7

Making Backups

Incremental Backups

Whether it's caused by system failure or accidental erasure, loss of stored data is a programmer's nightmare. Consequently, backups of files, programs, and disks are a normal part of existence. Backing up a hard disk is usually slow and tedious because the entire system is backed-up.

You can use incremental backups instead of full system backups. Incremental backups save only the files that have changed since the last backup. You must still perform a full system backup, but by using incremental backups you can perform them less often.

+ OS-9 provides two utilities that may be used with either tape or disk media to facilitate the use of incremental backups:

  - fsave
  - frestore

Certain terms must be defined to discuss incremental backups. A full system backup is referred to as a ***level 0 backup***. Consequent incremental backups are referenced by different level numbers. For example, a level 5 backup includes all files changed since the most recent backup with a level less than 5. While this sounds complex, it is actually quite easy to use and extremely helpful.

Two other terms need to be defined. A ***source device*** is the directory structure or file you are backing up. A ***target device*** is the tape or disk you are using to hold your backup information.

Making an Incremental Backup: The fsave Utility

The fsave utility performs an incremental backup of a directory structure to tape(s) or disk(s). The syntax for the fsave utility is:

```
fsave [<opts>] [<path>] [<opts>]
```

Typing fsave by itself on the command line makes a level 0 backup of the current directory onto a target device with the name /mt0.

+ **NOTE:** /mt0 is the default OS-9 device name for tape device just as /h0 is the default OS-9 device name for a hard disk.

/h0/sys/backup_date is a backup log file maintained by fsave. Each time you execute an fsave, the backup log is updated. The backup log keeps track of the name of the backup, the date it was created, and more importantly, the level of the backup. When you execute fsave, this backup log is examined to find the specified level of the current backup and the previous backups with the same name. Once the backup is finished, a new entry is made in the file indicating the date, name, level, etc. of the current backup.

During the discussion of the actual fsave procedure, references to fsave's options are made. The options are:

Option	Description
-?	Displays the use of fsave.
-b[=]<int>	Allocates <int>k buffer size to read files from the source disk.
-d[=]<dev>	Specifies the target device to store the backup. The default is /mt0.
-e	Does not echo file pathlists as they are saved to the target device.
-f[=]<path>	Saves to the file specified by <path>.
-g[=]<int>	Specifies a backup of files owned by group number <int> only.
-j[=]<number>	Specifies the minimum system memory request.
-l[=]<int>	Specifies the level of the backup to be performed.
-m[=]<path>	Specifies the pathlist of the date backup log file to use. The default is /h0/sys/backup_dates.
-p	Turns off the mount volume prompt for the first volume.
-s	Displays the pathlists of all files needing to be saved and the size of the entire backup without actually executing the backup procedure.
-t[=]<dirpath>	Specifies the alternate location for the temporary index file.

Option	Description
-u[=]<int>	Specifies a backup of files owned by user number <int> only.
-v	Does not verify the disk volume when mounted.
-x[=]<int>	Pre-extends the temporary file. <int> is given in kilobytes.

The fsave Procedure

Upon starting an fsave procedure, fsave prompts you to mount the first volume to use. Volume in this case refers to the disk or tape used to store the backup:

```
fsave: please mount volume.
(press return when mounted).
```

If a disk is used as the backup medium, fsave verifies the disk and displays the following information:

```
verifying disk

Bytes held on this disk: 546816
Total data bytes left:   62431
Number of Disks needed:  1
```

NOTE: The numbers above are used only as an example.

If a tape is used as the backup medium, no preliminary information is displayed and the backup begins at this point.

As each file is saved to the backup device, its pathlist is echoed to the terminal. If this is a long backup, you may want to use the -e option to turn off the pathlist echoing.

If fsave receives an error when trying to backup a file, it displays the following message and continues the fsave operation:

```
error saving <file>, error - <error number>, its incomplete
```

NOTE: The most common error found when executing fsave is a record lock error. Record lock errors are caused when another user has the file in question open.

+ To prevent record lock errors, perform fsave operations only when no one else is using the system.

If the backup requires more than one volume, fsave prompts you to mount the next volume before continuing.

