



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

## **Operator's Guide to Data Backup and Recovery**

**Revision 23.0**

**DOC10324-1LA**

---

# OPERATOR'S GUIDE TO DATA BACKUP AND RECOVERY

---

*First Edition*

**by Celia Bartlett**

*This guide documents the Data Backup & Recovery utilities  
MAGSAVIRST and Physical Save and Restore implemented as an  
Independent Product Release at Revision Levels 21.0 and 22.0 and as  
a Master Disk Release at Revision 23.0*

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 © 1990 by Prime Computer, Inc. All rights reserved.

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

ORACLE is a registered trademark of Oracle Corp.

This document was prepared in the United Kingdom by Technical Publications Department, 4 Bramley Road, Mount Farm, Milton Keynes, MK1 1PT, United Kingdom.

## **PRINTING HISTORY**

First Edition (DOC10324-1LA) June 1990

## **CREDITS**

Design: Richard Merry-West

Editorial: Fiona Carey

Project Support: Vaughn Meads and Stephen Randles

Production: Prime Technical Publications production unit

# CONTENTS

## ABOUT THIS BOOK

**Chapter Contents**

**Superseded Products**

**Related Documentation**

**Prime Documentation Conventions**

## PART I ENHANCED MAGSAV/RST

<b>1</b>	<b>OVERVIEW OF ENHANCED MAGSAV/RST</b>	<b>1-1</b>
	Introduction	1-1
	Major Features	1-1
	Command Line Options	1-5
	Tape Labels and Blocks	1-6
	Product Compatibility	1-7
<b>2</b>	<b>LOGICAL SAVE TO TAPE: MAGSAV</b>	<b>2-1</b>
	Introduction	2-1
	Overview	2-1
	Incremental Saves	2-2
	Command Line Options	2-3
	The MAGSAV Dialog	2-6
	Listing the Tape Contents	2-11
	Save Procedure for an Entire Partition	2-11
	Unattended Backup	2-13
<b>3</b>	<b>LOGICAL RESTORE FROM TAPE: MAGRST</b>	<b>3-1</b>
	Introduction	3-1

	<b>Overview</b>	<b>3-1</b>
	<b>Restoring Files, Directories, and Partitions</b>	<b>3-1</b>
	<b>File Security</b>	<b>3-2</b>
	<b>Command Line Options</b>	<b>3-3</b>
	<b>The MAGRST Dialog</b>	<b>3-4</b>
	<b>Procedure for a Partial Restore</b>	<b>3-7</b>
	<b>Procedure for a Checkpoint Restore</b>	<b>3-8</b>
	<b>Restore Procedure for an Entire Partition</b>	<b>3-8</b>
<b>4</b>	<b>CREATING AND USING A BOOT TAPE</b>	<b>4-1</b>
	<b>Introduction</b>	<b>4-1</b>
	<b>Overview</b>	<b>4-1</b>
	<b>Creating a List File</b>	<b>4-2</b>
	<b>The BOOT_CREATE Command</b>	<b>4-3</b>
	<b>Creating a System Boot Tape</b>	<b>4-4</b>
	<b>Booting from Tape</b>	<b>4-6</b>
<b>5</b>	<b>TAPE INDEXING</b>	<b>5-1</b>
	<b>Introduction</b>	<b>5-1</b>
	<b>Overview</b>	<b>5-1</b>
	<b>Summary Index Format</b>	<b>5-1</b>
	<b>Extended Index Format</b>	<b>5-2</b>
	<b>Examples of Indexes</b>	<b>5-3</b>
	<b>PART II</b>	
	<b>PHYSICAL SAVE &amp; RESTORE</b>	
<b>6</b>	<b>PHYSICAL SAVE TO TAPE: PSR</b>	<b>6-1</b>
	<b>Introduction</b>	<b>6-1</b>
	<b>Overview</b>	<b>6-1</b>
	<b>Invoking PSR</b>	<b>6-6</b>
	<b>Save Procedure for an Entire Partition</b>	<b>6-11</b>
	<b>Example Command Lines and Dialogs</b>	<b>6-12</b>
	<b>Unattended Backup</b>	<b>6-16</b>
	<b>Maximizing Performance</b>	<b>6-18</b>
<b>7</b>	<b>RESTORING A PHYSICAL SAVE: PSR</b>	<b>7-1</b>
	<b>Introduction</b>	<b>7-1</b>
	<b>Overview of PSR Restore</b>	<b>7-1</b>
	<b>Invoking PSR</b>	<b>7-5</b>

	<b>Restore Procedure for an Entire Partition</b>	<b>7-7</b>
	<b>Example Command Lines and Dialogs</b>	<b>7-9</b>
	<b>Maximizing Performance</b>	<b>7-10</b>
<b>8</b>	<b>PHYSICAL COPY BETWEEN DISKS</b>	<b>8-1</b>
	<b>Introduction</b>	<b>8-1</b>
	<b>Overview of PSR Copy</b>	<b>8-1</b>
	<b>Invoking PSR</b>	<b>8-4</b>
	<b>Copy Procedure for an Entire Partition</b>	<b>8-6</b>
	<b>Example Command Lines and Dialogs</b>	<b>8-7</b>
	<b>Maximizing Performance</b>	<b>8-8</b>
	<b>PART III</b>	
	<b>SYSTEM BACKUPS:</b>	
	<b>ADMINISTRATION</b>	
<b>9</b>	<b>DEFINING A BACKUP STRATEGY</b>	<b>9-1</b>
	<b>Introduction</b>	<b>9-1</b>
	<b>The Need for Backups</b>	<b>9-1</b>
	<b>Prime Backup Utilities</b>	<b>9-2</b>
	<b>Logical Backup Utility MAGSAV/RST</b>	<b>9-2</b>
	<b>Physical Backup Utility PSR</b>	<b>9-3</b>
	<b>Disk Mirroring</b>	<b>9-4</b>
	<b>Maximizing Backup Performance</b>	<b>9-5</b>
	<b>Retaining Backups</b>	<b>9-6</b>
	<b>Logging Backups</b>	<b>9-7</b>
	<b>Selecting a Backup Strategy</b>	<b>9-7</b>
<b>10</b>	<b>SYSTEM PREPARATION, SHUTDOWN, AND STARTUP</b>	<b>10-1</b>
	<b>Introduction</b>	<b>10-1</b>
	<b>Preliminary Preparations</b>	<b>10-1</b>
	<b>System Preparation</b>	<b>10-3</b>
	<b>Shutdown Procedures</b>	<b>10-3</b>
	<b>Startup Procedures</b>	<b>10-6</b>
	<b>APPENDIX</b>	
<b>A</b>	<b>GLOSSARY</b>	<b>A-1</b>
	<b>INDEX</b>	<b>Index 1</b>

## ABOUT THIS BOOK

The *Operator's Guide to Data Backup and Recovery* describes the features of the products MAGSAV, MAGRST and PSR at PRIMOS Revisions 21.0, 22.0 and 23.0, and gives comprehensive guidance on their use.

Enhanced MAGSAV/RST is issued as an independent product release (IPR) which runs on PRIMOS Revs. 21.0 and 22.0, and as a standard product implemented at Master Disk Rev. 23.0. Although the enhanced product is not supported on earlier PRIMOS revisions, you can use it to transfer data between systems running on revisions later than and including PRIMOS Rev. 19.0.

PSR is issued as an independent product release (IPR) which runs on PRIMOS Rev. 22.0, and as a standard product implemented at Master Disk Revision Level 23.0.

### Chapter Contents

#### **PART I            Enhanced MAGSAV/RST**

- |           |  |
|-----------|--|
| Chapter 1 | Describes the features of enhanced MAGSAV/RST. The advantages and capabilities of the product are discussed, as well as aspects that you should be aware of when using it. |
| Chapter 2 | Describes how to perform a logical save to tape using enhanced MAGSAV.   |
| Chapter 3 | Describes how to restore a logical tape save using enhanced MAGRST.  |
| Chapter 4 | Describes how to create and use a boot tape.   |
| Chapter 5 | Gives details on indexing the contents of tapes saved using MAGSAV.  |

#### **PART II            Physical Save and Restore**

- |           |   |
|-----------|---|
| Chapter 6 | Describes how to perform a physical save to tape using PSR. |
| Chapter 7 | Describes how to restore a physical tape save using PSR.    |

Chapter 8	Describes how to perform a physical copy between disks using PSR.
<b>PART III</b>	<b>System Backups: Administration</b>
Chapter 9	Describes the factors you should consider when defining a backup strategy.
Chapter 10	Gives step-by-step procedures to shut down and restart either the entire system or particular partitions before and after running MAGSAV/RST and PSR.
Glossary	Defines the terminology used in this guide.

## **Superseded Products**

Enhanced MAGSAV/RST supersedes the logical save and restore utilities available as standard products on PRIMOS Rev. 22.0 and earlier revisions.

PSR supersedes the physical save and restore utilities available as standard products on PRIMOS Rev. 22.0 and earlier revisions.

Manuals which refer to these earlier utilities are superseded by this guide.

## **Related Documentation**

Listed below are the manuals that contain information that may be useful to users of save and restore utilities.

- The PRIME CPU Handbook for your machine
- *MAGNET User's Guide* DOC10156-1LA (Rev. 21.0) and its update package UPD10156-11A (Rev. 22.0)
- *Operator's Guide to File System Maintenance* DOC9300-3LA (Rev. 21.0), DOC9300-4LA (Rev. 22.0), and DOC9300-6LA (Rev. 23.0)
- *Operator's Guide to System Commands* DOC9304-3LA (Rev. 21.0), DOC9304-4LA (Rev. 22.0), and DOC9304-5LA (Rev. 23.0)
- *Operator's Master Index* DOC10110-3LA (Rev. 21.0), DOC10110-4LA (Rev. 22.0), and DOC10110-5LA (Rev. 23.0)
- *Prime User's Guide* DOC4130-4LA, and its update packages UPD4130-41A, UPD4130-42A (Rev. 21.0) and *PRIMOS User's Guide* DOC4130-5LA (Rev. 22.0)
- *PRIMOS Commands Reference Guide* DOC3108-6LA (Rev. 21.0), DOC3108-7LA (Rev. 22.0)
- *System Administrator's Guide Vol. I* DOC10131-1LA (Rev. 21.0), DOC10131-2LA (Rev. 22.0), and DOC10131-3LA (Rev. 23.0)
- *System Administrator's Guide Vol. II* DOC10132-1LA (Rev. 21.0), DOC10132-2LA (Rev. 22.0), and RLN10132-21A (Rev. 23.0)]



## OPERATOR'S GUIDE TO DATA BACKUP AND RECOVERY

- *System Administrator's Guide Vol. III DOC10133-1LA* (Rev. 21.0), *DOC10133-2LA* (Rev. 22.0), and *DOC10133-3LA* (Rev. 23.0)
- *Rev. 23.0 Software Release Document DOC10001-7PA* (Rev. 23.0)

### Prime Documentation Conventions

The following conventions are used in command formats, statement formats, and in examples throughout this document. Command and statement formats show the syntax of commands, program language statements, and callable routines. Examples illustrate the uses of these commands, statements, and routines in typical applications. Terminal input may be entered in either upper case or lower case.

<i>Convention</i>	<i>Explanation</i>	<i>Example</i>
UPPERCASE	In command formats, words in uppercase indicate the actual names of commands, statements, and key words. They can be entered in either uppercase or lowercase.	MAGSAV
lowercase	In command formats, words in lowercase represent items for which you must substitute a value.	COPY pathname
Abbreviations in format statements	If an uppercase word in a command format has an abbreviation it is shown below.	{ ASSIGN } { AS }
<u>Underscore</u> in examples	In examples, user input is underscored but system prompts and output are not.	OK, <u>PROP -STATUS</u>
Brackets [ ]	Brackets enclose a list of one or more optional items. Choose none, one or more of these.	INDEX [pathname]
Braces { }	Braces enclose a list of optional items. Choose only one of these.	SPEED { -25 } { -100 }
Ellipsis ...	An ellipsis indicates that the preceding item may be repeated.	item-x [, item-y ] ...
Hyphen -	Wherever a hyphen appears in a command line option it is a required part of that option.	MAGSAV -INC
Angle Brackets < >	Angle brackets are used to separate the elements of a pathname.	< MFD > SPOOL* > DOC
Octal ' or ^	The ' or ^ symbol shows that the number is given as an octal value.	^240 or ^240 are equivalent to 160 (base 10)

---

**PART I**  
**ENHANCED MAGSAV/RST**

---

# OVERVIEW OF ENHANCED MAGSAV/RST

## Introduction

This chapter describes some of the more important features of Enhanced MAGSAV/RST. Enhanced MAGSAV/RST is available as an Independent Product Release (IPR) at PRIMOS Revs. 21.0 and 22.0, and as a standard product implemented at Master Disk Rev. 23.0.

## Major Features

Major features of Enhanced MAGSAV/RST are listed below; they are described in the sections which follow:

- Tape handling
- Tape format
- Support of attributes DTA (date/time accessed) and DTC (date/time created)
- Use of DTB (date/time backed up) for incremental control
- Support of labeled tapes
- Handling of unrecoverable read and write errors
- Extended index files
- Index library management
- Unattended backup

## Tape Handling

Enhanced MAGSAV/RST makes more efficient use of tape, and handles files faster than pre-Rev. 23.0 standard logical save/restore utilities. The increase in throughput when saving files depends on the size and type of file saved, and on the CPU type and disk devices that you are using.

Enhanced MAGSAV writes checkpoints to tape at intervals of 500 blocks, or, if you are using a 4595 helical scan tape drive, at intervals of 10,000 blocks. You can obtain information

about the checkpoints on tape by generating an extended index of the save/restore operation; details on extended indexes are given in Chapter 5, Tape Indexing.

If an unrecoverable error occurs during a save operation, the system rewinds the tape to the last checkpoint and writes end-of-tape labels; the save then continues on a new reel from this point. Further details on this are given in the section, Handling of Unrecoverable Read and Write Errors, later in this chapter.

When you use enhanced MAGRST to restore specific file objects from tape, you can use checkpoint positioning to locate the required object on tape by specifying the number of the checkpoint which immediately precedes the object. The system then positions the tape to this checkpoint before searching for the object. Enable checkpoint positioning by entering the appropriate response during the MAGRST dialog, and then specify the checkpoint number associated with each file system object you wish to restore. Details of this are given in Chapter 3, Logical Restore from Tape: MAGRST.

Checkpoint positioning considerably speeds up tape operations if you are using the Model 4595 helical scan tape drive, where one tape can hold up to 2GBytes of data.

You can also use checkpoint positioning on standard half-inch reel to reel tape drive hardware, but the improvement in performance is not as great. In this case, the speed of tape movement during operations such as moving forward to the next checkpoint is similar to that during a read operation. However, tape movement is slightly faster using checkpoint positioning due to the the fact that no data transfer takes place during the tape positioning operation.

### **Tape Format**

The tape format generated by enhanced MAGSAV provides the following benefits:

- An ANSI label is written at the start of each reel. This label conforms to ANSI standard X3.27-1987 and can be read on any system that supports this standard.
- You can use larger block sizes when saving to tape. This results in faster tape handling and reduces the amount of tape required. Enhanced MAGSAV writes 12K byte blocks by default. If you are using enhanced MAGSAV at PRIMOS Rev. 22.0 or at a later revision, you can select a tape block size of 12K bytes or 16K bytes.
- The block structure on tape saves time and tape when running save and restore operations.

Further details on tape format are given in the section, Tape Labels and Block Structure, later in this chapter.

### **Support of Attributes DTA and DTC**

When you save files or directories to tape using enhanced MAGSAV, the attributes DTA (date/time accessed), DTC (date/time created) and DTM (date/time modified) are automatically saved to tape.

When you restore files or directories from tape using enhanced MAGRST, the option -NO\_ATTRIBUTES allows you to choose whether or not to restore the saved values of DTA, DTC, and DTM from tape. Details on using this option are given in Chapter 4, Logical Restore from Tape: MAGRST.

You can use the enhanced MAGSAV option `-NO_DTA` to prevent the system from automatically updating the DTA of saved disk objects. Details on using this option are given in Chapter 2, Logical Save to Tape: MAGSAV.

### Use of DTB for Incremental Control

During an incremental save, only those files are saved that have been modified since the last base or incremental save. The standard pre-Rev. 23.0 MAGSAV product used the dumped bit switch setting to determine whether or not the file had been modified or created since the last save. Enhanced MAGSAV makes use of the attributes DTB and DTM for incremental control. These attributes give the date and time that the object was last backed up or modified. MAGSAV compares the value of DTB and that of DTM, and saves the object if the DTM is later than the DTB. When you use the option `-UPDT`, enhanced MAGSAV sets the DTB of saved files to the current date and time, but does not set the file's dumped bit, except in the case of Recovery Based Files, which are described below. Details on incremental saves are given in Chapter 2, Logical Save to Tape: MAGSAV.

When saving Recovery Based Files (RBFs), MAGSAV calls Recovery Oriented Access Method (ROAM) subroutines. At PRIMOS Revs. 21.0 and 22.0, these subroutines rely on the setting of the dumped bit to determine whether or not to save these files during an incremental save. Therefore, if you are using enhanced MAGSAV at PRIMOS Rev. 21.0 or 22.0 to save RBFs with the `-UPDT` option, the system updates the DTB and also sets the dumped bit of saved RBFs.

### Support of Labeled Tapes

When you save a tape using enhanced MAGSAV, the system automatically writes an ANSI label at the start of the reel of tape. You can specify the volume serial ID for the ANSI label using the enhanced MAGSAV option `-LABEL`. This means that each reel of tape contains an identifier on tape, as well as on the adhesive label. This helps considerably in the administration of large tape libraries. Details on this option are given in Chapter 2, Logical Save to Tape: MAGSAV.

### Handling of Unrecoverable Read and Write Errors

An unrecoverable read error occurs when the system fails to read a tape block after 20 attempts. The block is lost completely and the system skips past it to read the remaining data. If the block lost is a directory block, enhanced MAGRST recognizes this, and recreates the missing directory (with default security and attribute settings), so that subsequent tape files are restored at the correct level.

An unrecoverable write error occurs when the system fails to write a tape block after 20 attempts. You cannot write any more data to that reel of tape. When an unrecoverable error occurs, the system rewinds the tape to the last checkpoint, and overwrites the existing data with end-of-tape (EOT) labels. MAGSAV then prompts you to mount another tape and continues the save on the newly mounted reel from the point at which the previous reel was truncated.

**Note**

If you are using a 60 MB cartridge drive (QIC-02), the system cannot truncate the tape as described above. These drives allow you to append data to the tape only, and the system cannot backspace and overwrite previously written data with EOT labels. If the system encounters an error when writing to one of these drives, you are prompted to mount another tape and the system then resaves all the data written to the errored reel on the newly mounted reel. Details on saving cartridge tapes, are given in Chapter 2, Logical Save to Tape: MAGSAV.

**Extended Indexes**

When using enhanced MAGSAV/RST, you can generate extended indexes of save and restore operations in addition to the summary indexes available using standard pre-Rev. 23.0 logical save/restore utilities. The extended index includes tape checkpoint information and the value of the DTM attribute for each file system object that is saved or restored. You enable extended indexing by entering the appropriate response during the MAGSAV and MAGRST dialog. Further details on extended indexes are given in Chapter 5, Tape Indexing.

**Unattended backup**

When run under CPL control, enhanced MAGSAV sets three local variables which can be used to help eliminate the necessity for operator intervention in backup and restore operations. Details are given in Chapter 2, Logical Save to Tape: MAGSAV, and Chapter 3, Logical Restore from Tape: MAGRST.

## Command Line Options

The command line options supported by enhanced MAGSAV/RST are listed below. Also listed are those options that were supported by the standard pre-Rev. 23.0 logical save/restore product, but which have been withdrawn from the enhanced product.

### MAGSAV Options

Enhanced MAGSAV supports the following command line options:

<b>-BIG</b>	Writes a tape with 16K tape blocks.
<b>-BOOT</b>	Creates a boot tape.
<b>-HELP</b>	Displays the MAGSAV syntax and options.
<b>-INC</b>	Specifies an incremental save.
<b>-LABEL</b>	Invokes a prompt to enter volume serial IDs for any tape that is encountered during a save and does not contain an ANSI label.
<b>-NO_ACL</b>	Instructs MAGSAV not to save file access information.
<b>-NO_DTA</b>	Instructs MAGSAV not to modify the DTA of any file or directory being saved.
<b>-NO_RBF</b>	Instructs MAGSAV not to save Recovery Based Files.
<b>-QUERY</b>	Asks for confirmation before overwriting existing logical tapes.
<b>-REV19</b>	Writes tapes in a non-ANSI format for PRIMOS Revisions 19.0 to 19.4.14.
<b>-REV20</b>	Writes tapes in a non-ANSI format for PRIMOS Revisions 20, 21 and 22.
<b>-SAVE_UFD</b>	Instructs MAGSAV to save all directory entries, modified or not.
<b>-TTY</b>	Prompts for the entry of a tape unit number.
<b>-UNLOAD</b>	Instructs MAGSAV to rewind and unload tape.
<b>-UPDT</b>	Instructs MAGSAV to update the DTB of any saved file and directory.

The full description of the use of these options is described in Chapter 2, Logical Save to Tape: MAGSAV.

Enhanced MAGSAV does not support the following options:

<b>-P300</b>	Writes a tape readable by a P300 system.
<b>-7TRK</b>	Specifies that a 7-track tape is to be saved.
<b>-CAM_TO_DAM</b>	Converts CAM files to DAM files so that they can be restored on pre-Rev. 20 systems. The option <b>-REV19</b> automatically does this conversion for you.

If you use any of these options, an error message is displayed.

The options **-LONG** and **-VAR** available on PRIMOS Rev. 18.0 are now obsolete; if you attempt to use either of these options, a warning message is displayed.

### MAGRST Options

Enhanced MAGRST supports the following command line options:

## **OPERATOR'S GUIDE TO DATA BACKUP AND RECOVERY**

<b>-CAM_RBF</b>	Restores all RBF subfiles as CAM files.
<b>-HELP</b>	Displays MAGRST syntax and options.
<b>-QUERY</b>	Prompts for responses when restoring RBFs or when restoring an EPF currently in use.
<b>-NO_ATTRIBUTES</b>	Disables the restoring of DTA/DTC/DTM from tape.
<b>-TTY</b>	Causes a prompt for the entry of a tape unit number from the terminal.
<b>-UNLOAD</b>	Instruct MAGRST to rewind and unload tape.

The use of these options is described in Chapter 2, Logical Restore from Tape: MAGRST. Enhanced MAGRST does not support the following options:

<b>-7TRK</b>	Specifies that a 7-track tape is to be restored.
<b>-DAM_RBF</b>	Restores all RBF subfiles as DAM files.

If you use either of these options, an error message is displayed.

