

ULTRIX

Guide to Backup and Restore

Order Number: AA-ME92B-TE

June 1990

Product Version: ULTRIX Version 4.0 or higher

This manual provides information on how to develop a backup and restore strategy. Included in this manual is information using backup and restore commands and utilities.

digital equipment corporation
maynard, massachusetts

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

© Digital Equipment Corporation 1989, 1990
All rights reserved.

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation 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.

No responsibility is assumed for the use or reliability of software on equipment that is not supplied by Digital or its affiliated companies.

The following are trademarks of Digital Equipment Corporation:

digital	DECUS	ULTRIX Worksystem Software
CDA	DECwindows	VAX
DDIF	DTIF	VAXstation
DDIS	MASSBUS	VMS
DEC	MicroVAX	VMS/ULTRIX Connection
DECnet	Q-bus	VT
DECstation	ULTRIX	XUI
	ULTRIX Mail Connection	

UNIX is a registered trademark of AT&T in the USA and other countries.

Contents

About This Manual

Audience	vii
Organization	vii
Related Documents	viii
Conventions	viii

1 Backup and Restore Methods and Strategies

1.1 Selecting a Backup Method	1-1
1.1.1 Using the dump or rdump Commands	1-1
1.1.2 Using the mdtar or tar Commands	1-2
1.1.3 Using the opser Utility	1-2
1.2 Selecting a Restore Method	1-2
1.2.1 Using restore or rrestore Commands	1-2
1.2.2 Using tar or mdtar Commands	1-3
1.3 Developing a Backup Strategy	1-3
1.3.1 Selecting a Full or Incremental Backup	1-3
1.3.2 Determining How to Use the Backup Levels	1-4

2 Backup Commands and Procedures

2.1 Backing Up File Systems	2-1
2.2 Transferring Files to Tape or Diskette	2-2
2.3 Backing Up from the Operator Account	2-3
2.3.1 Running the opser Utility	2-3
2.3.2 Determining Who Is Logged In	2-4
2.3.3 Shutting Down Multiuser Mode	2-4
2.3.4 Unmounting File Systems	2-5
2.3.5 Checking File System Consistency	2-5

2.3.6	Backing Up File Systems	2-6
2.3.7	Escaping to the Shell	2-6
2.3.8	Restarting Multiuser Mode	2-7
2.3.9	Performing a Remote Backup	2-7
2.3.10	Halting the Processor	2-7
2.3.11	Exiting from opser	2-7
2.4	The Labeled Tape Facility	2-8
3	Local File Restoration	
3.1	Using the restore Program	3-1
3.2	Preparing for an Interactive Restore	3-1
3.2.1	Logging In	3-1
3.2.2	Mounting the Dump Media	3-2
3.2.3	Changing Your Working Directory	3-2
3.3	Restoring Files Interactively	3-2
3.3.1	Invoking the Interactive restore Program	3-2
3.3.2	Using Interactive Mode Commands	3-3
3.3.2.1	Listing the Commands	3-3
3.3.2.2	Listing the Pathname of the restore Working Directory	3-3
3.3.2.3	Getting a File and Directory Listing	3-3
3.3.2.4	Comparing Inode Numbers	3-4
3.3.2.5	Changing Directories	3-4
3.3.3	Specifying What You Want Restored from the Dump Image	3-4
3.3.3.1	Creating and Modifying the Extraction List	3-5
3.3.4	Restoring the Specified Files and Directories	3-6
3.3.5	Entering the extract Command	3-6
3.3.5.1	Responding to the 'Specify next volume #' Prompt	3-7
3.3.5.2	Responding to the 'Set owner/mode for '.' Prompt	3-7
3.3.6	Exiting from the Interactive Restore	3-7
3.4	Restoring Files Noninteractively	3-7
3.4.1	Preparing for a Noninteractive Restore	3-8
3.4.1.1	Logging in	3-8
3.4.1.2	Mounting the Dump Media	3-8
3.4.1.3	Changing Your Working Directory	3-8
3.4.2	Invoking the Noninteractive restore Program	3-8
3.4.3	Exiting the Noninteractive Restore	3-9
3.5	Restoring a Complete File System	3-9

3.6	Local Restoration of the root and /usr File Systems	3-11
3.7	Using the tar or mdtar Commands to Restore Files	3-13
3.8	The Labeled Tape Facility	3-14
4	Remote Backup	
4.1	The Network Environment	4-1
4.1.1	The Master/Slave Relationship	4-1
4.1.2	Hardware and System Setup Considerations	4-1
4.1.2.1	Hardware Considerations	4-1
4.1.2.2	System Setup Considerations	4-2
4.1.3	Staging Area or Direct Backup Determination	4-3
4.2	Performing Remote Backups	4-3
4.2.1	Escaping to the Slave Shell	4-6
4.2.2	Escaping to the Master Shell	4-6
4.2.3	Unmounting the Slave's File Systems	4-6
4.2.4	File System Consistency Check on the Slave	4-7
4.2.5	Backing Up the Slave's File Systems	4-9
4.2.5.1	Staging Area Backup Method	4-9
4.2.5.2	Direct Backup Method	4-11
4.2.6	Returning to the Local opser	4-11
4.2.7	Halting the Slave Processor	4-12
4.2.8	Stopping Remote Execution of opser	4-12
4.2.9	Exceptional Condition Handling	4-12
5	Remote File Restoration	
5.1	Using the rrestore Program	5-1
5.1.1	Network Considerations	5-1
5.2	Preparing for an Interactive Restore	5-2
5.2.1	Logging In	5-2
5.2.2	Mounting the Dump Media	5-2
5.2.3	Transferring Files to the Staging Area	5-2
5.3	Restoring Files Interactively	5-2
5.3.1	Using Interactive Mode Commands	5-3
5.3.1.1	Listing the Commands	5-3
5.3.1.2	Listing the Pathname of the restore Working Directory	5-3
5.3.1.3	Getting a File and Directory Listing	5-3
5.3.1.4	Comparing Inode Numbers	5-3

5.3.1.5	Changing Directories	5-4
5.3.2	Specifying What You Want Restored from the Dump Image	5-4
5.3.2.1	Creating and Modifying the Extraction List	5-4
5.3.3	Restoring the Specified Files and Directories	5-6
5.3.4	Entering the extract Command	5-6
5.3.4.1	Responding to the 'Specify next volume #' Prompt	5-6
5.3.4.2	Responding to the 'Set owner/mode for '.' Prompt	5-7
5.3.5	Exiting from the Interactive Restore	5-7
5.4	Restoring Files Noninteractively	5-7

A Device Mnemonics

Examples

4-1:	Performing a File System Consistency Check	4-8
4-2:	Staging Area Backup Method Display	4-10

Tables

A-1:	Devices Supported by MAKEDEV	A-2
------	------------------------------------	-----

About This Manual

The objective of this guide is to provide you with information on backing up and restoring files and file systems using *ULTRIX* system commands and utilities. The guide assists you in developing system backup strategies locally and in a network environment. It also presents guidelines from which you can develop specific procedures for your site.

Audience

The *ULTRIX Guide to Backup and Restore* is written for the person responsible for managing and maintaining an *ULTRIX* system. It assumes that this individual is familiar with *ULTRIX* commands, the system configuration, the system's controller/drive unit number assignments and naming conventions, and an editor such as *vi* or *ed*. You do not need to be a programmer to use this guide.

Organization

This manual consists of 5 chapters, one appendix, and an index. The chapters and the appendix are:

- Chapter 1: Backup and Restore Methods and Strategies
 Identifies backup and restore methods and suggests certain strategies for backing up and restoring files and file systems.
- Chapter 2: Backup Commands and Procedures
 Identifies and explains how to use the various *ULTRIX* backup commands and utilities.
- Chapter 3: Local File Restoration
 Describes how to use the restore command and provides procedures for restoring file systems.

- Chapter 4: Remote Backup
Explains the network environment and how to use the `opser` utility to perform remote backups.
- Chapter 5: Remote File Restoration
Explains how to restore files and file systems using the `rrestore` command. This chapter also contains procedures for restoring the root and `/usr` file systems after a head crash has occurred.
- Appendix A: Device Mnemonics
Lists the supported device mnemonics and explains how to obtain detailed reference page information on devices.

Related Documents

You should have the hardware documentation for your system and peripherals.

Conventions

The following conventions are used in this manual:

- `%` The default user prompt is your system name followed by a right angle bracket. In this manual, a percent sign (`%`) is used to represent this prompt.
- `#` A number sign is the default superuser prompt.
- `>>>`
`CPU n >>` The console subsystem prompt is two right angle brackets on RISC systems, or three right angle brackets on VAX systems. On a system with more than one central processing unit (CPU), the prompt displays two numbers: the number of the CPU, and the number of the processor slot containing the board for that CPU.
- user input** This bold typeface is used in interactive examples to indicate typed user input.
- `system output` This typeface is used in interactive examples to indicate system output and also in code examples and other screen displays. In text, this typeface is used to indicate the exact name of a command, option, partition, pathname, directory, or file.
- UPPERCASE
lowercase The ULTRIX system differentiates between lowercase and uppercase characters. Literal strings that appear in text, examples, syntax descriptions, and function definitions must be typed exactly as shown.
- `rlogin` In syntax descriptions and function definitions, this typeface is used to indicate terms that you must type exactly as shown.

macro	In text, bold type is used to introduce new terms.
<i>filename</i>	In examples, syntax descriptions, and function definitions, italics are used to indicate variable values; and in text, to give references to other documents.
[]	In syntax descriptions and function definitions, brackets indicate items that are optional.
{ }	In syntax descriptions and function definitions, braces enclose lists from which one item must be chosen. Vertical bars are used to separate items.
. . .	In syntax descriptions and function definitions, a horizontal ellipsis indicates that the preceding item can be repeated one or more times.
.	A vertical ellipsis indicates that a portion of an example that would normally be present is not shown.
cat(1)	Cross-references to the <i>ULTRIX Reference Pages</i> include the appropriate section number in parentheses. For example, a reference to <code>cat(1)</code> indicates that you can find the material on the <code>cat</code> command in Section 1 of the reference pages.
RETURN	This symbol is used in examples to indicate that you must press the named key on the keyboard.
CTRL/x	This symbol is used in examples to indicate that you must hold down the CTRL key while pressing the key <i>x</i> that follows the slash. When you use this key combination, the system sometimes echoes the resulting character, using a circumflex (^) to represent the CTRL key (for example, ^C for CTRL/C). Sometimes the sequence is not echoed.
ESC X	This symbol is used in examples to indicate that you must press the first named key and then press the second named key. In text, this combination is indicated as ESC-X.

This chapter provides an overview of the backup and restore processes. The backup and restore processes allow you to back up and restore files on the local system or over the network. By performing regular backups, you are ensured a retrieveable copy of your files should the originals be destroyed.

This chapter discusses how to select a backup or restore method. Also included in this chapter is information that can help you develop a backup strategy.

1.1 Selecting a Backup Method

A backup operation lets you copy or transfer file systems or individual files and directories to tape, disk, or diskettes. The command or utility that you choose to back up files is determined by the type and location of files that you want to back up.

You can perform either a local backup or a remote backup. A local backup allows you to copy or transfer file systems or individual files and directories from the system on which you reside to tape, disk, or diskette. A remote backup lets you copy file systems over the network to tape or disk. You cannot do a remote backup of individual files and directories.

To back up files on the local system, you can use the `dump`, `mdtar`, or `tar` commands, or you can use the `opser` utility. To back up file systems over the network, you can use the `rdump` command or the `opser` utility. The following sections briefly highlight the backup commands and the `opser` utility. For more information, see Chapter 2.

1.1.1 Using the `dump` or `rdump` Commands

The `dump` command copies all designated file systems or individual files and directories changed after a specified date to a file, pipe, magnetic tape, disk, or diskettes. To use the `dump` command, you must have superuser privileges. The system can be in either multiuser mode or single-user mode for a backup operation; however, if you perform a backup of a file system or files in multiuser mode, any user or system activity might corrupt that backup.

The `rdump` command copies all designated file systems changed after a specified date over the network to magnetic tape, disk, or a dump image file. You cannot back up individual files and directories over the network. To use the `rdump` command, you must have superuser privileges and the system must be in single-user mode.

Both commands operate by checking the inodes of the files you want to back up. The inodes contain data such as table entries and other statistics. When you use this command to back up file systems or files, an inode is attached to each file. If system or user activity changes a file after the inode data is recorded, but before the file is backed up, the backup may be corrupted. Whether you are doing a local dump or a remote dump of a file system, it is recommended that you perform the operation in single-user mode.

1.1.2 Using the `mdtar` or `tar` Commands

The `mdtar` and `tar` commands save multiple files on multiple archives. Both commands have the same functionality. While these commands let you back up individual files and directories, you cannot back up file systems, nor can you perform a remote backup using these commands. To use the `mdtar` or `tar` commands, you must have read access on the files you want to back up and write privileges on the output device. To back up files from root (`/`) or `/usr`, you must be root. The system can be in multiuser mode.