At the end of the backup, fsave prints the following information:

```
fsave: Saving the index structure

Logical backup name:
Date of backup:
Backup made by:
Data bytes written:
Number of files:
Number of volumes:
Index is on volume:
```

The index to the backup is saved on the last volume used.

fsave performs recursive backups for each pathlist if one or more directories are specified on the command line. You can specify a maximum of 32 directories on the command line.

The following options are provided:

-d	Specifies an alternate target device. The default device is /mt0.
-m	Specifies an alternative backup log file. The default pathlist is /h0/sys/backup_dates.
-l	Specifies different levels of backups. A higher level backup only saves files that have changed since the most recent backup with the next lower number. For example, a level 1 backup saves all files changed since the last level 0 backup.

! **WARNING:** When using disks for backup purposes, fsave does not use an RBF file structure to save the files on the target disk. It creates its own file structure. This makes the backup disk unusable for any purpose other than fsave and frestore without reformatting the disk. Any data stored on the disk before using fsave is destroyed by the backup.

Example fsave Commands

Typing fsave by itself on a command line specifies a level 0 backup of the current directory. This assumes the /mt0 device is used and that /h0/SYS/backup_dates is used as the backup log file for this backup.

The following command specifies a level 2 backup of the current directory using the /mt1 device. /h0/misc/my_dates is used as the backup log file:

```
$ fsave -l=2 -d=/mt1 -m=/h0/misc/my_dates
```

The following command specifies a level 0 backup of all files owned by user 0.0 in the CMDS directory, if CMDS is in your current directory:

```
$ fsave -pb=32 -g=0 -u=0 -d=/d2 CMDS
```

This backup uses /d2 as the target device and /h0/sys/backup_dates as the backup log file. The mount volume prompt is not generated for the first volume. A 32K buffer is used to read the files from the CMDS directory.

Restoring Incremental Backups: The frestore Utility

The frestore utility restores a directory structure from multiple volumes of tape or disk media. The syntax for the frestore utility is:

```
frestore [<opts>] [<path>] [<opts>]
```

Typing frestore by itself on the command line attempts to restore a directory structure from the /mt0 device to the current directory.

Specifying the pathlist of a directory on the command line causes the files to be restored in that directory. fsave creates the directory structure and an index of the directory structure.

If more than one tape/disk is involved in the fsave backup, each tape/disk is considered to be a different ***volume***. The volume count begins at one (1). When you begin an frestore operation, you must use the last volume of the backup first because it contains the index of the entire backup.

frestore first attempts to locate and read the index of the directory structure of the source device. frestore then begins an interactive session with you to determine which files and directories in the backup should be restored to the current directory.

During the discussion of the actual frestore procedure, references are made to frestore's options. The options are:

Option	Description
-?	Displays the use of frestore.
-a	Forces access permission for overwriting an existing file. You must be the owner of the file or a super user to use this option.
-b[=]<int>	Specifies the buffer size used to restore the files.
-c	Checks the validity of files without using the interactive shell.
-d[=]<path>	Specifies the source device. The default is /mt0.
-e	Displays the pathlists of all files in the index, as the index is read from the source device.
-f[=]<path>	Restores from a file.
-i	Displays the backup name, creation date, group.user number of the owner of the backup, volume number of the disk or tape, and whether the index is on the volume. This option will not cause any files to be restored. The information is displayed, and frestore is terminated.
-j[=]<int>	Sets the minimum system memory request.
-p	Suppresses the prompt for the first volume.

Option	Description
-q	Overwrites an already existing file when used with the -s option.
-s	Forces frestore to restore all files from the source device without an interactive shell.
-t[=]<dirpath>	Specifies an alternate location for the temporary index file.
-v	Displays the same information as the -i option, but does not check for the index. This option will not cause any files to be restored. The information is displayed and frestore is terminated.
-x[=]<int>	Pre-extends the temporary file. <int> is given in kilobytes.

The Interactive Restore Process

Once you call frestore, the following prompt is displayed:

```
frestore> mount the last volume
(press return when ready)
```

When you are ready, frestore attempts to read in the index and create the directory structure of the backup. It then displays the prompt:

```
frestore>
```

This prompt tells you that you are in the interactive shell. If the index is not on the mounted volume, frestore displays an error message and again prompts you to mount the last volume.