## **Tape Labels and Blocks**

The format of tapes generated by enhanced MAGSAV incorporates features that improve both the efficiency and performance of the utility. These features are summarized below and are discussed in the following sections:

- ANSI labels
- Block sizes
- Block structure

### **ANSI Labels**

Enhanced MAGSAV writes an ANSI label at the start of each reel of tape. This label conforms to ANSI standard X3.27-1987 and can be read on any system that supports this standard.

### **Block Sizes**

Enhanced MAGSAV writes tape blocks of 12K bytes by default. However, if you are using enhanced MAGSAV on systems running on PRIMOS Rev. 22.0 or a later revision you can use block sizes of 12K bytes or 16K bytes. Writing larger blocks to tape results in faster tape handling and reduces the amount of tape used when you save large files.

To use a block size of 16K bytes, you must first use the MTRS configuration directive to set the buffer size in the system configuration file to 16K or greater and then use the option **-BIG** when you invoke MAGSAV. You cannot write tape blocks larger than 16K bytes. If you do not use the MAGSAV option **-BIG**, the system defaults to writing 12K byte tape blocks irrespective of the MTRS setting in the system configuration file.



**Note**

You cannot use 16K byte tape blocks when running on systems using PRIMOS Rev. 21 or earlier, because the MTRS directive is not supported on these PRIMOS revisions.

For details of the MTRS directive, refer to the *System Administrator's Guide, Vol. I: System Configuration*.

---

**Caution**

If you create a tape using 16K byte blocks, this tape cannot be read on systems where the buffer size is less than 16K bytes. Tapes with 16K byte blocks cannot, therefore, be read on systems running on PRIMOS revisions earlier than PRIMOS Rev. 22.0, or on systems running on PRIMOS Rev. 22.0 or a later revision where the buffer size has not been set to 16K bytes (or greater) using MTRS

---

**Block Structure**

The block structure used by enhanced MAGSAV enables the system to handle tapes faster and more efficiently than previous versions of MAGSAV, especially when saving small files. When writing a file to tape, the system writes the pathname, data, and end-of-file (EOF) to tape without intervening inter-block gaps. Several files may be written to the same tape block, if their size allows.

There is a trade-off between performance and resistance to error. Packing more than one file into a single tape block does mean that unrecoverable read errors are more damaging; the loss of one tape block could potentially mean the loss of several small files. If there is an increased risk of tape errors occurring, you can still write the tape in non-ANSI format, by invoking enhanced MAGSAV with one of the command line options -REV20 or -BOOT.

**Product Compatibility**

Enhanced MAGSAV/RST minimizes any compatibility problems when transporting tapes between systems that use different PRIMOS revisions. Points that you should be aware of when you use the enhanced product with systems running on different revisions are described in the following sections.

**Enhanced MAGSAV/RST and PRIMOS**

You can implement all features of enhanced MAGSAV/RST when using systems running on PRIMOS Rev. 22.0 or a later revision. If you are running the product on PRIMOS Rev. 21.0, you can implement all features but MAGSAV writes tape blocks of 12K bytes only.

---

**Caution**

If you create a tape using 16K byte blocks, this tape cannot be read on systems where the buffer size is less than 16K bytes. Tapes with 16K byte blocks cannot, therefore, be read on systems running on PRIMOS revisions earlier than Rev. 22.0, or on systems running on PRIMOS Rev. 22.0 or a later revision, where the buffer size has not been set to 16K bytes (or greater) using MTRS.

---

### **Transporting Tapes and PRIMOS**

You can use enhanced MAGSAV to write tapes in non-ANSI format that can be read on pre-Rev. 21.0 systems. The MAGSAV options that allow you to do this are as follows:

- **-REV19:** writes a MAGSAV tape in non-ANSI format, readable on systems running on PRIMOS Revs. 19.0 to 19.4.14. Any CAM files saved are automatically converted to DAM files. (CAM files were introduced at PRIMOS Rev. 20.0).
- **-REV20:** writes a MAGSAV tape in non-ANSI format, readable on systems running on PRIMOS Revs. 20, 21, and 22.

If you use any of the above options, you cannot use the MAGSAV options **-LABEL**, or **-QUERY**, since these options rely on the tape being in ANSI format.

If you do not specify either of the options **-REV19** or **-REV20**, enhanced MAGSAV defaults to writing ANSI format tapes.

When you use enhanced MAGRST to restore a tape, you do not need to specify the format of the tape. MAGRST automatically determines the tape format for you, by reading the tape label.

### **Notes**

You cannot mix tape formats on the same reel of tape. If you try to do this, an error message is displayed.

Enhanced MAGRST will not restore BRMS tapes. These tapes must be read using BRMS restore commands (See the *Data Backup and Recovery Guide*).

### **Incremental Saves and PRIMOS**

Disks created on systems using PRIMOS revisions earlier than Rev. 19.3 do not support the DTB attribute. You cannot therefore use enhanced MAGSAV to perform incremental saves of these partitions.

## LOGICAL SAVE TO TAPE: MAGSAV

### Introduction

This chapter gives comprehensive details on the use of enhanced MAGSAV. Incremental saves and saves of entire partitions are described. Explanations are given of the command line options, and of the system prompts and user responses.

### Overview

MAGSAV is a utility that you use to save files and directories from disk to tape, so that each object is stored on tape as a logical entity, instead of its component records being stored in separate areas of the storage medium, as they might be on disk. When you use a physical save utility, the partition has the same format on tape as it had on disk. A logical save is slower than a physical save but does enable you to restore individual objects from the tape created.

### Creating a Boot Tape

You can use enhanced MAGSAV to create a boot tape. Boot tapes are used in the initial stages when booting from tape after a command partition crash. Full details on creating a boot tape and on booting from tape are given in Chapter 4, Creating and Using a Boot Tape.

### Access Rights

To save an object using MAGSAV, you must have the correct access rights to the object on disk. These are as follows:

- Read (R) rights for files
- List (L) and Use (U) rights for directories

If you wish to save an entire directory, you must have read (R) rights to all files within the directory. To make full use of the features available in MAGSAV, you should use the

command SPAC to set a priority Access Control List (ACL) on the partition containing the files you wish to save. To use all the available MAGSAV options, you must also be a member of the ACL group .BACKUP\$ that is set up by the System Administrator. If you invoke MAGSAV from the supervisor terminal, you automatically possess privileged user rights and can use all the available MAGSAV options without being a member of .BACKUP\$, but you must still set a priority ACL.

### The Save Process

Every time you invoke MAGSAV a separate *logical tape*, or *save*, is created. If the logical tape is very large, it may span more than one reel of tape. Conversely, one reel of tape can contain several logical tapes.

When you invoke MAGSAV to write to a previously written tape, you can choose either to append the data to the tape, or to overwrite the existing data on the tape.

Depending on the setting of the read/write lock (rwlock), you can usually save files that are open for reading, but you may not be able to save files that are open for writing; if the file cannot be saved, an informative message is displayed and the system passes on to the next file, if any, to be saved. For details of read/write locks, refer to the RWLOCK command in the *PRIMOS Commands Reference Guide*.

### File Security

When you save objects, MAGSAV first writes the object to tape and then writes access information about that object.

MAGSAV always saves specific ACLs attached to files or directories, unless you use the option -NO\_ACL to prevent this. However, MAGSAV does not save ACL information for file system objects that use default protection.

When you save a directory, MAGSAV automatically saves any access categories within the directory. However, when you save individual files by name, you *must* also specify the names of any access categories associated with the file, as these are not saved automatically.

Passwords attached to directories are automatically saved with the directory to which they belong.

### Incremental Saves

You can use MAGSAV for the following types of save:

- Base Saves: save all files within a partition or directory.
- Cumulative Incremental Saves: save all objects that have been modified since the last base save. When you subsequently restore the data, you need to restore the base save and the last cumulative incremental save.

- **True Incremental Saves:** save all objects that have been modified since the last save, which may have been a base save or an incremental save. When you subsequently restore the data, you need to restore the base save and all incremental saves.

The last two types of save are collectively referred to as incremental saves.

### Incremental Control

When you run an incremental save, MAGSAV uses the file system attributes Date/Time Modified (DTM), and Date/Time Backed Up (DTB), to determine which objects to save, and saves those objects whose DTM is later than the DTB.

Use the MAGSAV option `-UPDT` to update the DTB of saved objects. To use this option you must be a member of the ACL group `.BACKUP$`, or else you must invoke MAGSAV from the supervisor terminal.

#### Note

Enhanced MAGSAV uses the DTB and not the dumped bit setting to provide incremental control. On the first use of enhanced MAGSAV you should therefore perform a full base save which will set the DTB on the saved files.

### Command Line Options

MAGSAV is an interactive command that prompts you for information. Invoke MAGSAV by issuing the MAGSAV command using none, one, or more of the available command line options. These options are described below. Some options are restricted to members of the `.BACKUP$` ACL group. If you invoke MAGSAV from the supervisor terminal, you automatically possess privileged user rights and do not need to be a member of `.BACKUP$` in order to use all the available MAGSAV options.

#### ► MAGSAV [options]

<i>Option</i>	<i>Description</i>
<b>-BIG</b>	Writes a tape with 16K tape blocks, provided the MTRS directive in the system configuration file is set to 16K bytes or greater. You must set the MTRS directive before you invoke MAGSAV. You can use this option only if you are using a system running on PRIMOS Rev. 22.0 or a later revision, and you are writing an ANSI format tape. You cannot use this option if you are writing a non-ANSI tape; therefore, the option <code>-BIG</code> cannot be used with options <code>-REV19</code> and <code>-REV20</code> . If you omit the option, MAGSAV automatically writes 12K byte blocks to tape irrespective of the setting of the MTRS directive.
<b>-BOOT</b>	Creates a boot tape; see Chapter 4 for further details of <code>-BOOT</code> .
<b>-HELP</b>	Displays the MAGSAV syntax and options.

- INC** Specifies an incremental save. Only files and directories with a DTM later than the DTB are saved. If you do not specify **-INC**, all specified files and directories are saved. This option is used primarily by system operators.
- LABEL** Instructs MAGSAV to prompt you for a volume serial ID for any reel encountered during the save that does not contain an ANSI label. The volume serial ID is an alphanumeric string of a maximum length of 6 ANSI characters that forms part of the tape's ANSI label and which uniquely identifies a particular reel of tape. This is particularly useful if the save spans more than one tape reel. If you specify a string that contains non-ANSI characters you are prompted to reenter the volume serial ID. You cannot use this option if you are writing a non-ANSI tape; therefore, the option **-LABEL** cannot be used with options **-REV19** and **-REV20**. If you omit the option, MAGSAV automatically writes an ANSI label on tape that contains a null entry for the volume serial ID.
- { -NO\_ACL }  
{ -NA }** This option instructs MAGSAV not to save ACLs, passwords, or access categories (ACATs) associated with saved objects. If you omit this option, file access information is saved to tape.
- NO\_DTA** Instructs MAGSAV not to modify the DTA of any files or directories saved. This option uses the special backup key to open files. You can use this option only if you are a member of the ACL group **.BACKUP\$**, and also have protect (P) access rights to the disk object. You can also use this option if you invoke MAGSAV from the supervisor terminal. If you omit the option, the DTA of saved objects is set to the current date/time.
- NO\_RBF** Instructs MAGSAV not to save any Recovery Based Files (RBFs). Refer to the *ROAM Administrator's Guide* for more information on these files.
- Note**
- If you save an entire partition containing RBFs that also span other partitions, the system automatically accesses these partitions.
- QUERY** Instructs MAGSAV to prompt you for confirmation before overwriting existing logical tapes. You cannot use this option if you are writing a non-ANSI tape; therefore, the option **-QUERY** cannot be used with options **-REV19** and **-REV20**.
- REV19** Writes a tape in non-ANSI format, readable on systems running on PRIMOS Revisions 19.0 to 19.4.14. Automatically saves CAM files as DAM files. You cannot use this option with options **-BIG**, **-LABEL**, or **-QUERY**.

**-REV20**

Writes a tape in non-ANSI format, readable on systems running on PRIMOS Revisions 20, 21, and 22. You cannot use this option with options -BIG, -LABEL, or -QUERY.

**{-SAVE\_UFD}  
{-SUFD}**

Instructs MAGSAV to save all directory entries, whether or not the directories have been modified since the last save. The contents of these entries are saved only if they have been modified. This enables you to recreate the directory structure on disk when you restore the tape. Use this option to save the directory structure when running incremental saves where the save tape is written in non-ANSI format, that is, when running incremental saves using options -REV19 or -REV20.

**-TTY**

Use this option when running MAGSAV from CPL or command input (COMI) files. Instructs MAGSAV to prompt you to enter the tape unit number, and the response to the overwrite query (see option -QUERY), from your terminal. All other information is taken from the CPL or COMI files.

**-UNLOAD**

Instructs MAGSAV to rewind and unload the tape when the end of the tape is reached.

**-UPDT**

Instructs MAGSAV to update the DTB of any files or directories that are saved. You can use this option only if you are a member of the ACL group .BACKUP\$ or if you invoke MAGSAV from the supervisor terminal. This option is used primarily by system operators.

### Command Line Examples

Two examples of the MAGSAV command are shown below.

**Cumulative Incremental Save:** Use the following command to run a cumulative incremental save:

OK, MAGSAV -INC

The -INC option instructs the system to save only those files or directories whose DTB is later than the DTB. The system does not update the DTB of saved objects. The next time you run an incremental save, the system saves the same objects again, as well as any others that have been updated since this save. When you restore to disk, you need to restore the base save and the latest incremental save.

**True Incremental Save:** Use the following command to run a true incremental save:

OK, MAGSAV -INC -UPDT

The **-INC** option instructs the system to save objects in the same way as when running a cumulative incremental save. The **-UPDT** option instructs the system to update the DTB of objects saved. The next time you run an incremental save, the system saves only those objects that have been modified since this save. When you restore to disk, you need to restore the base save and all incremental saves since the base save.

### The MAGSAV Dialog

After you invoke MAGSAV by issuing the command and options, the MAGSAV dialog begins. Shown below are the system prompts in the order in which they appear, together with the appropriate user responses. Note that if you are saving to a 60MB cartridge tape, the dialog is slightly different from that shown below. These differences are described in the section, *Saving to a 60MB Cartridge Drive*, later in this chapter.

#### Prompt

#### User Response

Tape unit:

Enter the logical tape unit number for the tape drive you wish to use. The system assumes you are using a 9-track tape unit. Enter Q, or QUIT to quit the program.

If the reel is in ANSI format, the label on tape may contain a volume serial ID that identifies the reel of tape. If it does, a message of the following form is displayed:

Volume serial ID is MYVOL1.

Enter logical tape number:

Enter a number that denotes the sequential number on the tape for the logical tape you wish to save. Enter 1 for the first save on the reel, 2 for the second and so on. MAGSAV positions the tape to the start of the required logical tape. If you enter APPEND this will position the tape at the end of the last save on tape. If you enter the value 0 or press

**Return**

MAGSAV assumes that the tape is already correctly positioned. Enter Q or QUIT to quit the program.

If you used the **-QUERY** option and are writing to an ANSI reel that already contains data, a prompt of the form shown below appears:



## LOGICAL SAVE TO TAPE: MAGSAV

Logical tape no: 1, Tape name: MYTAPE  
Logical Tape 1 already exists on MYVOL1:  
OK to overwrite(Y/N):

Answer YES to overwrite the existing logical tape. If you answer NO, you are requested to enter another logical tape number.

If you used the -LABEL option, and are about to write the first logical tape on an unlabelled ANSI reel, the following prompt appears:

Enter volume serial ID:

Enter an alphanumeric string of a maximum length of 6 characters. The characters must belong to the ANSI character set. If you specify non-ANSI characters, the system prompts you to reenter the volume serial ID. This string is written to the tape label and is used to give the reel of tape a unique identity. Strings longer than the maximum length are truncated to the first 6 characters; strings shorter than 6 characters are padded with blanks.

You cannot mix formats on tape. If you write to a tape that already contains data in a format different from the one you are using, you must overwrite all the data already on tape by specifying a logical tape number of 1. Otherwise the system prompts you to reenter the logical tape number, as shown below:

Tape unit: 1  
Enter logical tape number: 2  
Tape formats do not match.  
Logical tape number must be 1  
Enter logical tape number: 1

Tape name:

Enter an alphanumeric string of a maximum length of 32 characters. For tapes written in non-ANSI format, the name is truncated to the first 6 characters when written to tape.

Date (mm dd yy):

Press **RETURN** to enter the displayed current date (represented here by *mm dd yy*). You can also enter the date in the format *mm dd yy*, where *mm*=month, *dd*=day, and *yy*=year. PRIMOS checks the date format and rejects it if it is invalid.

Rev. no:

Enter a decimal integer to specify the Revision number you are using. If you press

**RETURN**

the field is set to 0. Non-integer values are not allowed.

**Name or Command:**

Enter the name of the object to save, or enter one of the subcommands that modify the mode of operation of the save. The subcommands are discussed below.

**Notes**

The action of the subcommand applies to all objects that are saved after you issue the subcommand.

Pathnames are not allowed; therefore before specifying the object to be saved, you should be attached to the directory containing that object. You can change your attach point using the \$A subcommand.

After you enter the object name, the save begins. If MAGSAV cannot access a file during the save operation, the system continues with the next file or directory to be saved, if any.

When the save is complete, the prompt **Name or Command:** is redisplayed, and you can continue saving further objects or issue a subcommand. Enter the subcommand \$Q, \$R, or \$UNload to return to PRIMOS when you have finished.

You can make any one of the following responses to this system prompt:

*objectname*

\*

MFD

\$A *pathname*

\$Index [*pathname*] [*n*]

\$E *pathname* [*n*]

\$INC ON

\$INC OFF

\$Q

\$R

\$UNload

\$TTY

\$Updt ON

\$Updt OFF

*objectname* saves the file or directory specified. You should be attached to the directory containing the object *objectname*. The full pathname of the object is saved.

\* saves all files and directories within the current directory. The name of the directory to which you are attached is omitted from the pathnames on tape.

MFD saves the entire partition. You must be attached to the MFD of the partition. All saved objects have MFD as the first element of their pathname on tape.

**Note**

At Rev. 23, if you add a partition using the -MOUNT\_PATH option, so it is mounted lower in the file system, MAGSAV will not save the entire partition at the point to which you are attached.

\$A *pathname* attaches you to the specified directory *pathname* on disk. Supply a password if needed.

\$Index [*pathname*] [*n*] generates an index of the save operation. If you omit *pathname*, the index is displayed on the screen. If you specify *pathname*, the index is written to the file specified, and is not displayed on the screen. If the file already exists, the index is appended to that file. You cannot specify passwords in the *pathname*. The argument *n* is a decimal integer that specifies the number of directory levels in the index. If you omit *n*, the default is two levels.

\$E *pathname* [*n*] generates an extended index of the save operation. The index is written to the file specified by *pathname*. If the file already exists, the index is appended to that file. You cannot specify passwords in the *pathname*. The argument *n* is a decimal integer that specifies the number of directory levels in the index. If you omit *n*, the default is two levels. For further information on extended index format see Chapter 5, Tape Indexing.

\$INC ON turns on incremental save mode. Only those objects are saved whose DTM is later than their DTB. (At PRIMOS Revs. 21.0 and 22.0, RBFs are saved if the dumped bit is 0.) This subcommand has the same effect as the -INC command line option.

\$INC OFF turns off incremental save mode, whether you set incremental mode using option -INC, or subcommand \$INC ON. All objects subsequently specified are saved, irrespective of their DTB and DTM settings, (or the dumped bit setting for RBFs). This is the default.

\$Q terminates the logical tape and returns you to PRIMOS.

\$R terminates the logical tape, rewinds the reel, and returns you to PRIMOS.

**\$TTY** Use this subcommand when running MAGSAV from CPL or command input (COMI) files. Instructs MAGSAV to prompt you to enter the tape unit number from your terminal. All other information is taken from the CPL or COMI files.

**\$UNLOAD** terminates the logical tape, rewinds the tape, and unloads the reel.

**\$UPDT ON** turns on update mode. MAGSAV updates the DTB on objects subsequently saved. (At PRIMOS Revs. 21.0 and 22.0, the dumped bit of RBFs is set to 1.) This subcommand has the same effect as the **-UPDT** command line option. To use this subcommand, you must be a member of the **.BACKUP\$ ACL** group, or else you must have invoked MAGSAV from the supervisor terminal.

**\$UPDT OFF** turns off update mode whether you set update mode using option **-UPDT**, or subcommand **\$UPDT ON**. MAGSAV does not update the DTB on objects subsequently saved. (In the case of RBFs, the dumped bit is not set.) This is the default.

### **End of Tape Handling**

If the system encounters an end-of-tape (EOT) when writing to tape, an informative message is displayed, the system rewinds the tape (and unloads it, if you specified **-UNLOAD**), and prompts you to enter the tape unit number for the continuation tape. A typical output is shown below:

```
End of reel (Volume serial ID MYVOL1)
Mount continuation reel
Continuation tape unit: 1
```

The save then continues from the beginning of the newly mounted reel. The continuation tape unit does not have to be the same as the tape unit previously used. If you are running the save from a CPL or COMI file, use the option **-TTY** if you wish to enter the continuation tape unit number from your terminal.

### **Saving to a 60MB Cartridge Drive**

When you run MAGSAV to save to a 60MB cartridge drive you can use the same command line options as those described earlier in this chapter. After you invoke MAGSAV, the format of the dialog differs slightly from that described on the previous pages, because with these drives you can append data only at the end of the existing data on tape, otherwise you must overwrite the entire contents of the tape.

## Examples

The following two examples show typical outputs when running MAGSAV to save to a 60MB cartridge drive. Shown below is a typical output when overwriting the data on tape.

```
Tape Unit: 1
Volume serial ID is MYVOL1
Overwrite or Append (O/A): O
Tape Name: MYTAPE
Date (03 22 90): [RETURN]
Rev. no: 23
Name or Command: MYFILE
Tape initializing, please wait .....
```

MAGSAV erases all the existing data from the tape. This takes approximately two minutes, after which the save begins. Shown below is a typical output when appending to the data already on tape.

```
Tape Unit: 1
Volume serial ID is MYVOL1
Overwrite or Append (O/A): A
```

MAGSAV checks for end-of-tape (EOT) markers at the end of the last logical tape. If you aborted from the save when last writing to tape, these markers are not present; an error message is displayed, and you are prompted to re-enter the tape unit number.

When the system has located the EOT markers, the dialog continues with the prompts shown above in Example 1, and the data is appended to that already on the tape.

## Listing the Tape Contents

You can list the contents of a MAGSAV tape by using MAGRST command with the subcommand NW. This produces an index of your tape without actually restoring the contents of the tape. Further details are given in Chapter 4, Logical Restore from Tape: MAGRST.

## Save Procedure for an Entire Partition

You should first decide whether to shut down the entire system to users, or just the partition that you want to back up. For a full discussion of preparation procedures to follow before saving and restoring data, refer to Chapter 10 of this guide.

