- data management, considerations for use on robust partitions, 7-13
- DBS and RMA on first partition, 5-3
- DBS and RMA, location of, 8-5
- deletion of on standard and robust partitions, 7-3
- DSKRAT, 1-4
- DSKRAT, discussion of, 2-2
- DSKRAT, listing badspots in, 8-6
- DSM logging, 6-15

dynamically hashed and robust partitions, 7-12

DYNBSP, creation by MAKE, 5-2

DYNBSP, purpose of, 8-5

logical types, 7-2

logical typing, 7-7

logical-to-physical mapping, 7-2

maintenance of, 1-3

MAKE, created by, 1-2, 5-2

name, referring to by, 1-4

names, listing by FIX_DISK, 6-23

organization on robust partitions, 7-7

performance on robust partitions, 7-11

pointers and badspot handling, 2-6

PRIMOS.COMI, MIRROR_ON in, 9-8

PRIMOS.COMI, mirroring directives in, 9-8

record allocation, 5-18

records for on paging partition, 5-21

records for system in split first partition, 5-22

RMA, function of, 8-4

other on robust partitions, 7-14

SAM and DAM allocation, 10-6

SAM organization, 1-8

segment directory types, 7-9

size of segdir subfiles, 7-10

structures, 1-7

structures, relation among, 1-12

truncation and deletion of on robust partitions, 7-11

truncation of, 6-25

unknown entries in, 6-11

See also DBS file; File system

First partition,

access by MAKE and FIX_DISK, 8-9

converting before others, 6-17

creation of, 5-14

defined, G-4

discussion of, 8-7

function of, 5-3

geometry of, procedure for changing, 8-8

MAKE, warning on use, 8-8

mode switching, 6-14

remaking, example of, 5-33

split, 5-22

surfaces on, changing number of, 8-8

warning on changing geometry of, 5-7

See also Partitions

FIX_DISK options,

- ADD_BADSPOT and unrecoverable read error, 7-29
- FAST and MTTR, 7-14
- FAST, understanding, 7-5
- TRUNCATE, 7-29
- ADD_BADSPOT, discussion of, 6-12
- ALL_CONTROLLER, discussion of, 6-13
- AUTO_TRUNCATION, discussion of, 6-14
- categories of, 6-9
- CHECK, discussion of, 6-15
- COMDEV, assigning and shutting down partition, 6-10
- COMDEV, assigning COMDEV, 6-14
- COMDEV, discussion of, 6-15
- COMDEV, use of, 4-4
- COMDEV, when to use, 6-16
- CONVERT_19, discussion of, 6-17
- CONVERT_20, discussion of, 6-17
- CONVERT_21 and record allocation bit, 10-6
- CONVERT_21, discussion of, 6-17
- CONVERT_22.1, discussion of, 6-18, 7-18
- DISK and pdev, caution on, 6-11
- DISK, discussion of, 6-10
- DISK_TYPE, discussion of, 6-19
- DISK_TYPE, use with -CONVERT_21, 6-17
- DUFE and -SUFE options, caution on use, 6-12
- DUFE, discussion of, 6-11
- DUMP_DBS, discussion of, 6-19
- FAST, caution on use of, 6-21
- FAST, discussion of, 6-20
- FIX, caution on use of, 6-11
- FIX, discussion of, 6-11, 6-13
- FIX, need for, 6-12
- FIX, options to specify with, 6-12
- FIX, use with -CONVERT_22.1, 6-18
- IC and -AC, use with -CONVERT_21, 6-17
- INTELLIGENT_CONTROLLER, discussion of, 6-13
- INTERACTIVE, discussion of, 6-22
- LEVEL and -MAX_NESTED_LEVEL comparison, 6-22
- LEVEL, discussion of, 6-22

list of, 6-9

- LIST_BADSPOTS, discussion of, 6-23
- LIST_FILE, discussion of, 6-23
- MAX_EXTENT_SIZE, discussion of, 6-23, 7-30
- MAX_NESTED_LEVEL, discussion of, 6-14
- MIN_EXTENT_SIZE, discussion of, 6-23, 7-30
- NO_QUOTA, discussion of, 6-24
- NUMBER_OF_RETRIES, discussion of, 6-12
- ODI, discussion of, 6-24
- OVERRIDE_DEFAULT_INTERLEAVE, discussion of, 6-24
- RDI, discussion of, 6-25
- RESTORE_DEFAULT_INTERLEAVE, discussion of, 6-25
- SUFE and -DUFE options, caution on use, 6-12
- SUFE, discussion of, 6-11
- TRUNCATE, discussion of, 6-25
- UFD_COMPRESSION, discussion of, 6-11