The `mdtar` and `tar` commands operate by transferring files collectively. Unlike the `dump` command, these commands read the inode statistics but do not attach inodes to the files before transferring them to the output device. This lowers the possibility of file corruption when doing backups in multiuser mode.

1.1.3 Using the `opser` Utility

The `opser` utility is an interactive facility that lets you back up file systems on the local system or over the network. The local `opser` facility only supports tape as a backup media. The remote `opser` utility supports backups to files, disk, or tape.

To use `opser`, log in to the `operator` account; the `opser` utility is automatically invoked. This utility runs as a shell environment in place of the system shell. While in the `opser` environment, you can escape to the system shell, execute `ULTRIX` system commands, and return to the `opser` environment. The system must be in single-user mode when performing backups using the `opser` utility. A built-in help menu provides you with a list of the available commands that allow you to shut down the system, dismount devices, perform a file consistency check, and perform a backup. You can back up file systems only to tape.

This utility uses the `dump` and the `rdump` commands to perform a backup of your file systems. For information on how these commands work, see Section 1.1.1.

1.2 Selecting a Restore Method

A restore operation lets you retrieve file systems and individual files and directories that you copied to a designated dump device or dump image file. The method you choose to restore files is determined by the type of backup you performed.

If you used the `rdump` command or remote `opser` utility to back up files, you must use the `rrestore` command to retrieve files. If you used the `tar` or `mdtar` command to back up files, you must restore the files with the same command. If you used the `dump` command or local `opser` utility to back up files, you must use the `restore` command to restore files. The next two sections briefly describe these operations.

1.2.1 Using `restore` or `rrestore` Commands

The `restore` command retrieves, from magnetic tape, disk, dump image file, or a pipe, files that were backed up using the `dump` command or the `opser` utility. If you have write privileges on the files you are restoring, and if you do not plan to change file or directory attributes, you do not need superuser privileges to perform a restore operation.

The `rrestore` command retrieves from magnetic tape or disk, files that were backed up using the `rdump` command or the remote `opser` utility. To restore

individual files and directories, you do not need superuser privileges, but you must have write access to the files. To restore file systems, you must have superuser privileges.

Both the `restore` and `rrestore` commands operate by reading, from the dump media, the files that you specify. You can do an interactive restore or a noninteractive restore. The interactive restore process has a built-in help file and command options that allow you to do the restore. This is particularly helpful when you want to restore only parts of a dump. The noninteractive restore is most useful when you want to restore entire file systems or when you want to restore the entire dump. These commands do not perform consistency checks nor do they use the inode statistics to restore files.

1.2.2 Using tar or mdtar Commands

If you backed up your files with either the `tar` or `mdtar` command, the files must be restored using the same command. To restore files using `tar` or `mdtar`, you need write access to the files or directories that you want to retrieve. If you do not have write access, superuser privileges are necessary. As long as you do not attempt to access absolute pathnames, you can perform the restore in multiuser mode.

A restore operation, using the `mdtar` or `tar` command, transfers files from your backup medium to the directory or file you specify.

1.3 Developing a Backup Strategy

Regardless of whether you perform a local backup or a remote backup, you should develop a backup strategy that ensures you have a retrievable copy of the latest files should a mishap occur. Developing a backup strategy entails determining whether you need to perform a full backup of your file systems or determining whether an incremental backup is sufficient.

A full backup copies all files from a device to the backup medium you specify. An incremental backup copies a subset of files from a device to the backup medium you specify. The next two sections discuss how to select and specify a full or incremental backup.

1.3.1 Selecting a Full or Incremental Backup

To instruct the backup commands, `dump` and `rdump`, and the `opser` utility, to back up either all the files or a subset of files, you must specify a backup level. Backup levels range from 0 to 9. A level-0 backup instructs the commands or utility to back up all files. Level-1 to level-9 backup instructs the commands or utility to perform an incremental backup based on the dates and levels stored in the `/etc/dumpdates` file.

Any backup strategy should begin with a full backup of the file system. You must then determine how often you should perform incremental backups and when your next full backup should occur. This varies from site to site, as it depends on your work environment.

If there is constant testing or software development on your system, mishaps can occur that affect your file system. In this instance, setting up a monthly backup strategy is most feasible. You could begin with a full backup at the beginning of the month, followed by daily incremental backups during the week, and weekly incremental backups at the end of each week. If the environment that you work in

does not have much system activity and files are not updated or created frequently, you may feel comfortable performing a full backup followed by weekly incremental backups. Only you can determine the importance of your files and the need for backups.

1.3.2 Determining How to Use the Backup Levels

The backup levels range from 0 to 9. A level-0 backup always copies all files to the device you specify. For levels 1 through 9, the information stored in the `/etc/dumpdates` file determines how much of the file system is backed up. The `/etc/dumpdates` file contains the dates and level numbers of each backup performed on a file system. When performing a backup, the command or utility searches for the previous backup that is equivalent in level to that currently specified. If backups do not exist at that level, the backup command or utility searches for the next lowest level.

The following example demonstrates a monthly backup strategy. In this example, the files in the `root` file system are backed up to the default tape device. The `u` option instructs the `dump` command to update the `/etc/dumpdates` file after each successful backup. Level-0 backups are performed at the beginning of the month, level 9 backups are performed daily, and level 5 backups are performed weekly:

```
# /etc/dump 0u      1
# /etc/dump 9u      2
.
.
# /etc/dump 5u      3
# /etc/dump 9u      4
.
.
# /etc/dump 5u      5
# /etc/dump 9u
.
.
# /etc/dump 5u
# /etc/dump 9u
.
.
# /etc/dump 0u
```

The preceding example backs up files as follows:

- 1 Tells the `dump` command to do a full backup of the file system. This occurs as the first backup of the month.
- 2 Instructs the `dump` command to do an incremental backup. The `dump` command searches for the previous backup that occurred at level 9. Because a level 9 backup does not exist, it backs up all files created or changed from the next lowest level (level 0).
- 3 Initiates a weekly incremental backup. That is, the `/etc/dumpdates` file is searched for the previous backup that occurred at level 5. If, as in this example, the backup command or utility cannot locate a level 5, it searches for the next lowest level. In this instance, all files modified or changed since level 0 are backed up.

- 4 Continues daily backups. The `/etc/dumpdates` file is searched for the previous backup that occurred at level 9, and all files changed or created since that time are backed up.
- 5 Forces the `dump` command to back up those files changed since the last level 5 backup.

This chapter explains how to back up files and file systems using the ULTRIX backup and file copy/transfer commands and utilities. The backup and file transfer commands described in this chapter enable you to complete the following:

- Back up file systems
- Transfer files and file systems to tapes or diskettes
- Perform backups from the operator account

The following sections describe how to use ULTRIX commands and utilities to perform the above tasks.

2.1 Backing Up File Systems

Use the `dump` command to back up file systems. This command enables you to back up a file system to a tape or disk device. To back up file systems, the system must be in single-user mode. This section describes how to perform a file system backup of the root (/) and the /usr file systems:

1. Shut down the system. To shut down the system in five minutes and give users periodic warning messages, type:

```
# /etc/shutdown +5 'System going down to perform backups'
```

2. Determine the names of the devices containing the file systems to be backed up by listing the contents of the `/etc/fstab` file as follows:

```
# cat /etc/fstab
/dev/rz0a:/:rw:1:1:ufs::
/dev/rz0g:/usr:rw:1:3:ufs::
```

In this example, the / and /usr file systems are on the rz0 device, partitions a and g respectively.

If you have performed backups using the `dump` command with the `-u` option, you can also use the `dump` command with the `-w` option to determine which files systems need to be backed up. The `-u` option updates the `/etc/dumpdates` file.

3. Use the `umount` command with the `-a` option to dismount the files systems that you want to back up:

```
# umount -a
```

Note that the root (/) file system remains mounted.

4. Use the `fsck` command to perform a file system consistency check on the file system that you want to back up. The `fsck` command has the following format:

```
fsck device
```

Performing a file consistency check on the file system you want to back up ensures the integrity of the file system. The `fsck` command attempts to correct any inconsistencies.

5. Perform a level-9 backup of the root (`/`) file system to the default tape device. Use the following format:

```
# /etc/dump 9u /
```

6. Perform a level-9 backup of the `/usr` file system to a disk device. Use the following format:

```
/etc/dump 9uf device /usr
```

In these level-9 backups, the `dump` command provides you with several messages and prompts you to insert your media, depending on the device type that you specify. It also prompts you with a number of yes and no questions. Answer all of the questions.

Change the media when the program prompts you to do so. Be sure to label all your media with the date of the backup, the file system name, the starting inode number for that volume, and the backup level. This makes it easier to locate files and file systems that you want to restore. Store your backups in a safe location.

If you want to tailor the backup process so that certain options are in effect, see `dump(8)` in the *ULTRIX Reference Pages*.

2.2 Transferring Files to Tape or Diskette

The `tar` and `mtar` commands allow you to transfer an individual user's directories or files to tapes or diskettes. You do not need superuser privileges, nor do you have to be in single-user mode to use these commands. However, you must have read permission on the files you want to back up.

Use the `tar` command when magnetic tape is the primary backup medium. The `tar` command is a tape archive facility that supports single-volume and multivolume archives.

System users should invoke the `tar` command to back up their files, provided that the system has a tape drive. The following example shows how to back up a directory named `/usr/staff/ajw` to tape. This example assumes that tape drive 0, the default device, is being used:

```
# cd /usr/staff
# tar c ./ajw
```

To use the `tar` command along with the device name, type:

```
# tar cf /dev/rmt1h ./ajw
```

Use the `mtar` command when diskettes are the primary backup medium. More than one diskette may be needed. Given a directory pathname, the `mtar` command makes a list of the files and decides how to fit them on multiple diskettes.

The following example shows how to back up the directory `/usr/staff/ajw` to diskettes:

```
# cd /usr/staff
# mdtar c ./ajw
```


To use the `mdtar` command along with the device name, type:

```
# mdtar cf /dev/rmt1h ./ajw
```

The `mdtar` command prompts you to insert and remove additional diskettes as needed to back up the files.

2.3 Backing Up from the Operator Account

The `opser` utility, `/opr/opser` provides an interface for ULTRIX system maintenance, primarily file system backups. The `opser` utility runs as a shell environment in place of the system shell. To automatically invoke the `opser` utility, log in to the operator account. The operator password is set by the superuser.

This utility enables you to complete the following:

- Determine who is currently logged in to the system
- Shut down multiuser mode, enter single-user mode, and restart multiuser mode
- Check file system consistency
- Perform incremental or full file system backup to tape
- Perform incremental or full file system backup over the network
- Halt the processor
- Execute `opser` options on a remote system.

The `opser` utility also allows you to escape to the shell, execute ULTRIX system commands, and return to the `opser` utility. It also provides on-line help in the form of an options menu. For more information, see `opser(8)` in the *ULTRIX Reference Pages*.

To perform backups, run the `opser` utility from the operator's console. Running the `opser` services from a terminal limits you to the help, user, and shell functions.

The `opser` utility supports these tape devices for backup operations: TE16, TS11, TSV05, TU77, TU78, TU79, TU80, TU81, TU81E, TK50, and TK70. It also supports the network as a backup device.

2.3.1 Running the `opser` Utility

To run the `opser` utility, log in to the operator account. The following information is displayed on your screen when `opser` is invoked:

```
ULTRIX-32 Operator Services
Line editing: delete - erase one character, ^U - kill entire line
```

```
For help, type h followed by a return.
```

```
opr>
```

The `opr>` prompt informs you that the `opser` utility is running in place of the shell and is ready to accept `opser` options.

To use an `opser` option, type the appropriate letter or name. Then, press the RETURN key. For example, to display on-line help about all `opser` options, type:

```
opr> h
```

() - may use first letter in place of full name
Valid commands for Local Opser are:

```
!sh          - shell escape (execute ULTRIX-32 commands)
              (Type control d to return from shell)
(u)sers      - show logged in users
(s)utdown    - stop time-sharing
(d)ismount   - unmount file systems
(f)sck       - file system checks
(r)estart    - restart time-sharing
(h)elp       - print this help message
(b)ackup     - file system backup
halt         - halt processor
(n)etwork [Slave] - initiate Remote Opser
(q)uit       - exit from opser
```

To end an opser session and return to a login prompt, type q.

The next sections provide detailed discussions of the opser options.

2.3.2 Determining Who Is Logged In

To determine how many users currently are logged in to the system, enter the u (users) option at the opr> prompt. For example:

```
opr> u
```

When you enter the u option, the opser utility displays a list of users who are currently logged in.

2.3.3 Shutting Down Multiuser Mode

Use the s (shutdown) option to place the ULTRIX system in single-user mode. To back up the root (/) or /usr file systems, you must be in single-user mode. Other file systems that you may have mounted, such as /usr/src, can be unmounted before being backed up. You need to be in single-user mode only when backing up systems that are used in multiuser mode, such as /usr or user data areas. Even user data modes can be backed up if no users are logged in and you have /etc/nologins on.