Shown below are the steps you should follow to use MAGSAV to save an entire partition.

If you have decided to shut down the entire system, follow the procedure described in Chapter 10 of this guide under the section, Procedure for Shutting Down the System.

## OPERATOR'S GUIDE TO DATA BACKUP AND RECOVERY

If you have decided to shut down only the partition, follow the procedure described in Chapter 10 of this guide, under the section, Procedure for Removing a Partition from Service.

The following steps must be executed from your supervisor terminal if you have shut down the system, otherwise they may be executed from a user terminal.

1. Attach to the MFD of the partition that you are going to save. For example:

OK, ATTACH <DSK1>MFD

If the partition is mounted lower down in the file system you must use the mount point pathname.

2. Assign the tape unit for the tape that you are going to save. For example:

OK, ASSIGN MT1

3. Mount the reel of tape on the assigned drive. It is advisable to attach a physical label to the reel with the following information:

- Your initials
- The date and time of the save
- The name of the partition being saved
- The name of the system in use
- The PRIMOS Revision number
- The recording density on tape
- The block size used
- The reel number within the volume set, for example, TAPE 1 of N, where N represents the number of reels within the set, and should be filled in once the backup is complete
- The fact that MAGSAV has been used for the save

You should also leave room on the label for the volume serial ID, if it is not labeled permanently, and for the size of the save.

4. Invoke MAGSAV and enter the required response when prompted to do so. A typical example is as follows:

```
OK, MAGSAV -QUERY -LABEL
[MAGSAV Rev. 23.0 Copyright(c) 1990, Prime Computer, Inc.]
Tape unit: 1
Enter logical tape number: 1
Enter volume serial ID: MYVOL1
Tape name: MYTAPE
Date (03 22 90): [RETURN]
Rev. no: 23
Name or Command: MFD
Name or Command: SUNLOAD
OK,
```

In the above example, it is assumed that only one reel of tape was required for saving the entire partition. If more than one reel is required, the EOT system prompts would be displayed, as described in the section, End of Tape Handling, earlier in this chapter.

5. When MAGSAV has completed the save, remove the last tape. It is important to fill in the blanks on the tape reel labels to indicate the number of tapes in the set, the size of the save and to add the volume serial ID if the reel is not already labeled. Remember also to remove the write-enable rings from the backup tapes, to avoid accidental overwriting.
6. Unassign the tape unit by issuing the following command:

OK, UNASSIGN\_MT1

7. Dismount the tape reel.
8. If you have shut down the entire system, restart the system by following the procedure described in Chapter 10 of this guide, under the section, Procedure for Starting Up the System.

If you have shut down only the partition, return it to service by following the procedure described in Chapter 10 of this guide, under the section, Procedure for Returning a Partition to Service.

## Unattended Backup

The very high data capacity of the 8mm cartridge tape drive reduces the number of tapes required to store backup data. This may eliminate the need for operator intervention, and thus allow you to perform backup operations overnight.

Some systems have a total backup requirement of less than the capacity of a single 8mm cartridge tape. Larger systems require the use of multiple tapes. If you need to use multiple tapes for backup operations, the ability to run unattended backups will depend on your system having enough tape drives for you to mount all the required tapes before you initiate the backup. You can use the CPL program described in the following section to run unattended backups using multiple tapes, provided you mount and assign all the required tapes before you initiate the backup.

### CPL Program for Multiple Tape Backups

You can use the CPL program described below to run a backup which requires more than one tape. The program allows for the overflow of a single partition onto more than one tape.

If you run MAGSAV under CPL control, the system sets three local variables when you terminate the save with the \$Q, \$R, or \$UNLOAD commands. These variables are as follows:

- DR\$TAPE\_UNIT is the tape drive unit number on which the save finished.
- DR\$SAVE\_NUMBER is the number of the last save written.
- DR\$VOLID is the final volume id.

## OPERATOR'S GUIDE TO DATA BACKUP AND RECOVERY

These variables give the tape unit and save number of the save which has just completed, thus enabling you to calculate the tape unit and save number for the next save.

In the following CPL program, each of the partitions P1, P2, P3, P4, and P5 are saved to tapes which have been previously mounted on tape units numbered sequentially from 0. These units must be assigned prior to the start of the CPL program.

The first partition is saved as save number 1, and the succeeding partitions as save numbers 2, 3, 4, and 5 respectively. The save name for each save is the name of the partition. Any existing data on these tapes will be overwritten.

```
&set_var DR$TAPE_UNIT := 0           /*First tape unit number

&set_var DR$SAVE_NUMBER := 1        /*First save number

&do partition &list P1 P2 P3 P4 P5  /*Start of loop for each partition

    ATTACH <%partition%>mfd
    &data MAGSAV
    %DR$TAPE_UNIT%                   /*Tape unit no.
    %DR$SAVE_NUMBER%                /*Save no.
    %PARTITION%                     /*Save name
    [date -USA]                     /*Date of save
    23                              /*PRIMOS Rev. no.
    $I 3                            /*Index
    *                               /*Save MFD
    [calc %DR$TAPE_UNIT% + 1]       /*Increment tape unit no.
                                   /*Enter further continuation
                                   /*devices here if necessary

    $Q
    &end
    &set_var DR$SAVE_NUMBER := [calc %DR$SAVE_NUMBER% + 1]
                                   /*Increment save no.
&end                               /*End of loop
```

If the end of tape is reached during the partition save, the system prompts for the continuation unit number; this is returned to the system in the above program at the line where the tape unit number is incremented. The partition may overflow onto several physical tapes; in each case the next tape accessed is that on the next sequential tape unit number.

If the system does not reach the end of tape during the partition save, MAGSAV reads the continuation tape unit number, as calculated in the above program, in response to the Name or command prompt. The system rejects this number as an invalid response, and issues a reprompt, to which it reads the command \$Q to terminate the current logical tape. The next partition save will commence at the point on the tape where the previous save terminated.

In either case, the save number is incremented only when the next partition is saved.

More complex CPL programs may be written to generate tape unit numbers in some other order, for example, from a list. Note that due to the limit on the size of a partition, a partition save would not normally continue onto a third 8mm tape.



## LOGICAL RESTORE FROM TAPE: MAGRST

### Introduction

This chapter gives comprehensive details on the use of enhanced MAGRST. Aspects of restoring files, directories, and partitions are described. Explanations are given of the MAGRST command line options, and of the system prompts and user responses. Procedures are given for restoring individual files and directories, and for restoring entire partitions.

### Overview

MAGRST is a utility used for restoring file system objects from tape to disk. You can only use MAGRST to restore tapes previously saved using MAGSAV. Because MAGSAV saves objects as logical entities, you can use MAGRST to restore individual objects to disk. You cannot do this if you use a physical save utility to write to the tape.

### Restoring Files, Directories, and Partitions

You can either restore the entire contents of a tape, or just part of the tape. If you restore part of the tape only, this is termed a partial restore. You must specify the pathnames of objects to be restored as recorded on tape. Objects are written into the directory to which you are attached. When you restore an object you may give it a different name from that which it has on tape.

#### Restoring Directories and Files

When MAGRST restores a directory to disk, the entire directory structure is recreated on disk within the directory to which you are currently attached.

When MAGRST restores individual files to disk without restoring the associated directory, only the filename portion of the pathname on tape is retained. This means that if you restore two files that have different pathnames on tape but the same filenames, the second

## **OPERATOR'S GUIDE TO DATA BACKUP AND RECOVERY**

file restored overwrites the first. For example, SUE>LOGIN.CPL overwrites MEG>LOGIN.CPL.

When you run incremental saves and the tape is written in ANSI format, the directory structure is saved as well as the files that have been modified since the last incremental save.

When you run incremental saves using the options -REV19 and -REV20, which write tapes in non-ANSI format, the directory structure is not saved to tape unless you use the option -SUFD. If you omit the option -SUFD when writing incremental save tapes in non-ANSI format, you will not be able to recreate the directory structure when you restore this tape to disk. In practice you usually restore the associated base tape before an incremental save tape, and the entire directory structure is recreated when you restore the base tape.

### **Restoring Segment Directories**

When MAGRST restores a segment directory into an area that already contains a segment directory of the same name, the system automatically overwrites or deletes subfiles in the disk directory, and the restored directory is an exact copy of the segment directory on tape.

### **Restoring Partitions**

A partition may occupy the entire contents of one or several tapes. Before restoring a partition from tape, you must ensure that there is sufficient available space on disk. If you do not have sufficient room to restore the tape contents you should first recreate the disk using the MAKE command.

## **File Security**

MAGRST always tries to restore the ACL, access category, or password protection associated with the object restored from tape. However, you should be aware of the following points:

- Because MAGSAV writes an object to tape before it saves the ACLs associated with that object, MAGRST first restores the object to disk, and then restores the ACLs. This means that you can restore an object from tape even when you have no rights to the object, but you may not be able to use the restored object.
- When you are running a partial restore and you restore a file that is protected by an access category, MAGRST does not automatically restore the access category. An error message is displayed and the restored file has default protection on disk. To restore specific files with their access categories, you should first restore the access categories and then the protected files.
- The following table describes the situations which may arise when you restore ACL and Password directories:
  - ACL directory restored from tape:

- Overwrites existing ACL disk directory of the same name. Restored directory is an ACL directory, and the ACLs of objects protected by this directory are set to the values in the restored directory.
- Overwrites existing password disk directory of the same name. Restored directory is a password directory, and the ACLs of objects protected by this directory are lost.
- o Password directory restored from tape:
  - Overwrites existing password disk directory of the same name. Restored directory is a password directory, and the passwords of objects protected by this directory are set to the values in the restored directory.
  - Overwrites existing ACL disk directory of the same name. Restored directory is an ACL directory, and the ACLs of objects protected by this directory are set to their default values.

## Command Line Options

MAGRST is an interactive command that prompts you for information. Invoke MAGRST by issuing the MAGRST command using none, one, or more of the available command line options. These options are described below.

### ► MAGRST [options]

<i>Option</i>	<i>Description</i>
<b>{ -CAM_RBF } {-CRBF }</b>	Restore all RBF subfiles as CAM files.
<b>-HELP</b>	Displays MAGRST syntax and options.
<b>{ -NO_ATTRIBUTES } {-NATT }</b>	By default, MAGRST restores the attributes DTA/DTC/DTM from tape. This option disables the restoring of these attributes from tape and the attribute values of the restored object are set to those of the current date and time.
<b>-QUERY</b>	Use this option when restoring Recovery Based Files (RBFs). If a master RBF of the same name exists on disk, the system prompts you to state if you wish to overwrite the existing copy. If you answer YES, the system deletes the master and slave RBFs from disk, and restores the objects from tape; if you answer NO, the system continues with the next object to restore. If you omit this option, the system overwrites existing master RBFs of the same name, but not existing slave RBFs if they are slave files of a different master RBF; an error message is displayed and the restore is aborted. You can also use this option to instruct MAGRST to query you when restoring an EPF that is currently

in use. For further information on RBFs, refer to the *ROAM Administrator's Guide*.

**-TTY**

Instructs MAGRST to prompt you to enter the tape unit number from your terminal. All other information is taken from the CPL or COMI file. Use this option when running MAGRST from CPL or command input (COMI) files.

**-UNLOAD**

Instructs MAGRST to rewind and unload the tape when the end of the tape is reached.

## The MAGRST Dialog

After you invoke MAGRST by issuing the MAGRST command and options, the MAGRST dialog begins. Shown below are the system prompts in the order in which they appear, together with the appropriate user responses.

*Prompt*

*Response*

Tape unit:

Enter the logical tape unit number for the tape drive you wish to use. The system assumes you are using a 9-track tape unit. Enter Q, or QUIT to quit the program.

If the reel has a valid volume serial ID, a message of the following form is displayed:

Volume serial ID is MYVOL1

Enter logical tape number:

Enter a number that denotes the sequential number on tape for the logical tape (or save) that you wish to restore. Enter 1 for the first save on the reel, 2 for the second, and so on. MAGRST positions the tape to the start of the requested save number. If you enter the value 0, or press

**Return**

MAGRST assumes that the tape is already correctly positioned. Enter Q, or QUIT to quit the program.

A message of the following form is displayed, showing the details of the logical tape file:

Positioning to logical tape number 2...

Name: MYTAPE

Date: 12 10 90

Rev no: 23

Reel no: 1

If you enter a save number that is higher than that of the last save on the reel, an error message is displayed, and the system prompts you to re-enter the tape unit number.

Ready to Restore:

You can make any one of the following responses to this system prompt:

YES  
 NO\$  
 A *pathname*  
 \$Index [*pathname*] [*n*]  
 \$E *pathname* [*n*]  
 NW [*filename*] [*n*]  
 XNW *filename* [*n*]  
 Partial  
 CHecKpoint

YES restores the entire logical tape and returns you to PRIMOS. If MAGRST cannot access an object during a restore operation, that object is abandoned and the system continues with the next file or directory, if any, to restore.

NO requests a different tape unit and logical tape. MAGRST does not restore the previously specified tape, and prompts you to reenter the tape unit number.

\$A *pathname* attaches you to the specified directory *pathname* on disk. Supply a password if needed.

\$Index [*pathname*] [*n*] generates an index of the restored objects, and restores them to disk. If you omit *pathname*, the index is displayed on the screen. If you specify *pathname*, the index is written to the file specified and is not displayed on the screen. If the file already exists, the index is appended to that file. You cannot specify passwords in the pathname. The argument *n* is a decimal integer that specifies the number of directory levels in the index. If you omit *n*, the default is two levels.

\$E *pathname* [*n*] generates an extended index of the restored objects, and restores them to disk. The index is written to the file specified by *pathname*. If the file already exists, the index is appended to that file. You cannot specify passwords in the pathname. The argument *n* is a decimal integer that specifies the number of directory levels in the index. If you omit *n*, the default is two levels.

NW [*filename*] [*n*] generates an index of the tape contents, but does not restore any files or directories. If you omit *filename*, the index is displayed on the screen. If you specify *filename*, the index is written to the file specified and is not displayed on the screen. If the file already exists, the index is appended to that file. The argument *n* is a decimal integer that specifies the number of directory levels in the index. If you omit *n*, the default is all levels. Use this option to determine the contents of a tape.

XNW *filename* [*n*] generates an extended index of the tape contents, but does not restore any files or directories. The index is written to the file specified by *pathname*. If the file already exists, the index is appended to that file. The argument *n* is a decimal integer that specifies the number of directory levels in the index. If you omit *n*, the default is 100 levels. Use this option to determine the contents of a tape.

PARTIAL restores only specified files and directories. You are prompted to enter the the names of the objects you wish to restore, as shown below:

TREE NAME:

Enter the pathnames of the objects you wish to restore. These pathnames should be entered exactly as recorded on tape. You may change the name of the object you wish to restore by entering the response *name1 name2*, where *name1* is the pathname of the object on tape and *name2* is the pathname you wish to give to the restored object. After you enter each pathname, MAGRST repeats the prompt. You can specify any number of pathnames. Press  at the prompt to begin the restore process.

CHECKPOINT restores only specified files and directories in the same way as the PARTIAL subcommand, but in this case you are prompted to enter both the pathname of the object you wish to restore, and the number of the checkpoint which immediately precedes the object on tape. The prompt and response format are as shown below:

TREE NAME: *name1* *checkpoint\_no* [*name2*]

where *name1* is the pathname of the object on tape, *checkpoint\_no* is the number of the checkpoint which immediately precedes the object on tape, and *name2* is the (optional) pathname you wish to give to the restored object. The *checkpoint\_no* is obtained from the extended index.

### End of Tape Handling

If MAGRST encounters an end-of-tape (EOT) when restoring from tape, an informative message is displayed, the system rewinds the tape (and unloads it, if you specified

-UNLOAD), and prompts you to enter the tape unit number for the continuation tape. A typical output is shown below:

```
End of reel (Volume serial ID MYVOL1)
Mount continuation reel
Continuation tape unit: 1
```

The system continues restoring objects from the newly mounted reel. The continuation tape unit does not have to be the same as the tape unit previously used. If the continuation reel is out of sequence, a messages is displayed and you are prompted to reenter the continuation tape unit. If you are running the restore from a CPL or COMI file, use the option -TTY if you wish to enter the continuation tape unit number from your terminal.

## Procedure for a Partial Restore

Shown below are the steps you should follow when you use MAGRST to run a partial restore. It is not necessary to shut down either the system or the partition to users.

9. Attach to the directory into which you intend to restore the objects. For example:

```
OK, ATTACH <DSK1>MYDIR
```

10. Assign the tape unit for the tape that is to be restored. For example:

```
OK, ASSIGN MT1
```

11. Mount the reel of tape on the assigned drive. If the objects to restore are contained on a multireel tape volume, this must be the first reel of the volume.
12. Invoke MAGRST and enter the required response when prompted to do so. A typical example is as follows:

```
OK, MAGRST
[MAGRST Rev 23.0 Copyright (c) 1990, Prime Computer, Inc.]
Tape unit: 1
Volume serial ID is MYVOL1 Enter logical tape number: 2
Positioning to logical tape number 2 ...
Name: MYTAPE
Date: 12 10 90
Rev no: 23
Reel no: 1
Ready to Restore: PARTIAL
Tree name: A>FILE1
Tree name: A>FILE2
Tree name: [Return]
*** Starting Restore
*** Restore Complete
OK,
```

13. Unload the tape and unassign the tape unit by issuing the following command:

OK, UNASSIGN MT1 -UNLOAD

14. Dismount the tape reel.

## **Procedure for a Checkpoint Restore**

The procedure for restoring objects from tape using checkpoints to speed up the tape handling, is very similar to the procedure shown above for a partial restore. However, for a checkpoint restore, you enter the response **CHeckpoint** instead of **PARTIAL**, and then input both the object name and the associated checkpoint number at the **MAGRST** prompt

Tree name:.

## **Restore Procedure for an Entire Partition**

It is not essential to shut down either the system, or the partition before restoring a partition. However, it is advisable to do so if the partition has many origin directories, or if it holds information used by many users on the system. If you are restoring the command partition, you must shut down the system. Restoring the command partition is carried out as part of the process of booting from tape. For details of boot procedures, refer to Chapter 4, **Creating and Using a Boot Tape**. For a full discussion of preparation procedures to follow before saving and restoring data, refer to Chapter 10, **System Preparation, Startup, and Shutdown**.

Before you restore the partition from tape, you must ensure that there is sufficient available space on disk. If you do not have sufficient room to restore the tape contents you should first recreate the partition using the **MAKE** command.

Shown below are the steps you should follow to use **MAGRST** to restore an entire partition.

If you have decided to shut down the entire system, follow the procedure described in Chapter 10 of this guide under the section, **Procedure for Shutting Down the System**. If you have decided to shut down only the partition, follow the procedure described in Chapter 10 of this guide under the section, **Procedure for Removing a Partition from Service**. The following steps must be executed from your supervisor terminal if you have shut down the system, otherwise they may be executed from a user terminal.

1. Attach to the MFD of the partition that you are going to restore. For example:

OK, ATTACH <DSK1>MFD

2. Assign the tape unit for the tape that you are going to restore. For example:

OK, ASSIGN MT1



3. Mount the reel of tape on the assigned drive. If the objects to restore are contained on a multireel tape volume, this must be the first reel of the volume.
4. Invoke MAGRST and enter the required response when prompted to do so. A typical example is as follows:

```
OK, MAGRST
[MAGRST Rev. 23.0 Copyright (c) 1990, Prime Computer, Inc.]
Tape Unit: 1
Volume serial ID is MYVOL1 Enter logical tape number: 2
Positioning to logical tape number 2 ...
Name: MYTAPE
Date: 12 10 90
Rev no: 23 Reel no: 1
Ready to Restore: YES
*** Starting Restore
*** Restore Complete
OK,
```

5. Unload the tape and unassign the tape unit by issuing the following command:

```
OK, UNASSIGN MT1 -UNLOAD
```

6. Dismount the tape reel.
7. If you have shut down the entire system, restart the system by following the procedure described in Chapter 10 of this guide, under the section, Procedure for Starting Up the System.

If you have shut down only the partition, return it to service by following the procedure described in Chapter 9 of this guide, under the section, Procedure for Returning a Partition to Service.

## CREATING AND USING A BOOT TAPE

### Introduction

This chapter explains how to make a boot tape from which you can boot your system in the event of a system failure, and how to recreate the system from a boot tape.

---

#### Caution

The processes of using enhanced MAGSAV/RST to create a boot tape and to boot the system from tape vary slightly depending upon whether you are using PRIMOS Rev. 21.0 and 22.0 or PRIMOS Rev. 23.0. You should read this chapter carefully to ensure that you are aware of these differences.

---

### Overview

You use the `BOOT_CREATE` command to generate a boot tape. This command invokes a CPL program which reads a list file containing the pathnames of the directories and files you wish to write to the boot tape, and then calls MAGSAV with the `-BOOT` option to save the objects to tape. On systems using enhanced MAGSAV/RST at PRIMOS Revs. 21.0 and 22.0, the boot tape is saved in non-ANSI format. At PRIMOS Rev. 23.0, the boot tape is saved in ANSI format.

---

#### Caution

You cannot boot a Rev. 21.0 or 22.0 system using a boot tape that was generated using enhanced MAGSAV running on PRIMOS Rev. 23.0. This situation may arise, if, for example, you had both a PRIMOS Rev. 22.0 and PRIMOS Rev. 23.0 command disk on your system simultaneously.

---

At PRIMOS Rev. 23.0 you use `MAGRST` to restore boot tapes; this utility is stored on the system as `MAGRST.RUN`. On systems using enhanced MAGSAV/RST at PRIMOS Revs. 21.0 and 22.0, you restore a boot tape using the utility `BOOT_RESTORE`; this utility is

stored on the system as `BOOT_RESTORE.SAVE`. You must ensure that you write the correct utility to the boot tape when you generate the tape using `BOOT_CREATE`.

## **Creating a List File**

To make the system boot tape you must first create a list file to contain the pathnames of all the directories and files that are needed to restore your system.

You can only boot from the first logical tape, and this must be contained within a single tape reel. Therefore, the boot tape must contain the files necessary to constitute a minimum working system that allows you to restore the rest of the system from other backup tapes.

Use either the Prime line editor `ED` or the screen editor `EMACS` to create the list file. The object pathname you specify in the list file must be sufficient to identify the object from your current attach point when you run `BOOT_CREATE`. Put each filename on a new line of the list file. You can place the list file anywhere in the system.

### **List File for PRIMOS Rev. 23.0**

If you are using PRIMOS Rev. 23.0, you should include, as a minimum, the following files and directories in the list file:

- `CMDNC0>MAKE.SAVE`
- `CMDNC0>MAGRSTRUN`
- `PRIRUN`

### **List File for Enhanced MAGSAV/RST on PRIMOS Revs. 21.0 and 22.0**

If you are using enhanced MAGSAV/RST running on a PRIMOS Revs. 21.0 or 22.0 system, you should include, as a minimum, the following files and directories in the list file:

- `CMDNC0>MAKE.SAVE`
- `CMDNC0>BOOT_RESTORE.SAVE`
- `PRIRUN`

### **List File Options**

When `BOOT_CREATE` runs, it reads the list file twice. The first pass checks that the files listed are present and accessible; the second pass calls `MAGSAV`, which writes the files to tape.

