

Backing Up and Restoring File Systems [6]

This section describes how to maintain file systems by creating backup copies of them regularly (also called *backing up* a file system). It also describes how to restore your file systems. *Backing up* a file or file system means to create another copy of it on different storage media (using `dump`); the copy could then be used to replace the original if the original had been damaged or destroyed. *Restoring* a file or file system means to overwrite the current disk file or file system (using `restore`) with the back-up copy.

Note: Backing up large file systems is a resource-consuming task. File saving procedures ideally should be performed in single-user mode with file systems unmounted; therefore, frequent backups mean less time available for user processing. You must adopt a file-backup schedule that is best for your site.

Backing up your file systems on a regular basis ensures users against the loss of time, effort, and valuable information if a file system is corrupted or disk crash occurs. New users (occasionally even experienced ones) may sometimes remove files by mistake. As the system administrator, you must develop and maintain adequate backup procedures.

The following utilities are available for partial file system backup tasks:

- Archiving and extracting files with tape (using `tar`)
- Copying file archives while maintaining status and path names (using `cpio`)

6.1 Related backup and restore documentation

The following documentation contains information covered in this section:

- *General UNICOS System Administration*, Cray Research publication SG-2301, section on file system planning
- *UNICOS Configuration Administrator's Guide*, Cray Research publication SG-2303
- *UNICOS File Formats and Special Files Reference Manual*, Cray Research publication SR-2014: `dump(5)` and `fstab(5)` man pages
- *UNICOS Administrator Commands Reference Manual*, Cray Research publication SR-2022: `dump(8)`, `rdump(8)`, `restore(8)`, and `rrestore(8)` man pages

- *Tape Subsystem Administration*, Cray Research publication SG-2307: tape-related information

6.2 Tape devices referenced in `/dev/tape`

To use the various UNICOS utilities to back up and restore files not using the `tpdaemon`, the tape devices are in the `/dev/tape` directory in which the `tpdaemon` addressable devices also reside. For more information on naming tape devices, see the *Tape Subsystem Administration*, Cray Research publication SG-2307.

6.3 Backup and restore utilities

The following utilities have somewhat different capabilities to back up or restore your file systems. This subsection also recommends when to use each of the utilities. (This section of the guide describes using the `dump` and `restore` utilities for standard file system maintenance.) The `dump` and `restore` utilities are excellent utilities to use because they function based on the concepts of file systems.

6.3.1 `dump` and `restore` utilities

The `dump` and `restore` utilities are recommended for performing file system backups and restores because you can examine the contents of a tape of dumped files without actually reading the entire tape.

The `dump` utility writes a header, which lists the contents of the dump tape on a tape volume. The `restore` utility can read this tape header. The `restore` utility has a simple interactive option that allows an administrator to select some or all of the tape contents for restoration by marking desired files listed in the header.

6.3.2 `rdump` and `rrestore` utilities

The `rdump` and `rrestore` utilities are used to perform the same tasks as the `dump` and `restore` commands across a TCP/IP network.

6.3.3 dd utility

The `dd` utility is used for copying data directly from a disk partition. `dd` is a good tool for creating absolute block-by-block copies of entire file systems. The `dd` utility converts and copies a file to the specified output device (disk-to-disk backup).

6.3.4 tar and cpio utilities

The `tar` and `cpio` utilities copy regular files or directories to disk or tape. These utilities are best suited for saving portions of file systems (a series of files or directories of files) that can be written to one tape (round or cartridge). One limitation is that you must read the entire contents of the tape to determine what files reside on the media. `tar` archives files to tape, and `cpio` copies files; `cpio` uses standard input and standard output so it generally is used in conjunction with I/O redirection and/or command-line pipeline.