To shut down multiuser mode and leave the system in single-user mode, enter the s (shutdown) option at the opr> prompt. For example:

```
opr> s
```

When you enter the s option, the opser utility first displays the names of those users currently logged in to the system. Then, it prompts for the number of minutes to delay before shutting down from multiuser to single-user mode. The number of minutes that you enter must be between 1 and 99. This delay gives users enough time to finish their work and to log out. During this delay period, the opser utility broadcasts warnings of the impending shutdown. Five minutes prior to the designated time, it disables logins. Finally, at the designated time, it broadcasts a final message and then shuts down multiuser mode.

Prior to entering single-user mode from multiuser mode, the opser utility kills all running processes and synchronizes the file systems (using the sync utility). When the opser utility redisplayes the opr> prompt, you are automatically in a single-user environment. You can now enter other opser options and continue with system maintenance.

2.3.4 Unmounting File Systems

To unmount the file systems when running `opser` in single-user mode, enter the `d` (dismount) option at the `opr>` prompt. For example:

```
opr> d
```

When you enter the `d` option, the `opser` utility invokes the `umount` command to unmount file systems listed in the `/etc/fstab` file. For more information on the `umount` command, see `mount(8)` in the *ULTRIX Reference Pages*.

To ensure that all of the file systems have been unmounted, escape to the shell, using the `!sh` option as described in Section 2.3.7, and enter the `mount` command.

Without options, the `mount` command displays all of the currently mounted file systems. At this point, you will have to unmount the file systems by issuing the appropriate `umount` command.

The following example shows the sequence of commands that you would enter assuming that the `/usr/staff3` file system is still mounted after you entered the `d` option:

```
opr> !sh
Password:
type ^D to return to opser

# /etc/mount
/dev/rz0a on / type ufs
/dev/rz1h on /usr/staff3 type ufs
# /etc/umount /dev/rz1h
# <CTRL/D>
opr>
```

In this example, the shell escape option, `!sh`, returns you to the system shell prompt (`#`) after the root password is entered. Entering the `mount` command without options displays the list of mounted file systems, in this instance, `root (/)` and `/usr/staff3`. The `root` file system should remain mounted; however, `/usr/staff3` is dismounted by issuing the `umount` command on the `/dev/rz1h` device. Type `CTRL/D` to return to the `opr>` prompt.

Note

When you unmount the `/usr` file system, some *ULTRIX* commands may be unavailable.

2.3.5 Checking File System Consistency

The file systems should be checked prior to performing a backup. You should not back up a corrupted file system. To check the consistency of your file systems when running `opser` in single-user mode, enter the `f` (file system check) option at the `opr>` prompt. For example:

```
opr> f
```

When you enter the `f` option, the `opser` utility invokes the `fsck` command to check the file systems indicated in the `/etc/fstab` file.

2.3.6 Backing Up File Systems

You should unmount and then check all file systems before backing them up. Use the `d` option to unmount the file systems. Use the `f` option to check the file systems for consistency. To shut down multiuser mode prior to unmounting and checking file systems, specify the `s` (shutdown) option.

To back up file systems when running the `opser` utility in single-user mode, enter the `b` (backup) option at the `opr>` prompt. Enter the option using the format:

```
b command file name
```

The *command file name* argument specifies the command file that contains the ULTRIX system commands required for that backup. If you do not use the *command file name* argument, the `opser` utility calls the default backup command file, `/opr/backup`. This script enables you to perform level-0 or level-9 backups to supported tape drives.

The `/opr/backup` command file prompts you to specify the tape drive number, the logical unit number of the tape drive, and the tape density. For example:

```
opr> b backup
```

```
ULTRIX-32 File System Backup Procedure
```

```
start errlog daemon - elcsd
elcsd:
```

```
Please specify the tape drive type. (eg. tel6):
Please specify the logical unit number for the tape drive. (eg. 0):
Please specify tape density. (eg. 1600):
```

If you specify either a TK50 or a TK70 for the tape drive type, this backup script will not prompt you for a tape density. If any of your entries are invalid, the command file responds with the message:

```
Error in specifying tape drive - repeating questions.
```

When you have correctly specified the device, the backup script prompts:

```
Backup to be full or incremental? < f or i >:
```

If you enter an `f`, a level-0 backup is performed. If you enter an `i`, a level-9 backup is performed.

After you have correctly responded to the prompts, the file system dump process begins.

The default backup command file also invokes the error logger daemon in single-user mode, so that error logging is enabled during backups. The error logger daemon is described in the *ULTRIX Guide to the Error Logger*.

As the superuser, you should set up backup command files in the way that you want your backups performed.

2.3.7 Escaping to the Shell

To escape to the shell at any time when running the `opser` utility, enter `!sh` at the `opr>` prompt. The `opser` utility prompts you to enter the root password. This is the only password you can enter. For example:

```
opr> !sh
Password:
type ^D to return to opser
```

```
#
```

After receiving a shell prompt, you can execute any ULTRIX command. The `!sh` command is useful for using the `talk` command to tell users that you intend to shut down the system. After you finish entering ULTRIX commands, press `CTRL/D` to return to the `opr>` prompt.

2.3.8 Restarting Multiuser Mode

To restart multiuser mode when running the `opser` utility in single-user mode, enter the `r` (restart) option at the `opr>` prompt. For example:

```
opr> r
```

This option causes the system to restart multiuser mode. The system displays the same startup messages and starts the same daemons as though a normal startup had occurred. When the system is back in multiuser mode, it again displays the standard login prompt.

2.3.9 Performing a Remote Backup

The `opser` utility enables you to back up a file system over the network. Chapter 4 explains how to use the `opser` utility to perform a remote backup.

2.3.10 Halting the Processor

To halt the processor when running the `opser` utility in single-user mode, enter the `h` (halt) option at the `opr>` prompt. For example:

```
opr> halt
```

This command synchronizes the disks (writes out in-memory file system information to the disks) and then halts the processor. You must then reboot your system manually as described in the *ULTRIX Guide to Shutdown and Startup*.

Note

To shut down multiuser mode prior to halting the processor, specify the `s` (shutdown) option.

2.3.11 Exiting from opser

You can exit the `opser` utility and return to the login prompt by using the `q` option, provided that you have not shut the system down to single-user mode. Once you shut the system down, you must use the `r` (restart) option to exit. Section 2.3.9 describes how to use the `r` option. If you enter `q` at the `opr>` prompt after shutting down the system, `opser` displays the message:

```
Time-sharing stopped
opr>
```

If you have not shut the system down and you want to exit `opser`, enter the `q` (quit) option at the `opr>` prompt. For example:

opr> q

The `opser` utility logs you out of the operator account and the system will redisplay the login prompt.

2.4 The Labeled Tape Facility

In addition to the backup methods already described, the system contains a labeled tape facility command, `ltf`. The `ltf` command reads and writes single-volume, versions three and four, ANSI-compatible tape volumes. This provides you with a way of accurately transferring information between ULTRIX and other systems. For information on this command, refer to `ltf(1)` and `ltf(5)` in the *ULTRIX Reference Pages*.

This chapter describes how to restore files and file systems using ULTRIX commands. Specifically, this chapter enables you to do the following:

- Restore files, directories, or file systems on different media using the `restore` command
- Transfer files using the `tar` and `mdtar` commands
- Restore a complete file system
- Restore the root (/) and /usr file systems after a catastrophic event has occurred

3.1 Using the restore Program

Use the `restore` program to restore files, file systems, and directories that you backed up with the `dump` program.

Note

If you used the `tar` or `mdtar` command to back up your files, you must use the `tar` or `mdtar` command again to restore the backup media. See Section 3.7 for details.

With the `restore` program, you can do either an interactive or a noninteractive restore operation. The following sections explain both methods with examples of how to perform each.

3.2 Preparing for an Interactive Restore

Before starting the actual restoration, you should prepare in three ways:

- Log in as yourself or as superuser to root.
- Mount the dump media.
- Change your working directory to the dump media mount point.

The following sections describe each of these preparatory tasks.

3.2.1 Logging In

Some `restore` functions require superuser privileges. For example, to change file and directory attributes during a restore operation, you must be logged in as superuser. See `restore(8)` in the *ULTRIX Reference Pages* for details on the `restore` command options and required permissions.

3.2.2 Mounting the Dump Media

Before invoking the `restore` program, ensure that the backup media containing the dump image is mounted on the appropriate device. This is usually the default tape device, `/dev/rmt0h`, but it could be any device or a file system in the form of an on-line backup file. In a multivolume dump set, you must mount volume 1 first, because it contains information on the directory structure and inodes for all files on the dump image. When you have established the restore environment, you can mount other volumes in the set.

3.2.3 Changing Your Working Directory

Change your current working directory to the mount point of the file system that was backed up. For example, if the `/usr` file system was backed up from `root`, first position yourself at `root`, then enter the command to invoke `restore` from this location.

Note

If you are unsure of the mount point, enter the `/etc/restore` command with the `-t` option for a complete listing of the dump media contents.

Be aware that when you invoke the `restore` program, you enter the `restore` environment. While in this environment, you get directory information about the dump media only, not about your working environment.

The `restore` program restores files to the current working directory (the directory from which you invoked the program) using the relative pathname of the dump media files that are being restored.

3.3 Restoring Files Interactively

To restore individual files or directories from the dump image on the backup media, use the `/etc/restore` command with the `-i` option. The `-i` option specifies that you want to do the restore in interactive mode.

3.3.1 Invoking the Interactive restore Program

The location of the dump image determines what command format you use to invoke the `restore` program:

- If the dump image is located on a device other than the default (`/dev/rmt0h`), use the format:

```
/etc/restore -if device
```

The `-i` option specifies interactive mode. The `-f` option and `device` argument let you specify the name of the archive. Include the `-f` option and `device` argument when the archive is a device other than `/dev/rmt0h`. If the device is the default tape drive `/dev/rmt0h`, exclude the `-f` option and `device` argument on the command line as follows:

```
# /etc/restore -i
```


- If the dump image is an on-line backup file on a mounted disk, use the format:

```
/etc/restore -if file
```

Here, *file* specifies the dump image backup file. Specify either the full or relative pathname, depending on your current working directory. For example, to restore the `/usr` file system from a dump image named `/usr/backups`, position yourself at root and type:

```
# /etc/restore -if /usr/backups
```

3.3.2 Using Interactive Mode Commands

When you type the `/etc/restore` command with the `-i` option, the `restore >` prompt appears on your screen. The prompt signals that the `restore` program is waiting for a command.

3.3.2.1 Listing the Commands – To get a listing of the available commands, enter an `h` or a `?` at the `restore >` prompt. The program responds with a listing. For example:

```
restore >h
Available commands are:

ls [arg] - list directory
cd arg - change directory
pwd - print current directory
add [arg] - add 'arg' to list of files to be extracted
delete [arg] - delete 'arg' from list of files to be extracted
extract - extract requested files
quit - immediately exit program
verbose - toggle verbose flag (useful with "ls")
help or '?' - print this list
If no 'arg' is supplied, the current directory is used
restore >
```

3.3.2.2 Listing the Pathname of the restore Working Directory – To list the pathname of the working directory within the `restore` environment, type `pwd` at the `restore >` prompt. For example:

```
restore >pwd
/lib
restore >
```

3.3.2.3 Getting a File and Directory Listing – To get a listing of the files and directories that can be restored at the current directory level, type `ls` at the `restore >` prompt. For example:

```
restore >ls
adm/      hosts/    man/      skel/
bin/      include/ mdec/     spool/
dict      kits     msgs/     src/
restore >
```

To get a listing of the files and directories that can be restored at a different directory level, enter the pathname of that directory as an argument to the `ls` command.

3.3.2.4 Comparing Inode Numbers – Inode numbers increase with each tape. The largest inode numbers are on the last tape and the smallest numbers are on the first tape. By comparing the displayed inode numbers with the starting inode numbers recorded on a multivolume dump set, you can find the tape or disk that contains the file you want to restore. To see the inode numbers of the files contained within your listing, first use the `verbose` command to enter verbose mode, then use the `ls` command to get your listing. The `restore` program displays the files with the inode numbers to the left of the dumped file names. For example:

```
restore >verbose
verbose mode on
restore >ls
   2 *./      11971 hosts      14253 spool/
   2 *../      2177 include/    4145  src/
11969 adm/     11427 mdec/       4253  staff
 1633 bin/     2722  msgs/       3266  sys/
restore >
```

In this example, the `hosts` file has an inode number of 11971 and the `spool` directory has an inode number of 14253.

Note

Sometimes the inodes for a given file span multiple tapes or disks. When this happens, you will need more than one tape or disk to restore the file.

After checking the inode numbers, you can turn off verbose mode. To do this, type `verbose` one more time. For example:

```
restore >verbose
verbose mode off
restore >
```

3.3.2.5 Changing Directories – Change directories in the `restore` environment with the `cd` command. To specify a particular directory, enter the `cd` command with the directory pathname given as an argument to the command. For example:

```
restore >cd /lib
restore >
```

3.3.3 Specifying What You Want Restored from the Dump Image

When working interactively with the `restore` program, you must first decide which files or file systems you want restored from the dump media. Use the `ls` command to get a file or directory listing of the dump media. When you know what is on the media, you can specify what you want restored.