FIX_DISK utility,

- access to first partition by, 8-9
- ADBADS option and mirroring, 9-15
- ADBADS option for uncorrectable read error, 8-4
- ADD_BADSPOT option, 6-12
- ALL_CONTROLLER option, 6-13
- assigning COMDEV, 6-14
- AUTO_TRUNCATION option, 6-14
- backups, running during, 6-5
- changing controller modes, example of, 6-36
- CHECK option, 6-15
- checking partition shutdown, example of, 6-35
- cold start after, 6-16
- COMDEV option and conversions, 8-14
- COMDEV option, 6-15
- COMDEV option, example of, 6-29
- command device, running on, 4-4
- command format, 6-9
- conversion to Dynamic Badspot Handling (-IC) mode, 8-13
- conversion to Nondynamic Badspot Handling (-AC) mode, 8-13
- CONVERT_19 option, 6-17

FIX_DISK utility, (continued)

- CONVERT_20 option, 6-17
 - CONVERT_21 option, 6-17
 - CONVERT_21 option, badspot remapping, 8-6
 - CONVERT_22.1 option, 6-18, 7-18
 - converting partitions with, 6-3
 - converting partitions, example of, 6-28
 - DBS file display, example of, 6-37
 - design of, 7-5
 - detecting file structure corruption, 7-2
 - discussion of, 1-3
 - DISK option, 6-10
 - DISK_TYPE option, 6-19
 - display of DBS file, 8-14
 - display of, 6-22
 - DSKRAT reconstruction, example of, 6-33
 - DUFE option, 6-11
 - DUMP_DBMS option, 6-19
 - dynamic badspot handling and, 8-12
 - error handling, example of, 6-27
 - error messages indicating disk problems, 6-6
 - error on COMDEV, 6-16
 - examples of, 6-26
 - fast and data record headers, 7-5
 - fast and integrity verification, 7-2
 - FAST option, 6-20
 - fast, and some errors, caution on, 7-29
 - fast, defined, 6-20
 - fast, operation of, 7-5
 - fast, use of, 7-3
 - fast, use on standard partition, 6-21
 - fast, when to use, 7-28
 - FIX option, 6-11, 6-13
 - FIX option, example of using without, 6-31
 - FIX option, need for, 6-12
 - full and record header verification, 7-6
 - full, defined, 6-20
 - full, when to use, 7-27
 - functions of, 6-1
 - halts, running after, 6-5, 7-29
 - hardware problems, warning on, 6-5
 - HELP option, 6-21
 - INTELLIGENT_CONTROLLER option, 6-13
 - INTERACTIVE option, 6-22
 - invoking from magnetic tape, 6-38
 - LEVEL option, 6-22
 - LIST_BADSPOTS option, 6-23
 - LIST_FILE option, 6-23
 - MAX_EXTENT_SIZE option, 6-23
 - MAX_NESTED_LEVEL option, 6-14
 - messages from, B-1
 - MIN_EXTENT_SIZE option, 6-23
 - mode (-AC and -IC) conversion, 8-2
 - MTRESUME, using to invoke from tape, 6-38
 - NO_QUOTA option, 6-24
 - normal display, example of, 6-26
 - NUMBER_OF_RETRIES option, 6-12
 - ODI option, 6-24
 - operation of, 6-2
 - options, summary of, F-5
 - OVERRIDE_DEFAULT_INTERLEAVE option, 6-24
 - pathname of, 6-9
 - pdev and -DISK, caution on, 6-11
 - pdev specification, 6-10
 - procedure after running, 6-25
 - procedure before running, 6-7
 - procedure flowchart, 6-8
 - procedure for running, 6-7
 - RDI option, 6-25
 - recommendations for running, 6-5, 7-28, 7-29
 - record read attempts, 6-12
 - record truncation, example of, 6-32
 - RESTORE_DEFAULT_INTERLEAVE option, 6-25
 - robust partitions and, 7-27
 - SUFE option, 6-11
 - syntax summary, F-5
 - TRUNCATE option, 6-25
 - UFD_COMPRESSION option, 6-11
 - usage display, 6-21
 - using MTRESUME to invoke from tape, 6-39
 - using, 6-29
 - when to use, 6-4
 - See also* FIX_DISK options
- FIX_DISK,
See FIX_DISK utility; FIX_DISK options
- Fixed-Media Devices,
See FMDs; Disks
- FIXRAT utility, warning on use of, 1-3
- Flaw maps, 2-2

Flaw maps, (continued)

- badspots, list of known, 5-20
- default checking, 5-18
- defined, G-4
- ignoring, 5-17
- processing of, 5-16
- static badspots, handling by intelligent controller, 8-4
- use of, 5-16
- See also* Badspots

FMDs,

- 60MB, partitioning, 3-15
- 68MB, partitioning, 3-15
- 84MB, partitioning, 3-15
- 120MB, partitioning, 3-16
- 158MB, partitioning, 3-16
- 160MB, partitioning, 3-16
- 258MB (Model 4719), partitioning, 3-16
- 315MB (Model 4475), partitioning, 3-17
- 328MB (Model 4721), partitioning, 3-17
- 496MB (Model 4735), partitioning, 3-17
- 675MB, partitioning, 3-17
- 770MB (Model 4845), partitioning, 3-18
- 817MB (Model 4860), partitioning, 3-18
- defined, G-5
- disk type, 2-2
- drive unit numbers for, 3-4
- See also* Disks