6.3.5 root and usr file systems

There is a minor, but significant change to the disk space allocation scheme for CRAY J90se systems. At initial installation, two sets of production disk partitions named `roota/usra/srca` and `rootb/usrb/srcb` are created. This lets you install into another set of partitions in multiuser mode without disturbing the running system. The CRAY J90se installation utility is designed to perform upgrade installations into the alternative set of partitions. These procedures makes references to `root` and `usr` disk partitions; therefore, if you are using a CRAY J90se system, replace `root` and `bkroot` with `roota` and `rootb`, respectively, and replace `usr` and `bkusr` with `usra` and `usrb`, respectively, in the creation and booting procedures.

At initial installation, the `bkroot` and `bkusr` disk partitions are created automatically for you, except for CRAY J90se systems that have extremely limited disk space. If these disk partitions are not defined, you must define them before continuing with the creation and booting procedures.

Procedure 10: Creating `bkroot` and `bkusr` file systems

You can create a bootable copy of your production `root` and `usr` file systems into file systems called `bkroot` and `bkusr`. You should perform this procedure before upgrading an operating system as a fall-back preparation measure, or when you are sure that no outstanding problems exist with your current production system. You should not run this procedure by using the `cron` utility. This procedure also is not a substitute for making regular backups to

tape. You should perform this procedure when the activity on the `root` and `usr` file systems is at a minimum.

The following are the steps for creating a bootable copy of your production `root` and `usr` file systems into file systems called `bkroot` and `bkusr`.

1. Create the directories and file structure as follows, replacing values in italics with values appropriate to your system. Use an `mkfs` big file allocation option suitable for the disk types on which `bkroot` and `bkusr` reside for *DEVTYPE*. If you use the UNICOS MLS product, you may have to add security-related options for system levels and compartments (*SECURITY_OPTIONS*).

```
# export PATH=$PATH:/etc
# mkdir -p /mnt
# mkdir -p /mnt2
# mkfs -q -ADEVTYPE SECURITY_OPTIONS /dev/dsk/bkroot
# mkfs -q -ADEVTYPE SECURITY_OPTIONS /dev/dsk/bkusr
```

2. Label the file systems, as follows:

```
# labelit /dev/dsk/bkroot bkroot cray
# labelit /dev/dsk/bkusr bkusr cray
```

3. Check file system consistency, as follows:

```
# fsck -u /dev/dsk/bkroot # fsck -u /dev/dsk/bkusr
```

4. Mount the file systems, as follows:

```
# mount /dev/dsk/bkroot /mnt
# mount /dev/dsk/bkusr /mnt2
```

5. Flush data from all logical device caches to disk and dump the file system by executing the following sequences of commands:

```
# cd /mnt
# ldsync; sync; sleep 4
# dump -t 0 -f - /dev/dsk/root | restore -r -f - &;
# cd /mnt2
# ldsync; sync; sleep 4
# dump -t 0 -f - /dev/dsk/usr | restore -r -f - &;
```

Note: Using dump piped to restore to copy the file systems mean that you will get the benefits of defragmentation and file system validity checking, but you may lose the ability to continue batch work that was checkpointed before the system backup.

6. Unmount and check the bkroot file system, as follows:

```
# cd /mnt
# rm restoresymtabl
# echo "root backed up to bkroot on `date`" >> bkroot.log
# cd /
# umount /dev/dsk/bkroot
# fsck -u /dev/dsk/bkroot
```

7. Unmount and check the bkusr file system, as follows:

```
# cd /mnt2
# rm restoresymtabl
# echo "usr backed up to bkusr on `date`" >> bkusr.log
# cd /
# umount /dev/dsk/bkusr
# fsck -u /dev/dsk/bkusr
```

Procedure 11: Booting bkroot and bkusr into production

The following are procedures to boot the bkroot and bkusr file systems into production.

Note: If your production `root` and/or `usr` file systems are damaged beyond the repair of `fsck`, and you have booted on a `bkroot` and `bkusr` file system to single-user mode, you can either restore your production `root` and `usr` file systems from a recent backup tape or apply the `bkroot` creation procedure in reverse order to create a new production `root` and/or `usr` file system. Boot to multiuser mode on `bkroot` and `usr`. To aid the handling of checkpointed work, you may want to disable the automatic startup of NQS while in single-user mode by editing the `/etc/config/daemons` file.