You specify what you want restored by creating and modifying an extraction list. Creating an extraction list involves working from the dump image and using the `restore` program's `add` command. Modifying an extraction list involves working from the dump image and using the `restore` program's `add` and `delete` commands.

3.3.3.1 Creating and Modifying the Extraction List – To create and modify an extraction list, you must be within the `restore` environment. When you invoke the `restore` program, follow these steps:

1. Use the `pwd` command to determine your position within the dump media directory structure.
2. Use the `ls` command to identify which files and directories are on the dump image at the current level.
3. Decide which files or directories you want to extract from the dump image.
4. Use the `cd` command to move around the dump image and confirm the location of files and directories.
5. Position yourself in the dump media directory structure at a point that is one level above the desired item. For example, to restore `/usr/lib`, position yourself at `/usr`. If you attempt to identify `/usr/lib` while positioned within `/usr/lib`, the program issues an error message.
6. Create the extraction list by typing the `add` command and the name of each desired file or directory as an argument to the command.

For instance, if you are in the `/usr` directory and you want to extract the files `lpf` and `lpd` and the subdirectory `refer`, enter the `add` command followed by the file and directory names:

```
restore >add lpd lpf refer
restore >
```

Note

When you add a file to the extraction list, the program marks that file as one to restore. When you add a directory to your extraction list, the program marks that directory, its files, and all subordinate directories and files as ones to restore.

7. Use the `ls` command to see how the program has marked these files. For example:

```
restore >ls
diffh          libc_p.a          *lpd
dnet/          libcg.a          *lpf
how_pix        libpc_p.a        *refer/
restore >
```

Notice the asterisks (*) in front of the items that you added to your extraction list. When you view the directory listing, the asterisks point out what you put on the list. Similarly, when the `restore` program reads an asterisk during the extraction process, it transfers a copy of that file or directory from the dump image to your current working directory.

8. Use the `delete` command with the file name given as an argument to remove a particular file from the list. For example:

```
restore >delete lpf
restore >
```

9. Use the `ls` command to see what the program did in response to the `delete` command. For example:

```
restore >ls
diffh                libc_p.a                *lpd
dnet/                libcg.a                lpf
how_pix             libpc_p.a             *refer/
restore >
```

Notice that the `lpf` file has no asterisk now. This change indicates that the program took `lpf` from the list and will not extract it from the dump image.

10. When you create or modify the extraction list, you request a transfer of the files to your system. To do this, use the `extract` command.

3.3.4 Restoring the Specified Files and Directories

The `restore` program's `extract` command initiates the actual restore. When you enter the `extract` command, the program transfers a copy of each marked item to the current working directory using the relative pathname found on the dump media.

Note

The current working directory is the one from which you invoked the `restore` program initially. Your position within the dump media directory structure when you enter the `extract` command has no influence on the final location of the restored files.

For example, assume that the file `/usr/lib/lpf` was dumped to tape with `/usr` as the mount point of the backup. In this case, `./lib/lpf` is the relative pathname on the dump media. If you invoked the `rrestore` program from `/usr` (in other words, with `/usr` as your current working directory), the program restores `/lib/lpf` as subordinate to `/usr` (`/usr/lib/lpf`). If you invoked the `rrestore` program from `/lib` (in other words, with `/lib` as your current working directory), the program restores `/lib/lpf` as subordinate to `/lib` (`/lib/lib/lpf`).

3.3.5 Entering the `extract` Command

To extract the files that you placed on your extraction list, enter the `extract` command at the `restore >` prompt. For example:

```
restore >extract
```

In response, the `restore` program displays several statements and prompts you for some answers. For example:

```
You have not read any tapes yet.
Unless you know which volume your file(s) are on you
should start with the last volume and work forward
towards the first.
Specify next volume #:
set owner/mode for '.'? [yn]
```

3.3.5.1 Responding to the 'Specify next volume #' Prompt – The first prompt, Specify next volume #:, assumes that you are working from a multivolume backup set.

- If you are working from a multivolume set, and you know that the files you want extracted start on a certain volume number, then enter that volume number. If you do not know which volume contains the files that you want extracted, enter the number of the last volume of the set and work your way towards the first. For example, for a three-volume backup set, enter 3 at the initial prompt. The program returns a second prompt to which you respond by entering 2, and so on.
- If you are working from a single-volume backup tape, enter 1 in response to the initial prompt.

In response to your entry, the `restore` program searches the dump image for the items listed on the extraction list. When it finds the individual files and directories, it restores them.

3.3.5.2 Responding to the 'Set owner/mode for .' Prompt – The second prompt, set owner/mode for '.', lets the superuser change the attributes of the current directory (shown as '.' within the prompt) to the attributes found for that entry on the dump image. The current directory's attributes include access permissions, ownership, and the accessed and update times.

- If you want to change the attributes, enter a y.

The `restore` program responds by changing the current directory's attributes to those found on the dump image for the corresponding directory. It then redisplay the `restore >` prompt.

If you are not the superuser and you enter a y in response to the prompt, the program responds with an error message and does not change the attributes.

- If you do not want to change the attributes or if you do not have superuser privileges, enter an n.

The `restore` program leaves the attributes of the current directory unchanged. It then redisplay the `restore >` prompt.

3.3.6 Exiting from the Interactive Restore

When you finish restoring files, type `quit` to get back to the system prompt:

```
restore >quit
#
```

3.4 Restoring Files Noninteractively

Using a noninteractive restore method is useful when you want to restore an entire file system or the entire dump image.

The following sections describe how to prepare for a restoration and how to invoke the noninteractive `restore` program.

3.4.1 Preparing for a Noninteractive Restore

Before starting the actual restoration, you should prepare in three ways:

- Log in as yourself or as superuser to root.
- Mount the dump media.
- Change your working directory to the dump media mount point.

The following sections describe each of these preparatory tasks.

3.4.1.1 Logging in – Some restore functions require superuser privileges. For example, if you intend to change directory attributes when restoring, you must be superuser. See `restore(8)` in the *ULTRIX Reference Pages* for details on the `restore` command options and required permissions.

3.4.1.2 Mounting the Dump Media – Before invoking the `restore` program, ensure that the backup media containing the dump image is mounted on the appropriate device. This is usually the default tape device, `/dev/rmt0h`, but it could be any device or file system in the form of an on-line backup file. In a multivolume dump set, you must mount volume 1 first, because it contains information on the directory structure and inodes for all files on the dump image. When you have established the `restore` environment, you can mount other volumes in the set.

3.4.1.3 Changing Your Working Directory – Before invoking `restore`, change your current working directory to the mount point of the file system that was backed up. For example, if the `/usr` file system was backed up from `root`, first position yourself at `root` and then enter the command to invoke `restore` from this location.

Note

If you are unsure of the mount point, enter the `/etc/restore` command with the `-t` option for a complete listing of the dump media contents.

Be aware that, when you invoke the `restore` program, you enter the `restore` environment. While in this environment, you get directory information about the dump media only, not your working environment.

The program restores files to the current working directory (the directory from which you invoked the `restore` program), using the relative pathname contained on the dump media. If you are in the wrong directory, the program could overwrite the files in that area if they happen to have the same file name.

3.4.2 Invoking the Noninteractive restore Program

To restore files noninteractively, enter the `/etc/restore` command with the options and arguments that specify what you want restored. See `restore(8)` in the *ULTRIX Reference Pages* for a description of the command, its options, restrictions, and diagnostics.

The following examples demonstrate various command line entries:

- If the dump image is located on the default device, and you want to restore the entire image, use the format:

```
#/etc/restore -x
```

The `-x` option specifies that the root directory and all subordinate items are to be extracted.

- If the dump image is on a device other than the default, and you want to restore the entire image, use the format:

```
/etc/restore -xf device
```

The `-x` option specifies that the entire image is to be restored. The `-f` option and *device* argument let you specify the name of the archive. For example, if the dump media resides on a device named `/dev/rmt1h`, and you want all items restored, enter:

```
#/etc/restore -xf /dev/rmt1h
```

- If the dump image is located on the default device, and you want to restore specific files from the image, use the format:

```
/etc/restore -v pathname1 pathname2
```

Here, the `-v` option specifies that you want the program to display the name and type of each file it extracts. The *pathname* arguments specify that the named files are to be extracted from the dump media. If these files match a directory on the dump media, the directory and subordinate files and directories are also extracted. For example, to restore (in verbose mode) the files `/usr/lib/a` and `/usr/bin/b` which reside on the default device, enter:

```
#/etc/restore -v /usr/lib/a /usr/bin/b
```

3.4.3 Exiting the Noninteractive Restore

At the end of a noninteractive restore, the `restore` program exits automatically and the system redisplay its prompt.

3.5 Restoring a Complete File System

To restore a complete file system, you must first unmount it. For information on unmounting a file system, see `mount(8)` in the *ULTRIX Reference Pages*. When the file system is unmounted, perform these steps:

1. Create the target file system. To create the file system, use the `newfs` command on an unmounted (raw) device.

Caution

When re-creating a file system, the `newfs` command destroys all existing data on the partition. Therefore, you should back up important data before issuing `newfs`. To check the partitions, use the `chpt` command with the `-q` option.

For example, to use the `newfs` command to create a new file system on partition `g` of an RZ55 disk on drive 0, enter:

```
# /etc/newfs /dev/rz0g.rz55
```

For more information on the `newfs` command, see `newfs(8)` in the *ULTRIX Reference Pages*.

2. Check and mount the target file system. After creating a target file system, you must check it using the `fsck` command and then mount it using the `mount` command. To perform file consistency check, use the following format:

```
fsck device
```

To mount the file device onto the target directory `/usr`, use the following format:

```
/etc/mount device /usr
```

- When you invoke the `fsck` command, it displays certain information on the file system. Refer to `fsck(8)` in the *ULTRIX Reference Pages* for further details.

3. Change to the directory on the target file system. When the target file system is available for use, use the `cd` command to position yourself at the top of the directory, in this case, `/usr`:

```
# cd /usr
```

4. Restore the dumped file system from the backup media. Having sufficiently prepared the target file system, restore the file system from its most recent and most complete dump tapes or disks. Use the `/etc/restore` command with the `r` option. For example:

```
# /etc/restore r
```

This command restores all of the files and directories in the file system from a dump tape mounted on the default tape device, `/dev/rmt0h`, to the current directory, `/usr`.

5. Remove the restore table by removing the file named `restoresymtable` that `restore` created in the current directory. Then reposition yourself at the root (`/`) directory. For example:

```
# rm restoresymtable  
# cd /
```

6. Unmount the restored file system by using the following format:

```
/etc/umount device
```

This command unmounts the file system `/usr` which was mounted in step 2, and prepares the file system for the next step. Refer to `mount(8)` in the *ULTRIX Reference Pages* for more information on the `umount` command.

7. Check the file system. After unmounting the restored file system, use the following format to check for inconsistencies:

```
/etc/fsck device
```

The `fsck` command checks the named file system, notifies you of all inconsistencies, occasionally prompts you for a response to its suggested course of action, and proceeds accordingly. When unsure of the consequences of your response, you should answer `no`. By answering `no`, you leave the condition uncorrected, but create a summary from which you can decide on a plan of action. You can also use the `fsck` command with either the `-p` or `-P` options.

Refer to `fsck(8)` in the *ULTRIX Reference Pages* for information on these as well as other `fsck` command options.

Note

If you are using `fsck` on the root partition and `fsck` finds and corrects errors, you should halt the system without synchronizing the disks. See the documentation for your processor for information on how to halt the processor. Halting the processor returns you to the console subsystem and allows you to reboot the system. It also ensures that the system buffer cache does not overwrite the `fsck` corrections. You must then reboot your system to get the corrected root file system into memory.

For further details, see the information on maintaining file system consistency in the *ULTRIX Guide to System Crash Recovery*.

3.6 Local Restoration of the root and /usr File Systems

This section describes a procedure for restoring the root (/) and /usr file systems of your system disk.

The procedure in this section requires you to have access to the most recent dump image of your root (/) and /usr file systems. The dump information should be available either in the form of a dump tape or as a sequence of dump diskettes.

You should use this procedure only when a catastrophic error occurs on the system disk, such as a disk crash, or when the inadvertent deletion of either the root (/) or /usr file systems renders the system inoperative.

The following example assumes that you are restoring your root (/) and /usr file systems from a level 0 TK50 dump tape.

Disk device file names are processor specific. Appendix A lists the disk device special file name mnemonics and provides information on how to find their associated special file names.

1. Boot the TK50 device where the standalone system volume is mounted. The boot command is described in the *Basic Installation Guide* for your specific processor.

After entering the boot command, the system prompts you to enter 1 for the Basic Installation, 2 for the Advanced Installation, or 3 for System Management.