You can control the result of the checks carried out by `BOOT_CREATE` during the first pass through the list file, by attaching an option to each file specified in the list file. You can use only one option with each file. These options are described below.

<i>Option</i>	<i>Description</i>
<b>-CHECK</b>	Checks whether a file exists in a directory without saving the file or directory. If the file exists, no messages are displayed, and the object is not saved to the boot tape during the subsequent creation of the boot tape. If the file is not found, an error message is displayed and <b>BOOT_CREATE</b> aborts.
<b>-NO</b>	<b>BOOT_CREATE</b> displays a warning if it cannot locate or access the specified file or directory and that object is not written to tape. The program continues and writes all successfully located objects to the boot tape.
<b>-YES</b>	<b>BOOT_CREATE</b> displays an error message if it cannot locate or access the specified file or directory. The program terminates and no objects are written to the boot tape.

### Example List File

The following example shows how you can use ED to create a typical list file when running PRIMOS Rev. 23.0.

In this example, the list file is created in a directory called **TEST>MASTERFILE**, and is filed as **BOOT\_LIST**.

```
OK, ATTACH TEST>MASTERFILE
OK, ED
INPUT
CMDNCO>MAKE_SAVE
CMDNCO>MAGRST_RUN
PRIRUN
EDIT
FILE BOOT_LIST
OK,
```

### The **BOOT\_CREATE** Command

**BOOT\_CREATE** is an interactive command that prompts you for information. To invoke **BOOT\_CREATE** issue the command using none, one, or more of the command line options. The command syntax is as follows:

► **BOOT\_CREATE** [pathname] [options]

where *pathname* is the name of the list file. If you have created the list file in a password-protected directory and you want to use the **-NO\_QUERY** option, you must include the password in the pathname. In this case, enclose the entire pathname in single quotes, write it in upper case, and separate the password from the directory name by one space. If you omit *pathname*, the system prompts you to enter it.

## OPERATOR'S GUIDE TO DATA BACKUP AND RECOVERY

The remaining options are described below.

<i>Options</i>	<i>Description</i>
<b>{ -HELP -H }</b>	Displays BOOT_CREATE syntax and options.
<b>-MT [n]</b>	Specifies the tape drive unit number. If you omit <i>n</i> the unit number defaults to 0. If you omit the option, the system prompts you to enter the required unit number when you issue the BOOT_CREATE command.
<b>{ -NO_QUERY -NQ }</b>	Suppresses any prompt for the list file password. If you use this option and your list file is in a password-protected directory whose password has either not been specified or else has been incorrectly specified in the command line, BOOT_CREATE displays an error message and aborts.

### Creating a System Boot Tape

This section describes how to create a system boot tape. The command dialog is described, and an example is given of a typical BOOT\_CREATE session.

#### BOOT\_CREATE Command Dialog

You must assign a tape drive for the boot tape. You can do this in one of the following ways:

- Before you invoke BOOT\_CREATE, by using the ASSIGN command
- When you invoke BOOT\_CREATE, by using the -MT option on the command line
- During the BOOT\_CREATE dialog: BOOT\_CREATE prompts you to enter the tape drive unit number

Mount the reel of tape on the tape unit you intend to use, for example, tape unit number

1. It is advisable to label the reel physically with the following information:

- Your initials
- The date and time of the save
- The fact that this is a boot tape
- The name of the system in use
- The PRIMOS Revision number
- The recording density on tape
- The block size used

Issue the `BOOT_CREATE` command, for example,

```
OK, BOOT_CREATE test>masterfile>boot_list -MT 1
```

The above example instructs `BOOT_CREATE` to read the specified list file, and to write to the tape mounted on tape unit 1.

When you invoke `BOOT_CREATE`, you receive a series of prompts. These prompts and the appropriate responses are as follows.

## Prompt

## User Response

Does <diskname>dirname>subdirname have a password ?

`BOOT_CREATE` asks whether the directory that holds the list file is protected by a password. You do not receive this prompt if you have used the `-NO_QUERY` option in the command line.

If you answer YES, `BOOT_CREATE` asks you for the password. If you either fail to give the password or give an incorrect password, `BOOT_CREATE` aborts.

If the directory is not protected by a password, answer NO.

Enter List File name:

Enter the name of the list file. You receive this prompt only if you did not specify the list file on the `BOOT_CREATE` command line.

Which drive are you using

Give the number of the tape drive you want to assign. You do not receive this prompt if you have used the `-MT [n]` option in the command line.

Have you mounted the tape on drive <device number> ?

If you answer YES, `BOOT_CREATE` continues.

If you answer NO, `BOOT_CREATE` asks you to mount the correct tape on the tape drive.

To suppress all these prompts, specify the list file and options `-MT` and `-NO_QUERY` on the command line.

## Example BOOT\_CREATE Session

The following example shows the screen display for a typical `BOOT_CREATE` session. It is assumed that you are using PRIMOS Rev. 23.0. If you are using enhanced MAGSAV/RST at PRIMOS Rev. 21.0 or 22.0, the display would differ only in that the program `BOOT_RESTORE.SAVE` would be included in the list file instead of `MAGRSTRUN`. In this example, `BOOT_CREATE` reads the list file `BOOT_LIST`; the use of the command line options `-NO_QUERY` and `-MT 0` suppresses the command prompts. The example assumes that file `BOOT_LIST` is the example file described earlier in this chapter.

## OPERATOR'S GUIDE TO DATA BACKUP AND RECOVERY

OK. BOOT\_CREATE TEST>MASTERFILE>BOOT\_LIST -MT 0 -NO\_QUERY

[BOOT\_CREATE Rev.23.0 Copyright (c) Prime Computer, Inc. 1990]

No Warnings

Assigning Tape drive 0 to your job .....

Device MTO assigned.

Calling MAGSAV .....

WARNING: Only one reel can be loaded during system boot

\*\*\*\*\* because MTBOOT does not handle continuation reels.

[MAGSAV Rev.23.0 Copyright (c) Prime Computer, Inc. 1990]

Tape unit (9 Trk): 0

Volume serial ID is MYVOL1

Enter logical tape number: 1

Tape name: BOOT1

Date (MM DD YY):

Rev no:

Name or Command: \$TTY

Name or Command: \$A CMDNCO

Name or Command: MAKE.SAVE

Name or Command: MAGRST.RUN

Name or Command: \$A MFD

Name or Command: PRIRUN

Name or Command: \$R

Device released.

## Booting from Tape

There are two main reasons for booting from tape:

- The command device partition has crashed.
- The user is installing a new PRIMOS revision.

This section describes how to use the boot tape to recreate the entire system. It does not, however, give detailed explanations of the procedures, and for a comprehensive explanation of all aspects of booting from tape you should refer to your CPU Handbook. The success of these procedures depends upon regular backups to enable you to recover lost files.

## Recreating a System

This section describes how to recreate your entire system after a command device crash. To boot the system from tape, you need to have a boot tape that contains a minimum working system, and a tape that contains a copy of the command partition. If you are using enhanced MAGSAV/RST at PRIMOS Rev. 21.0 or 22.0, both these tapes must be in non-ANSI format; if you are using PRIMOS Rev. 23.0, the tapes must be in ANSI format. These conditions are automatically ensured if you use BOOT\_CREATE.

When booting from tape, you must first boot a minimum operating system from tape, and then restore the command device partition. When you have done this, you can restore the other backup tapes.

## Procedure for Booting from Tape

This section describes the procedure for booting a PRIMOS Rev. 23.0 system from tape. If you are using enhanced MAGSAV/RST on a Rev. 21.0 or 22.0 system, the procedure is almost identical; the only difference is that the utility `BOOT_RESTORE.SAVE` would be stored on the boot tape and this utility would be booted from tape in Step 4 below.

To recreate the system after a command partition crash, you need the following:

- A disk on which to restore the command partition
- A boot tape containing `MAKE.SAVE`, `MAGRSTRUN`, and `PRIRUN`
- A backup tape that contains the command partition

The procedure below outlines the steps you should follow to boot the system from tape. For full details on booting from tape, refer to your CPU Handbook or to the *Operator's Guide to File System Maintenance*.

1. Boot `MAKE.SAVE` from the boot tape to create and format the command device. A typical command sequence is shown below. Refer to your CPU Handbook for details on switch settings to use for the `BOOT` command.

```
CP>SYSCLR
CP>BOOT 10205
DPM007: System booting, please wait
RUN FILE TREENAME=MAKE.SAVE
BOOTING FROM MT1 MAKE.SAVE
ENTER COMMAND LINE OPTIONS: -DISK pdev -DT dsktyp -FORMAT -NO_INIT
PARTITION NAME?ptnam
```

where *pdev* is the physical device number of the command partition you are creating with `MAKE`, *dsktyp* is the disk type you are formatting, and *ptnam* is the partition name. The system now creates and formats the command device.

2. If your system does not have a paging device, boot `MAKE.SAVE` again to create and format the paging device as shown below.

```
CP>SYSCLR
CP>BOOT 10205
DPM007: System booting, please wait
RUN FILE TREENAME=MAKE.SAVE
BOOTING FROM MT1 MAKE.SAVE
ENTER COMMAND LINE OPTIONS: -DISK pdev -DT dsktyp -FORMAT -NO_INIT -SPLIT
PARTITION NAME?ptnam
```

where *pdev* is the physical device number of the paging partition, *dsktyp* is the disk type you are formatting, and *ptnam* is the partition name. The system now creates and formats the paging device after prompting you for the number of paging records.

3. Boot `PRIMOS.SAVE` from the boot tape to create a minimum operating system. The octal parameter 100000 instructs the system to prompt you for the configuration parameters. A typical display is shown below:

```
CP>SYSCLR
CP>BOOT 100205
```



## OPERATOR'S GUIDE TO DATA BACKUP AND RECOVERY

```
DPM007: System booting, please wait
PHYSICAL DEVICE=MT1
RUN FILE TREENAME=PRIRUN>PRIMOS.SAVE
BOOTING FROM MT1 PRIRUN>PRIMOS.SAVE
Enter Command Device: 2420
Enter PAGING device: 100421
Enter Number Terminal Users: 4
```

The numbers 2420 and 100421 represent the physical device numbers for the command and paging devices respectively, and 4 is the number of terminal users on the system during the booting process. You should enter the correct numbers for your system.

4. Assign the tape drive on which the boot tape is mounted, and resume MAGRST as shown below:

```
OK, ASSIGN MT1
Device MT1 assigned
OK, ATTACH MFD
OK, MTRESUME MT1 MAGRST.RUN
```

5. When the prompt `tape unit:` appears, replace the boot tape on MT1 with the backup tape containing the command partition, and restore that to disk.
6. Alternatively, you can use a different tape unit to restore the command partition backup tape. In this case, assign the unit at the same time as you assign the unit for the boot tape, and mount the command partition tape on this unit. Issue the `MTRESUME` command as described in step 4, and when the prompt `tape unit:` appears, enter the unit number for the command partition tape to start the restore.
7. When you have restored the command partition, you can use `NSED` to edit the `PRIMOS.COM1` and `CONFIG` files and to specify physical device numbers for the new paging and command devices.
8. You can now invoke `MAGRST` from disk to restore the rest of the system.
9. Reboot `PRIMOS` from disk following the procedure described in your CPU Handbook.

## TAPE INDEXING

### Introduction

This chapter describes the format of the indexes you can generate during save and restore operations.

The MAGSAV and MAGRST options that you use for generating the indexes are described in Chapters 2 and 4, respectively.

### Overview

You can generate indexes which give details of the tape contents of both save and restore operations. You can generate the index during the operation, or you can generate an index of the contents of a previously saved tape, without actually restoring the tape.

The index generated may be a summary index, or may be an extended index that contains more detailed information about the tape contents. Summary indexes may be written to file or displayed on the screen. Extended indexes are always written to file.

### Summary Index Format

This type of index can be written to a specified file during save or restore operations, or can be displayed on your screen during the operation. You can also generate an index of this type in order to determine the contents of a tape previously saved using MAGSAV, without actually restoring the tape. The format of a summary tape index is as follows:

- |        |   |
|--------|---|
| Header | There is one header entry for each save. The header contains the following information: <ul style="list-style-type: none"><li>● The Volume serial ID</li><li>● The logical tape (or save) name and number</li></ul> |
|--------|---|

- The save date
- The PRIMOS Revision number
- The reel number of the current reel
- The maximum directory level

**Object Information**      This contains the following details for each object:

- The object type, for example, SAM, DAM, CAM, UFD
- The name of the saved object

## **Extended Index Format**

The format of an extended tape index is as follows:

**Header**      There is one header entry for each save. The header contains the following information:

- Date and time of creation of index
- The Volume serial ID and save number of the save
- The save type, for example, Base, Cumulative Increment, or True Increment
- The save date and time
- The Volume ID of the current reel
- The reel number of the current reel if this is a continuation reel

**Reel Trailer**      Contains information about continuation volumes.

**Checkpoints**      Whenever a checkpoint is generated (during save operations) or encountered (during restore operations), the checkpoint number is written to the index.

**Object Information**      This contains the following details for each object:

- The Date and Time Modified (DTM) for the saved file, in the form YYMMDD.HHMMSS
- The object type, for example, SAM, DAM, CAM, DIR
- The name of the saved object, indented to indicate its level in the directory hierarchy

### **Note**

In the case of incremental saves, all directory names are stored in the index, irrespective of whether or not the directory has been modified. The DTM of modified directories is set to the appropriate value; that of unmodified directories is left blank.

## Examples of Indexes

This section gives four examples of an index for a save/restore operation on the same partition. These examples are as follows:

- Summary index for a base save
- Summary index for an incremental save
- Extended index for a base save
- Extended index for an incremental save

### Example 1

The following example shows the contents of a summary index for a base save.

```
MAGSR save index:- Volume serial ID: TAPE01 Logical tape name: MYSAVE
Logical tape no: 1 Save date: 22 03 90 Rev: 0 Reel number of save: 1
(Maximum directory level: 100)
```

```
STAFF_DIR (aclufd)
STAFF_DIR>BOB (aclufd)
STAFF_DIR>BOB>GOAL_LIST (dam)
STAFF_DIR>BOB>DATA_FILE (cam)
STAFF_DIR>BOB>DATABASE (aclufd)
STAFF_DIR>BOB>DATABASE>SEGDIR1 (segsam)
STAFF_DIR>BOB>DATABASE>SEGDIR2 (segdam)
STAFF_DIR>BOB>PROJECT_DESCRIPTION (sam)
STAFF_DIR>JANE (aclufd)
STAFF_DIR>JANE>DATA (sam)
STAFF_DIR>JANE>MYPROGRAM (dam)
STAFF_DIR>PROJECT_LIST (dam)
STAFF_DIR>FRED (aclufd)
STAFF_DIR>STAFF_LIST (sam)
```

End of save index

### Example 2

The following example shows the contents of a summary index for an incremental save.

```
MAGSR save index:- Volume serial ID: TAPE02 Logical tape name: INCSAV
Logical tape no: 1 Save date: 03 22 90 Rev: 0 Reel number of save: 1
(Maximum directory level: 10)
```

```
STAFF_DIR (aclufd)
STAFF_DIR>BOB (aclufd)
STAFF_DIR>BOB>GOAL_LIST (dam)
STAFF_DIR>BOB>DATABASE (aclufd)
STAFF_DIR>JANE (aclufd)
STAFF_DIR>PROJECT_LIST (dam)
STAFF_DIR>FRED (aclufd)
STAFF_DIR>HOLIDAY_LIST (sam)
```

End of save index

## OPERATOR'S GUIDE TO DATA BACKUP AND RECOVERY

### Example 3

The following example shows the contents of an extended index for a base save.

MAGSR Index File Version 2 . Created on 22 Mar 90 at 14:20:21  
Volume TAPE01 Save 001, Base Save on 22 Mar 90 at 13:51:28 Checkpoint 0001  
Start of save MYSAVE

```

                                <DSK01>MFD>
900322.134332 ADIR      STAFF_DIR>
900322.134638 ADIR      BOB>
900305.101944 DAM      GOAL_LIST
900322.132638 CAM      DATA_FILE
900322.133040 ADIR      DATABASE>
900322.133012 SEGS      SEGD1R1
900322.133040 SEGD      SEGD1R2
900114.184404 SAM      PROJECT_DESCRIPTION
900322.134900 ADIR      JANE>
900322.134704 SAM      DATA
900322.134756 DAM      MYPROGRAM
900305.101944 DAM      PROJECT_LIST
900322.133952 ADIR      FRED>
900114.184404 SAM      STAFF_LIST
```

End of Save MYSAVE on Volume TAPE01

### Example 4

The following example shows the contents of an extended index for an incremental save.

MAGSR Index File Version 2 Created on 22 Mar 90 at 14:22:24  
Volume TAPE02 Save 001, Cum Inc on 22 Mar 90 at 14:01:12 Checkpoint 0001  
Start of save INCSAV

```

                                <DSK01>MFD>
900322.135840 ADIR      STAFF_DIR>
                                BOB>
900305.135808 DAM      GOAL_LIST
900305.135744 DAM      PROJECT_LIST
900114.135818 SAM      HOLIDAY_LIST
```

End of Save INCSAV on Volume TAPE02

---

**PART II**  
**PHYSICAL SAVE & RESTORE**

---

## PHYSICAL SAVE TO TAPE: PSR

### Introduction

The following three chapters, 6, 7 and 8, document the operation of the Physical Save and Restore (PSR) utility at PRIMOS Revs. 22.0 and 23.0.

The PSR (Physical Save and Restore) command enables you to

- Save a physical disk partition to tape
- Restore a physical disk partition from tape
- Copy a physical disk partition to another disk partition
- Protect a partition from being accidentally overwritten by a restore or copy

This chapter describes how to use PSR to save a partition to tape. For details of how to restore and copy partitions, refer to Chapter 7, Restoring a Physical Save: PSR, and Chapter 8, Physical Copy between Disks.

### Overview

In save mode, PSR saves an entire physical disk partition to tape. It does so without reference to the logical structure of the data on disk and a partition is therefore the smallest unit that you can restore from a PSR save.

A physical save has the advantage of being considerably faster than a logical save of an entire partition but is not as convenient if you subsequently want to restore only a small number of files. However, a physical restore of an entire partition is much faster than a logical restore of a partition. You make physical saves in order to be able to recover as quickly as possible from, for example, a disk crash.

If you save a partition that has related files on another partition, remember to save this other partition also, to ensure that the files remain a logically consistent set. For example, a ROAM file may consist of a master file and several slave files, and the slave files might not be on the same partition as the master file.

**Note**

PSR cannot save the command device (COMDEV). To save the command device, use the logical utility MAGSAV.

When you use PSR to save a partition, PSR offers the following features that are specific to a save:

- Base saves
- Incremental saves
- Archive saves
- Multiple save streams
- Unique partition identifiers

PSR also has general features, applicable to a save, restore, or copy:

- Partition write protection
- Interactive interface
- Error handling
- Comprehensive help system

These features are described in the following sections.

### **Base, Incremental, and Archive Saves**

This section defines base saves, incremental saves, and archive saves. It also explains when you should use each type of save.

**Base Save:** A base save is a save of an entire partition, excluding any unused records on the partition. After a base save, PSR resets a date/time field in the partition's Record Availability Table (RAT); this field records the date and time of the base save. A base save provides the basis for subsequent incremental saves.

**Incremental Save:** An incremental save is a save of all the records on a partition that have changed since the most recent base save of that partition. An incremental save is faster than a base save, because it saves fewer records, and needs fewer tapes. The PSR incremental saves are cumulative saves. This means that an incremental save records all changes since the associated base save, not just changes since the previous incremental save.

**Archive Save:** When you run an archive save, PSR saves the same data as for a base save; that is, PSR saves the entire partition, excluding any unused records. However, for an archive save, PSR does not update the base save date/time field, which continues to record the date and time of the most recent base save. An archive save thus enables you to save a partition without affecting the cycle of base and incremental saves for that partition. It is useful, for example, when you want to save a partition to tape in order to restore it onto another system.



**Running Base and Incremental Saves:** After you have performed any of the following operations you must run a base save of a partition, rather than an incremental save.

- Run MAKE on the partition
- Run PSR to restore data to the partition
- Run PSR to copy data to the partition

All these operations set the base save date/time field in the partition's RAT to an invalid state, and you can only return the field to a valid setting by making a base save.

After a partition base save, future saves of that partition can be either base saves or incremental saves.

If you have made an incremental save, and you subsequently want to restore the partition, you must do this in two stages:

1. Restore the most recent incremental save.
2. Restore the most recent base save.

It is thus quicker to restore a partition on which you have run only base saves. However, this advantage is outweighed by the fact that it takes considerably longer to perform base saves than it does incremental saves. Save operations must be performed on a regular basis, in order to maintain an up-to-date backup of your system, whereas restore operations need to be performed only occasionally, for example, after a disk crash. You should, therefore, endeavor to reduce the speed of saves, in preference to that of restores. The section, Maximizing Performance, later in this chapter, suggests several ways of doing this.

#### Notes

If the most recent save of a partition was an incremental save, then when you use PSR to restore the partition, you restore the most recent incremental save *first*, and then the base save.

You cannot run incremental saves on pre-Rev. 22 partitions, because PSR cannot identify changed records on such partitions.

### Multiple Save Streams

PSR allows you to make parallel saves of the same partition to tape through a single invocation of PSR. Parallel saves are faster than several separate saves, and they guarantee that you have identical copies of the save. They enable you, for example, to write a security copy for offsite storage at the same time as you make a local copy.

When you invoke PSR it prompts you to define the first save, which is always on save stream 1. After you have defined the save on stream 1, you have the opportunity to specify further target devices for that save. For each target tape drive, PSR establishes another save stream. For example, you could specify that you want parallel saves to tape drives MT0, MT1 and MT4. For each drive, PSR would set up a separate save stream: stream 1 (for MT0), stream 2 (for MT1), and stream 3 (for MT4).

### Note

The tape volumes (reels) written on by a save stream are a single entity. When you make multiple parallel saves, take care not to mix the volumes written by different save streams. If you restore, for example, a two-volume save, you cannot use the first volume from save stream 1, and the second volume from save stream 2. This is because the point at which the break between volumes occurs is usually different on each stream.

### Unique Partition Identifier

PSR allows you to use a unique partition identifier to specify the partition that you want to save. This identifier consists of the disk volume ID and the partition's physical device number (pdev), in the format *volume\_id:pdev*. *volume\_id* is a user-created string that must comply with normal PRIMOS filename standards, and the System Administrator has to decide what naming conventions the IDs should follow. For example, a partition identifier could be PACK9:20662, where PACK9 is the name that the Administrator has given to the disk pack, and 20662 is the physical device number of a partition on that pack.