1. Shut down UNICOS by executing the following commands:

```
# /etc/shutdown
# sync
# sync
# sync
```

2. Edit the `/opt/CYRIos/sn9xxx/param` file and boot the SIO by executing the following commands (this may be done on the CRAY J90se system console, in which case the prompt will be `sn9 xxx>`):

```
sn9xxx> cd /opt/CYRIos/sn9xxx
sn9xxx> cp param param.prd
sn9xxx> cp param param.bkr

sn9xxx> ed param.bkr
  1
  /rootdev/
  s/ldd root /ldd bkroot/p
    rootdev is ldd bkroot;
  w q
sn9xxx> cp param.bkr param
sn9xxx> cd /
sn9xxx> bootsys -c
```

3. Check the `bkusr` file system and mount the `bkusr` file system, as follows:

```
# fsck -u /dev/dsk/bkusr
# mount /dev/dsk/bkusr /usr
```

4. Edit the `/etc/fstab` UNICOS configuration files by using the `vi` command and change the following displayed lines:

```
# vi /etc/fstab
```

change root, usr, bkroot and bkusr lines from:

```
/dev/dsk/root      /          NC1FS  CRI_RC=NO,rw  1  1
/dev/dsk/bkroot    /mnt       NC1FS  CRI_RC=NO,rw  1  4
/dev/dsk/usr       /usr       NC1FS  CRI_RC=NO,rw  1  2
/dev/dsk/bkusr     /mnt/usr   NC1FS  CRI_RC=NO,rw  1  2
```

to the following:

```
/dev/dsk/root      /mnt       NC1FS  CRI_RC=NO,rw  1  4
/dev/dsk/bkroot    /          NC1FS  CRI_RC=NO,rw  1  1
/dev/dsk/usr       /mnt/usr   NC1FS  CRI_RC=NO,rw  1  2
/dev/dsk/bkusr     /usr       NC1FS  CRI_RC=NO,rw  1  2
```

5. Edit the `/etc/config/rcoptions` file by using the `vi` command and change the following displayed lines:

```
# vi /etc/config/rcoptions
```

change ROOTDEV, PIPEDEV, and USRDEV lines from:

```
ROOTDEV='root'
PIPEDEV='root'
USRDEV='usr'
```

to the following:

```
ROOTDEV='bkroot'
PIPEDEV='bkroot'
USRDEV='bkusr'
```

6. Unmount the `/usr` file system by executing the following command:

```
# umount /usr
```

7. Enter multiuser mode by executing the following command:

```
# /etc/init 2
```

Procedure 12: Backing up the SWS

To create a backup copy of the SWS database, see the “SWS Database and File System Backups” section of the *SWS-ION Administration and Operations Guide*, Cray Research publication SG-2204.

6.4 `/etc/dump` utility

The `/etc/dump` utility provides either full or incremental file system dumps. Dump level numbers 0 through 9 are used to determine the files that will be dumped. Dump level 0 causes the entire file system to be dumped. You can arbitrarily assign levels 1 through 9 (9 is considered the lowest level). A description of some important options follows. For a complete description of all the options, see the `dump(8)` man page.

<u>Option</u>	<u>Description</u>
<code>-A altfile</code>	Specifies the name of a file to contain a second copy of the output from the beginning of dump.
<code>-c</code>	Writes to cartridge tape.
<code>-f file</code>	Places the dump in a disk file, rather than on tape.
<code>-t dump_level</code>	Specifies the dump level. Default is level 9.
<code>-u</code>	Writes the date and time of the beginning of dump in the <code>/etc/dumpdates</code> file. A separate date is maintained for each file system and dump level.
<code>-v vsn_list</code>	Specifies a list of volume serial numbers (VSNs) to use for output. If you omit this option, dump prompts the operator for a list of VSNs.