Formatting,

- defined, 1-2, G-3
- disks, 5-1
- See also* MAKE utility

Fragmentation,

- CAM file, 1-11
- disk, preventing, 7-21
- partitions, criteria for determining, 7-25
- partitions, discussion of, 7-23
- procedure for handling, 7-27

Full FIX_DISK,

- See* FIX_DISK utility; FIX_DISK options

Functions, OCTAL, converting ldev, 7-24

G

Geometry,

- disk, 2-1
- disk, defined, G-3
- first partition, changing, 8-8

H

Halts,

- recovery, 7-2
- robust partitions and, 7-5
- running FIX_DISK, 6-5, 7-29

Hardware, problems and FIX_DISK, 6-5

Hashed directories,

- See* Directories, hashed

Hashed,

- defined, 5-2
- directories, 5-2
- index, 1-7
- See also* Directories, hashed

Hashing, 1-7

- defined, G-5

Heads,

- defined, G-5
- entering badspots by, 5-24

I

IAP, user directory, 1-5

ICOP software,

- disk controller, D-1
- intelligent disk controllers, 2-3

Index, hash, 1-7

INFORMATION on robust partitions, 7-12

Intelligent Channel Order Protocol,

- See* ICOP software; Dynamic badspot handling; Intelligent disk controllers
- See also* Controllers; Disk controllers

Intelligent disk controllers,

- defined, G-5
- downline load message, 5-14
- downloading, 5-9
- dynamic badspot handling, 2-3
- error correction by, D-11
- error handling by, D-4
- ICOP, 2-3
- logical status word, D-7, D-8
- mirroring, 5-14
- need for, 8-9
- OPCODE, D-6
- physical status word, D-9, D-10
- read error, correctable, handling, 8-4
- static badspot handling, 2-3, 8-4
- warm start problem, 9-7
- write-error, handling, 8-4

Interleave factor, 10-1, 10-2

Interleave factor, (continued)
 changing with `FIX_DISK`, 10-7
 default, 10-5
 defined, G-5
 setting on pre-Rev. 22.1 partitions, 10-6
 setting, 6-24, 10-6
 specifying, 5-18

L

LCB command,
 format and display, 7-23, 7-24
 options, 7-24
 ldev,
 converting octal to decimal, 7-24
 defined, G-5
 logical device 0, 5-14
 LED, disk controller, D-11
`LIST_CONTIGUOUS_BLOCKS` command,
See LCB command
 Logging, `FIX_DISK` and directories, 6-15
 Logical device 0 (ldev 0), 5-14
 Logical file type, 7-2
 Logical save, 2-4
 defined, G-6
 Login server,
`FIX_DISK` on `COMDEV`, 6-14, 6-16

M

`MAGSAV` command, badspot handling,
 2-4
 MAKE options,
 -AC, use with MAKE standalone, 5-14
 -ALL_CONTROLLER (-AC), discussion
 of, 5-9
 -BADLEV and -NOFLMP, combinations
 of, 5-18
 -BADSPOT_LEVEL, discussion of, 5-9
 -BAUD_RATE, discussion of, 5-10
 -CPYDEV, discussion of, 5-11
 -CPYNAM, discussion of, 5-10
 -DISK and pdev, caution on, 5-6
 -DISK, discussion of, 5-6
 -DISK_REVISION, discussion of, 5-11
 -DISK_TYPE (-DT), discussion of, 5-7
 -DISK_TYPE, use of for flaw map
 processing, 5-16
 -DISK_TYPE, use with -IC option,
 5-14

-FLAW_MAP, discussion of, 5-17
 -FORMAT, discussion of, 5-11
 -FORMAT, use with -IC option, 5-14
 -FORMAT, warning on use of, 5-12
 -HELP, discussion of, 5-13
 -INIT, discussion of, 5-13
 -INTELLIGENT_CONTROLLER (-IC),
 discussion of, 5-14
 -MAP_UNCORR, discussion of, 5-14
 -MAX_EXTENT_SIZE, discussion of,
 5-15
 -MIN_EXTENT_SIZE, discussion of,
 5-15
 -NEW_DISK, caution on, 5-10
 -NEW_DISK, discussion of, 5-11
 -NEWDSK, use with -IC option, 5-14
 -NO_INIT, discussion of, 5-13
 -NOFLMP and -BADLEV, combinations
 of, 5-18
 -ODI, use of, 10-6
 -OVERRIDE_DEFAULT_INTERLEAVE
 (-ODI), changing record allocation,
 10-5
 -OVERRIDE_DEFAULT_INTERLEAVE
 (-ODI), discussion of, 5-19
 -PARTITION, discussion of, 5-20
 -QUERY_BADSPOTS, discussion of,
 5-20
 -RDI, use of, 10-6
 required, 5-27
 -RESTORE_DEFAULT_INTERLEAVE
 (-RDI), changing record allocation,
 10-5
 -RESTORE_DEFAULT_INTERLEAVE
 (-RDI), discussion of, 5-19
 -ROBUST, discussion of, 5-20
 -SPLIT, discussion of, 5-21
See also MAKE utility
 MAKE utility,
 -AC option, use with MAKE
 standalone, 5-14
 access to first partition by, 8-9
 ACL, default on MFD, 5-2
 -ALL_CONTROLLER (-AC) option, 5-9
 -BADSPOT_LEVEL option, 5-9
 -BAUD_RATE option, 5-10
 booting from disk, 5-37
 booting from tape, 5-40
 command format, 5-5, 5-28
 converting to Rev. 22.1, using for, 5-4