Once in the interactive shell, the frestore commands and options are displayed when you type a return at the prompt:

```
frestore> commands:
  add [<path>] [-g=<#> -u=<#> -r -a] -- marks files for restoration
  del [<path>] [-g=<#> -u=<#> -r -a] -- unmarks files for restoration
  dir [<dir names>] [-e] -- displays a directory or directories
  chd <path> -- changes directories within the restore file structure
  pwd -- gives the pathlist to current dir in the restore file structure
  cht <path> -- changes directories on target system
  rest [<path>] [-f -q] -- restores marked files in and below the current dir
  check [-f] -- checks validity if marked files in and below the current dir
  dump [<file>] -- dumps the contents of a file to stdout
  $   -- forks a shell
  quit -- quit frestore program
options:
  -g=<group#> -- only mark files with 'group#'
  -u=<user#> -- only mark files with 'user#'
  -r -- mark directories recursively
  -e -- display directory with extended format
  -f -- force restoration of already restored files
  -q -- overwrite already existing files without question
  -a -- force marking or unmarking of an already restored file or dir
   * -- matches any string of characters on 'add' or 'del' only
   ? -- matches any single character on 'add' or 'del' only
frestore>
```

The index from the source device sets up a restore file structure that parallels the usual OS-9 file/directory structure.

Use the dir and chd shell commands to display the restore file structure. For example:

```
frestore>dir
            Directory of .
DIR1       file1       file2       file3
```

All files to be backed up on to the source device appear in the restore file structure regardless of what volume they appear in. Information concerning the file structure is available using the -e option with the dir command:

```
    frestore>dir -e

              Directory of .

      Owner   Last modified  Attributes Volume Block Offset   Size    Name
       ------  -------------- ----------- ------ ----- ------  -----   ------
  1.23    89/08/22 16/14  ----r-wr       1     0      0  CF12    file1
  1.23    89/08/25 11/00  ----r-wr       1     2      0  A356    file2
  1.23    89/08/21 11/12  ----r-wr       1     4      0  45F0    file3
  1.23    89/08/24 10/57  d-ewrewr       0     5      0   120    DIR1
```

The interactive shell allows you to mark the files you want restored with the add command. You can mark groups of files using the options of the add command:

-g	Marks files by group number.
-u	Marks files by user number. You can mark all directories within a specified directory using the -r option.

+
- Files may be marked one at a time by specifying relative or complete pathlists within the restore file structure.
- Entire directories may be marked by specifying a pathlist of a directory.

Marking files does not restore them. It merely marks them as ***to be restored***. You can see this when you use the dir command. Each file added to the "to be restored" list is marked by a plus sign (+) by its filename.

For example, the following directory has file1 and file2 marked for restoration, but file3 is not marked. The directories DIR1 and DIR2 also have marked files:

```
frestore>add file1 file2 dir1/file5 dir1/file6 dir2/file7

frestore>dir
        Directory of .
+DIR1           +DIR2           +file1          +file2
file3

frestore>dir dir1
        Directory of DIR1
file4           +file5          +file6

frestore>dir dir2
        Directory of DIR2
+file7          file8
```

The del command can unmark files. Entire directories may be unmarked by specifying the directory's name on the command line. If the -r option is also used, all files and directories included in the specified directory are unmarked. For example:

```
frestore>del -r dir2

frestore>dir
        Directory of . 10:42:32
+DIR1             DIR2              +file1            +file2
file3

frestore>dir dir2
        Directory of DIR2
file7             file8
```

Once files are marked, the rest command may be used to restore the target device's current directory.

Files existing on the target system with the same name are overwritten without prompting if del -q is used. Otherwise, frestore displays the following prompt:

```
frestore> file1 already exists
      write over it or skip it (w/s)
```

The cht command allows you to change directories on the target device. This allows you to selectively restore files to specific directories.

After restoring files, you may continue marking and unmarking files. Files previously restored have a hyphen (-) displayed next to their names in the restore file structure:

```
frestore>dir
        Directory of . 10:42:32
-DIR1             DIR2              -file1            -file2
file3

frestore>dir dir1
            Directory of DIR1
file4             -file5            -file6
```

> + An asterisk (*) preceding the name of a file in a dir listing indicates an error occurred while backing up this file. This file is incomplete and should not be restored.