2. Enter 3 for System Management.
3. Run MAKEDEV in the /dev directory to make the system disk special file and the tape device special file. The command line formats for making the special files are as follows:

```
cd /dev
MAKEDEV mnemonic
```

The *mnemonic* argument refers to a device mnemonic. See Appendix A for a listing of the supported device mnemonics. These commands create the necessary special files in /dev for the system disk and the TK50 tape drive.

4. Change to the root (/) directory.

```
# cd /
```

5. Make a new root file system using the `mkfs` command. This command requires you to use the following format:

```
mkfs dev name 32768 sectors tracks
```

The *dev name* variable specifies the raw device special file. The number 32768 specifies the size of the root file system in 512-byte blocks. This is a constant used by all disks. The *sectors* and *tracks* variables specify the disk-specific number of sectors and number of tracks for the target disk. The following table lists the supported disk drives, the *sectors*, and the *tracks* entries.

Disk	Sectors	Tracks
ra60	42	4
ra70	33	11
ra80	31	14
ra81	51	14
ra81	57	15
ra90	69	13
rd53	18	8
rd54	17	15
rm05	32	19
rp07	50	32
rf30	37	6
rf31	50	8
rf71	37	16
rz55	36	15
rz56	54	15

As described in the `mkfs(8)` reference page, additional entries can be made for the block size and the fragment size. However, the accepted approach is to let `mkfs` use its defaults of 8192 and 1024 respectively.

The following example shows you the correct `mkfs` entry for making the root file system on partition a of an `rd54` disk. In this example, raw device is `/dev/rra0a`. If this disk drive had been connected to a VAXstation 2000, the device entry would have been `/dev/rrd0a`.

```
# /etc/mkfs /dev/rra0a 32768 17 15
```

6. Reload the boot block. This step is necessary only if the disk is physically damaged or replaced for some other reason. The boot block should normally be operational. Use the following command line format to reload the boot block:

```
dd if=/vaxboot of=device conv=sync
```

7. Check the new file system for consistency using the following format:

```
/etc/fsck device
```

If you do not do a file consistency check at this point, you will be unable to mount the file system later.

8. Mount the root file system using the following format:

```
/etc/mount device /mnt
```

9. Change to the /mnt directory and restore the root file system using the following format:

```
cd /mnt
restore -rf device
```

If you are restoring the root dump image to the default tape device, you do not have to specify the `f` option or the device name.

10. Unmount the file system and check it for consistency using the following format:

```
cd /
/etc/umount device
/etc/fsck device
```

11. Halt and reboot the system as follows:

```
# sync
# sync
# halt
```

```
>>>b device
```

In the boot command line, you must specify the name of the newly restored disk.

12. Use the steps in Section 3.5 to restore the /usr file system.

3.7 Using the tar or mdtar Commands to Restore Files

If you backed up your files and directories with the `tar` and `mdtar` commands, you must use these commands to do your restore. You do not need superuser privileges, nor do you have to be in single-user mode to use these commands.

The following procedure assumes that a relative pathname was used when the original `tar` image was created. To restore individual files:

1. Change the directory. Use the `cd` command to go to the home directory of the file or files that you want to extract. For instance, if you want to extract files in the /usr directory, type:

```
# cd /usr
```

When you are in the /usr directory, you can list the files stored on the archive media by typing:

```
# tar t
```

If you want to display the file pathnames on a specific device, type:

```
# tar tf /dev/rmt1h
```

The `tar` command displays the pathnames of the files in the `tar` image.

2. Extract the files. To extract a single file or a series of named files from tape, type:

```
# tar xp file1 file2 file3...
```

The `tar` command extracts the listed files using the path name that was placed on the `tar` archive media. The `p` option restores the named files to their original protection codes. The `p` option only takes effect if you are logged in as the superuser.

If you specify the `tar` command with the `xp` options and do not specify file names, you will extract all of the files on the archive media by default. For example:

```
# tar xp
```

There are many other options that you can use with the `tar` and `mdtar` commands. For more information, refer to `tar(1)` and to `mdtar(1)` in the *ULTRIX Reference Pages*.

3.8 The Labeled Tape Facility

In addition to the restore methods already described, the system contains a labeled tape facility, `ltf`. The `ltf` command reads and writes single-volume, versions three and four, ANSI-compatible tape volumes. This provides you with a way of accurately transferring information between ULTRIX and non-ULTRIX systems. For information on this command, refer to `ltf(1)` and `ltf(5)` in the *ULTRIX Reference Pages*.

This chapter introduces conceptual and setup information needed to perform a network backup and explains how to back up file systems over the network. The chapter also explains how to use the remote `opser` utility.

The chapter contains the following topics:

- The network environment
- Performing remote backups

4.1 The Network Environment

This section introduces concepts that you need in order to perform a remote backup in a master/slave environment and identifies hardware and system setup considerations. This section also explains the concept of performing a network backup either directly to a tape device or through the use of a staging area.

4.1.1 The Master/Slave Relationship

A master system is the local system that you use to initiate a remote backup. You initiate a remote backup by running the remote `opser` utility as described in Section 4.2. You must be logged in to the master system to perform a remote backup. The master system should have a tape drive and a disk drive that you can use to transfer data from the remote system.

The slave system is the system from which the master system takes the backup data. In some cases, a slave system may have only fixed disks and no tape drive device. Therefore, the only way to back up the slave system files is from another system. It can also be more efficient and economical to centralize the backup process, regardless of the slave systems' hardware.

4.1.2 Hardware and System Setup Considerations

Before performing a network backup, there are several hardware and system setup considerations to take into account. The hardware considerations described in this section focus on the tape and disk drives that you use to perform a network backup. The system setup considerations described in this section focus on the system files that have to be in place to perform a network backup.

- 4.1.2.1 Hardware Considerations** – The primary hardware consideration deals with the type of tape drive connected to the master system. Tape drives can have either a streaming mode capability or a start/stop mode. Although having a tape drive with a streaming mode capability is not a requirement, this type of tape drive will effect a faster backup process when moving a staging file to tape. All Tape Mass Storage Control Protocol (TMSCP) tape drives are classified as streaming tape drives. Refer to `tms(4)`, `tu(4)`, and `ts(4)` in the *ULTRIX Reference Pages* for information on

specifying these tape devices.

The tape drives that have a streaming mode capability are:

- TA81
- TA78
- TA79
- TA90
- RV20
- TK50
- TK70
- TS05 (When extended characteristics are set)
- TU80
- TU81
- TU81E

The other tape drives that you can use are classified as start/stop mode tape drives. The start/stop mode tape drives are:

- TE16
- TS05 (When extended characteristics are not set)
- TS11
- TU77
- TU78
- TU79

If you have a start/stop mode tape drive, use the direct backup method described in Section 4.2.5.2.

The other hardware consideration concerns the disk drives in your system configuration. To perform a remote backup using a staging area as discussed in Section 4.1.3, you must ensure that you have a partition on the master system disk that is large enough to receive all of the backup data. You must know the partition sizes of the disks connected to your system and how much free space they contain. The free disk space on the master system disk must be at least 12 percent greater than the amount of slave system backup data. This is because the software that performs the network backup adds an additional 12 percent of overhead to the data being transferred.

4.1.2.2 System Setup Considerations – There are two system files that you must set up on both the master and slave systems before you can perform a network backup. These system files are:

- The `/etc/hosts` file
- The `/.rhosts` file in the root directory

The `/etc/hosts` file on the master system must specify all of the slave systems on which you intend to perform remote backups. As described in the *ULTRIX Guide to Networking*, each entry in the master system `/etc/hosts` file must contain the

slave system's Internet address, the official host name, and any aliases for that system. Additionally, the `/etc/hosts` file on the slave systems must contain an entry for the master system. Refer to the *ULTRIX Guide to Networking* and to `hosts(5)` in the *ULTRIX Reference Pages* for more information on the format of this file.

The `/.rhosts` file in the `root` directory on the master system must list the slave systems on which you intend to perform remote backups. The `/.rhosts` files on the slave systems must specify the official host name of the master system. Note that the `/.rhosts` file on the master system specifies access to the `root` directory of all the slave systems, while the `/.rhosts` file on the slave system specifies access to the `root` directory of the master system. This enables the master system to perform superuser activities on the slave systems during the remote backup process. If you try to perform a network backup without this information in the `/.rhosts` files, the system responds with the message:

```
Permission denied
```

If either of these two files is set up incorrectly, you will be unable to perform a slave system backup using the remote `opser` capability described in Section 4.2.

4.1.3 Staging Area or Direct Backup Determination

The remote `opser` utility described in Section 4.2 provides you with two methods of performing a network backup:

- Staging area backup
- Direct backup

A staging area is designated space in a file system on the master system that has enough free disk space to receive the slave system's backup data. The staging area backup method requires you to back up the slave system's data to the master system's disk and then to transfer the data to a tape device. Transferring the backup data to tape from the master system's disk enables you to take advantage of the streaming mode capability of the tape drive.

In the streaming mode, a tape drive continuously writes data to tape and does not stop tape movement until all of the data has been written. To support the streaming mode, data must be transferred at a rate at least equal to the tape's write transfer rate. Yet, because data cannot be transmitted over a network at this rate, it must first be transmitted to a staging area on the master system's disk drive and then copied to the tape drive. Transferring the staging area file to a tape drive in this way takes advantage of this type of drive's streaming mode capability. The disadvantage is that the file restoration process has the added step of having to transfer the dump image from the tape back to the staging area on the disk.

The direct backup method allows you to back up directly to a tape or disk drive. Use the direct backup method if you have a start/stop mode tape drive. The direct backup method has a simpler file restoration process, because you can restore directly from the tape or disk without having to transfer the dump image to a staging area.

4.2 Performing Remote Backups

Remote backups are performed by using the `opser` utility and selecting the `n` (network) option from the `opser` menu.

To run the `opser` utility, log in to the operator account (operator login name). When you log in to the operator account, the system automatically invokes the `opser` utility in place of the shell. Once running, the `opser` utility prints a utility sign-on message, two informational messages, and a command options prompt:

```
ULTRIX-32 Operator Services
```

```
Line editing: delete - erase one character, ^U - kill entire line
```

```
For help, type h followed by a return
```

```
opr>
```

The `opr>` prompt informs you that the `opser` utility is running in place of the shell and is ready to accept `opser` command options.

To use an `opser` option, type the appropriate option letter or name and press the RETURN key. For example, to display on-line help about all `opser` options, press the RETURN key without a command option letter, or type `h` and press the RETURN key:

```
opr> h
```

```
() - may use first letter in place of full name  
Valid commands for Local Opser are:
```

```
!sh                - shell escape (execute ULTRIX-32 commands)  
                    (Type control d to return from shell)  
(u)sers            - show logged in users  
(s)hutdown         - stop time-sharing  
(d)ismount         - unmount file systems  
(f)sck             - file system checks  
(r)estart          - restart time-sharing  
(h)elp            - print this help message  
(b)ackup           - file system backup  
halt              - halt processor  
(n)etwork [Slave] - initiate Remote Opser  
(q)uit            - exit from opser
```

```
opr>
```

To end an `opser` session and return to a login prompt, type `q` and press the RETURN key.

Except for the `n` (network) option, all of the local `opser` menu selections are discussed in Chapter 2. Refer to that chapter for information on other menu selections.

The remainder of this section discusses the functions and capabilities of the remote `opser` network option.

To enable remote `opser`, enter `n` at the `opr>` prompt. You must also know the name of the slave system on which you want the remote `opser` running. After you enter `n`, the `opser` utility prompts you to enter the name of the slave system; however, it only allows you to enter slave system names that are stored in the master system's `/etc/hosts` file.

The following example shows how `opser` responds when you enter an invalid slave system name.


```
opr> n
Enter Slave System name: proton
Slave system name not found in /etc/hosts file
```

The following example shows how `opser` responds when you enter a valid slave system name.

```
opr> n
Enter Slave System name: proton
Network: slave = proton.
alvin   tty00   Jun 15 09:24  (nucleon)
simon   tty01   Jun 15 08:30
teddy   tty02   Jun 15 08:33
Shutdown at 09:27  (in 3 minutes)
proton_opr>
```

This example shows that after you enter a valid slave system name, the `opser` utility displays the name of the slave system followed by a list of users currently logged in to that system. The `opr>` prompt becomes `proton_opr>`, which tells you that you are now running `opser` on the slave system named `proton`. You can also enter the slave system name and the `n` on the `opser` command line and achieve the same results. For example:

```
opr> n proton
Network: slave = proton.
alvin   tty00   Jun 15 09:24  (nucleon)
simon   tty01   Jun 15 08:30
teddy   tty02   Jun 15 08:33
Shutdown at 09:27  (in 3 minutes)
proton_opr>
```

Regardless of how you invoke the remote `opser`, it performs a shutdown of the slave system in three minutes; all users on the slave system must be logged out. When the shutdown completes, the slave system is in single-user mode and connected to the network. The slave system is then able to receive `opser` commands across the network from the master system.