MAKE utility, (continued)

- CPYDEV option, 5-11
- CPYNAM option, 5-10
- DISK option, 5-6
- disk type, specifying, 5-7
- DISK_REVISION option and badspot handling mode, 8-12
- DISK_REVISION option, 5-11
- DISK_TYPE (-DT) option, 5-7
- DISK_TYPE option, for flaw map processing, 5-16
- DT option, use with -IC option, 8-12
- dynamic badspot handling and, 8-11
- examples of, 5-32
- FLAW_MAP option, 5-17
- FORMAT option, 5-11
- FORMAT option, use of, 8-6
- FORMAT option, use with -IC option, 8-12
- FORMAT option, warning on use of, 8-8
- hashed directories, 5-2
- HELP option, 5-13
- INIT option, 5-13
- INTELLIGENT_CONTROLLER (-IC) option, 5-14
- MAP_UNCORR option, 5-14
- MAX_EXTENT_SIZE option, 5-15
- messages from, A-1
- MFD creation, 5-1
- MIN_EXTENT_SIZE option, 5-15
- mode (-AC, -IC) conversion, 8-3
- NEW_DISK option, 5-11
- NEW_DISK option, caution on, 5-10
- NEWDSK option, use with -IC option, 8-12
- NO_INIT option, 5-13
- options, required, 5-27
- options, summary of, 5-5, F-1
- OVERRIDE_DEFAULT_INTERLEAVE (-ODI) option, 5-19
- PARTITION option, 5-20
- pathname of, 5-2
- pdev and -DISK, caution on, 5-6
- pdev specification, 5-6
- procedure before running, 5-27
- procedure for running, 5-27
- QUERY_BADSPOTS option, 5-20
- RESTORE_DEFAULT_INTERLEAVE (-RDI) option, 5-19

- ROBUST option, 5-20
- running standalone, 5-9
- SPLIT option, 5-21
- syntax summary, F-1
- versions, caution on use of, 5-2
- warning on use of pre-Rev. 22.1 version on first partition, 5-3
- warning on use of pre-Rev. 22.1, 8-8
- See also* MAKE options

MAKE,

See MAKE utility; MAKE options

MAKE_ROBUST utility,

- discussion of, 7-19
- messages from, E-1
- operation of, 7-19
- options and arguments, 7-19
- syntax of, 7-19

Maximum extent size, discussion of, 5-15

Memory, virtual, 1-2

MESSAGE command, 4-3

Messages,

- COPY_DISK and badspots, 2-7
- disk controller, D-11
- disk error, 1-4, D-1
- dynamic badspot handling, associated with, 8-15
- example, intelligent disk controllers, D-5
- FIX_DISK related, B-1
- format, intelligent disk controllers, D-4
- format, nonintelligent disk controllers, D-2
- MAKE related, A-1
- MAKE_ROBUST utility, E-1
- mirroring related, C-1
- mirroring, 9-16
- PHYRST and badspots, 2-7
- PRIMOS, free contiguous space, 7-26
- problem with disk, 6-6
- read- and write-errors, 6-6
- See also* Errors

MFDs, 1-4

- creation by MAKE, 5-1
- discussion of, 1-5
- files in, 1-5
- FIX_DISK and hashed, 6-3
- protection by MAKE, 5-2