PSR also allows you to specify a partition by its pdev alone, but a pdev is not a unique identifier. For example, if you mount a disk pack on another drive, the pdevs of the partitions on that pack will probably change; or, if you remove a disk pack from a given drive, and mount a new pack, you may have partitions with the same pdevs as partitions on the previously mounted pack. To avoid such situations, it is advisable to use the unique identifier, *volume\_id:pdev*, when you save a partition.

When you restore a partition, PSR displays messages that use the partition identifier you supplied when you saved that partition. If you have used the unique identifier, the PSR messages output during a restore enable you to be sure that you are restoring the correct partition from the correct disk pack, and not a partition that just happens to have the same pdev as the partition you want to restore.

### Partition Write Protection

When you create a partition, using MAKE, it is automatically write protected against PSR. PSR allows you to remove this write protection before you restore or copy a partition and to reinstate the protection after the restore or copy operation has ended. For details of how to use PSR to control partition write protection, refer to Chapter 7, Restoring a Physical Save: PSR, and Chapter 8, Physical Copy between Disks.

If you save a partition that has been left in an unprotected state, PSR issues the warning message

```
Source device <id> is backup write enabled
```

where *id* is in the format *volume\_id:pdev*, or *pdev*, depending on how you identified the partition to PSR when you ran the save.

To reinstate write protection, issue the command

OK, PSR -PROTECT [id]

If you omit *id*, PSR prompts you for the partition ID.

#### Note

PSR write protection only restricts PSR restores and copies. It does not affect any other disk I/O operations.

### Interactive Interface

When you invoke PSR, you receive a series of prompts that enable you to control the save interactively. The sequence of prompts and responses is termed the PSR dialog, and is explained in the section, Invoking PSR, later in this chapter.

### Error Handling

PSR first attempts to recover from errors when writing to tape by rewriting the data. If this fails, PSR repositions the tape to the previous checkpoint, terminates the tape volume at that point, and prompts for a new volume. Checkpoints are markers that PSR writes to tape at regular intervals, to aid error recovery.

### PSR Help

To obtain general information about PSR, issue the command

OK, PSR -HELP

The following is an example of PSR -HELP:

[PSR Rev. 23.0 Copyright (c) 1990, Prime Computer, Inc.]  
One of the following must be supplied:

-SAVE	Save a disk partition
-RESTore	Restore a disk partition
-COPY	Copy a partition
-PROtect	Set DRB write protection on partition
-No_PROtect	Clear DRB write protection on partition

For help on one of the above, use -HELP with the option.  
For example, PSR -SAVE -HELP gives more details for -SAVE.

If you require specific help about how to use PSR to save a partition, issue the command

OK, PSR -SAVE -HELP

During the PSR dialog, you can obtain help about how to respond to a PSR prompt by entering HELP in response to the prompt.

The following is an example of PSR -SAVE -HELP:

## OPERATOR'S GUIDE TO DATA BACKUP AND RECOVERY

[PSR Rev. 23.0 Copyright (c) 1990, Prime Computer, Inc.]

The **-SAVE** option is used to save a disk partition to tape.

The command line format is

**PSR -SAVE** [source{-BASE | -INC | -ARCHIVE}] [-No\_Volume\_ID] [-NO\_RAT] [-UNLOAD]

where

<b>source</b>	Identifies the source disk partition
<b>-BASE</b>	Performs a base save
<b>-INC</b>	Performs an incremental save
<b>-ARCHIVE</b>	Performs an archive save
<b>-No_Volume_ID</b>	Tapes do not have unique volume IDs
<b>-NO_RAT</b>	Only use this option at the direction of your Prime Customer Service representative, to prepare a tape to send to another site for specialized data recovery
<b>-UNLOAD</b>	Unload the tape at the end of reel

### Invoking PSR

The format of the PSR command to save a partition is:

► **PSR -SAVE**  $\left[ \begin{array}{l} \{ \text{vol\_id:pdev} \\ \text{pdev} \} \end{array} \right] \left\{ \begin{array}{l} \text{-ARCHIVE [-NO\_RAT]} \\ \text{-INC} \\ \text{-BASE} \end{array} \right\} \left[ \begin{array}{l} \{ \text{-No\_Volume\_ID} \} \\ \text{-NVID} \\ \text{-UNLOAD} \end{array} \right]$

The command options are described below:

<i>Option</i>	<i>Description</i>
<b>-SAVE</b>	Tells PSR that you want to save data to tape.
<b>volume_id</b>	A user-created string that names the disk pack, for example PACK6. This is an optional part of the command line, but is recommended as a means of uniquely identifying each disk pack.
<b>pdev</b>	The partition's physical device number. You can specify the partition by just <i>pdev</i> , or you can use <i>pdev</i> in conjunction with <i>volume_id</i> :. If you specify both <i>volume_id</i> : and <i>pdev</i> , the format is <i>volume_id.pdev</i> .
<b>-BASE</b>	Tells PSR to run a base save that is to be the start of a sequence of incremental saves.
<b>-INC</b>	Tells PSR to make an incremental save.
<b>-ARCHIVE</b>	Tells PSR to make an archive save.
$\left\{ \begin{array}{l} \text{-NO\_VOLUME\_ID} \\ \text{-NVID} \end{array} \right\}$	Tells PSR that you are not using uniquely identified tapes, and that it should not prompt you for the volume ID.

**-NO\_RAT** This option can only be used in conjunction with both the **-SAVE** and **-ARCHIVE** options. The option causes PSR to save all the records on the partition, rather than only those marked as in use in the partition's Record Availability Table. You should only use this option at the direction of your Prime Customer Service representative, to prepare a tape to send to another site for specialized data recovery.

**-UNLOAD** If this option is specified, when the end of the tape is reached the tape is rewound and then unloaded.

Valid command lines are:

OK, PSR -SAVE (the default command)

OK, PSR -SAVE pdev -BASE

OK, PSR -SAVE pdev -INC

OK, PSR -SAVE pdev -ARCHIVE

OK, PSR -SAVE pdev (pre-Rev. 22.0 partitions only:  
generates an archive save)

**Note**

If you save a pre-Rev. 22.0 partition, you can only make archive saves. You cannot make either base or incremental saves

After you issue a PSR command, you enter the PSR dialog, which is described later in this chapter in the section, PSR Dialog. The content of the dialog depends upon the PSR command line.

For example, if you have a Rev. 22.0 partition, and you issue the default PSR command

OK, PSR -SAVE

the PSR dialog begins by prompting you to identify which partition you want to save, and to specify the type of save you require. With a pre-Rev. 22.0 partition, this default command would also initiate the PSR dialog, but PSR would not prompt for the type of save; it would assume you require an archive save.

The **-BASE**, **-INC**, and **-ARCHIVE** options enable you to specify on the command line that you want to run a base, an incremental, or an archive save. For example, the commands

OK, PSR -SAVE 20662 -BASE

and

OK, PSR -SAVE PACK9:20662 -BASE

## OPERATOR'S GUIDE TO DATA BACKUP AND RECOVERY

both tell PSR that you want to run a base save of physical device number 20662. The PSR dialog that follows the command does not prompt for the source disk, nor for the type of save. In the second example, the command line identifies the disk pack as PACK9.

To specify on the command line that you want to run an incremental save, use the `-INC` option instead of the `-BASE` option. For example, the command

```
OK, PSR -SAVE PACK9:20662 -INC
```

tells PSR that you want to run an incremental save of pdev 20662.

To specify that you want to run an archive save, use the `-ARCHIVE` option on the command line. For example, the command

```
OK, PSR -SAVE PACK9:20662 -ARCHIVE
```

tells PSR that you want to archive pdev 20662.

If you have a pre-Rev. 22.0 partition, you can only make archive saves, and you do not need to include the `-ARCHIVE` option on the PSR command line. The command

```
OK, PSR -SAVE PACK9:20662
```

tells PSR that you want to archive the pre-Rev. 22.0 partition pdev 20662. If you specify a partition that is not in pre-Rev. 22.0 format, you receive an error message.

### PSR Dialog

This section describes the PSR prompts and messages you receive when you run a save. You do not receive all the prompts listed below, because the dialog depends upon the PSR command line, and upon the format of the partition (pre-Rev. 22.0, or Revs. 22.0 and later). There are also slight variations in the dialog if you use a cartridge tape drive; the description of the dialog identifies these variations.

For online help about how to respond to a prompt, enter `HELP`. To exit from the dialog, enter `QUIT` in response to any of the PSR prompts. Refer to the section, Example Command Lines and Dialogs, later in this chapter, for examples of the PSR dialog.

#### Prompts & Messages

#### User Response

Enter source disk:

You receive this prompt if you have not specified the partition on the PSR command line. Enter the partition ID in the format `volume_id:pdev` or `pdev`.

Is this a Base (B)  
Incremental (I)  
the previous  
Archive (A) save?

You receive this prompt if you issued the default PSR command, and you specified a Rev. 22.0 or later partition in response to the prompt, Enter source disk.  
Enter B to run a base save.  
Enter I to run an incremental save.  
Enter A to run an archive save.

Enter media for save  
stream 1 Use physical device:

Enter the tape drive identifier, for example MT0.

Volume ID is <serial-id>

If the tape is labeled, PSR reports the volume serial ID. No user action is needed, other than to check that you have mounted the correct tape.

Media does not contain  
a valid volume label  
Enter volume ID:

You receive this prompt if the tape is not labeled, and you have not included the `-NO_VOLUME_ID` option on the PSR command line.

Enter the serial ID of the tape volume you have mounted. The ID is a string of 1 through 6 characters that uniquely identify the reel.

The string is ANSI-87 standard and can be alphanumeric characters or symbols excepting `# $ \ [ ] ^ _ { } |` and `'`.

The ID should be the same as the name written on the paper label on the reel.

If you specify fewer than six characters, they are blank-padded to the right.

Displays details of tape's  
position, the save at  
that position, and the  
date the save was made

PSR displays several messages to inform you about the content of the tape. No user action is required, other than to check that you have mounted the correct tape.

Enter required save number  
or Append

If the tape holds data, PSR prompts for the point at which you want to position the tape. Enter the save number at which you want to position the tape. For example, first save is 1, second, 2, etc. If you enter a save number that already exists on the tape, PSR overwrites the existing save without asking for confirmation.

If you want to append to the tape, enter A.

If you save a partition to a cartridge tape, you do not receive a prompt for a save number. You can only write from the start of the tape or append to the tape and you receive the following prompt:

## **OPERATOR'S GUIDE TO DATA BACKUP AND RECOVERY**

**Append (A) or Overwrite (O)  
the media?**

You receive this prompt, instead of the save number prompt above, if you save a partition to a cartridge tape.

Enter A if you want PSR to append the save to the tape.

Enter O if you want PSR to write from the beginning of the tape.

**Enter recording density, (bpi):**

If the save begins at the start of the tape, PSR prompts for the density at which you want to write the data. Enter the density you want, or press the RETURN key to select the density for the tape drive previously specified when using the ASSIGN command.

You do not receive this prompt if you save to a cartridge tape drive that only records at one density.

**Enter media for**

**save stream 2**

**Use physical device:**

To define a parallel save on stream 2, enter the tape drive unit number for this save stream, for example MT1. PSR then repeats the prompts described above, for the volume ID, save number, and recording density.

If you want to terminate the definition of save streams, and to begin the save, press the RETURN key at this prompt.

**This save is not part of an incremental save sequence.**

**OR**

**This base save is the start of a new incremental save sequence.**

**OR**

**The base save for this incremental was made on <date-and-time>**

PSR issues one of these messages reminding you of the type of save invoked: archive, base, or incremental.

**Operation complete.**

PSR issues this message when the save has ended successfully.

**Please record the following information for use when restoring:**

**<details of save type and saved partition>**

**<details of stream, volume serial ID, and save number>**

You should record the information that PSR displays about the save. You need this information if you have to restore the save.



## End of Volume

If a save reaches the end of a volume, PSR initiates the following dialog.

### Prompts & Messages

### User Response

End of media detected on:  
<serial-ID [tape-drive-ID]>  
Enter media for continuation.  
Use physical device

Mount the continuation volume, and enter the unit number of the tape drive on which it is mounted, for example MT2.

Displays details of the  
tape's position, the save  
at that position, and the  
date the save was made

PSR displays several messages to inform you about the content of the tape. No user action is needed, other than to check that you have mounted the correct tape.

OK to overwrite the media?

You cannot append to a continuation volume. If the volume already holds data, PSR requests permission to overwrite it. To continue the save, and overwrite the tape, enter Y.

If you do not want to overwrite the tape, answer N. PSR then repeats the cycle of prompts, starting with the prompt use physical device:. Mount a new continuation reel, and begin the end-of-volume procedure again.

## Save Procedure for an Entire Partition

You should first decide whether to shut down the entire system to users, or just the partition that you want to back up. For a full discussion of preparation procedures to follow before saving and restoring data, refer to Chapter 10, System Preparation, Shutdown, and Startup.

Shown below are the steps you should follow to use PSR -SAVE to save an entire partition.

If you have decided to shut down the entire system, follow the procedure described in Chapter 10 of this guide under the section, Procedure for Shutting Down the System.

If you have decided to shut down only the partition, follow the procedure described in Chapter 10 of this guide, under the section, Procedure for Removing a Partition from Service.

The following steps must be executed from your supervisor terminal if you have shut down the system, otherwise they may be executed from a user terminal.

1. Assign the tape units for the tape that you are going to save. For example:

```
OK, ASSIGN_MTO
OK, ASSIGN_MT1
```

2. Mount the reel of tape on the assigned drive. It is advisable to physically label the reel with the following information:

## **OPERATOR'S GUIDE TO DATA BACKUP AND RECOVERY**

- Your initials
- The date and time of the save
- The disk pack identifier, if you intend to use this on the PSR command line (for example, PACK2)
- The pdev of the partition being saved
- The name of the partition being saved
- Whether the partition is robust or standard
- The name of the system in use
- The disk format Revision number
- The recording density on tape
- The block size used
- The reel number within the volume set, for example, TAPE 1 of N, where *N* represents the number of reels within the set, and should be filled in once the backup is complete
- The fact that PSR -SAVE has been used for the save

You should also leave room on the label for the volume serial ID, if it is not labeled permanently, and for the size of the save.

3. Invoke PSR -SAVE and enter the required response when prompted to do so. Examples of command lines and the ensuing dialog are described in the section, Example Command Lines and Dialogs, later in this chapter.
4. When PSR has completed the save, remove the last tape. It is important to fill in the blanks on the tape reel labels to indicate the number of tapes in the set, the size of the save and to add the volume serial ID if the reel is not already labeled. Remember also to remove the write-enable rings from the backup tapes, to avoid accidental overwriting.
5. Unassign the tape unit(s). For example:

```
OK, UNASSIGN_MTO  
OK, UNASSIGN_MT1
```

6. If you have shut down the entire system, restart the system by following the procedure described in Chapter 10 of this guide, under the section, Procedure for Starting Up the System.  
  
If you have shut down only the partition, return it to service by following the procedure described in Chapter 10 of this guide, under the section, Procedure for Returning a Partition to Service.

### **Example Command Lines and Dialogs**

This section gives examples of PSR command lines and dialogs to:

- Make a base save that spans two volumes of tape
- Append an archive save to a tape

- Make an incremental save

There is also an example CPL program that drives a PSR save.

### Making a Base Save to Two Volumes

The following example is of a base save that spans two reels. The continuation reel already contains data and PSR prompts for confirmation to overwrite it.

```
OK, PSR -SAVE PACK7:20882 -BASE
[PSR Rev. 23.0 Copyright (c) 1990, Prime Computer, Inc.]
Enter media for save stream 1.
Use physical device: MT0
Volume ID is 117202.
This volume begins with save number 1,
a physical base save of 20882
created on 22 Mar 90 15:55:24 Tuesday.
Enter required save number or Append (A): 1
Positioning to beginning of media ...
Enter recording density (bpi) : 8250
Enter media for save stream 2.
Use physical device: RETURN
```

```
This base save is the start of a new incremental save sequence.
Saving...
End of media detected on 117202 (MT0).
Enter media for continuation.
Use physical device: MT1
Positioning to beginning of media ...
Volume ID is 117205.
This volume begins with save number 1,
a continuation of a physical base save of 20882
created on 22 Mar 90 15:55:24 Tuesday.
```

```
OK to overwrite the media ? Y
Positioning to beginning of media ...
Operation complete.
Transferred 25350 records.
```

```
Please record the following information for use when restoring:
Media contains base save of Rev 23 partition PACK7:20882
Stream 1 initial volume is 117202, save number 1
and continues on volume(s) 117205
OK,
```

### Appending an Archive Save

The following example shows the prompts you receive when you use the default PSR command. If there were only one save on tape IKT001, this example would append save 2, an archive save, to the tape. However, if save 2 already existed, PSR would automatically overwrite it with the archive save. PSR would not ask you for confirmation that you wanted

## OPERATOR'S GUIDE TO DATA BACKUP AND RECOVERY

to overwrite save 2, and the dialog would be the same as if PSR were appending the save. Alternatively, if you entered append (A), PSR would not overwrite any existing saves, but would start after the last save on tape.

OK, PSR -SAVE

[PSR Rev. 23.0 Copyright (c) 1990, Prime Computer, Inc.]

Enter source disk: 20662

Is this a Base (B), Incremental (I) or Archive (A) save? A

Enter media for save stream 1.

Use physical device: MTQ

Volume ID is IKT001.

This volume begins with save number 1,

a physical base save of 20662

created on 22 Mar 90 15:55:24 Tuesday.

Enter required save number or Append (A): 2

Positioning to required save ...

Enter media for save stream 2.

Use physical device: RETURN

This save is not part of an incremental save sequence.

Saving...

Operation complete.

Transferred 25350 records.

Please record the following information for use when restoring:

Media contains archive save of Rev 23 partition 20662

Stream 1 initial volume is IKT001, save number 2

OK,

### Making an Incremental Save

In the following example, the PSR command line specifies that PSR is to run an incremental save of partition 10662, which is on the disk named PACK3. This incremental save is save 2 on tape DRB001, which already holds a previous incremental save of the partition.

OK, PSR -SAVE PACK3:10662 -INC

[PSR Rev. 23.0 Copyright (c) 1990, Prime Computer, Inc.]

Enter media for save stream 1.

Use physical device: MTQ

Volume ID is DRB001.

The volume is positioned at the end of save number 1,

a physical cumulative incremental save of PACK3:10662

created on 24 Mar 90 11:45:19 Thursday.

Enter required save number: 2

Enter media for save stream 2.

Use physical device: RETURN

The base save for this increment was made on 22 Mar 90 15:55:24

Saving ...

Operation complete.  
Transferred 2500 records (10% of records in use).

Please record the following information for use when restoring:  
Media contains cumulative incremental save of Rev 23 partition PACK3:10662  
Stream 1 initial volume is DRB001, save number 2  
OK,

## CPL Program to Drive a PSR Base Save

The following is a very simple CPL program example, that would drive a PSR base save. There is no error checking or assigning of disks, tape drives, and so on included. You have to customize the program accordingly, depending on your particular circumstances. The program is named PSR\_SAVE.CPL.

```
/* Example of how to invoke PSR -SAVE from CPL
/*
/* START-DESCRIPTION
/*
/* This is a very simple program to demonstrate the use of CPL to
/* run the physical backup product, PSR, to do a base save of a
/* partition.
/*
/* Usage: PSR_SAVE <pdev> -TO <MTn>
/*
/* <pdev> is the physical device number of the partition to be
/* saved. <MTn> is the drive number on which the save will be
/* started; if the tape already contains data this save will be
/* appended to the end. Continuation devices are prompted
/* for when required. The user must assign the partition to be
/* saved, and the tape drives to be used, before running this
/* program. All media used is assumed to be labelled either
/* explicitly by the LABEL command or by a previous use by PSR.
/*
/* END-DESCRIPTION
/*
/* START-CODE
/*
/* Parse the command line
/*
/*args pdev:oct; -
/*    to_arg:-TO first_mt:char
/*
/* Check if the source pdev and the first MT unit were supplied
/*
/*if [null %pdev%] | [null %first_mt%] -
/*then -
/*    &stop 1 &message Usage: PSR_SAVE <pdev> -TO <MTn>
/*
/* Set variable for null responses
/*
/*set_var null :=
/*
/* Run PSR to do the save
/*
/*data PSR -SAVE
/*    %pdev%                                /*source device
```

## OPERATOR'S GUIDE TO DATA BACKUP AND RECOVERY

```
Base                               /* save tape
%first_mt                         /* device for stream 1
Append                            /* save number
%null%                           /* density
%null%                           /* device for stream 2
&tty                             /* continuation devices, if required
&end
/*
/* Check error severity
/*
  &if %severity% = 0 -
  &then -
    &stop 0 message * * * Save of %pdev% completed OK.
  &else -
    &stop %severity% &message * * * Save of %pdev% failed.
/*
/* END-CODE
/*
```

### Unattended Backup

The very high data capacity of the 8mm cartridge tape drive reduces the number of tapes required to store backup data. This may eliminate the need for operator intervention, and thus allow you to perform backup operations overnight.

Some systems have a total backup requirement of less than the capacity of a single 8mm cartridge tape. Larger systems require the use of multiple tapes. If you need to use multiple tapes for backup operations, the ability to run unattended backups will depend on your system having enough tape drives for you to mount all the required tapes before you initiate the backup. You can use the CPL program described in the following section to run unattended backups using multiple tapes, provided you mount and assign all the required tapes before you initiate the backup.

### CPL Program for Multiple Tape Backups

You can use the CPL program described below to run a backup which requires more than one tape. The program allows for the overflow of a single partition onto more than one tape.

If you run PSR -SAVE under CPL control, the system sets three local variables when the operation terminates. These variables are as follows:

- DR\$TAPE\_UNIT is the tape drive unit number on which the save finished.
- DR\$SAVE\_NUMBER is the number of the last save written.
- DR\$VOLID is the final volume ID.

These variables give the tape unit and save number of the save which has just completed, thus enabling you to calculate the tape unit and and save number for the next save.

The following CPL program performs a base save on each of three partitions, whose physical device numbers are 10664, 20664, 30664. The saves are written to tapes which have been

previously mounted on tape units numbered sequentially from 0. These units must be assigned prior to the start of the CPL program.