There are two methods of restoring files more than once. The first method uses the -a option with the add command. This forces the file(s) previously marked as restored to be marked as "to be restored." The second method requires the -f option to be used with the rest command. This forces any file previously marked as restored to be restored in the current directory.

The -s option forces frestore to restore all files/directories of the backup from the source device without the interactive shell.

Using the -d option allows you to specify a source device other than /mt0. For example, to restore all files/directories found on the /mt1 source device to the directory BACKUP without using the interactive shell, type:

```
$ frestore -d=/mt1 -s BACKUP
```

The -v option causes frestore to identify the name and volume number of the backup mounted on the source device. The date the backup was made and the group.user number of the person who made the backup is also displayed. This option does not restore any files. For example:

```
$ frestore -v

Backup: DOCUMENTATION
Made:   1/16/91 10:10
By:     0.0
Volume: 0
```

The -i option displays the above information and also indicates whether the index is on the volume. Both the -v and -i options terminate frestore after displaying the appropriate information. These options are useful when trying to locate the last volume of the backup if any mix-up has occurred.

The -e option echoes each file pathlist as the index is read off the source device.

Example Command Lines

To restore files/directories from the source device /mt0 to the current directory by way of an interactive shell, type:

```
$ frestore
```

The following example restores files/directories from the source device /d0 to the current directory using a 32K buffer to write the restored files. As each file is read from the index, the file's pathlist is echoed to the terminal.

```
$ frestore -eb=32 -d=/d0
```

Incremental Backup Strategies

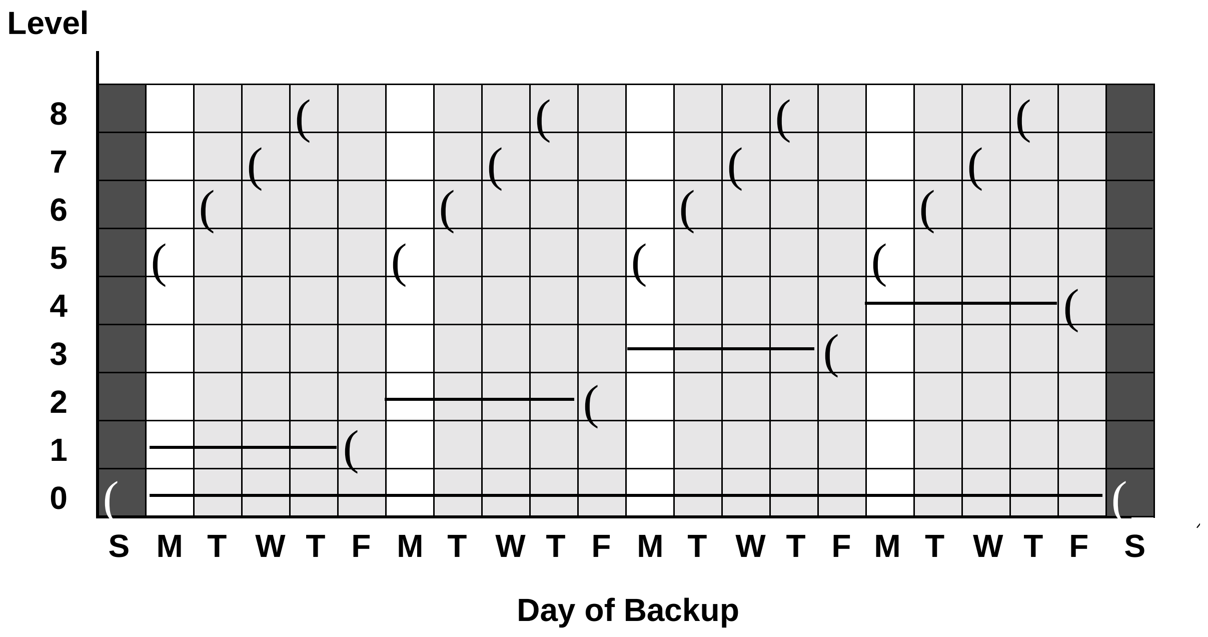
Many different strategies are available for those concerned with regularly scheduled backups. Most strategies are well documented in computer books and magazines. The following two strategies are offered as examples of methods that can be used.