MIDASPLUS on robust partitions, 7-12

Minimum extent size, discussion of, 5-15

MIRROR configuration directive, 9-3

- MIRROR_OFF command,
 - caution on use in RESUS, 9-8
 - format of, 9-10
 - options of, 9-10
 - MIRROR_ON command,
 - caution on use in RESUS, 9-8
 - configuration directives necessary for, 9-9
 - format of, 9-8
 - options of, 9-9
 - PRIMOS.COMI file, 9-8
 - warning on use of, 9-5
 - Mirroring,
 - AC mode controller, 9-14
 - ADDISK command example, 9-7
 - breaking, defined, 9-13, G-2
 - breaking, errors causing, 9-13
 - catch-up copy server operation,
 - warning on, 9-5
 - catch-up copy server, operation of, 9-5
 - catch-up copy server, startup of, 9-6
 - catch-up copy, 9-5
 - catch-up copy, caution on, 9-11
 - catch-up copy, failure of, 9-16
 - commands for, 9-8
 - configuration directives, 9-2, 9-3, 9-9
 - COPY_DISK or PHYSAV, use of, 9-5
 - defined, 9-1, G-3
 - directives, configuration, 9-2, 9-3
 - dual-ported disks, 9-9
 - error handling by PRIMOS, 9-13
 - error messages, C-1
 - errors associated with, 9-13
 - FIX_DISK -ADBADS option, 9-15
 - intelligent disk controllers, 5-14
 - messages, 9-16, C-1
 - paging partitions, 9-2
 - partial disk, caution on, 9-3
 - partitions, age of, 9-5
 - partitions, maximum number of, 9-2, 9-10
 - partitions, primary and secondary, 9-1
 - performance of, 9-3
 - physical problems and, 9-15
 - PRIMOS commands, 9-8
 - purpose of, 9-1
 - read errors, recovery for, 9-15
 - requirements for, 9-2
 - robust partitions, 7-31
 - SHUTDN command example, 9-7
 - shutdown stamp, 9-6
 - starting with a COMI file, 9-10
 - STATUS command example, 9-7
 - survivor, defined, 9-13
 - system halts, 9-16
 - uncorrectable read-error, 9-15
 - write errors, recovery for, 9-14
 - write-protect switch, action if set, 9-14
 - See also* Partitions
 - Modes,
 - AC and -IC, using MAKE and FIX_DISK, 8-3
 - AC, converting to with FIX_DISK, 8-13
 - badspot handling (-IC or -AC), default, 8-11
 - badspot handling (-IC), 6-13
 - conversion between -IC and -AC, 8-2
 - default -IC or -AC, 5-9
 - Dynamic Badspot Handling (-IC) and first partition, 5-3
 - Dynamic Badspot Handling (-IC) and pre-Rev. 20.0 partitions, 8-7
 - Dynamic Badspot Handling (-IC), 2-3, 5-9, 8-2
 - IC or -AC, selecting, 6-3
 - IC, converting to with FIX_DISK, 8-13
 - IC, converting to with MAKE, 5-14
 - mixing, caution on, 8-7
 - Nondynamic Badspot Handling (-AC), 2-3, 5-9, 8-2
 - switching and first partition, 6-14
 - MTRESUME command, for invoking FIX_DISK, 6-38
 - MTTR, defined, 7-14
- N**
- Nondynamic badspot handling mode,
 - defined, G-6
 - See also* Controllers; Disk controllers
 - Nonintelligent disk controllers,
 - defined, G-6
 - error handling by, D-1
 - status word, D-3
 - Numbers,
 - disk drive unit, 3-3
 - drive unit for FMDs, 3-4
 - drive unit for SMDs and CMDs, 3-3

Numbers, (continued)