The first partition is saved as save number 1, the second as save number 2, and the third as save number 3. Any existing data on these tapes will be overwritten. All media used is assumed to have been labelled, either explicitly by the LABEL command, or by previous use of PSR.

```
&set_var partition_list := 10864 20864 30864 /*Set pdevs of partitions

&set_var DR$TAPE_UNIT := 0                /*First tape unit number

&set_var DR$SAVE_NUMBER := 1              /*First save number

&set_var null :=                          /*Set variable for null responses

&do pdev &list %partition_list%           /*Start of loop for each partition

    &data PSR -SAVE
        %pdev%                            /*Source device
        Base                             /*Save type
        MT%DR$TAPE_UNIT%                  /*Tape unit no. for stream 1
        %DR$SAVE_NUMBER%                  /*Save no.
        &if %DR$SAVE_NUMBER% = 1 -
        &then
            8250                          /*Density for first save only
            %null%                        /*No second stream
            MT[calc %DR$TAPE_UNIT% + 1]    /*Increment tape unit no.
            Yes                            /*OK to overwrite continuation tape
                                           /*Enter further continuation
                                           /*devices here if necessary

    &end
    &set_var DR$SAVE_NUMBER := [calc %DR$SAVE_NUMBER% + 1]
                                           /*Increment save no.

&end                                       /*End of loop
```

If the end of tape is reached during the partition save, the system prompts for the continuation unit number; this is returned to the system in the above program at the line where the tape unit number is incremented. The partition may overflow onto several physical tapes; in each case the next tape accessed is that on the next sequential tape unit number.

If the system does not reach the end of tape during the partition save, PSR -SAVE reads the continuation tape unit number, as calculated in the above program, and saves the next partition onto this tape.

In either case, the save number is incremented only when the next partition is saved.

More complex CPL programs may be written to generate tape unit numbers in some other order, for example, from a list. Note that due to the limit on the size of a partition, a partition save would not normally continue onto a third 8mm tape.

## **Maximizing Performance**

There are a number of ways in which you can maximize the speed of a backup:

- Use alternate tape drives.
- For multistream saves, use tape drives on different controllers.
- Use incremental saves.
- Schedule the save for a time when the system is lightly loaded.
- Close the entire system to users before you begin the backup.

If a save requires more than one reel of tape, you can reduce the time that a save takes by using alternate tape drives. Mount the continuation reel on the unused drive while the save proceeds on the other drive. When PSR reaches the end of the reel and prompts for the next reel, you can quickly resume the save on the standby drive.

If you run a multistream save, the save will be faster if you select tape drives on different controllers.

An incremental save is normally faster than a base save, because the incremental only saves records that have changed since the most recent base save; PSR saves fewer records than for a base save, and needs fewer reels. You should therefore assess the rate of change of the data you want to save, and only run base saves when a large proportion of the records have changed since the previous base save.

The save also proceeds more quickly if the system is lightly loaded, or closed entirely to users. The section, Procedure for Shutting Down the System, in Chapter 10 explains how to close the system to users as part of the save procedure.



## RESTORING A PHYSICAL SAVE: PSR

### Introduction

The PSR (Physical Save and Restore) command enables you to

- Save a physical disk partition to tape
- Restore a physical disk partition from tape
- Copy a physical disk partition to another disk partition
- Protect a partition from being accidentally overwritten by a restore or a copy

This chapter describes how to use PSR to restore a partition from tape. For details of how to save and copy a partition refer to Chapter 6, Physical Save to Tape: PSR, and Chapter 8, Physical Copy between Disks.

### Overview of PSR Restore

In restore mode, PSR restores an entire partition from tape. The partition must have been saved to tape by PSR in its save mode. You cannot restore parts of a partition with PSR.

Typically, you use PSR to restore a partition after a disk crash, in order to return the data to a known point in time. Because a physical restore of an entire partition is much faster than a logical restore, using PSR offers you the fastest means of recovery from a disk crash.

#### Note

PSR cannot restore to the command device. To save and restore a command device, you must use logical save and restore utilities.

### Restoring Incremental and Base Saves

If the most recent save of a partition was an incremental save, you must *first* restore the incremental save and then restore the associated base save. When you have restored the incremental save, PSR identifies the associated base save, and prompts you to restore

that save. PSR does allow you to restore a base save without restoring an associated incremental save, but the restored partition would not include any data changes made since the base save.

### **Related Files**

If you restore a partition that has related files on another partition, remember to restore this other partition also, to ensure that the files remain a logically consistent set. For example, a ROAM file may consist of a master file and several slave files, and the slave files might not be on the same partition as the master file.

### **PSR Features**

In restore mode, PSR has the following features:

- The target disk can differ in size and geometry from the source disk.
- Partitions can be protected against an accidental restore.

PSR also has general features, applicable to a save, restore, or copy:

- Interactive interface
- Error handling
- Comprehensive help system

These features are described in the following sections.

### **Valid Source and Target Partitions**

PSR is very flexible, in that it allows you to restore a partition to a disk that differs from the source disk in terms of size and geometry. The term disk geometry covers the number of

- Sectors per track
- Cylinders
- Heads

The only restrictions on a restore are that

- You cannot restore to the command device.
- The target partition must be large enough to hold the restored data.
- The source and target partitions must be the same revision.
- You must restore a robust partition to another robust partition, and a standard partition to a standard partition.

### Partition Write Protection

When you create a partition using MAKE, it is automatically write protected against PSR. This guards the partition against an accidental overwrite, for example, if you attempt to restore a partition to the wrong disk pack.

PSR enables you to remove the write protection before you restore a partition, and to reinstate the write protection for that partition after the restore has ended.

If you attempt to restore to a protected partition, PSR displays the warning message

```
Target device <id> is backup write protected
```

where *id* is the partition identifier and you will receive the prompt to identify the partition you want to restore again. In response you can enter another partition identifier or enter QUIT to exit from the restore (for more information see Chapter 6).

If you exit from the restore, you can remove write protection by issuing the command

```
OK, PSR -NO_PROTECT pdev
```

where *pdev* is the physical device number of the partition from which you want to remove write protection.

When you have completed the restore, you should reinstate the protection on the partition by issuing the command

```
OK, PSR -PROTECT pdev
```

The section, Restore Procedure, later in this chapter, explains how to remove and reinstate partition write protection as part of the step-by-step restore procedure.

#### Note

PSR write protection only restricts PSR restores and copies. It does not affect any other disk I/O operations.

### Interactive Interface

When you invoke PSR, you receive a series of prompts that enable you to define the restore interactively. The sequence of prompts and responses is termed the PSR dialog, and is explained in the section, Invoking PSR, later in this chapter.

### Error Handling

If unrecoverable tape read errors occur during a restore, PSR displays a detailed error message and asks:

```
Do you wish to continue with the restore?
```

## OPERATOR'S GUIDE TO DATA BACKUP AND RECOVERY

If you answer NO, the restore aborts. You must then run MAKE on the target partition, because it will be in an inconsistent state. You are not able to rerun the restore until you have run MAKE on the target partition.

If you answer YES, the restore continues, and PSR displays the warning

Disk records have been lost

PSR also checks for errors on the target partition. You can ask PSR, via either the PSR dialog or a command line option, to read back the data it has written to the target partition. The read is a means of checking whether the disk has new badspots. If PSR cannot read some of the data, it recovers by writing the data elsewhere on the partition.

---

### Caution

If errors are detected and recovered by PSR, you must run FIX\_DISK to reset record pointers at the end of the restore.

---

## PSR Help

To obtain general information about PSR, issue the command

OK, PSR -HELP

The following is an example of PSR -HELP

[PSR Rev. 23.0 Copyright (c) 1990, Prime Computer, Inc.]  
One of the following must be supplied:

-SAVE	Save a disk partition
-REStore	Restore a disk partition
-COPY	Copy a partition
-PROtect	Set DRB write protection on partition
-No_PROtect	Clear DRB write protection on partition

For help on one of the above, use -HELP with the option.  
For example, PSR -SAVE -HELP will give more details for -SAVE.

If you require specific help about how to use PSR to restore a partition, issue the command

OK, PSR -RESTORE -HELP

During the PSR dialog, you can obtain help about how to respond to a PSR prompt by entering HELP in response to the prompt.

The following is an example of PSR -RESTORE -HELP

[PSR Rev. 23.0 Copyright (c) 1990, Prime Computer, Inc.]

The **-RESTore** option is used to restore a disk partition from tape.

The command line format is

**PSR -RESTore [-CHECK] [-UNLOAD]**

where

**-CHECK**      Performs read after write verify

**-UNLOAD**     Unload tape at end of reel

## Invoking PSR

The format of the PSR command to restore a partition is:

► **PSR -RESTORE [option ...]**

The command options are described below:

<i>Option</i>	<i>Description</i>
<b>-RESTORE</b>	Tells PSR that you want to restore data from tape.
<b>-CHECK</b>	Specifies that you want PSR to read back the data it restores to the target partition in order to check that the restored data is readable. The restore operation is slower if you use this option.
<b>-UNLOAD</b>	Specifies that when the end of tape is reached, the tape is rewound and unloaded.

After you issue the PSR command, you begin the PSR dialog, which is described in the next section.

## PSR Dialog

This section describes the PSR dialog when you invoke the default PSR restore command.

If you are unsure how to respond to any PSR prompt, enter **HELP** at that prompt and the online help information will be displayed.

To exit from the dialog, enter **QUIT** in response to any of the PSR prompts.

Refer to the section, **Example Command Lines and Dialogs**, later in this chapter, for examples of the PSR dialog.

**Restoring an Archive or a Base Save:** This section describes the dialog for restoring either an archive or a base save. It also describes the first part of the dialog for restoring an incremental save; if you restore an incremental save you follow the dialog as far as the prompt **check read after disk writes?**, and after that prompt you continue with the dialog described in the next section, **Restoring an Incremental and a Base Save**.

## OPERATOR'S GUIDE TO DATA BACKUP AND RECOVERY

### Note

If you are restoring a base and incremental save combination, you should restore the incremental save first.

#### Prompts & messages

#### User Response

Enter media for restore  
stream. Use physical device:

Enter the tape drive identifier, for example MT0.

Displays details of  
the volume serial ID,  
the tape's position,  
the save at that position,  
and the date  
the save was made

PSR displays several messages to inform you about the content of the tape. No user action required, other than to check that you have mounted the correct tape.

Enter required save number:

Enter the save number at which you want to position the tape. For example, first save is 1, second, 2, etc.

<details of the save>

PSR displays details of the save you have selected. No user action is needed, other than to check that you have selected the correct save.

Target disk:

Enter the physical device number of the target partition.

Check read after  
disk writes?

Answer Y if you want PSR to attempt to read back the data it has written to disk. The read is a means of checking for new badspots on the target disk and it adds to the total time required for the restore.

If you do not want PSR to read back the data, answer N.

If you are restoring an incremental save, the dialog continues as described in the next section.

Operation complete.

PSR issues this message when the restore has ended successfully.

**Restoring an Incremental and a Base Save:** If you restore an incremental save, the dialog is as described above, up to the prompt check read after disk writes?. After this prompt, the dialog continues as follows:

**Prompts and Messages**

**User Response**

End of restore of increment.

PSR has restored the incremental save. No user action is needed.

Please load the volume containing the base save of <partition-id> created on <date>

Enter the ID of the tape drive on which the base save tape is mounted, for example, MT1.

Use physical device:

<details of the volume serial ID, and the base save number>

PSR displays several informational messages. No user action is needed, because PSR finds the base save associated with the incremental save you have just restored.

Operation complete.

PSR has restored the base save, and the whole restore operation has ended successfully.

**Error Recovery**

If you answered YES to the prompt check read after disk writes, PSR displays the following message if a read error occurs:

Unrecovered disk read error reading record <nnn> from device <pdev>.  
Record <nnn> on device <pdev> has been added to the badspot file.

---

**Caution**

If you receive this message, you must run FIX\_DISK on the target partition after the restore has ended, and before you use the partition. FIX\_DISK corrects pointers to records that have been remapped to avoid the badspots.

---

**End of Volume:** If a save that you restore spans more than one reel, PSR prompts for the continuation reel, and displays the prompt Use physical device:. Enter the unit number of the tape drive on which you have mounted the continuation reel.

**Restore Procedure for an Entire Partition**

You should first decide whether to shut down the entire system to users, or just the partition that you want to restore. For a full discussion of preparation procedures to follow before saving and restoring data, refer to Chapter 10, System Preparation, Shutdown, and Startup.

Shown below are the steps you should follow to use PSR -RESTORE to restore an entire partition.

If you have decided to shut down the entire system, follow the procedure described in Chapter 10 of this guide under the section, Procedure for Shutting Down the System.

## OPERATOR'S GUIDE TO DATA BACKUP AND RECOVERY

If you have decided to shut down only the partition, follow the procedure described in Chapter 10 of this guide under the section, Procedure for Removing a Partition from Service.

The following steps must be executed from your supervisor terminal if you have shut down the system, otherwise they may be executed from a user terminal.

1. If the target partition is write protected, remove that protection by issuing the command

```
OK, PSR -NO_PROTECT pdev
```

2. Assign the tape units for the tape that you are going to restore. For example:

```
OK, ASSIGN_MT0  
OK, ASSIGN_MT1
```

3. Mount the reel of tape on the assigned drive. If the objects to restore are contained on a multireel tape volume, this must be the first reel of the volume.

### Note

Ensure that the tapes you intend to restore were all written on the same save stream. Refer to Chapter 6 for an explanation of save streams.

4. Invoke PSR -RESTORE and enter the required response when prompted to do so. Examples of command lines and the ensuing dialog are described in the section, Example Command Lines and Dialogs, later in this chapter.
5. When PSR has completed the restore, remove the last tape.
6. PSR may print the following message

```
FIX_DISK required.  
Please run FIX_DISK on device <pdev> before using it.
```

If you receive this message, continue with the following step. If you do not receive this message, omit the following step.

For detailed information about badspots and the way that utilities handle badspots, refer to the *Operator's Guide to File System Maintenance*.

7. Run FIX\_DISK on the restored partition. Include the -FIX option on the command line.

```
OK, FIX_DISK -DISK pdev -FIX
```

8. To protect the target partition from accidental overwrite when using PSR, you should issue the command

```
OK, PSR -PROTECT pdev
```

9. Unassign the tape units. For example:



OK, UNASSIGN MT0  
OK, UNASSIGN MT1

10. If you have shut down the entire system, restart the system by following the procedure described in Chapter 10 of this guide, under the section, Procedure for Starting Up the System.

If you have shut down only the partition, return it to service by following the procedure described in Chapter 10 of this guide, under the section, Procedure for Returning a Partition to Service.

## Example Command Lines and Dialogs

This section gives examples of PSR command lines to

- Restore a base save
- Restore an incremental save and the associated base save

### Restoring a Base Save

In the following example, PSR restores save number 3 from the tape on drive MT4 to the partition with physical device number 10662. Save number 3 is a base save of pdev 20662, from the disk pack named PACK12.

```
OK, PSR -RESTORE
[PSR Rev. 23.0 Copyright (c) 1990, Prime Computer, Inc.]
Enter media for restore stream.
Use physical device: MT4
Volume ID is DRB001.
The volume is positioned at the start of save number 1,
a physical archive save of PACK12:20662
created on 14 Mar 90 17:05:16 Monday.
Enter required save number: 3
Positioning to required save ...
About to restore save number 3,
a physical base save of PACK12:20662
created on 15 Mar 90 12:20:52 Tuesday.
Enter target disk: 10662
Check read after disk writes? N
Restoring...
Operation complete.
Transferred 25350 records.
OK,
```

### Restoring an Incremental Save and a Base Save

In the following example, PSR restores an incremental save of partition 20662, from tape DRB002 on drive MT4, to physical device number 10662. After it has restored the incremental save, PSR displays messages that identify the base save associated with this incremental save. The user mounts tape DRB001, which holds the base save, on tape drive MT0, and

## OPERATOR'S GUIDE TO DATA BACKUP AND RECOVERY

enters MT0 in response to the PSR prompt Use physical device:. PSR then restores the base save, which in this example is save 1.

```
OK, PSR -RESTORE
[PSR Rev. 23.0 Copyright (c) 1990, Prime Computer, Inc.]
Enter media for restore stream.
Use physical device: MT4
Positioning to beginning of media ...
Volume ID is DRB002.
The volume is positioned at the start of save number 1,
a physical cumulative incremental save of 20862
created on 15 Mar 90 10:15:27 Tuesday.
Enter required save number: 2
Positioning to required save ...
About to restore save number 2,
a physical cumulative incremental save of 20862
created on 18 Mar 90 17:09:38 Friday.
Enter target disk: 10862
Check read after disk writes? N
Restoring...
End of restore of increment.
Please load the volume containing the base save of 20862
created on 11 Mar 90 17:05:18 Friday.
Use physical device: MT0
Volume ID is DRB001.
Positioning to required save ...
Restoring base save number 1.
Restoring...
Operation complete.
Transferred 25350 records.
OK.
```

## Maximizing Performance

There are three ways in which you can maximize the speed of the restore:

- Use alternate tape drives.
- Schedule the restore for a time when the system is lightly loaded.
- Close the entire system to users before you begin the restore.

If the restore is from more than one reel of tape, you can reduce the time the restore takes by using alternate tape drives. Mount the continuation reel on an unused standby drive while the restore proceeds on the first drive. When PSR reaches the end of the first reel and prompts for the next reel, you can quickly resume the restore on the standby drive.

The restore also proceeds more quickly if the system is lightly loaded, or closed entirely to users. The section, Procedure for Shutting down the System, in Chapter 10, explains how to close the system to users as part of the restore procedure.

## PHYSICAL COPY BETWEEN DISKS

### Introduction

The PSR (Physical Save and Restore) command enables you to

- Save a physical disk partition to tape.
- Restore a physical disk partition from tape.
- Copy a physical disk partition to another disk partition.
- Protect a partition from being accidentally overwritten by a restore or a copy.

This chapter describes how to use PSR to copy a partition to another partition. For details of how to save and restore a partition on tape refer to Chapter 6, Physical Save to Tape: PSR, and Chapter 7, Restoring a Physical Save: PSR.

### Overview of PSR Copy

In copy mode, PSR copies an entire partition to another partition. It has the following features:

- The source disk can differ in size and geometry from the target disk.
- Partitions can be protected against an accidental overwrite by PSR.

PSR also has general features, applicable to a save, restore, or copy:

- Interactive interface
- Error handling
- Comprehensive help system

These features are described in the following sections.

**Notes**

You cannot copy to or from the command device. If you want to either save, or restore the command device, use a logical utility.

If you copy a partition that has related files on another partition, remember to copy this other partition also, to ensure the files remain a logically consistent set. For example, a ROAM file may consist of a master file and several slave files, and the slave files might not be on the same partition as the master file.

**Valid Source and Target Partitions**

PSR is very flexible, in that it allows you to copy a partition to a disk that differs from the source disk in terms of size and geometry. The term disk geometry covers the number of:

- Sectors per track
- Cylinders
- Heads

The only restrictions on what you can copy are:

- You cannot copy to or from the command device.
- The target partition must be large enough to hold the data you want to copy.
- The source and target partitions must be the same revision.
- You must copy a robust partition to another robust partition, and a standard partition to a standard partition.
- The source and target partitions must not be the same partition.

**Partition Write Protection**

When you create a partition using MAKE, it is automatically write protected against PSR. This guards against the risk of an accidental overwrite, for example if you copy a partition in the wrong direction.

PSR enables you to remove the write protection before you copy to a partition, and to reinstate that protection after the copy has ended.

If you attempt to copy to a protected partition, PSR displays the warning message

Target device <id> is backup write protected

and again prompts you to identify the partition you want to copy. You can enter another partition ID, or enter `quit` to exit from the copy.

If you exit from the copy, you can remove write protection by issuing the command

OK. `PSR -NO_PROTECT pdev`

where pdev is the physical device number of the partition from which you want to remove write protection.

When you have completed the copy, you can reinstate the protection on the partition by issuing the command

OK, PSR -PROTECT pdev

The section entitled Copy Procedure, later in the chapter, explains how to remove and reinstate partition protection as part of the step-by-step copy procedure.

#### Note

PSR write protection only restricts PSR restores and copies. It does not affect any other disk I/O operations.

### Interactive Interface

If you invoke PSR without identifying the source and target partitions on the PSR command line, PSR prompts you to identify the partitions.

The sequence of prompts and responses is termed the PSR dialog, and is explained in the section Invoking PSR, later in this chapter.

### Error Handling

You can instruct PSR, via either the PSR dialog or a command-line option, to read back the data it has written to the target partition. The read is a means of checking whether there are new badspots on the target partition. If PSR cannot read some of the data, it recovers by writing the data elsewhere on the partition.

---

#### Caution

If errors are detected and recovered by PSR, you must run FIX\_DISK to reset record pointers at the end of the restore.

---

### PSR Help

To obtain general information about PSR, issue the command

OK, PSR -HELP

The following is an example of PSR -HELP

[PSR Rev. 23.0 Copyright (c) 1990, Prime Computer, Inc.]  
One of the following must be supplied:

-SAVE            Save a disk partition

## OPERATOR'S GUIDE TO DATA BACKUP AND RECOVERY

<b>-REStore</b>	Restore a disk partition
<b>-COPY</b>	Copy a partition
<b>-PROtect</b>	Set DRB write protection on partition
<b>-No_PROtect</b>	Clear DRB write protection on partition

For help on one of the above, use **-HELP** with the option.  
For example, **PSR -SAVE -HELP** will give more details for **-SAVE**.

If you require specific help about how to use PSR to copy a partition, issue the command

OK, **PSR -COPY -HELP**

During the PSR dialog, you can obtain help about how to respond to a PSR prompt by entering **HELP** in response to the prompt.

The following is an example of **PSR -COPY -HELP**

[PSR Rev. 23.0 Copyright (c) 1990, Prime Computer, Inc.]

The **-COPY** option is used to copy one disk partition to another.

The command line format is

**PSR -COPY [ source -TO target [-CHECK] ]**

where

<b>source</b>	Identifies the source disk partition
<b>target</b>	Identifies the target disk partition
<b>-CHECK</b>	Performs a read after write verify

## Invoking PSR

The format of the PSR command to copy a partition is

► **PSR -COPY [source -TO target [-CHECK] ]**

The command options are described below:

<i>Option</i>	<i>Description</i>
<b>-COPY</b>	Tells PSR that you want to copy a partition.
<b>source</b>	Identifies the pdev of the <i>source</i> partition that you want to copy.
<b>-TO target</b>	Identifies the pdev of the <i>target</i> partition to which you want to make the copy.

**-CHECK** Specifies that you want PSR to read back the data it copies to the target partition, in order to check that the copied data is readable. The restore operation is slower if you use this option.

For example, the default PSR command line

OK, PSR -COPY

would initiate a short PSR dialog, which is described in the next section, PSR Dialog. PSR prompts you for the source and target partition pdevs, and asks whether you want PSR to read back the data.

Alternatively, you can identify the partitions via the PSR command line. For example, the command

OK, PSR -COPY 10662 -TO 20662

copies the partition with a pdev of 10662 to the partition with a pdev of 20662, without verifying the copy.

## PSR Dialog

This section describes the PSR dialog when you invoke the default PSR copy command. Online help is available for each prompt if you respond to the prompt by entering HELP.

To exit from the dialog, enter QUIT in response to any of the PSR prompts.

Refer to the next section, Example Command Lines and Dialogs, for an example of the PSR dialog.