If you press the RETURN key, or enter an `h` for help at the `proton_opr>` prompt, the `opser` utility displays the remote `opser` menu. For example:

```
proton_opr> h

() - may use first letter in place of full name
Valid commands for Remote Opser are:

!sh      - Slave System shell escape (execute ULTRIX-32 commands)
lsh      - Master System shell escape (execute ULTRIX-32 commands)
          (Type control d to return from shell)
(d)ismount - unmount file systems
(f)sck    - file system checks
(r)estart - restart time-sharing
(h)elp    - print this help message
(b)ackup  - file system backup
halt     - halt processor
(q)uit    - exit Remote Opser (do not restart Slave System)

proton_opr>
```

The following sections explain how each of these `opser` options work. Read these sections before trying to use them.

4.2.1 Escaping to the Slave Shell

To escape to the slave shell, enter `!sh` at the slave system `opr>` prompt. The `opser` utility prompts you to enter a password. You must enter the root password for the slave system. For example:

```
proton_opr>!sh
Password:
type ^D to return to opser
#
```

After you enter the slave's root password, the system places you in the slave's shell and displays the shell prompt. You can now execute any ULTRIX command, excluding editors. This option is useful for doing manual file system checks with the `fsck` command, displaying mounted file systems with the `df` command, or unmounting file systems with the `umount` command.

To return to the remote `opser`, type CTRL/D at the shell prompt.

4.2.2 Escaping to the Master Shell

To escape to the master shell, type `!sh` at the slave system `opr>` prompt. The `opser` utility prompts you to enter a password. You must type the root password for the master system. For example:

```
proton_opr>!sh
Password:
type ^D to return to opser
#
```

After you enter the master's root password, the system places you in the master's shell and displays the shell prompt. You can now execute any ULTRIX command. This option is useful for using the `talk` command to inform users that you intend to shut down the system. It is also useful for ensuring that the staging area that you are going to use is large enough to perform a staging area backup with the `df` command.

This option is also used for invoking script files before and after performing a staging area backup. As described in Section 4.2.5.1, script files can be used to clear the staging area of unnecessary files or to transfer staging area files to tape.

To return to the remote `opser`, type CTRL/D at the shell prompt.

4.2.3 Unmounting the Slave's File Systems

To unmount the file systems on the slave system, type `d` at the slave system `opr>` prompt. For example:

```
proton_opr> d
```

When you use the `d` option, the `opser` utility invokes the `umount` command to unmount file systems listed in the slave system's `/etc/fstab` file. The `umount` command is described in `mount(8)` of the *ULTRIX Reference Pages*.

To ensure that you have unmounted all of the file systems, use the `!sh` option as described in Section 4.2.1 and enter the `mount` command. Without options, the `mount` command displays any file systems that are currently mounted. If any file systems are still mounted, unmount them by issuing the appropriate `umount` command.

The following example shows the sequence of options and commands that you would enter, assuming that the `/usr/staff3` file system is still mounted after you enter

the d option:

```
proton_opr> d
proton_opr> !sh
Password:
type ^D to return to opser
# /etc/mount
/dev/rz0a on / type ufs
/dev/rz1h on /usr/staff3 type ufs
# /etc/umount /dev/rz1h
# <CTRL/D>
proton_opr>
```

Note

When you unmount the /usr file system, some ULTRIX commands may be unavailable.

The sequence of commands shows the shell escape, !sh, which causes the system to prompt you for the slave system root password, after which it displays the # shell prompt. Next, the sequence shows the system response to the mount command. The file system /usr/staff3 is unmounted by unmounting /dev/rz1h. This is the device, rz1, and partition, h, on which the /usr/staff3 file system is mounted. Lastly, the sequence shows the how typing CTRL/D returns to the proton_opr> prompt.

4.2.4 File System Consistency Check on the Slave

Before you back up any file systems, you should check them for consistency. To check file systems for consistency, enter f at the slave system opr> prompt. The opser utility invokes the fsck command at the slave system to check the file systems listed in the slave's /etc/fstab file. If the fsck command finds any file system inconsistencies, it displays a message informing you that you must run fsck manually.

To run fsck manually, escape to the slave shell and issue the appropriate fsck command.

If fsck fails on the slave's root file system, you must:

1. Quit network opser by typing q at the slave system opr> prompt. As described in Section 4.2.8, the slave system is shutdown and the system is placed in single-user mode.
2. Run fsck on the slave system. Use the following format:
/etc/fsck device
3. On the slave system, halt the processor so that you can return to the console subsystem. This allows you to reboot the system and ensures that the system buffer cache does not overwrite the fsck corrections. The hardware documentation for your processor describes how to return to the console subsystem.
4. You must then reboot the slave system to get the corrected root file system into memory.
5. Reestablish network opser by typing n at the local opser menu at the master system.

Caution

Whenever `fsck` corrects root file system inconsistencies, either through the `f` menu option or manual `fsck`, you must halt and reboot the slave system. See the hardware documentation for your processor for information on how to halt your processor.

For further details, see the information on maintaining file system consistency in the *ULTRIX Guide to System Crash Recovery*.

Example 4-1 shows a sequence of commands with a file system inconsistency encountered on the `/dev/rz0h` file system.

Example 4-1: Performing a File System Consistency Check

```
proton_opr> d

proton_opr> f

/dev/rz0a: 489 files, 6050 used, 631 free (93 frags, 160 blocks)
/dev/rz0d: 306 files, 43542 used, 51325 free (122 frags, 327 blocks)

/dev/rz0h: UNEXPECTED INCONSISTENCY; RUN fsck MANUALLY.
/dev/rz0f: 155 files, 41487 used, 147657 free (98 frags, 126 blocks)

proton_opr> !sh

Password:
type ^D to return to opser

# fsck /dev/rz0h
** Last mounted on /usr/users
** Phase 1 - Check Blocks and Sizes
** Phase 2 - Check Pathnames
** Phase 3 - Check Connectivity
** Phase 4 - Check Reference Counts
** Phase 5 - Check Cyl groups
/dev/rz0h: 3427 files, 95983 used, 14774 free (184 frags, 3362 blocks)
# /etc/mount -a
# df
Filesystem      total    kbytes  kbytes  percent
   node         kbytes  used    free   used   Mounted on
/dev/rz0a        7423    6050    631    91%    /
/dev/rz0d     105407  43542  51325    46%   /usr
/dev/rz0h     123063  95983  14774    87%   /usr/users
/dev/rz0f     210159  41487 147657    22%   /usr/server
# /etc/umount -a
# <CTRL/D>
proton_opr>
```

Example 4-1 shows that the `fsck` command was executed on the raw device, `/dev/rz0h`, after receiving the following message:

```
UNEXPECTED INCONSISTENCY; RUN fsck MANUALLY.
```

The `mount` command was then invoked with the `-a` option to mount all the file systems in the `/etc/fstab` file so that the `df` command could be invoked. The `df` command was issued to display and confirm the contents of the file systems. The `umount` command with a `-a` option was issued to dismount the file systems so that a network backup of the file systems could be performed without having to use the remote `opser` dismount option.

Refer to `fsck(8)`, `mount(8)`, and `df(8)` in the *ULTRIX Reference Pages* for more information.

4.2.5 Backing Up the Slave's File Systems

There are two methods that you can use to perform a remote backup of file systems. One is the staging area backup method and the other is the direct backup method. You will usually use the staging area backup method if you have a streaming mode tape drive. Use the direct backup method if you have a start/stop mode tape drive.

Note

You must unmount and check all file systems prior to backing them up. Use the `opser d` option to unmount the file systems; use the `opser f` option to check them for consistency.

The following two sections explain how to use the two backup methods. The sections contain complete examples of how the `opser` utility responds when using either method.

4.2.5.1 Staging Area Backup Method – The staging area backup method requires you to back up the slave system data to a designated file system on the master system's disk, and then to perform a tape transfer of the backed-up data.

Before performing a staging area backup, you must know:

- The backup level, which is a number from 0 to 9. Use the number 9 for a daily backup, the number 5 for a weekly backup, and the number 0 for a monthly or full backup.
- The file system that you will be using for the staging area on the master system, for example, `/staging`.
- The name of the file system that you want to dump to the staging area, for example, `/`.
- The amount of slave system data that you want to dump to the master system's staging area and the amount of free disk space in the staging area. The amount of free disk space must be at least 12 percent greater than the amount of slave system data that you want to dump.

When the backup is complete, you must use the `tar` command to transfer the staging area file to tape, or the `mdtar` command to transfer the staging area file to another disk. In either case, you should remove the staging area file from the staging area once you have transferred it to a removable media.

To use either `tar` or `mdtar` without exiting `opser`, you must escape to the master's shell using the `lsh` option as described in Section 4.2.2. If an operator is performing the staging area backup and does not have access to the superuser password, you must create some script files to perform the tape or disk transfer and to remove the staging area file from the staging area. The script files must be located in the `/opr` directory and have `root` ownership.

For example, to perform a tape transfer of the staging area file, a script file named `saveit` would have a one-line entry:

```
tar -cv /staging/*
```

Similarly, to remove the staging area file from the /staging directory, a script file named `removeit` would have a one-line entry:

```
rm -rf /staging/*
```

Having script files available to operators who do not have access to the superuser password keeps them from having to exit the `opser` utility to perform the tape or disk transfers or to remove files from the staging area.

Example 4-2 shows a backup sequence of the root (/) file system from the slave system `proton` to the master system named `nucleon`. The sequence shows that before performing the backup, the `removeit` script file was invoked to clear the /staging directory of any files. The example also shows that the root file system was transferred to the master's /staging directory. When the name of the file system is entered, the `opser` utility determines if there is enough disk space in the specified staging area to back up the slave file system. If there is enough room, the backup process begins. If there is not enough room, the system sends a beep signal to your terminal and displays the message:

```
Couldn't allocate ##### bytes.  
proton_opr>
```

The number signs (#) represent the number of bytes required to perform the backup.

When the `opser` utility has determined that there is enough disk space in the staging area, it starts the dump sequence. The `opser` utility assigns a file name to the data being transferred to the staging area. The naming convention that it uses for the file created on the master system is:

slave system name.filesystem name.time stamp

The `opser` utility also changes any slashes (/) in the file system name to tildes (~). In Example 4-2, the file name is:

```
proton.~.MON-Jun-24-10:32:26-1989
```

The dump image file is located in the master system `nucleon`'s /staging directory. In the example, the staging area file name breaks down as the root file system of the slave system `proton`, which was backed up on Monday, June 24, 1989 at 10:32 am. The tilde sign (~) equals the root (/) file system.

Example 4-2: Staging Area Backup Method Display

```
proton_opr> lsh removeit  
proton_opr> b  
Enter backup level (0 - 9): 9  
Use staging file or go directly to device(s = staging/d = direct)? s  
Enter directory to place staging file: /staging  
Enter name of filesystem to dump: /
```

```
Sizing file system to determine number of bytes to pre-allocate.
```

```
DUMP: Estimated 68608 bytes output to file named  
nucleon:/staging/proton.~.Mon-Jun-24-10:32:26-1989
```

```
Allocating 68608 bytes.
```

```
DUMP: Date of this level 9 dump: Mon Jun 24 10:32:23 1989  
DUMP: Date of last level 0 dump: Wed Jun 12 09:19:58 1989  
DUMP: Dumping /dev/rra0a (/) to  
/staging/proton.~.Mon-Jun-24-10:32:26-1989 on host nucleon  
DUMP: Mapping (Pass I) [regular files]  
DUMP: Mapping (Pass II) [directories]
```

Example 4-2: (continued)

```
DUMP: Estimated 68608 bytes output to file named
      /staging/proton.~.Mon-Jun-24-10:32:26-1989
```

```
DUMP: Dumping (Pass III) [directories]
DUMP: Dumping (Pass IV) [regular files]
```

```
DUMP: 58368 bytes were dumped to file
      /staging/proton.~.Mon-Jun-24-10:32:26-1989
```

```
DUMP: level 9 dump on Mon Jun 24 10:32:23 1989
DUMP: Dump is done
```

```
proton_opr>!sh saveit
```

Example 4-2 shows that when the dump is complete, the file named `proton.~.Mon-Jun-24-10:32:26-1989` is transferred to tape by using the script file `saveit`.

4.2.5.2 Direct Backup Method – The direct backup method enables you to back up a slave's file systems directly to a master system's tape drive or disk drive if your system configuration has the capacity to designate a disk drive for the purpose of backing up the slave systems.

Whether you are backing up to a tape or a disk drive, you must know the following information before you can use the direct backup method:

- The backup level number, which is a number from 0 to 9. Use the number 9 for a daily backup, the number 5 for a weekly backup, and the number 0 for a monthly or full backup.
- The device name where you want the slave system's backup data to go, for example, `/dev/rmt0h`.
- The name of the slave's file system that you want to dump.

You must also ensure that you have enough of the chosen media (tapes, disks, or diskettes) on hand when the backup process begins. Depending on the size of the backup, you may have multiple backup volumes.