The Small Daily Backup Strategy

This strategy requires making a level 0 backup once every four weeks. Level 1, level 2, level 3, and level 4 backups are made on the weeks following the level 0 backup. Between each major backup, four daily backups would be made: level 5, 6, 7, and 8. A recommended daily schedule is graphically presented below.

This strategy is ideal for small microcomputer systems backed up by floppy disks. Mounting disks is much easier and faster than tapes. Each daily backup can usually be kept on one disk to make warehousing simple. This strategy is perfect for small timely backups with little redundancy in the backups.

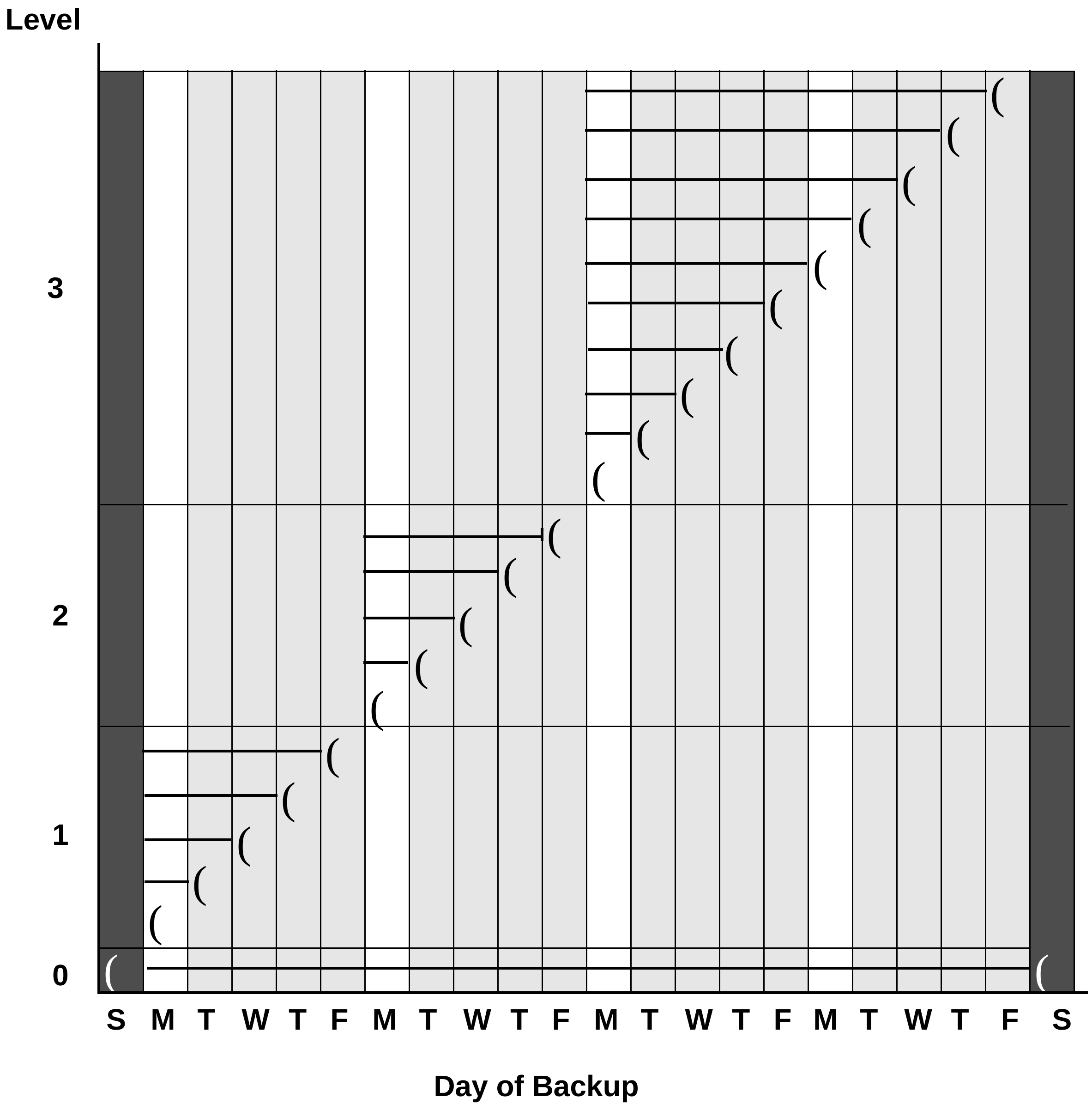
One major disadvantage of this scheme is the restore time necessary in case of a major system failure such as a hard disk being formatted, erased, or corrupted. Because of the lack of redundancy, more frestore operations are necessary to re-create the systems file structure. On large systems with tape backups, this is a major consideration.