<i>Prompts &amp; Messages</i>	<i>User Response</i>
Enter source disk:	Enter the source partition pdev.
Enter target disk:	Enter the target partition pdev.
check read after disk writes?	Answer Y if you want PSR to attempt to read back the data it writes to disk. The read is a means of checking for badspots on the target disk, and adds to the total time required for the restore.  Answer N if you do not want PSR to read the data back.
operation complete.	PSR issues this message when the copy has ended successfully.

## Error Recovery

If you have answered YES to the prompt check read after disk writes, PSR displays the following message if a read error occurs:

Unrecovered disk read error reading record <nnn> from device <pdev>.  
Record <nnn> on device <pdev> has been added to the badspot file.

---

### Caution

If you receive this message, you must run **FIX\_DISK** on the target partition after the copy has ended, and before you use the partition. **FIX\_DISK** corrects pointers to records that have been remapped to avoid the badspots.

---

## Copy Procedure for an Entire Partition

You should first decide whether to shut down the entire system to users, or just the partitions that you want to copy. For a full discussion of preparation procedures to follow before saving and restoring data, refer to Chapter 10 of this guide.

Shown below are the steps you should follow to use **PSR -COPY** to copy an entire partition to another partition.

If you have decided to shut down the entire system, follow the procedure described in Chapter 10 of this guide under the section entitled Procedure for Shutting Down the System.

If you have decided to shut down only the source and target partitions, follow the procedure described in Chapter 10 of this guide, under the section entitled Procedure for Removing a Partition from Service.

The following steps must be executed from your supervisor terminal if you have shut down the system, otherwise they may be executed from a user terminal.

1. If the target partition is write protected, remove that protection by issuing the command

```
OK, PSR -NO_PROTECT pdev
```

2. Invoke **PSR -COPY** and enter the required response when prompted to do so. Examples of command lines and the ensuing dialog are described in the section entitled Example Command Lines and Dialogs, later in this chapter.
3. **PSR** may print the message

```
FIX_DISK required.  
Please run FIX_DISK on device <pdev> before using it.
```

If you receive this message, continue with the following step. If you do not receive this message, omit the following step.

For detailed information about badspots and about the way that utilities handle badspots, refer to the *Operator's Guide to File System Maintenance*.

4. Run **FIX\_DISK** on the target partition; include the **-FIX** option.



OK, FIX\_DISK -DISK pdev -FIX

5. If you want to protect the target partition from accidental overwrite, issue the command

OK, PSR -PROTECT pdev

6. If you have shut down the entire system, restart the system by following the procedure described in Chapter 10 of this guide, under the section entitled Procedure for Starting Up the System.

If you have shut down only the source and target partitions, return them to service by following the procedure described in Chapter 10 of this guide, under the section entitled Procedure for Returning a Partition to Service.

## Example Command Lines and Dialogs

This section has examples of

- A PSR copy command that identifies the source and target partitions on the command line
- A PSR copy command in which the source and target partitions are identified interactively

In both examples, PSR copies partition 10662 to partition 20662.

### Copy Without Prompts

```
OK, PSR -COPY 10662 -TO 20662
[PSR Rev. 23.0 Copyright (c) 1990, Prime Computer, Inc.]
Copying...
Operation complete.
Transferred 25350 records.
OK,
```

### Copy With Prompts

```
OK, PSR -COPY
[PSR Rev. 23.0 Copyright (c) 1990, Prime Computer, Inc.]
Enter source disk: 10662
Enter target disk: 20662
Check read after disk writes? N
Copying...
Operation complete.
Transferred 25350 records.
OK,
```

## **Maximizing Performance**

There are two ways in which you can maximize the speed of the copy:

- Schedule the copy for a time when the system is lightly loaded.
- Close the entire system to users before you begin the copy.

The section entitled Procedure for Shutting Down the System, in Chapter 10, explains how to close the system to users as part of the copy procedure.

---

**PART III**  
**SYSTEM BACKUPS:**  
**ADMINISTRATION**

---

## DEFINING A BACKUP STRATEGY

### Introduction

This chapter gives guidance about when and how to carry out backups. It is for System Administrators who have yet to decide on a backup strategy and for those Administrators who want to reconsider the suitability of their current approach to backups.

The chapter discusses the reasons for taking backups and describes the utilities provided by Prime for performing backups. Factors that you should consider in order to maximize backup performance are discussed and the process of selecting a backup strategy is described.

### The Need for Backups

Data in computer systems can be lost for several reasons. This section is restricted to a discussion of loss of data on disk; data loss from memory is not included.

Data resident on disk can be lost due to:

- User error
- Software error
- Storage device failure
- Environmental disaster

User errors are the most common cause of data loss. Software failures are very similar to user errors in effect, but are caused by programming errors, and may go undetected for some time. Storage device failure is now a relatively infrequent cause of data loss, as modern disk drives are very reliable. However, the increasing capacity of a single disk does mean that the consequences of disk failure can be very severe. Environmental disasters are the most potentially damaging causes of data loss.

The usual way to protect against data loss is to take periodic copies, or backups, of the data. The more frequently you do this, the more up-to-date the backup copy will be. Taking backups is necessarily time consuming. Some data on the system may be of insufficient value to warrant frequent saving, or may not be worth saving, if it can be easily regenerated.

By carefully choosing the data and the frequency for backups, the administrator can minimize the time taken in performing backups, while at the same time maintaining adequate coverage against data loss.

### **Prime Backup Utilities**

The two main backup utilities provided by Prime are:

- The logical save and restore utility MAGSAV/RST
- The physical save and restore utility PSR

These utilities are discussed in detail in earlier chapters of this guide; their main features are summarized in the following sections.

### **Logical Backup Utility MAGSAV/RST**

MAGSAV provides a flexible way to back up your data to tape. A logical save is slower than a physical save for a given amount of data, but has the advantage that you can subsequently restore individual objects from the save tape, as well as the entire save. You can use MAGSAV for performing tape backups, and for transporting data between 50-series systems.

#### **Logical Backups**

You can use MAGSAV to perform both base saves and incremental saves. These are discussed in the following paragraphs.

**Base Saves:** A base save is a copy of all files within the specified directory or directories.

**Incremental Saves:** An incremental save contains only those files that have been modified since the last save, and may be of two types. A cumulative incremental save contains all files changed since the last base save. A true incremental save contains only those files that have changed since the last incremental or base save. A true incremental save is quicker than a cumulative incremental save, but a full restore takes longer, as the base save and all ensuing true incremental saves have to be restored, instead of just the base save and the most recent cumulative incremental save.

#### **Incremental Control**

MAGSAV uses the DTB (Date/Time Backed up) attribute to control the incremental cycle. When you run an incremental save, MAGSAV compares the value of the DTB attribute with that of DTM (Date/Time modified), and saves the file if the DTM is more recent than the DTB.

If you are running a true incremental save, you use the command line option **-UPDT** to reset the DTB value of saved files to the current date and time. The next time you run an incremental save, the file will be saved only if it has been modified since the previous incremental save.

If you are running a cumulative incremental save, you do not use the **-UPDT** option in the command line. MAGSAV does not update the DTB of saved files, and the next time you run an incremental save, the file will be saved whether or not it has been modified in the intervening period.

### **Restoring a Logical Backup**

When you restore the tapes written by MAGSAV, you restore the base save first, and then any incremental saves in order.

## **Physical Backup Utility PSR**

A physical save is performed without reference to the logical hierarchy of the file system, and saves all used disk records within a partition. The advantage in using a physical save utility is the greater speed of save, compared with a logical save utility; the disadvantage is that it is only possible to restore the entire partition, rather than selected file system objects.

### **Physical Backups**

A physical save is an exact copy of the contents of a partition. File system objects are spread across the backup medium (disk or tape) in exactly the same way as on the source partition. Physical backups are faster than logical backups for a given amount of data, and are thus particularly useful when you wish to back up large amounts of data. However, you cannot restore individual file system objects from a physical backup; you must restore the entire saved partition.

PSR allows you to make backup copies on either tape or disk; use the command **PSR -SAVE** to make a backup on tape, and **PSR -COPY** to make a backup on disk. You can copy and restore data to a partition of different size, or to a disk of different geometry from the source partition. In addition, it is possible to restore to a partition that is smaller than the source partition, provided there is sufficient space to hold all the used records.

#### **Note**

You cannot use PSR to save, restore, or copy the command device partition. You must use the logical save and restore utility MAGSAV/RST for this.

If you have related files spread across several partitions, for example, ROAM master and slave files, you must back up all partitions that hold the related files, to ensure that they remain a logically consistent set.

### **Physical Backup to Tape**

When saving to tape, you can run both base and incremental saves. These are discussed in the following section.

**Base Saves:** A base save is a copy of all records within the specified disk partition.

**Cumulative Incremental Saves:** A cumulative incremental save saves only those disk records that have changed since the last base save. This is highly suited to a system with a large database, or with very large files. A logical incremental save would save the entire database even if only one record had changed; a physical incremental save of the same system would save only those records within the database that had changed. You cannot perform incremental saves with PSR if you are saving to disk.

### **Restoring a Physical Backup from tape**

Use the command PSR -RESTORE to restore a physical backup from tape. If you have done an incremental save, you must restore the cumulative incremental save *first*, and then restore the base save. PSR controls the entire process, so that it is not possible to leave the disk in a partially restored state.

### **Copying to Disk using PSR**

Use the command PSR -COPY to make a backup on disk. This has the advantage that you do not need to run a restore operation; the pack can be loaded for immediate use, although this does expose the backup disk to the risk of loss of data.

Incremental disk-to-disk copying is not available because failure of the source disk during such an operation would result in the loss of both the source and backup disks.

## **Disk Mirroring**

Disk mirroring provides an alternative means of backing up data. In this case, the system keeps a duplicate online copy of the disks for which mirroring is enabled. A mirrored disk partition is a pair of partitions, and each disk holds an identical image of the data. All writes are executed on both disks, but reads are shared between the two (See the *Operator's Guide to File System Maintenance*).

#### **Note**

Mirroring provides protection *only* against storage device failure. User or software errors are duplicated on both disks, and an environmental disaster would probably destroy both disks. If there is a device failure, PRIMOS breaks the mirroring and continues with the surviving disk. Once the drive has been fixed, it can be brought back into service without interrupting the application; a disk-to-disk copy takes place automatically alongside the normal operation.

## Maximizing Backup Performance

It is important to reduce the time taken to run backup operations, but still maintain an adequate level of data security. Formerly, computer systems were often run on a 24 hour basis, and backups were taken overnight. 24 hour operation has become less common, and backups frequently have to be performed while many other applications are running during the day.

The following sections highlight some of the factors you should consider in order minimize the time taken to run backup operations. Choosing a method that reduces the backup time may result in the associated restore taking longer, but as backups are performed more frequently than restores, it is more important to reduce the time taken to perform the backup.

### Reducing the Backup Time

There are several ways in which you can reduce the time taken for backup operations. Some of these are described in the following paragraphs.

### Grouping Data on the System

You could group data so that those objects requiring frequent backup are placed together within one or more partitions, and less important data is grouped together on separate partitions. This reduces the amount of data to be saved regularly. If your system has large databases, it may be advantageous to group the databases together on one or more partitions, so that you can run physical base and incremental saves on these partitions; further details on these types of save are given in the next paragraph.

Grouping data on the system in order to reduce backup time may not be practical given the constraints of the applications you are running; conflicts may arise between maximizing application performance and backup performance.

### Using Incremental Saves

Saving only that data which has been modified since the last save also reduces the amount of data saved during backup operations. Restoring the base save and the incremental save(s) obviously takes longer than if you backed up the data using base saves only, but this is more than outweighed by the reduced time of the save operations.

If your system has large databases, it may be preferable to run physical base and incremental saves, rather than logical saves. The time taken to run the physical base save will be similar to the time taken to run a logical base save, but when you run the physical incremental save, the system saves only those records within the database that have been modified since the base save. In the case of a logical incremental save, even if only one record has been modified, the system saves the entire database. The inability of PSR to restore selected files from a physical save is irrelevant in this case, as the entire database must be restored.



### **Scheduling Backups**

The MAGSAV and PSR utilities run best on lightly-loaded systems. Ideally you should close the entire system to users before you begin the backup. To improve the performance of these utilities, while minimizing the inconvenience to users, you should therefore consider scheduling the backups for off-peak hours. Details on running unattended backups are given in the following section.

### **Unattended Backup**

The very high data capacity of the 8mm cartridge tape drive reduces the number of tapes required to store backup data. This may eliminate the need for operator intervention, and thus allow you to perform backup operations overnight.

Some systems have a total backup requirement of less than the capacity of a single 8mm cartridge tape. Larger systems require the use of multiple tapes. If you need to use multiple tapes for backup operations, the ability to run unattended backups will depend on your system having enough tape drives for you to mount all the required tapes before you initiate the backup.

You can run an unattended backup operation using the CPL program described in the section, Unattended Backup, in Chapters 2 and 6 of this guide.

### **Compacting the File Structure**

MAGSAV runs more slowly when it saves heavily fragmented file structures. If a partition has been heavily used for a period of time, without having been logically restored from tape, it may be worthwhile to run a logical save and restore on the entire partition, to speed up subsequent backups of this partition. The best way to do this is as follows:

- Use MAGSAV without either the -INC or -UPDT options to save all the partition objects to tape
- Run MAKE on the partition
- Restore the partition objects with MAGRST

### **Retaining Backups**

This section discusses how long you need to retain your tape or disk backups, and where you should keep them.

It is a good idea to retain three generations of backups and to keep each backup in a separate place. When you make a new backup, rotate the three generations and delete the oldest version. You should keep the most recent backup in an easily accessible place: this is the backup that you use if you need to restore data. Ideally, you should keep the intermediate backup in a secure, fireproof location somewhere in the same building, and place the oldest copy offsite.

You can use MAGSAV to archive inactive files that you might need at some time in the future. In that case, the life of the archived data is indefinite: retain the archive tapes for as long as the archived data may be useful.

**Note**

Do not retain any tape for longer than a year, because after that time tapes begin to deteriorate. You should regularly transfer your data to new tapes, to be sure of being able to read the data when you need it.

## **Logging Backups**

After you have adopted a backup strategy, it is a good idea to record the backup procedures in the system log book. It is also advisable to record in the log book the details of each backup and any problems that the operator has encountered during a backup.

## **Selecting a Backup Strategy**

The backup strategy you choose will depend on what is most appropriate for your system. Outlined below are some of the factors you should consider before you decide on a backup strategy.

- Decide on the level of security that your system requires. Typically, you may divide your files into different classes depending on whether they require daily or weekly backup, or no backup at all. When you do this, consider the different causes of data loss.
- Distinguish between backups and archives. A backup is used to recover from a data loss on the system; archives provide a historical record of the system. Archives usually have a longer life and are accessed less frequently than backups. Archive tapes may require periodic regeneration, due to the gradual deterioration of the magnetic image on tape.
- Group files into partitions that reflect their backup requirements. This may be difficult to do, in view of conflicts with application requirements. In particular, try to group large databases together so that you can use PSR to back up these databases.
- If some files are sufficiently important, it may be worthwhile to use disk mirroring with these files. However, remember that this only protects the files against system device failure.
- If files have become fragmented on disk, due to frequent modifications, run a logical save, remake the partition and then restore of the disk to compact the file structure and thus speed up subsequent logical save/restore operations.
- Consider the possibility of using 8mm cartridge tape drives to enable you to run backups overnight without operator intervention.
- Having decided on a policy, review it against the causes of data loss. Envisage how you would recover from the various causes of data loss.

## **OPERATOR'S GUIDE TO DATA BACKUP AND RECOVERY**

- Examine your hardware configuration. Is your backup policy adversely constrained by lack of disks or tape drives? The extra cost of additional drives may be offset by reduced operations costs.
- Check that your backup policy does not require concurrent use of two tape drives on a single controller, as the system can transfer data to only one drive at a time on the controller. You may be able to schedule the use of the drives, so that, for example, the tape on one drive is rewinding while data is being transmitted to the other.
- Consider using intelligent disk controllers. These offer higher transfer rates than hard wired controllers, especially where data on disk is being accessed sequentially, as is the case with backups.
- Ensure that your hardware configuration is sensibly matched. Performance may be seriously impaired if, for example, you have a small processor to drive a high speed streaming tape.

When you have decided on a policy, ensure that users of the system are made aware of it. If, for example, you have decided on grouping files in order to maximize backup performance, users must know where to place new files.

Decide where to store backups. It is essential to keep an offsite copy, in a secure location.

Consider the impact on backups of new applications.

The following pages describe some examples of user configurations and describe suitable backup strategies for these configurations.

### **Example 1**

The following example is of an imaginary financial services company. It runs a large transaction processing (TP) application on a Prime 9955. The system has about 7GB of data, including a 5GB database. About 5% of the database changes each day. The company uses after-imaging with DBMS, to roll the database forward to the last completed transaction. The operations staff are experienced, and are employed specifically to perform operations tasks. A scheduled downtime of up to eight hours per day is acceptable.

An appropriate backup strategy for this type of system would be as follows:

- Weekly physical base save of the database, using PSR -SAVE -BASE
- Daily physical incremental save of the database using PSR -SAVE -INC
- Weekly full logical save of the non-database files, using MAGSAV -UPDT
- Daily incremental backup of the non-database files, using MAGSAV -INC

To maximize the speed of the PSR -SAVE operation, the system should have at least two high-speed, high-density tape drives. There should also be a free disk drive on the system; this speeds up a physical restore from tape.

The performance of MAGSAV could be improved by having the files for the backups on the same part of the system, rather than having them spread thinly across many partitions.

### **Example 2**

The following example is of an imaginary engineering company that runs a CAD system on a Prime 9750. There are several CAD databases, totalling approximately 1.2GB, and they change by 5% a day. There are a small number of operations staff, but they are experienced.

An appropriate backup strategy for this type of system would be

- Weekly full logical backup, using MAGSAV
- Daily incremental backup, using MAGSAV

The system should include a high-speed, high-density tape drive, to maximize the speed of the backup.

### **Example 3**

The following example is of an imaginary small advertising company that runs a Prime INFORMATION application on a Prime 2450, and uses a 60MB cartridge tape drive for backups. The total volume of data on the system is about 300MB, 2% to 5% of which changes each day. Computer expertise amongst the staff is low. The operators who perform the backups are inexperienced, and do not specialize in computer operation tasks.

An appropriate backup strategy for this system would be

- Weekly full backup, using MAGSAV
- Daily incremental backup, using MAGSAV

An alternative approach would be to perform a full backup every day. This would simplify the backup and restore procedures for inexperienced operators.

### **Example 4**

The following example is of an imaginary university that uses a Prime 9650 system for both teaching and general administration. There are a large number of users, and most of the files on the system are small. The total volume of data is about 3.6GB, and approximately 5% of the data changes each day. Operations staff are experienced.

An appropriate backup strategy for such a system would be

- Weekly full logical backup, using MAGSAV
- Daily incremental backups, using MAGSAV

An alternative strategy for such a system would be to perform a physical backup of the system each day, using PSR -SAVE. However, logical restores of single files would not be possible.

# SYSTEM PREPARATION, SHUTDOWN, AND STARTUP

## Introduction

This chapter describes the preparatory stages of the save or restore process. It also gives procedures for shutting down the system before starting the save or restore, and to restart the system once the save or restore is complete.

Operators frequently need to follow these procedures when saving and restoring partitions to and from tape or disk during routine backup and restore operations.

## Preliminary Preparations

Outlined below are the factors you should take into consideration before commencing the save or restore operation.

### Preparing for a Tape Backup

Before you begin a tape backup, you need to know the following information:

- The location of the data you want to save, either the logical file system object names, or the physical device number.
- The tape drive unit number.
- The volume serial ID, if the tape is not already labeled.
- Whether the tape already holds data. If it does, you need to know how many saves there are already on tape, and which saves, if any, you want to overwrite.

### Preparing for a Tape Restore

Before you begin a tape restore, you need to know the following information:

- Which tape volume holds the save you want to restore
- The number of the save you want to restore

- Where you want to restore the data

In the case of a physical restore or copy, ensure that the source and target partitions are valid partitions, as defined in the section, Valid Source and Target Partitions, in Chapter 7, Restoring a Physical Save: PSR. You need to know the following information:

- The size of the source and target partitions
- Whether the partitions are robust or standard
- The PRIMOS revision of the source and target partitions

All the required information about the source partition should be on the adhesive label on the backup tape.

To display the size of the target partition, use the AVAIL command and ensure that the partition is large enough to hold the restored data.

To determine whether a partition is robust or standard, use the LIST\_DISKS, STATUS DISKS, or LD command. You must restore a robust partition to another robust partition and a standard partition to another standard partition.

The revision number of the source partition and the save type should have been noted on the tape adhesive label when the save was completed. The source and target partitions must have the same revision number. The revision number of a partition is displayed when you run either FIX\_DISK or ADDISK.

### **Preparing for a Disk to Disk Copy**

Before you begin a PSR copy, you need to know the following information:

- The physical device number of the source partition
- The physical device number of the target partition

Ensure that the source and target partitions are valid partitions, as defined in the section entitled Valid Source and Target Partitions, in Chapter 8, Physical Copy between Disks. You need to know the following information:

- The size of the source and target partitions
- Whether the partitions are robust or standard
- The revision number of the source and target partitions

To display the size of the target partition, use the AVAIL command, and ensure that the partition is large enough to hold the data.

The revision number of the partitions and whether they are robust or standard should be already documented, for example in your system logbook.

To determine whether a partition is robust or standard, use the LIST\_DISKS, STATUS DISKS, or LD command. You must restore a robust partition to another robust partition, and a standard partition to another standard partition.

The source and target partitions must have the same revision number. The revision number of a partition is displayed when you run either FIX\_DISK or ADDISK.

## System Preparation

Before you begin a save, restore, or copy operation, you should decide whether to

- Close the entire system to users
- Close selected partitions only

If you want to maximize the speed of the operation, take the entire system out of service. If you close only selected partitions, this may be more convenient to users, but will cause the operation to run more slowly.

Regardless of performance considerations, it is advisable to close the entire system to users if any one of the following considerations applies to the partition you are saving, restoring, or copying:

- It holds many users' origin directories (20% or more of system users).
- It holds files accessed by most of the users on the system.
- It is on the same disk as a paging partition.

In the first two cases, users will not be able to work on the system at all, and in the third situation, any work carried out by users during the operation will be very slow.

If the operation accesses any master or slave ROAM files, you should either shut the system to users entirely, or shut down all partitions that hold related ROAM files.

## Shutdown Procedures

Procedures for shutting the entire system to users, and for shutting down individual partitions are given in the following sections.

### Procedure for Shutting the System to Users

If you decide to shut down the entire system before you run the save, restore, or copy operation, execute the following steps from the supervisor terminal.