`-w` Prints the file systems that must be dumped. This information is gathered from the dump frequency field in the `/etc/fstab` and `/etc/dumpdates` files.

Note: The `/etc/dump` utility is slow. Files are dumped (written) to tape in inode number order. The `/etc/dump` utility begins by traversing across and down the directory hierarchy of the file system, creating an index. This index is written to the first tape preceding data. The `restore` utility uses this index information.

6.5 Routine backup (`dump`) strategy

You can make two different types of backups: *full backups* or *partial backups*. Which type of backup you choose to use depends on your site, the time involved to make the backups, and the amount of media you can use for the backups. Perform file system dumps when the system is as quiet as possible. You do not have to be in single-user mode to perform file system backups. A *full backup* copies all user areas, UNICOS files, and any other special files. Full backups are often done to document the system status at a particular point in time (for example, immediately before a software update). A *partial backup* is usually more appropriate for copying everyday work; it is easily customized to individual sites.

The `dump` utility dumps all file system files that have been modified since the most recent dump that was performed at a lower level. For example, if a level 9 dump was performed on Sunday, a level 8 dump was performed on Monday, a level 9 dump was performed on Tuesday, and a level 9 dump was performed on Wednesday, then Monday's level 9 dump tapes would contain all changes since Sunday's level dump. Tuesday's level 8 dump tapes also would contain all changes since Sunday's level dump (Sunday's level dump is the most recent dump with a dump level value less than 8). Wednesday's level 9 dump tapes would contain all changes since Tuesday (Tuesday's level 8 dump is the most recent dump with a dump level less than 9).

You should save all of the tapes that would be required to recover a given week's work for at least two weeks. Some sites use five different sets of tapes, one set for each week of a month, and the fifth set for the first week of the following month. For the second week of the following month, the first of the five sets of tapes is overwritten. With this strategy, only five sets of back-up tapes are required, and a one-month rolling window of file system contents is preserved.

Most sites perform a full file system dump (level 0) once a week and level 9 dumps every day until the next week's full level dump.

The following is the recommended routine back-up (dump) strategy. It involves performing a full (level 0) dump to cartridge tape on a weekly basis and incremental (level 9) dumps on a daily basis. If you follow this plan, you need only two sets of tapes to reload a file system: the weekly dump and the most recent daily dump.

- Once per week: You should do a full (level 0) dump.
 - Repeat for each file system you want to copy.
 - Because the dump command can read unmounted file systems, you can unmount the file system to be dumped before you begin.
- Daily: You should perform an incremental (level 9) dump:
 - Dump everything that has been modified since the last dump performed with a lower dump level.
 - Repeat for each file system you want to copy.
 - Do this each day that you do not perform a level 0 dump.

6.6 Restoring file systems

The restore utility (`/etc/restore`) processes tapes produced by `/etc/dump`. The main options are as follows (for a complete description of all options, see the `restore(8)` man page):

<u>Options</u>	<u>Description</u>
<code>-c</code>	Reads from cartridge tape.
<code>-f file</code>	Reads the dump from a disk file, rather than tape.
<code>-i</code>	Initiates interactive restoration. A shell-like interface is provided that lets the user traverse through the directory tree and select files to be restored.
<code>-r</code>	Reads entire tape and loads into current directory. Do this only if you run <code>mkfs</code> on the file system first. You usually should do a full dump after a full restore.
<code>-t</code>	Lists the specified file names if the files are on the tape. If no file names are specified, all files on the tape are displayed.

- v *vsn* Causes `restore` to type the name of each file it treats, preceded by its file type.
- x Extracts specified files from the tape (creates subdirectories as necessary).

Note: Because of the use of synchronous write operations, `restore` is slow. `restore` wants to ensure that directory files were created before trying to write files into directories. Because the interface on the `restore` utility also is rather limited, be sure to use the `-i` (interactive) option when possible.