maximum starting surface for pdev,
3-6

See also pdev

O

OCTAL function, converting ldev, 7-24

Online storage, 1-1

OPCODE, intelligent disk controller, D-6

Options,

FIX_DISK, available for, 6-9

MAKE, available for, 5-5

MAKE, required for, 5-27

sectoring, 5-19

ORACLE on robust partitions, 7-13

Order, record allocation, 10-3

P

Packs, disk,

defined, 2-2, G-3

See also Disk drives

Paging,

creating partition, example of, 5-35

creating partitions for, 5-21

defined, 1-2, G-6

device, mirroring at startup, 9-3

disks, 5-1

partition, mirroring, 9-2

See also Mirroring

PAGINM configuration directive, 9-3

example of use, 9-4

Partitioning,

defined, 1-2, G-7

disks, 5-1

specific disk types, 3-14

See also MAKE utility; MAKE
options

Partitions, 1-1

accessing Rev. 22.1, 1-7

accessing, 5-2

added, defined, G-1

assigned, defined, G-1

assigning, error message, 6-10

backup before MAKE, 5-27

booting from, 7-17

controller mode, setting, 6-13

conversion, need for, 8-5

converting to Rev. 21.0, 8-12

converting to Rev. 22.1, 5-3

converting to robust partitions, 7-19

creating pre-Rev. 21.0, 8-6

DBS and RMA on first, 5-3

defined, G-6

errors on, 7-3

first and mode switching, 6-14

fragmentation of, 7-23

fragmentation, criteria for determining,
7-25

mirror, age of, 9-5

mirror, shutdown stamp of, 9-6

mirroring at startup, 9-8

mirroring, maximum number for, 9-2,
9-10

name of, changing, 5-20

naming, 1-2, 5-20

paging, creating, 5-21

paging, creating, example of, 5-35

paging, defined, G-6

paging, mirroring of, 9-2

physical copy of, 2-5

primary, 9-1

primary, defined, G-7

secondary, 9-1

secondary, defined, G-8

shutting down, 4-3

source, 2-4

specific disk types, 3-14

split first, minimum records for files,
5-22

split, and user files, 5-23

split, defined, G-8

standard, allocation direction for, 10-5

standard, default allocation, 10-6

standard, defined, G-8

target, 2-4

See also First partition; Disks; Robust
partitions

Password directories, hashing, 1-7

Password,

changing partition's, 5-30

pdev,

basic for CMDs, 3-14

basic for CMDs, table of, 3-19

basic for FMDs and SMDs, table of,
3-8

binary breakdown of, 3-12

commands that use, 3-1

pdev, (continued)
 construction of, 3-6
 construction, example of, 3-13
 defined, 3-1, 3-12, G-7
 determining, example of, 3-9
 drives and controllers, example
 combinations of, 3-9
 example worksheet, 3-10
 examples, SMD, 3-11
 information needed for, 3-5
 legitimate, 3-5
 maximum starting surface number, 3-6
 odd number, 3-6
 partition address, 4-3
 Phantoms, logging out, 4-4
 PHYRST command,
 badspot handling by, 2-5
 dynamic badspot handling and, 8-9
 valid data transfers, 8-10
 PHYSAV command,
 badspot handling, 2-4
 caution on use of, 8-9
 dynamic badspot handling and, 8-9
 mirrored partitions, use of for, 9-5
 valid data transfers, 8-10
 Physical copy, 2-4
 Physical device number,
 See pdev
 Physical save, 2-4
 defined, G-7
 See also PHYSAV command
 PRIMENET, accessing partitions, 5-2
 PRIMOS II, 1-3
 PRIMOS,
 mirroring actions, 9-1
 revisions, features of, 5-4
 PRISAM on robust partitions, 7-13
 Procedures,
 assigning and unassigning disks, 4-4
 converting partitions to Rev. 22.1, 5-4,
 7-18
 converting to robust partitions, 7-19
 first partition, changing geometry of,
 8-8
 FIX_DISK flowchart, 6-8
 FIX_DISK, 6-7
 FIX_DISK, after running, 6-25
 FIX_DISK, before running, 6-7
 fragmented partitions, handling, 7-27
 MAKE utility, 5-27

MAKE, after running, 5-30
 MAKE, before running, 5-27
 MAKE, stopping, 5-6
 pdev, construction of, 3-10
 Using DLINFO to check disk controller
 download file, D-12
 Prompts, DBS file change, 8-6
 PROTECT argument, ADDISK command,
 7-30
 Protocols, ICOP, 2-3

Q

Quota system,
 check by FIX_DISK, 6-21
 disabling checking for pre-Rev. 19,
 6-24
 errors in, 6-3
 fast FIX_DISK, 6-21
 message to run FIX_DISK, 6-6

R

RAT, defined, G-7
 Read, mirroring operations, 9-2
 Record allocation, 5-18
 algorithm for, 10-1
 choosing method of, 10-7
 default, 10-5
 dependencies, 10-4
 direction of, 10-5
 discussion of, 10-1
 example of, 7-21, 10-2
 order, 10-3
 SAM and DAM files, 10-6
 setting direction of, 6-24
 See also Allocation
 Records,
 beginning address (BRA), 1-5
 contiguous, 10-2
 contiguous, free, 7-21
 defined, 3-2
 header information, 8-6
 initializing headers, 5-11
 initializing, 5-13
 minimum for paging partition files,
 5-21
 number, entering badspots by, 5-23
 read attempts by FIX_DISK, 6-12

Records, (continued)

- retrieval by intelligent controller, 10-3
- split first partition, recommended
 - minimum number for, 5-22
- table of allocation order, 10-4
- truncation of, 6-25

See also Sectors

Recoverability, understanding the concept,
7-5

Remapped areas,

See RMA

RESUS command, caution on use of
MIRROR_ON and MIRROR_OFF, 9-8

Rev. 20.0, features of, 5-4

Rev. 21.0, features of, 5-4

Rev. 22.0, features of, 5-4

Rev. 22.1, features of, 5-4

Revisions, features of, 5-4

RMA,

- 80% full, 8-11
- badspot remapping, 2-3
- badspots in, 8-8
- damage to, 8-8
- defined, G-7
- function of, 8-4
- location of, 8-5
- sizes of, 5-22

Robust Partitions Evaluation Form, 7-15,
7-17

Robust partitions,

- access to, 7-4
- advantages of, 7-2
- boot procedure, 7-3
- candidates for conversion to, 7-17
- converting to, 6-4, 7-19
- creating, 5-20, 7-19
- criteria for, 7-17
- data management packages on, 7-11
- default allocation, 10-5
- defined, 7-1, G-8
- determining if a partition is robust,
7-7
- directories on, 5-2, 7-10
- directory structure on, 7-4
- evaluating use of, 7-14
- evaluation form, 7-17
- file administration, 7-20
- file organization, 7-7
- file system objects, 7-8
- file system performance, 7-11

file truncation and deletion, 7-11

files inappropriate for, 7-14

FIX_DISK, when to run, 7-27

halts and fast FIX_DISK, G-10

logical file typing, 7-7

mirroring, 7-31

questionnaire, 7-14

record allocation on, 6-25

restrictions on use of, 7-3

sectoring on, 7-4

sectoring, 10-4

segment directories on, 7-10

space needed for conversion to, 7-17

space needed for files on, 7-4

See also Partitions

S

SAM files, 1-8

operation of FIX_DISK on, 7-6

record allocation, 10-6

structure, 1-8

Sectoring,

default for controller/CPU combination,
5-19

forward, 10-2

forward, defined, G-5

options, meaning of, 5-19

recommendations for, 10-3

reverse, 10-2

reverse, defined, G-7

setting forward or reverse, 5-19, 6-24

setting on pre-Rev. 22.1 partitions, 10-6

specifying, 5-18

Sectors,

conversion of byte numbers to, 5-24

defined, G-8

entering badspots by, 5-24

See also Records; Directories,
directories

Segment directories, 1-7, 7-6

discussion of, 7-10

Servers,

catch-up, operation of, 9-5

copy, warning on use of, 9-5

SHUTDOWN command, 4-1

FIX_DISK, use with, 6-7

mirroring, 9-7

renaming partitions, 5-20

Shutdown stamp, mirrored partitions, 9-6

Size,
 default for extents, 5-15, 6-24
 maximum and minimum extent,
 discussion of, 5-15