4.2.6 Returning to the Local opser

To return to the local `opser` main menu, enter `r` at the slave system `opr>` prompt. For example:

```
proton_opr> r
```

This causes the remote `opser` running on the slave system to restart multiuser mode, disable the network connection to the slave system, and return you to the local `opser` that you initiated at the master system.

Caution

Do not use the `q` command to end a remote `opser` session. The `q` command stops remote execution of the `opser` utility and should only be used in the event of an emergency by someone with superuser privileges. Issuing the `r` command is the only way to end a remote `opser` session properly.

4.2.7 Halting the Slave Processor

To halt the slave system processor, enter `halt` at the slave system `opr>` prompt. This command causes the remote `opser` to synchronize the slave system disks (which means that in-memory file system data is written to the disks), then halt the slave system processor.

When you enter `halt`, the remote `opser` checks to make sure that you want to halt the slave system processor:

```
proton_opr>halt
This will leave the Slave System halted. You must then manually reboot
the Slave System.
Do you really want to do this?(Enter y/n)y
Leaving network mode.
opr>
```

When you enter a `y`, the remote `opser` utility exits remote `opser` execution and leaves the slave system halted. Consequently, someone at the slave system has to reboot it manually. The master system then returns to local execution of the `opser` utility, as indicated by the `opr>` prompt, and the network connection that was established automatically disconnects. If you enter an `n`, the remote `opser` utility returns the slave system `opr>` prompt.

4.2.8 Stopping Remote Execution of opser

In an emergency, it may be necessary to stop running the `opser` utility at the slave system. To stop remote execution of `opser`, enter `q` at the slave system `opr>` prompt. For example:

```
proton_opr> q
This will leave the Slave System shutdown in single user mode.
Do you really want to do this?(Enter y/n)y
Leaving network mode.
opr>
```

The preceding example shows that, when you enter `q`, the remote `opser` checks to make sure that you want to quit the remote `opser` session. When you enter a `y`, the remote `opser` utility exits remote `opser` execution and leaves the slave system in single-user mode. Consequently, someone at the slave system has to restart multiuser mode. The master system then returns to local execution of the `opser` utility, as indicated by the `opr>` prompt, and the network connection that was established automatically disconnects. If you enter an `n`, the remote `opser` utility returns the slave system `opr>` prompt.

4.2.9 Exceptional Condition Handling

You can abort either a remote `opser` process (such as `rdump`) on the slave or a local `opser` process (such as `dump`) on the master. To abort any processes running under the local `opser` on the master system, type `~ CTRL/C`. Typing `~ CTRL/C` causes the `opser` utility to return the `opr>` prompt. Type `CTRL/C` to abort any process running under the remote `opser` on the slave system.

You can restore files, directories, and file systems over the network with the `rrestore` program. This chapter describes how to:

- Restore files, directories, or file systems on different media using the `rrestore` program
- Restore the `root` and `/usr` file systems after a catastrophic event has occurred

5.1 Using the `rrestore` Program

You use the `rrestore` program to perform a remote file restoration, either interactively or noninteractively. Subsequent sections of this chapter explain both methods, with examples of how to perform each.

5.1.1 Network Considerations

Although the `rrestore` program is similar in function to the `restore` program, certain rules apply when working across a network:

- You must make a distinction between the master and slave systems, as described in Chapter 4. Typically, the dump media is located on the host or master system. You invoke the `rrestore` program from the slave system and specify the host name where the dump media resides.
- You do not have to be the superuser to restore files remotely. However, to restore a complete file system properly, you must be logged in as the superuser.
- It is important to remember where the file system that you want to restore was originally backed up.
- If the file system was backed up to a staging area file and is still intact, then you must specify the staging area file name to the `rrestore` program.
- If the staging area file was moved to tape using the `tar` command, then you must first use `tar` to move the staging area file from the tape to the staging area and then remote restore the file system from the staging area file. See Chapter 4 for a description of the staging area.
- Whatever procedure that you used to back up a file system must be repeated in reverse order.
- You can only specify a device to the `rrestore` command when the files that you want to restore were originally backed up using the direct backup method. See Chapter 4 for a description of backup methods.

5.2 Preparing for an Interactive Restore

To prepare for a remote restore operation, you must do the following:

- Log in as yourself or as superuser to `root`.
- Mount the first dump tape on the appropriate device.
- Transfer the backed up file from tape to a staging area using the `tar` command.

The following sections describe each of these tasks.

5.2.1 Logging In

Some `rrestore` functions require superuser privileges. For example, if you intend to perform an interactive file restoration of files that you do not own, you must be superuser. Similarly, if you intend to change file and directory attributes when restoring, you must be superuser. See `rrestore(8)` in the *ULTRIX Reference Pages* for details on the command options and required permissions.

5.2.2 Mounting the Dump Media

Before invoking the `rrestore` program, ensure that the backup media containing the first dump tape is mounted on the appropriate device. This is normally the default tape device, `/dev/rmt0h`, but it could be any device or file system in the form of an on-line backup file. The first tape is critical, because it contains directory and inode information about the contents of the dump image.

5.2.3 Transferring Files to the Staging Area

If the remote backup was to a staging area file and then was transferred to tape using the `tar` command, you must first ensure that the file is transferred back to the staging area from the tape by using the `tar` command with the `-x` option. For example, if the file name is `proton.~.Mon-Jun-24-10:32:26-1989`, you would type the following:

```
# cd /staging
# tar -x proton.~.Mon-Jun-24-10:32:26-1989
```

The `proton.~.Mon-Jun-24-10:32:26-1989` file name is the staging area file name assigned by the `rdump` command, as described in Chapter 4. When the staging area file has been transferred from tape to the staging area on the master system, you can invoke the `rrestore` command.

5.3 Restoring Files Interactively

Once you are logged in and the dump media is mounted on the appropriate device, use the following command format:

```
/etc/rrestore options host:device
```

The `-i` option specifies interactive mode. The `-f` option followed by the `host:device` string specifies the host and host device from which to obtain the data. For example, to restore files that were backed up to the master system `nucleon` using the direct backup method, you would enter:

```
# /etc/rrestore -if nucleon:/dev/rmt0h
```

To restore a file to the staging area from tape with the `tar` command, enter a command line similar to the following:

```
# /etc/rrestore -if nucleon:/staging/proton.~.Mon-Jun-24-10:32:26-1989
```

5.3.1 Using Interactive Mode Commands

When you enter the `/etc/rrestore` command with the `-i` option, the `rrestore >` prompt appears on your screen. The prompt signals that the program is waiting for a command.

5.3.1.1 Listing the Commands – To get a listing of the available commands, enter an `h` or a `?` at the `rrestore >` prompt. For example:

```
rrestore >h
Available commands are:

ls [arg] - list directory
cd arg - change directory
pwd - print current directory
add [arg] - add 'arg' to list of files to be extracted
delete [arg] - delete 'arg' from list of files to be extracted
extract - extract requested files
quit - immediately exit program
verbose - toggle verbose flag (useful with "ls")
help or '?' - print this list
If no 'arg' is supplied, the current directory is used
rrestore >
```

5.3.1.2 Listing the Pathname of the restore Working Directory – To list the pathname of the working directory within the `rrestore` environment, type `pwd` at the `rrestore >` prompt. For example:

```
rrestore >pwd
/lib
rrestore >
```

5.3.1.3 Getting a File and Directory Listing – To get a listing of the files and directories that can be restored at the current directory level, type `ls` at the `rrestore >` prompt. For example:

```
rrestore >ls
adm/          hosts/       man          skel/
bin/          include/    mdec/       spool/
dict         kits        msgs/       src/
rrestore >
```

To get a listing of the files and directories that can be restored at a different directory level, enter the pathname of that directory as an argument to the `ls` command.

5.3.1.4 Comparing Inode Numbers – Inode numbers increase with each tape. The largest inode numbers are on the last tape and the smallest numbers are on the first tape. By comparing the displayed inode numbers with the starting inode numbers recorded on a multivolume dump set, you can find the tape or disk that contains the file you want to restore. To see the inode numbers of the files contained within your listing, first use the `verbose` command to enter verbose mode, then use the `ls`

command to get your listing. The `rrestore` program displays the files with the inode numbers to the left of the dumped file names. For example:

```
rrestore >verbose
verbose mode on
rrestore >ls
  2 */          11971 hosts      14253 spool/
  2 */          2177 include/    4145  src/
11969 adm/      11427 mdec/       4253  staff
 1633 bin/      2722  msgs/       3266  sys/
rrestore >
```

In this example, the `hosts` file has an inode number of 11971 and the `spool` directory has an inode number of 14253.

Note

Sometimes the inodes for a given file span multiple tapes or disks. When this happens, you will need more than one tape or disk to restore the file.

After checking the inode numbers, you can turn off verbose mode. To do this, type `verbose` one more time. For example:

```
rrestore >verbose
verbose mode off
rrestore >
```

5.3.1.5 Changing Directories – Change directories in the restore environment with the `cd` command. To specify a particular directory, enter the `cd` command, with the directory pathname given as an argument, to the command. For example:

```
rrestore >cd /lib
rrestore >
```

5.3.2 Specifying What You Want Restored from the Dump Image

When working interactively with the `rrestore` program, you must first decide which files or file systems you want restored from the dump media. Use the `ls` command to get a file or directory listing of the dump media. Once you know what is on the media, you can specify what you want restored.

You specify what you want restored by creating and modifying an extraction list. Creating an extraction list involves working from the dump image and using the `rrestore` program's `add` command. Modifying an extraction list involves working from the dump image and using the `rrestore` program's `add` and `delete` commands.

5.3.2.1 Creating and Modifying the Extraction List – To create and modify an extraction list, you must be within the `rrestore` environment. When you have invoked the `rrestore` program, follow these steps:

1. Use the `pwd` command to determine your position within the dump media directory structure.
2. Use the `ls` command to identify which files and directories are on the dump image at the current level.

3. Decide which files or directories you want to extract from the dump image.
4. Use the `cd` command to move around the dump image and confirm the location of files and directories.
5. Position yourself in the dump media directory structure at a point that is one level above the desired item. For example, to restore `/usr/lib`, position yourself at `/usr`. If you attempt to identify `/usr/lib` while positioned within `/usr/lib`, the program issues an error message.
6. Create the extraction list by typing the `add` command and the name of each desired file or directory as an argument to the command.

For instance, if you are in the `/usr` directory and you want to extract the files `lpf` and `lpd` and the subdirectory `refer`, enter the `add` command followed by the file and directory names:

```
rrestore >add lpd lpf refer
rrestore >
```

Note

When you add a file to the extraction list, the program marks that file as one to restore. When you add a directory to your extraction list, the program marks that directory, its files, and all subordinate directories and files as ones to restore.

7. Use the `ls` command to see how the program has marked these files. For example:

```
rrestore >ls
diffh          libc_p.a          *lpd
dnet/          libcg.a          *lpf
how_pix        libpc_p.a          *refer/
rrestore >
```

Notice the asterisks (*) in front of the items that you added to your extraction list. When you view the directory listing, the asterisks point out what you put on the list. Similarly, when the `rrestore` program reads an asterisk during the extraction process, it transfers a copy of that file or directory from the dump image to your current working directory.

8. Use the `delete` command with the file name given as an argument to remove a particular file from the list. For example:

```
rrestore >delete lpf
rrestore >
```

9. Use the `ls` command to see what the program did in response to the `delete` command. For example:

```
rrestore >ls
diffh          libc_p.a          *lpd
dnet/          libcg.a          *lpf
how_pix        libpc_p.a          *refer/
rrestore >
```

The asterisk has been removed from the `lpf` file. This indicates that the program removed the `lpf` file from the list and will not extract it from the dump image.

10. Use the `extract` command to request a transfer of files from the extraction list to your system.

5.3.3 Restoring the Specified Files and Directories

The `rrestore` program's `extract` command initiates the actual restore. When you enter the `extract` command, the program transfers a copy of each marked item to the current working directory using the relative pathname found on the dump media.

Note

The current working directory is the one from which you invoked the `rrestore` program initially. Your position within the dump media directory structure when you enter the `extract` command has no influence on the final location of the restored files.

For example, assume that the file `/usr/lib/lpf` was dumped to tape with `/usr` as the mount point of the backup. In this case, `./lib/lpf` is the relative pathname on the dump media. If you invoked the `rrestore` program from `/usr` (in other words, with `/usr` as your current working directory), the program restores `/lib/lpf` as subordinate to `/usr` (`/usr/lib/lpf`). If you invoked the `rrestore` program from `/lib` (in other words, with `/lib` as your current working directory), the program restores `/lib/lpf` as subordinate to `/lib` (`/lib/lib/lpf`).

5.3.4 Entering the `extract` Command

To extract the files that you placed on your extraction list, enter the `extract` command at the `rrestore >` prompt. For example:

```
rrestore >extract
```

In response, the `rrestore` program displays several statements and prompts you for some answers. For example:

```
You have not read any tapes yet.  
Unless you know which volume your file(s) are on you  
should start with the last volume and work forward  
towards the first.  
Specify next volume #:  
set owner/mode for '.'? [yn]
```

5.3.4.1 Responding to the 'Specify next volume #' Prompt – The first prompt, Specify next volume #:, assumes that you are working from a multivolume backup set.