6.7 Increasing and decreasing file system space

Reorganizing file systems can involve one or more of the following activities: increasing and decreasing file system space, and/or reducing file system fragmentation.

A file system can become fragmented. The amount and occurrence of fragmentation occurs with a combination of factors: changes in a file system, low free space in a file system, and amount of time a newly created file system is in use.

Not all file systems suffer from fragmentation. For example, `/root` and `/usr` contain many directories and commands that never change; but the user and spooling (if separated) file systems are in constant change. When you want to decrease file system fragmentation, perform a full dump and restore of that file system.

6.8 Tape devices

As of the UNICOS 9.0.2 release, the following tape device names are no longer supported: `/dev/rss00`, `/dev/rmt00`, `/dev/rpe02`, `/dev/rpq01`, `/dev/rpd03`. The new naming convention uses tape daemon character devices as configured in the `/etc/config/text_tapeconfig` file. An example would be `/dev/tape/t120`, where `t=tape`, `2=SCSI bus number`, `1=SCSI target number`, `0=logical unit number`.

The list of supported tape device names changes periodically, and to remain up-to-date on supported tape devices, refer to Product Availability Notice (PAN) #111. See your Cray service representative for details on how to access this information.

6.9 Procedures included in this section

This subsection includes the following procedures:

- Backing up (dumping) a file system without `tpdaemon`
- Restoring a file system without `tpdaemon`
- Backing up (dumping) a file system by using `tpdaemon`
- Restoring a full file system by using `tpdaemon`
- Restoring a partial file system by using `tpdaemon`

Note: To back up and restore in batch requires you to set limits on the user database (UDB) account being used and on the Network Queuing System (NQS) queue. You also must use the `qsub -lU` command.

Procedure 13: Backing up (dumping) a file system without `tpdaemon`

In this method of doing a dump, you will use the UNIX-accessible logical tape devices that are defined in the `/dev` directory, as opposed to the `tpdaemon`-accessible devices defined in the `/dev/tape` directory.

Note: For this example, a square (CART) tape device, with the name `/dev/tape/CART`, is used.

For the rest of this example, a square-tape (CART) device with the name `/dev/tape/CART` is used.

1. To determine whether the tape is in a physically writable condition, load the device and enter the following command:

```
sn5111# mt -f device status
```

If necessary, physically alter the tape so that you can write to it.

For round (TAPE) tapes, if a plastic ring is clipped to the inner diameter of the tape, you can write to the tape. If no ring is clipped to the inner diameter of the tape, you cannot write to the tape.

For square (CART) tapes, you can roll a small plastic wheel back and forth. If the wheel is rolled so that the dot shows, you cannot write to the tape. If you roll the wheel so that the dot does not show, you can write to the tape.

For quarter-inch cartridge (QIC) tapes, a small black dial near one corner of the tape has a raised piece of plastic in the shape of a `>`. If the point of the shape

points at the word *SAFE* (for example, *> SAFE*), you cannot write to the tape. If you twist the dial so that the point of the shape points away from the word *SAFE* (for example, *< SAFE*), you can write to the tape.

For EXABYTE tapes (type *EXB*), on the edge of the tape you can pull a small red piece of plastic along the length of the tape so that it covers a small hole. If the piece of red plastic shows and the hole is covered, you cannot write to the tape. If you slide the piece of red plastic back so that it cannot be seen and the hole is exposed, you can write to the tape.

For digital audio tapes (*DAT*), a small white plastic slide covering the hole indicates you can write to the tape. If this white plastic slide exposes the hole, you cannot write to the tape.