SMDs,
 defined, G-8
 disk type, 2-2
 drive unit numbers for, 3-3
 partitioning 300MB, 3-15
 partitioning 80MB, 3-14
 See also Disks

Source partition, 2-4
 badspots on, 2-6

Stamp, shutdown, of mirrored partitions,
 9-6

Standalone, defined, G-8

Standard partition,
 See Partitions,

START_DSM command, 4-4
 running FIX_DISK on COMDEV, 6-16

START_LSR command, 4-4
 running FIX_DISK on COMDEV, 6-16

Startup,
 mirroring, 9-3
 system, and mirroring, 9-8

Static badspot handling,
 See Badspots

STATUS DISKS command,
 and mirroring, 9-6
 determining pdev for FIX_DISK, 6-9

Status words,
 intelligent disk controller, logical, D-7,
 D-8
 intelligent disk controller, physical,
 D-9, D-10
 nonintelligent disk controller, D-3

Status, disk controller LED, D-11

STOP_DSM command, 4-4
 running FIX_DISK on COMDEV, 6-16

STOP_LSR command, 4-4
 running FIX_DISK on COMDEV, 6-16

Storage Module Disks,
 See SMDs; Disks

Subfiles,
 See Files; Segment directories

Subroutines,
 CAM file management, 7-9
 CF\$SME, for CAM files, 7-21

Surfaces,
 changing number of, 8-8

 defined, G-9
 offset, 3-3

Survivor, mirroring, defined, 9-13

Switches, write-protect and mirroring,
 9-14

Syntax,
 FIX_DISK, summary of, F-5
 MAKE, summary of, F-1

System,
 availability, 9-1
 halts and mirroring, 9-16

T

Tables,
 conversion of byte numbers to sectors,
 5-24
 default record allocation, 10-5
 disk size data, 3-2
 disk types and arguments, 5-8
 extent sizes, default maximum and
 minimum, 5-15, 6-24
 FIX_DISK, recommendations for using,
 6-5, 7-29
 MAKE and FIX_DISK, using for mode
 (-AC and -IC) switching, 8-3
 MAKE options, use of with flaw
 maps, 5-18
 pdev, additions to, 3-7
 pdev, basic and adjusted example, 3-12
 pdev, basic for CMDs, 3-19
 pdev, basic for SMDs and FMDs, 3-8
 pdev, examples of, 3-9
 PRIMOS revisions, features of, 5-4
 record allocation, 10-4
 records, number recommended for files
 in split first partition, 5-22
 sectoring options, 5-19

Tapes,
 booting MAKE from, 5-40
 invoking FIX_DISK from, 6-38

Target partition, 2-4
 badspots on, 2-6

Tracks, entering badspots by, 5-24

Transaction recovery, 7-17

Tree structure,
 directory, 1-5
 file, 1-6

U

UFD,
 See Directories
UNASSIGN DISK command, 4-5
Unit number, disk drive,
 discussion of, 3-3
 table of, 3-7
User disks, 5-1

V

Volume, 1-1
 defined, G-9

W

Warm starts, problem with adding disks,
 9-7
Warnings,
 dual-ported disks and priority selecting,
 9-9
 FIX_DISK and hardware problems, 6-5
 FIXRAT, use of, 1-3
 initialization of file system, 5-13
 MAKE, use of -FORMAT option, 5-12
 MAKE, use of pre-Rev. 22.1 version on
 first partition, 5-3
 MAKE, use on first partition, 8-8
 mirroring, copy server operation, 9-5
 pdev, changing basic of first partition,
 5-7
Winchester disks,
 See FMDs; Disks
Write, mirroring operations, 9-1

SURVEYS

READER RESPONSE FORM

Operator's Guide to File System Maintenance DOC9300-5LA

Your feedback will help us continue to improve the quality, accuracy, and organization of our 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*

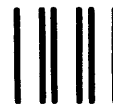
5. Which other companies' manuals have you read?