- If you are working from a multivolume set, and you know that the files you want extracted start on a certain volume number, then enter that volume number. If you do not know which volume contains the files that you want extracted, enter the number of the last volume of the set and work your way towards the first. For example, for a three-volume backup set, enter 3 at the

initial prompt. The program returns a second prompt to which you respond by entering 2, and so on.

- If you are working from a single-volume backup tape, enter 1 in response to the initial prompt.

In response to your entry, the `rrestore` program searches the dump image for the items listed on the extraction list. When it finds the individual files and directories, it restores them.

5.3.4.2 Responding to the 'Set owner/mode for '.' Prompt – The second prompt, Set owner/mode for '.', lets the superuser change the attributes of the current directory (shown as '.' within the prompt) to the attributes found for that entry on the dump image. The current directory's attributes include access permissions, ownership, and the accessed and update times.

- If you want to change the attributes, type `y`.

The `rrestore` program responds by changing the current directory's attributes to those found on the dump image for the corresponding directory. It then redisplay the `rrestore >` prompt.

If you are not the superuser and you type `y` in response to the prompt, the program responds with an error message and does not change the attributes.

- If you do not want to change the attributes or if you do not have superuser privileges, type `n`.

The `rrestore` program leaves the attributes of the current directory unchanged. It then redisplay the `rrestore >` prompt.

5.3.5 Exiting from the Interactive Restore

When you finish restoring files, type `quit` to get back to the system prompt:

```
rrestore >quit
#
```

5.4 Restoring Files Noninteractively

Noninteractive restoration of data is accomplished with a command line entry. The files or directories to be restored are entered as part of the command line. Before issuing the `rrestore` command in this way, ensure that you are in the correct directory and that you have the correct media mounted.

The following example shows you how to restore three files from the master system named `nucleon`. The example shows that the files named in the command line are restored from the dump tape mounted on device `/dev/rmt0h` on the master system `nucleon`.

```
#!/etc/rrestore -xf nucleon:/dev/rmt0h file1 file2 file3
#
```

When the restoration process is complete, the system redisplay the system prompt and the specified files are in the current working directory. Like the interactive restoration method, you can specify directories as well as files. The `rrestore` command will restore all of the files and subdirectories beneath the specified directory.

In addition to file names, you can specify file systems. For example:

```
#!/etc/rrestore -rf nucleon:/dev/rmt0h  
#
```

In this example, the `rrestore` command restores all files on the dump tape to the current working directory. This shows a file system restore, which is normally performed after a new file system is created and mounted, and a change directory is executed.

This appendix identifies and defines the mnemonics that are used to attach any hardware or software device to your system. The mnemonics are used by the `/dev/MAKEDEV` shell script to create the character or block special files that represent each of the devices. The mnemonics also appear in the system configuration file as described in *ULTRIX Guide to Configuration File Maintenance*.

Table A-1 lists the mnemonics in nine categories: generic, systems, consoles, disks, tapes, terminals, modems, printers, and others. The generic category lists the mnemonics of a general nature and includes memory, null, trace, and tty devices. The systems category lists the mnemonic for the DECstation 3100 system setup. The consoles category lists the system console devices that the ULTRIX operating system uses. The disks, tapes, terminals, modems, and printers categories identify the appropriate mnemonics for those devices. The others category lists the mnemonic for DECstation 3100 devices.

The description heading in Table A-1 identifies the corresponding device name. It does not define the mnemonic's use. For detailed information on the use of each mnemonic in relation to both the `MAKEDEV` script and the system configuration file, refer to the reference pages in Section 4 of the *ULTRIX Reference Pages*. If on-line reference pages are available, you can also use the `man` command. For instance, enter the following command at the system prompt to display the reference page for the Mass Storage Control Protocol (MSCP) disk controller driver:

```
% man ra
```

Where appropriate, the SYNTAX section of the reference page defines the device's syntax as it should appear in the `config` file. Refer to `/dev/MAKEDEV` for additional software device mnemonics that `MAKEDEV` uses. Refer to `MAKEDEV(8)` in the *ULTRIX Reference Pages* for a description of the `MAKEDEV` utility.

Table A-1 uses the convention of an asterisk (*) beside a mnemonic and a question mark (?) beside a device name to mean a variable number. The value of the variable number is dependent on the particular device.

Table A-1: Devices Supported by MAKEDEV

Category	Mnemonic	Description
Generic	boot*	Boot and std devices by cpu number; for example, boot750
	mvax*	All MicroVAX setups; for example, mvax2000
	vaxstation*	A VAXstation 2000 setup; for example, vaxstation2000
	std	Standard devices with all console subsystems:
	drum	Kernel drum device
	errlog	Error log device
	audit	Audit log device
	kUmem	Kernel Unibus/Q-bus virtual memory
	kmem	Virtual main memory
	mem	Physical memory
	null	A null device
	trace	A trace device
	tty	A character terminal device
	local	Customer-specific devices
Systems	DECstation	A DECstation 3100 setup
Consoles	console	System console interface
	crl	Console RL02 disk interface for VAX 86?0
	cs*	Console RX50 floppy interface for VAX 8??0
	ctu*	Console TU58 cassette interface for VAX 11/725/730/750
	cty*	Console extra serial line units for VAX 8??0
	cfl	Console RX01 floppy interface for 11/78?
	ttycp	Console line used as auxiliary terminal port
Disks	hp*	MASSBUS disk interface for RM?? drives and RP?? devices
	ra*	UNIBUS/Q-bus/BI/HSC/DSSI MSCP disk controller interface
	rb*	UNIBUS IDC RL02 disk controller interface for RB?? drives
	rd*	VAXstation 2000 and MicroVAX 2000 RD type drives
	rz	SCSI disks (RZ22/RZ23/RZ55/RRD40)
	rk*	UNIBUS RK?? disk controller interface
	rl*	UNIBUS/Q-bus RL?? disk controller interface
	rx*	VAXstation 2000 and MicroVAX 2000 RX type drives
Tapes	mu*	TU78 MASSBUS magtape interface
	tms*	UNIBUS/Q-bus/BI/HSC/DSSI TMSCP tape controller interface
	rv*	UNIBUS/Q-bus/BI TMSCP optical disk
	ts*	UNIBUS/Q-bus TS11/TS05/TU80 magtape interface
	tu*	TE16/TU45/TU77 MASSBUS magtape interface
	st*	VAXstation 2000 and MicroVAX 2000 TZK50 cartridge tape
	tz*	SCSI tapes (TZ30/TZK50)
Terminals	cx*	Q-bus cxa16
	cx*	Q-bus cxb16
	cx*	Q-bus cxt08
	dfa*	Q-bus DFA01 comm multiplexer
	dhq*	Q-bus DHQ11 comm multiplexer
	dhu*	UNIBUS DHU11 comm multiplexer

Table A-1: (continued)

Category	Mnemonic	Description
	dhv*	Q-bus DHV11 comm multiplexer
	dmb*	BI DMB32 comm multiplexer including dmbsp serial printer/plotter
	dhb*	BI DHB32 comm multiplexer
	dmf*	UNIBUS DMF32 comm multiplexer including dmfsp serial printer/plotter
	dmz*	UNIBUS DMZ32 comm multiplexer
	dz	UNIBUS DZ11 and DZ32 comm multiplexer
	sh*	MicroVAX 2000, 8 serial line expansion option
	ss*	VAXstation 2000 and MicroVAX 2000 basic 4 serial line unit
	dzq*	Q-bus DZQ11 comm multiplexer
	dzv*	Q-bus DZV11 comm multiplexer
	lta*	Sets of 16 network local area terminals (LAT)
	pty*	Sets of 16 network pseudoterminals
	qd*	Q-bus VCB02 (QDSS) graphics controller/console
	qv*	Q-bus VCB01 (QVSS) graphics controller/console
	sm*	VAXstation 2000 monochrome bitmap graphics/console
	sg*	VAXstation 2000 color bitmap graphics console
	lx	VAXstation 8000 color high-performance 3D graphics
	fg*	VAXstation 60 color bitmap graphics/console
Modems	dfa*	DFA01 integral modem communications device.
Printers	dmbsp*	BI DMB32 serial printer/plotter
	dmfsp*	UNIBUS DMF32 serial printer/plotter
	lp*	UNIBUS LP11 parallel line printer
	lpv*	Q-bus LP11 parallel line printer
Packet filter	pfilt	Packet filter devices; set of 64
Other	pm*	mono/color bitmap graphics/mouse/modem /printer/terminals for DECstation 3100

B

backup

- developing a strategy, 1-3
- full, 1-3
- incremental, 1-3
- selecting a method, 1-1

backup levels, 1-4

backup operation

- developing a strategy, 1-3
- local, 1-1
- performing a full backup, 1-3
- remote, 1-1
- selecting a method, 1-1
- using backup levels, 1-4

D

device mnemonics, A-1 to A-3

- reference list, A-2t to A-3t
- using with MAKEDEV, A-1
- using with man command, A-1

direct backup method

- See also* staging area backup method
- defined, 4-3
- description, 4-11
- prerequisites, 4-11

directory

- restoring dump image, 3-1

dump command

- backing up file systems, 2-1
- listing file systems, 2-1
- using, 1-1, 2-1 to 2-2

dump program

- See also* restore command

F

file

- restoring, 3-1 to 3-9
- restoring dump image, 3-1 to 3-9

file system

- See also* target file system
- backing up, 2-1 to 2-8
- checking restored, 3-10
- restoring, 3-9 to 3-10
- unmounting restored, 3-10

files

- transferring using mdtar, 2-2
- transferring using tar, 2-2

fsck command

- checking restored file system, 3-10
- root partition and, 3-10n

full backup, 1-3

H

hosts file

- network backup and, 4-2

I

incremental backup, 1-3

L

lrf command, 3-14

M

magnetic tape drive

- with start/stop mode, 4-1
- with streaming mode, 4-1

master system

- defined, 4-1
- tape drives and, 4-1

mdtar command

- restoring files, 3-13
- transferring files, 2-2
- using, 1-2, 2-2

N

newfs command

- creating file system, 3-9e
- using, 3-9

O

opser utility

- aborting, 4-12
- backing up local file systems, 2-6
- backing up remote file systems, 4-9
- checking file system consistency, 2-5
- checking remote file system consistency, 4-7, 4-8e
- displaying users, 2-4
- enabling remote opser, 4-4
- ending remote session, 4-11c
- escaping to local shell, 2-6
- escaping to master shell, 4-6
- escaping to slave shell, 4-6
- exiting local opser, 2-7
- halting slave system, 4-12
- halting the local processor, 2-7
- help menu, 2-4e, 4-4e
- local usage, 2-3 to 2-8
- network options help menu, 4-5e
- performing local backups, 2-3
- performing remote backups, 4-3 to 4-12
- restarting multiuser mode, 2-7
- shutting down multiuser mode, 2-4
- stopping remote execution, 4-12
- unmounting file systems, 2-5

opser utility (cont.)

- using, 1-2

R

rdump command

- using, 1-1

remote file

- restoring, 5-1 to 5-8
- restoring interactively, 5-2 to 5-3
- restoring noninteractively, 5-7

remote file system

- backing up, 4-1, 4-3 to 4-12
- backup prerequisites, 4-2

restore

- choosing a method, 1-2
- using mdtar, 1-3
- using tar, 1-3

restore command, 1-2

- extracting files, 3-6
- extraction list, 3-4
- restoring dump image, 3-1
- restoring file system, 3-10
- running interactively, 3-7
- running noninteractively, 3-7
- using, 3-1 to 3-9

restore operation

- selecting a method, 1-2
- using mdtar, 1-3
- using tar, 1-3

restore program

- command menu, 3-3e
- list command, 3-3e
- verbose command, 3-4e

restoresymtable file

- removing, 3-10

.rhosts file

- network backup and, 4-2

root file system

- fsck and, 4-7c
- restoring, 3-11 to 3-13

rrestore command, 1-2

- extracting files, 5-6
- extraction list, 5-4

restore command (cont.)

- options, 5-3
- running interactively, 5-2
- running noninteractively, 5-7
- using, 5-1

restore program

- command menu, 5-3e
- list command, 5-3e
- verbose command, 5-4e

S

slave system

- defined, 4-1

staging area

- defined, 4-3

staging area backup method

- See also* direct backup method
- defined, 4-3
- description, 4-9 to 4-11
- display, 4-10e
- prerequisites, 4-9

staging area file

- deleting, 4-9e
- transferring to tape, 4-9, 4-9e

streaming mode

- defined, 4-3

system

- backing up, 2-1 to 2-8

T

tar command

- restoring files, 3-13
- transferring files, 2-2
- using, 1-2, 2-2

target file system

- checking, 3-10
- creating, 3-9
- mounting, 3-10

U

/usr file system

- restoring, 3-11 to 3-13