2. Physically mount a tape in the tape drive that matches the logical tape device you want to use (for this example, a square (*CART*) tape was mounted in a drive that corresponds to the logical device `/dev/tape/CART`).
3. Rewind the tape. Round-type tape drives rewind the tape automatically when you push the button to select the tape to be loaded. It is a good practice to be cautious and rewind other types of tapes when they are used. Because round tapes are used in this example and they rewind automatically, you probably would exclude this step; however, to rewind other types of tapes, enter the `mt -f /dev/tapename rew` command and specify the tape device you are rewinding (`-f /dev/tapename` specifies the raw tape device to be activated). For example, to rewind an EXABYTE (`/dev/rpe02`) tape, type the following command line:

```
sn5111# mt -f /dev/tape/CART rew
```

Note: When dumping an open file, the updated file will not be dumped until the file is written to disk. If you want to ensure that all files are dumped, you should unmount the file system.

4. Dump the file system to tape. In this case, a full level (`-t 0`) dump (as opposed to a partial dump) is performed. The `-u` option is highly recommended. If you invoke this option, the date and time of the beginning of the dump will be written to a file called `/etc/dumpdates`, and a separate entry for each file system and each dump level will be recorded.

If this is the first time the `dump` command has been used on your system with the `-u` option, the `/etc/dumpdates` file probably does not exist. This causes the following error message at the end of the `dump` command screen output:

```
dump (/src to /tmp/dumpfile): dump has completed, 23618 blocks
dump (/src to /tmp/dumpfile): cannot open an existing /etc/dumpdates file
dump (/src to /tmp/dumpfile): The dump is aborted.
```

To prevent this error, you must create an empty file named `/etc/dumpdates` before executing the `/etc/dump` command. One way to do this follows:

```
sn5111# touch /etc/dumpdates
```

The following example shows a full file system dump (`-t 0`) to a square tape (`-f /dev/tape/rss00`):

```
sn5111# /etc/dump -t 0 -u -f /dev/tape/CART /dev/dsk/src
```

Note: You may want to append an `&` symbol to the end of the `/etc/dump` command line so that this command operates as a background process and you can still perform other operations (such as responding to operator messages) while the `dump` command is running.

5. Physically alter the tape to prevent the tape from being overwritten (see information included in step 2).
6. Attach a physical label to the tape that states the file systems that have been dumped to the tape and the date the tape was written. It may be useful to add the command that was used to write the tape. It also may be useful to add the commands necessary to restore the tape.

Your file system backup (`dump`) is now complete.

Procedure 14: Restoring a file system without `tpdaemon`

In this method of doing a restore, you will use the UNIX-accessible logical tape devices that are defined in the `/dev` directory, as opposed to the `tpdaemon`-accessible devices defined in the `/dev/tape` directory. A *full file system restore* means that the entire contents of a file system will be read in from tape and will overwrite the current disk version of that file system. A *partial file system restore* restores only a file or directory or some subset of a file system to the logical device; the rest of the file system remains untouched.

1. If it was not already unmounted, unmount the file system to be restored by using the `/etc/umount` command. The `/dev/dsk/src` file system is used in this sample procedure.



Caution: Before you can unmount it, the file system must be idle. To determine whether the file system is idle, you can use the `/etc/fuser` utility.

To determine whether the file system in question is currently mounted, examine the output of the `/etc/mount` command:

```
sn5111# /etc/mount
/ on /dev/dsk/root read/write on Fri Feb 11 10:38:15 1994
/tmp on /dev/dsk/tmp read/write on Fri Feb 11 10:40:56 1994
/usr on /dev/dsk/usr read/write,rw,CRI_RC="NO" on Fri Feb 11 10:40:59 1994
/usr/home on /dev/dsk/home read/write,rw,CRI_RC="YES" on Fri Feb 11 10:41:02 1994
/usr/src on /dev/dsk/src read/write,rw,CRI_RC="YES" on Fri Feb 11 10:41:04 1994
```

In this case, the last line of the output from the `/etc/mount` command shows that the `/dev/dsk/src` file system is currently mounted on the mount point `/usr/src`.