Name: _____ Position: _____

Company: _____

Address: _____

_____ Postal Code: _____



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

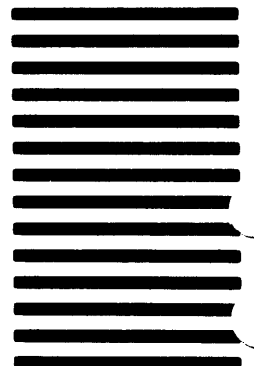
First Class Permit #531 Natick, Massachusetts 01760

BUSINESS REPLY MAIL

Postage will be paid by:



Attention: Technical Publications
Bldg 10
Prime Park, Natick, Ma. 01760



READER RESPONSE FORM

Operator's Guide to File System Maintenance DOC9300-5LA

Your feedback will help us continue to improve the quality, accuracy, and organization of our 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: _____



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

First Class Permit #531 Natick, Massachusetts 01760

BUSINESS REPLY MAIL

Postage will be paid by:



PrimeTM

Attention: Technical Publications
Bldg 10
Prime Park, Natick, Ma. 01760



READER RESPONSE FORM

Operator's Guide to File System Maintenance DOC9300-5LA

Your feedback will help us continue to improve the quality, accuracy, and organization of our 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: _____

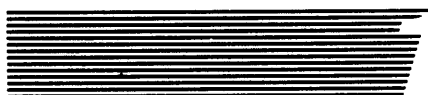
Address: _____

Postal Code: _____

First Class Permit #531 Natick, Massachusetts 01760

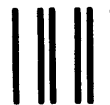
BUSINESS REPLY MAIL

Postage will be paid by:



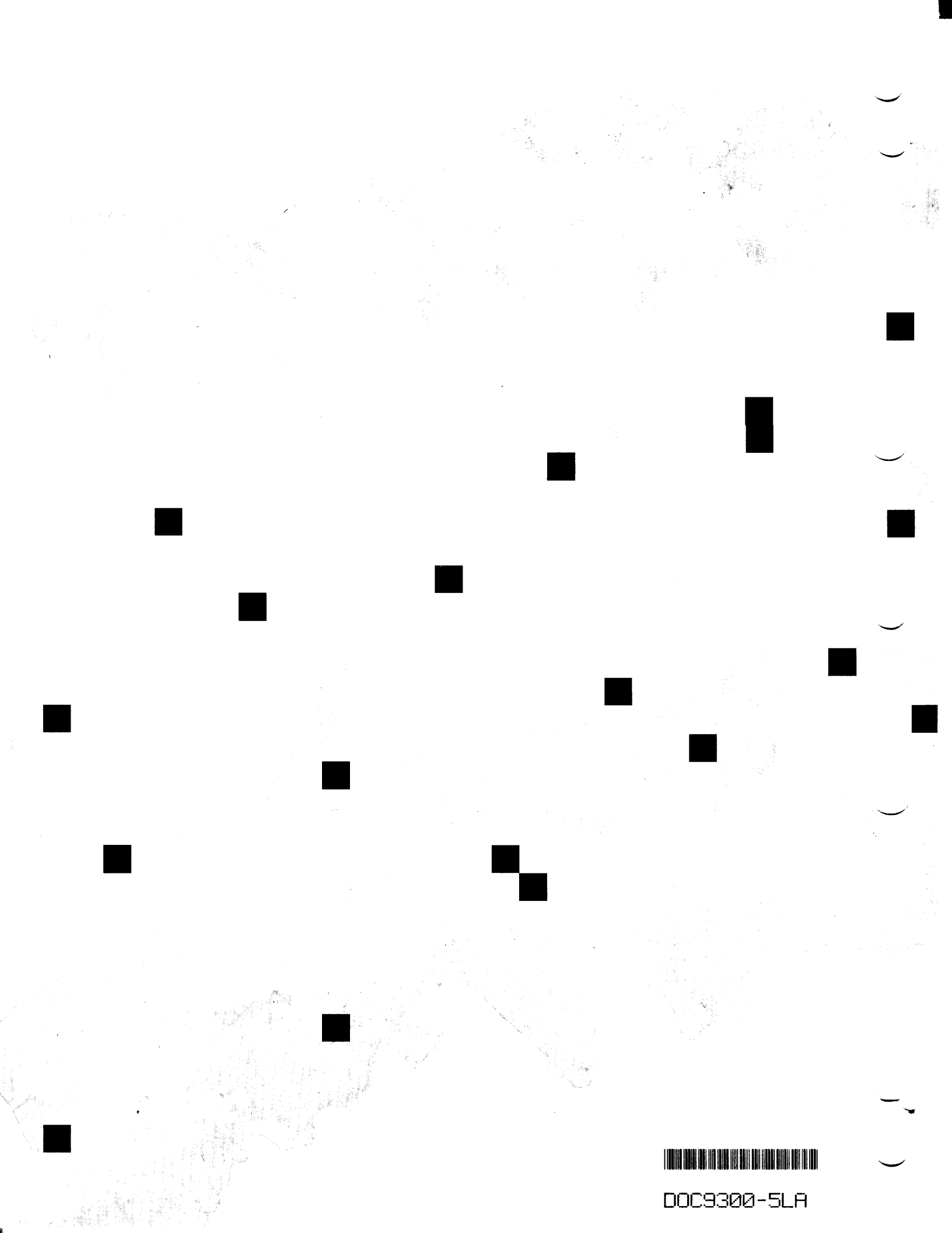
PrimeTM

Attention: Technical Publications
Bldg 10
Prime Park, Natick, Ma. 01760



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES





DOC9300-5LA