1. Inform users of the impending shutdown by issuing a series of messages. The first message should give users plenty of advance notice of the shutdown. For example:

```
OK, MESSAGE ALL -NOW -FORCE  
SYSTEM CLOSING DOWN AT 18:00 HOURS TODAY  
.  
.  
.  
OK, MESSAGE ALL -NOW -FORCE  
SYSTEM CLOSING DOWN IN 5 MINUTES - PLEASE LOG OUT
```

2. Prevent users logging in by issuing the following command:

```
OK, MAXUSR 0
```

3. Force the termination of all currently running system services. The commands that you need to issue to do this depend on the services that are installed on your system. For example, your system may have the following services installed:

- Batch
- Spooler
- File Transfer Service
- Network Service
- Ethernet
- DSM (Distributed Systems Management)

Stop these services by issuing the following commands:

```
OK, BATCH -STOP
OK, PROP prt -STOP
OK, FTOP -STOP SRVR ftp
OK, FTOP -STOP_MNGR
OK, STOP_NET
OK, STOP_NTS
OK, STOP_DSM
```

where *prt* is the printer device number and *ftp* is the server name for the file transfer service. If your system has several printers you should issue the PROP command for each printer.

4. Wait until all phantoms associated with these services have finished. You can check this by issuing the command STATUS USER.
5. Issue a message to remind users about the impending shutdown.
6. Log out all users by issuing the following command:

```
OK, LOGOUT_ALL
```

7. If you are using MAGSAV/RST, give yourself priority access to the partition you wish to save/restore. The command to do this is shown below.

```
OK, SPAC ptnam SYSTEM:ALL $REST:NONE
```

where *ptnam* is the partition name. Continue with the logical save/restore as described in procedures in Chapters 2 and 3 of this guide.

8. If you are using PSR proceed as follows: shut down the partitions for the save/restore/copy, add the partitions to the Assignable Disks Table, and then assign the partitions. The commands to do this are shown below:

```
OK, SHUTDN pdev
OK, DISKS pdev
OK, ASSIGN_DISK pdev
```

where *pdev* is the physical device number for the partition. Continue with the physical save/restore/copy as described in procedures in Chapters 6, 7, and 8 of this guide.



9. Shut down the system using

OK, SH ALL

and cold start the system to get all the processes running and enable user log in.

**Procedure for Removing a Partition from Service**

If you decide to remove an entire partition from service before running a save, restore, or copy operation, execute the following steps from the supervisor terminal.

1. Inform users of the impending shutdown by issuing a series of messages. The first message should give users plenty of advance notice of the shutdown. For example:

OK, MESSAGE ALL -NOW -FORCE  
PARTITION <DSK1> CLOSING DOWN AT 18:00 HOURS TODAY

.

OK, MESSAGE ALL -NOW -FORCE  
PARTITION <DSK1> CLOSING DOWN IN 5 MINUTES

2. Use the command STATUS USER to check if there are any system phantoms currently running on the partition you wish to shut down. If there are, issue the relevant commands to stop these phantoms. For example, if DSM is currently running on the partition you are shutting down, issue the command STOP\_DSM.
3. If you are using MAGSAV/RST, remove users from the partition you wish to save/restore, by removing and then restoring the partition to the file system, and give yourself priority access to the partition. The commands to do this are shown below.

OK, SHUTDN pdev  
 OK, ADDISK pdev  
 OK, SPAC ptnam user-id:ALL \$REST:NONE

where *pdev* is the physical device number for the partition, *ptnam* is the partition name and *user\_ID* is SYSTEM if you intend to run the operation from the supervisor terminal, or it is your own user identification if you intend to run the operation from from a user terminal. Continue with the logical save/restore as described in the procedures in Chapters 2 and 3 of this guide.

4. If you are using PSR proceed as follows: Shut down the partitions for the save/restore/copy, add the partitions to the Assignable Disks Table, and then assign the partitions. The commands to do this are shown below.

OK, SHUTDN pdev  
 OK, DISKS pdev  
 OK, ASSIGN DISK pdev

where *pdev* is the physical device number for the partition. Continue with the physical save/restore/copy as described in procedures in Chapters 6, 7, and 8 of this guide.

## Startup Procedures

Procedures for starting up the entire system, and for shutting down individual partitions are given in the following sections.

### Procedure for Starting Up the System

If you shut down the entire system to users before you ran a save, restore, or copy operation, then once the operation is finished, execute the following steps from the supervisor terminal to restart the system.

1. If you are using MAGSAV/RST, restore normal access to the partitions closed. The command to do this is shown below.

OK, RPAC\_ptnam

where *ptnam* is the partition name.

2. If you are using PSR proceed as follows: unassign the partitions you have just saved/restored/copied, remove the partitions from the Assignable Disks Table, and make the partitions available to users. The commands to do this are shown below.

OK, UNASSIGN DISK\_pdev  
OK, DISKS\_NOT\_pdev  
OK, ADDISK\_pdev

where *pdev* is the physical device number for the partition.

3. Issue the required commands to restart the system processes. The commands that you need to issue depend on the processes that are installed on your system. For example, to start the DSM, Ethernet, network, file transfer, spooler, and batch services, issue the following commands:

OK, START\_DSM  
OK, START\_NIS  
OK, START\_NET  
OK, FTOP -START\_MNGR  
OK, FTOP -START\_SRVR ftp  
OK, PROP prt -START  
OK, BATCH -START

where *prt* is the printer device number and *ftp* is the server name for the file transfer service. If your system has several printers you should issue the PROP command for each printer.

4. Issue the following command to allow users to log in:

OK, MAXUSR

5. Issue a message advising users that the system is now running again. For example:

OK, MESSAGE ALL -NOW -FORCE  
SYSTEM IS NOW UP AND RUNNING - PLEASE LOGIN

Alternatively, perform a system shutdown by issuing the command

OK, SH ALL

then cold start the system, which would then start all the system services running and allow user log in.

### Procedure for Returning a Partition to Service

If you removed a partition from service before you ran a save, restore, or copy operation, then once the operation is finished, execute the following steps from the supervisor terminal to return the partition to service.

1. If you are using MAGSAV/RST, restore normal access to the partitions by issuing the following command for each partition:

OK, RPAC ptam

where *ptam* is the partition name.

2. If you are using PSR proceed as follows: unassign the partitions you have just saved/restored/copied, remove the partitions from the Assignable Disks Table, and make the partitions available to users. The commands to do this are shown below.

OK, UNASSIGN DISK pdev  
OK, DISKS NOT pdev  
OK, ADDISK pdev

where *pdev* is the physical device number for the partition.

3. Start any system phantoms that you shut down when removing the partition from service. For example, to start the DSM service, issue the command START\_DSM.
4. Issue a message advising users that the partition is now available again, for example:

OK, MESSAGE -ALL -NOW -FORCE  
PARTITION DSK1 IS NOW AVAILABLE

---

## **APPENDICES**

---

**GLOSSARY**

Explanations are given on the following pages of some of the terms used in this guide.

**ACAT**

**Access Category.** This is a file object that contains the access control parameters associated with one or more objects in the same directory.

**ACL**

**Access Control List.** This is a set of user rights associated with individual file system objects, that define the user access rights to these objects. These include, for example, read (R), list (L), delete (D), write (W).

**ANSI**

**American National Standards Institute.**

**Base Save**

**A save of all data.**

**CAM**

**Contiguous Access Method.** This is a method by which the system accesses files, and also a method of file storage. Groups of file records are stored contiguously on the disk. This results in faster file access. Files accessed in this way are called CAM files.

**CPL**

**Command Processing Language.** This is a programming language used primarily for processing commands.

**Cumulative Incremental Save**

**An incremental save recording all changes since the base save.** This form of save requires only the base save and the last incremental save for restore.

## **OPERATOR'S GUIDE TO DATA BACKUP AND RECOVERY**

### **DAM**

**Direct Access Method.** This is a method by which the system accesses files, and also a method of file storage. Individual file records may be stored in separate areas of the storage medium. Files accessed in this way are called DAM files.

### **DTA**

**Date/Time Accessed.** This is an attribute associated with a file system object specifying the date and time that object was last accessed.

### **DTB**

**Date/Time Backed Up.** This is an attribute associated with a file system object specifying the date and time that object was last backed up.

### **DTC**

**Date/Time Created.** This is an attribute associated with a file system object specifying the date and time that object was created.

### **DTM**

**Date/Time Modified.** This is an attribute associated with a file system object specifying the date and time that object was last modified.

### **EOT**

**End of Tape.** This is a physical reflective strip on the tape close to the end of the tape, that the tape drive recognizes as signifying the end of the tape reel.

### **EPF**

**Executable Program Format.** This is one format in which programs that you run are stored on the system.

### **File System Object**

See *object*.

### **Home Attach Point**

The point to which you are attached when you issue a command to start an operation such as a save or restore operation.

### **Incremental Save**

Used to describe both cumulative and true incremental saves – see also *Cumulative Incremental Save* and *True Incremental Save*.

### **IPR**

**Independent Product Release.** This term describes a product release that is not tied to a PRIMOS revision release.

**Logical Tape or Logical Save**

The data recorded by a single invocation of MAGSAV. The data may be contained in one or several files or directories, and may occupy one or more reels of tape. If the data saved occupies more than one reel, the continuation logical tape is logical tape 1 on the continuation reel.

**MFD**

Master File Directory. This is the top-level directory of a partition.

**Object**

Is used throughout this guide to indicate either a file or a directory.

**Partition**

A separate logical area of file system storage on disk.

**PDev**

Physical Device Number. A numeric identifier describing the location and size of a partition.

**RAT**

Record Availability Table. Holds information about which records are in use on a partition.

**RBF**

Recovery Based File. These files are stored on the system as a master and several slave files. This method of storage enables data updates to be recovered even if there is a system crash during the updating session.

**ROAM**

Recovery Oriented Access Method. This is the method by which the system accesses Recovery Based Files.

**Save Stream**

One of the devices being written to when PSR writes multiple copies of a single save.

**Tape Block**

The system writes data to tape in discrete blocks, rather than as a continuous stream. MAGSAV writes 12K byte blocks by default. You can write tape blocks of 16K bytes when running on PRIMOS Rev. 22.0 or later revisions, using the configuration directive MTRS and the MAGSAV option -BIG. Refer to Chapters 1 and 2 for details of block sizes.

**Top Level Directory**

A directory immediately below an MFD in the file structure.

## ***OPERATOR'S GUIDE TO DATA BACKUP AND RECOVERY***

### **True Incremental Save**

An incremental save that records changes since the last base or incremental save. This form of save requires the base save and all incremental saves to restore the data.

### **Volume**

A physical object of storage media, for example, a reel of tape, or a disk.

### **Volume Serial ID**

An alphanumeric string of a maximum length of 6 ANSI characters contained within an ANSI tape label, that uniquely identifies the reel of tape. This identification is particularly useful in the case of multireel volumes.



---

## INDEX

---

# INDEX

## Symbols

BACKUP\$ MAGSAV/RST, 2-2, 2-3, 2-4, 2-5,  
2-10

## A

Access categories MAGSAV/RST

- No\_Acl, 2-4
- restoring, 3-2
- saving, 2-2

ACL

- No\_Acl, 2-2, 2-4
- restoring, 3-2
- saving, 2-2
- saving objects, 2-1

Appending to tape, PSR save dialog, 6-9

Archive saves

- physical save to tape, 6-2
- PSR -SAVE -ARCHIVE, 6-6
- PSR examples, 6-13
- PSR restore dialog, 7-5
- PSR save dialog, 6-8

Attributes MAGSAV/RST

- No\_Attributes, 1-2, 3-3
- NO\_DTA, 1-2, 2-4
- DTA, 1-2, 2-4, 3-3
- DTB, 1-3, 1-8, 2-3, 2-4, 2-5, 2-10, 3-3, 9-2
- DTC, 1-2, 3-3
- DTM, 1-2, 1-3, 2-3, 2-4, 3-3, 5-3, 9-2
- incremental control, MAGSAV control, 2-3
- logical restore, 1-2
- logical save, 1-2
- restoring, 3-3

## B

BACKUP\$ MAGSAV/RST, 2-3

Backups

- base saves, 9-2, 9-4
- cartridge tapes, 9-6
- CPL program, 6-16
- disk mirroring, 9-4
- examples, 9-8
- incremental control, 9-2
- incremental saves, 9-2, 9-4, 9-5
- logging, 9-7
- logical restore, 9-3
- MAGSAV/RST, 9-2
- need for, 9-1
- performance, 9-5
- physical copy to disk, 9-4, 10-2
- physical restore, 9-4
- policy, 9-8
- preparations for tape backup, 10-1
- preparations for tape restore, 10-1
- PSR, 9-3
- retaining, 9-6
- scheduling, 9-6
- strategy, 9-7
- system preparation, 10-3
- unattended, 6-16
- unattended MAGSAV, 1-4, 2-13, 9-6
- unattended PSR, 9-6

Backups MAGSAV, CPL program, 2-13

Base Saves, PSR -SAVE -BASE, 6-6

Base saves

- backups, 9-2, 9-4
- MAGSAV description, 2-2

## HEADER LEFT

- physical restore from tape, 6-3, 7-1
- physical save to tape, 6-2, 6-3
- PSR examples, 6-13, 6-15, 7-9
- PSR restore dialog, 7-5, 7-6
- PSR save dialog, 6-8

### Boot tapes

- BOOT, 2-1, 2-3, 4-1
- BOOT\_CREATE, 4-3
- BOOT\_RESTORE, 4-1
- creating, 2-1, 4-1, 4-2, 4-4
- format, 4-1
- list files, 4-2
- PRIMOS revisions, 4-1

### BOOT\_CREATE

- command dialog, 4-4
- command format, 4-3
- command options, 4-4
- description, 4-1
- examples, 4-5

### BOOT\_RESTORE, 4-1, 4-6, 4-7

### Booting from tape

- BOOT\_RESTORE, 4-6
- description, 4-6
- procedure, 4-7

### BRMS tapes, 1-8

## C

- CAM files MAGSAV/RST, 1-5, 1-6, 1-8, 2-4
- CAM\_RBF, 3-3

### Cartridge tapes

- MAGSAV, 2-10
- MAGSAV/RST, 1-4
- PSR save dialog, 6-9
- unattended backup, 9-6

### Check disk

- Physical copy to disk, 8-5
- physical restore from tape, 7-5
- PSR copy dialog, 8-5
- PSR restore dialog, 7-6

### Checkpoints

- MAGRST, 1-2
- MAGRST dialog, 3-6
- MAGRST procedure, 3-8
- MAGSAV, 1-1, 1-3
- MAGSAV/RST tape indexing, 5-2
- physical save to tape, 6-5

### Command device

- physical copy to disk, 8-2
- physical restore from tape, 7-1
- physical save to tape, 6-2, 9-3

### Commands

- BOOT\_CREATE, 4-1, 4-3
- MAGRST, 3-3
- MAGSAV, 2-3
- MTRESUME, 4-7, 4-8
- MTRS, 1-6, 1-8, 2-3
- PSR -COPY, 8-4
- PSR -RESTORE, 7-5
- PSR -SAVE, 6-6

### CPL programs

- TTY, 2-5, 3-4
- MAGSAV dialog, 2-10
- Physical saving to tape, 6-15
- unattended backup, 6-16
- unattended backup MAGSAV, 2-13

## D

- DAM files MAGSAV/RST, 1-5, 1-6, 1-8, 2-4

- Disk mirroring, 9-4

## E

### End of tape

- MAGRST, 3-6
- MAGSAV, 2-10
- physical restore from tape, 7-7
- physical save to tape, 6-10

### EPF, MAGRST, 3-3

### Error handling

- MAGSAV/RST, 1-3, 1-7
- physical copy to disk, 8-3, 8-5
- physical restore from tape, 7-3, 7-7
- physical save to tape, 6-5

## H

- Hardware considerations, controllers, 9-8

### Help

- MAGRST, 3-3
- MAGSAV, 2-3
- physical copy to disk, 8-3
- physical restore from tape, 7-4
- physical save to tape, 6-5

## I

- Incremental control, backups, 9-2

### Incremental saves

- INC, 2-4

- Save\_UFD, 2-5, 3-2
- UPDT, 2-3, 2-5
- backup performance, 9-5
- backups, 9-2, 9-4
- control MAGSAV, 1-3
- DTB, 1-3, 1-8, 2-3
- MAGSAV description, 2-2
- MAGSAV dialog, 2-9, 2-10
- MAGSAV examples, 2-5, 2-6
- physical restore from tape, 6-3, 7-1
- physical save to tape, 6-2, 6-3, 9-5
- PRIMOS Revisions MAGSAV/RST, 1-8
- PSR examples, 6-14, 7-9
- PSR restore dialog, 7-6
- PSR save dialog, 6-8
- PST -SAVE -INC, 6-6
- ROAM files, 1-3, 2-10

## L

- Logical tapes MAGSAV/RST
  - description, 2-2
  - MAGRST dialog, 3-4
  - MAGSAV dialog, 2-7
  - name, 2-7

## M

- MAGRST
  - command format, 3-3
  - command options, 1-5, 3-3
  - description, 3-1
  - dialog, 3-4
  - directory structure, 3-1
  - procedure for partial restore, 3-7
  - procedure for partition restore, 3-8
  - restoring directories, 3-2
- MAGRST options, 1-5, 3-3
  - No\_Attributes, 1-2
  - not supported, 1-6
- MAGSAV
  - command format, 2-3
  - command options, 1-5, 2-3
  - description, 2-1
  - dialog, 2-6
  - directory structure, 2-5, 3-2
  - procedure for partition save, 2-11
- MAGSAV options, 1-5, 2-3
  - BIG, 1-6
  - BOOT, 2-1, 4-1
  - INC, 2-5, 2-6
  - LABEL, 1-3, 2-7
  - No\_Acl, 2-2

- NO\_DTA, 1-3
- QUERY, 2-6
- REV19, 1-8
- REV20, 1-8
- Save\_UFD, 3-2
- UPDT, 1-3, 2-3, 2-6, 9-3
- not supported, 1-5

MTRESUME, 4-7

MTRS, 1-6, 1-8, 2-3

Multiple save streams, PSR dialog, 6-10

Multiple save streams PSR, description, 6-3

## O

- Object names
  - MAGRST, 3-1, 3-6
  - MAGSAV, 2-8

## P

- Partial restore
  - checkpoints, 3-6
  - MAGRST dialog, 3-6
  - procedure, 3-7

Partial restore MAGRST, description, 3-1

Partition, identifier, 6-6

Partition identifier PSR, 6-4

Partitions

- MAGRST, 3-2
- MAGRST procedure, 3-8
- MAGSAV, 2-9
- MAGSAV procedure, 2-11
- physical copy to disk, 8-1, 8-6
- physical restore from tape, 7-1
- physical restore from tape procedure, 7-7
- physical save to tape, 6-1, 6-11
- procedure for removing from service, 10-5
- procedure for returning to service, 10-7
- robust PSR, 7-2, 8-2
- valid source and target PSR, 7-2, 8-2
- write protection PSR, 6-4, 7-3, 8-2

Passwords MAGSAV/RST

- No\_Acl, 2-4
- restoring, 3-2
- saving, 2-2

Performance

- backups, 9-5
- physical copy to disk, 8-8
- physical restore from tape, 7-10
- physical save to tape, 6-17

**Physical copy to disk**

- backups, 9-4
- command format, 8-4
- command options, 8-4
- description, 8-1
- dialog, 8-5
- examples, 8-5, 8-7
- performance, 8-8
- preparations, 10-2
- valid source and target PSR, 8-2

**Physical restore from tape**

- command format, 7-5
- command options, 7-5
- description, 7-1
- dialog, 7-5
- examples, 7-9
- preparations, 10-1
- procedure, 7-7, 8-6
- valid source and target PSR, 7-2

**Physical save to tape**

- command format, 6-6
- command options, 6-6
- description, 6-1
- dialog, 6-8
- examples, 6-12
- performance, 6-17
- preparations, 10-1
- PRIMOS revisions, 6-7
- procedure, 6-11

**Physical saves to tape, examples, 6-7**

**PRIMOS Revisions**

- incremental saves MAGSAV/RST, 1-8
- MAGSAV tape compatibility, 1-7
- MAGSAV tape format, 1-8
- MAGSAV/RST, ix
- physical save to tape, 6-3
- PSR, ix
- running MAGSAV/RST, 1-7, 1-8

**PRIMOS revisions**

- boot tapes, 4-1
- physical save to tape, 6-7

**Procedures**

- booting from tape, 4-7
- checkpoint restore MAGSAV/RST, 3-8
- partial restore, 3-7
- partition restore MAGRST, 3-8
- partition save MAGSAV, 2-11
- physical copy to disk, 8-6
- physical restore from tape, 7-7
- physical save to tape, 6-11
- removing a partition from service, 10-5
- returning a partition to service, 10-7
- system shutdown, 10-3

- system startup, 10-6

**PSR options**

- HELP, 6-5, 7-4, 8-3
- NO\_PROTECT, 7-3, 8-2
- PROTECT, 6-4, 7-3, 8-3

**R**

- Record availability table, PSR -SAVE -NO\_RAT, 6-7

**Record availability table PSR**

- base saves, 6-2
- invalid state, 6-3

**Related documentation, x**

**ROAM, 6-1**

**ROAM files**

- CAM\_RBF, 3-3
- NO\_RBF, 2-4
- QUERY, 3-3
- incremental saves MAGSAV, 1-3
- incremental saves MAGSAV/RST, 2-10
- physical copy to disk, 8-2, 10-3
- physical restore from tape, 7-2, 10-3
- physical save to tape, 6-1, 9-3, 10-3

**Robust partitions, 7-2, 8-2**

**S**

**Save number**

- PSR restore dialog, 7-6
- PSR save dialog, 6-9

**Superseded products, x**

**System shutdown procedure, 10-3**

**System startup procedure, 10-6**

**T**

**Tape blocks MAGSAV/RST**

- sizes, 1-6, 1-8, 2-3
- structure, 1-7

**Tape density, PSR dialog, 6-10**

**Tape format MAGSAV/RST**

- REV19, 1-8
- REV20, 1-8, 2-5
- block sizes, 1-6, 1-8
- block structure, 1-7
- PRIMOS Revisions, 1-8

**Tape indexing, extended, 1-4**

Tape indexing MAGSAV/RST  
 description, 5-1  
 examples, 5-3  
 format, 5-1, 5-2  
 Listing tape contents, 3-5, 3-6  
 listing tape contents, 2-11  
 MAGRST dialog, 3-5  
 MAGSAV dialog, 2-9

Tape labels  
 -LABEL, 1-3, 2-4, 2-7  
 ANSI, 1-6  
 MAGSAV dialog, 2-7  
 PSR save dialog, 6-9

Tape units  
 MAGRST dialog, 3-4

MAGSAV dialog, 2-6  
 PSR restore dialog, 7-6  
 PSR save dialog, 6-9

## **V**

Volume serial ID  
 -LABEL, 1-3, 2-4, 2-7  
 MAGRST dialog, 3-4  
 MAGSAV dialog, 2-6  
 PSR restore dialog, 7-6  
 PSR save dialog, 6-9