Small Daily Backup Strategy

The Single Tape Backup Strategy

While most strategies rely on scheduled backup level changes, the ***single tape backup*** strategy depends on the size of the backup. The idea behind this strategy is to increase the level of the backup only when the backup cannot fit on a single tape. The only scheduled level backup is the level 0 backup. The level 0 backup occurs only when a higher level backup would not fit on a single tape or once a month, whichever occurs first. An example month's schedule is graphically presented below.



Single Tape Backup Strategy

This strategy is designed for tape backups of larger systems. Tapes are used efficiently because a question as to how many tapes are needed never arises. This strategy also cuts down on person hours, tape mounting, and storage space used for tapes. It allows for enough redundancy to make restoring a full system fairly painless.

Disadvantages, however, do exist. Each time you do a backup, you must determine the size of the back using `fsave -s`. As you near a full tape's worth of data, this takes an increasing amount of time.

Use of Tapes/Disks

Whatever strategy you use, you must make a decision concerning the number of tapes or disks to use. This decision must weigh the emphasis placed on redundancy, resources, person-hours, and storage. It must be offset with the possibility of tape or disk failure and system restoration.

In the first example strategy, you must make the daily backups on different volumes to overcome the lack of redundancy. You can use the four daily volumes week after week as daily backup volumes because of the lower level backups at the beginning of each week.

In the second example, theoretically, you could use the same tape for each day until a new level backup is reached. This insures no redundancy and minimal storage. It is also the most dangerous in case of tape failure. Using a number of alternating tapes for each level cuts down on storage and still allows a safety net in the case of tape failure. Using alternating level 0 tapes is another possibility.

The tape Utility

OS-9 provides a tape controller utility to facilitate setting up, reading, and rewinding tapes from the terminal. When using tape media to backup or restore your system, the tape utility is very practical. The syntax of the tape utility is:

```
tape {<opts>} [<dev>] {<opts>}
```

tape uses the default device /mt0 if you do not specify the tape device <dev> on the command line and you do not use the -z option.

tape has the following available options:

Options	Description
-?	Displays the use of tape.
-b[=<num>]	Skips a specified number of blocks. Default is one block. If <num> is negative, the tape skips backward.
-e=<num>	Erases a specified number of blocks of tape.
-f[=<num>]	Skips a specified number of tapemarks. Default is one tapemark. If <num> is negative, the tape skips backward.
-o	Puts tape off-line.
-r	Rewinds the tape.
-s	Determines the block size of the device.
-t	Retensions the tape.
-w[=<num>]	Writes a specified number of tapemarks. Default is one tapemark.
-z	Reads a list of device names from standard input. The default is /mt0.
-z=<file>	Reads a list of device names from <file>.

If you specify more than one option, tape executes each option function in a specific order. Therefore, it is possible to skip ahead a specified number of blocks, erase, and then rewind the tape all with the same command. The order of options executed is as follows:

¿ Gets device name(s) from the -z option.
¡ Skips the number of tapemarks specified by the -f option.
¬ Skips the number of blocks specified by the -b option.
Ð Writes a specified number of tapemarks.
ƒ Erases a specified number of blocks of tape.
Ý Rewinds the tape.
ý Puts the tape off-line.

For example, the following command skips four files on the /mt0 device, erases the next two blocks, rewinds the tape, and takes the tape off-line:

```
tape -e=2 -f=4 -ro
```

The next example determines the block size of the device:

```
tape -s
```

The next example retensions the tape, rewinds it, and then takes it off-line:

```
tape -rot
```

End of Chapter 7