The `/etc/umount` command unmounts the file system. You can specify either the mount point or the file system logical device name after the `umount` command for it to be effective. In the following example, the logical device name was used:

```
sn5111# /etc/umount /dev/dsk/src
```

Now the output of the `mount` command shows that the `/dev/dsk/src` file system is no longer mounted:

```
sn5111# /etc/mount
/ on /dev/dsk/root read/write on Fri Feb 11 10:38:15 1994
/tmp on /dev/dsk/tmp read/write on Fri Feb 11 10:40:56 1994
/usr on /dev/dsk/usr read/write,rw,CRI_RC="NO" on Fri Feb 11 10:40:59 1994
/usr/home on /dev/dsk/home read/write,rw,CRI_RC="YES" on Fri Feb 11
10:41:02 1994
```



Warning: The following step deletes all information on this file system.

2. **Complete this step only if you are doing a full file system restore. If you are doing a partial file system restore, skip to step 4.** Remake the file system structure on the `/dev/dsk/src` logical device by using the `/etc/mkfs` command.

The `-q` option on the `mkfs` command shown in the following example is optional syntax. Using this option bypasses the disk surface check and speeds the `mkfs` process, but it is not the most thorough way to prepare the disk for a file system structure. The first time you use the `/etc/mkfs` command to format a logical disk area for a file system structure, do not use the `-q` option. For more information about the `mkfs` command, see section Chapter 5, page 35, and the `mkfs(8)` man page.

```
sn5111# /etc/mkfs -q /dev/dsk/src
```

3. Check the file system by using the `/etc/fsck` command. Before mounting the file system, you **must** perform this command:

```
sn5111# /etc/fsck /dev/dsk/src
```

4. Make sure that no other file system is mounted on the directory in which you intend to mount your file system. You must mount the file system being restored on a mount point where you can perform the remaining administrative tasks without users being affected or interfering. Traditionally, the mount point that administrators use for such tasks is `/mnt`, because `/mnt` is not a directory users are likely to access. If the system is in multiuser mode, users probably will not interrupt administrative tasks being performed in that directory.

To check that no other file system is mounted on the directory in which you intend to mount your file system, examine the output of the `etc/mount` command, which lists all file systems currently mounted and their mount points, as shown in the following example:

```
sn5111# /etc/mount
/ on /dev/dsk/root read/write on Mon Feb 14 19:09:25 1994
/tmp on /dev/dsk/tmp read/write on Mon Feb 14 19:10:01 1994
/usr on /dev/dsk/usr read/write,rw,CRI_RC="NO" on Mon Feb 14 19:10:04 1994
/usr/home on /dev/dsk/home read/write,rw,CRI_RC="YES" on Mon Feb 14 19:10:07 1994
```


The mount command output shows that no file systems are mounted on the /mnt mount point.

5. Mount the file system on the mount point you have selected by using the `/etc/mount` command. In this example, the mount point /mnt is used. If any user is in the directory or any of its subdirectories, the mount command will not be successful (to remove users from a file system forcibly, see the `fuser(8)` man page). If you are the one in the directory or a subdirectory, change to a directory that is not part of the directory tree, including the mount point directory or any of its subdirectories, as follows:

```
sn5111# cd /
sn5111# /etc/mount /dev/dsk/src /mnt
```

6. Change directories to the mount point directory on which the file system is mounted. In step 6, /mnt was selected to be used for this directory:

```
sn5111# cd /mnt
```

7. Physically mount a tape in the tape drive that matches the logical tape device you want to use. In this example, a square (CART) tape was mounted in a drive that corresponds to the logical device `/dev/rss00`. The contents of this tape should include the dumped file system you want to restore, in this case, `/dev/dsk/src`.
8. Rewind the tape. Round-type tape drives rewind the tape automatically when you push the button to select the tape to be loaded. You should be cautious and rewind other types of tapes when they are used. However, to rewind other types of tapes, enter the `mt -f /dev/tapename rew` command and specify the tape device you are rewinding (`-f /dev/tapename` specifies the raw tape device to be activated). For example, to rewind a square (`/dev/tape/CART`) tape, type the following command line:

```
sn5111# mt -f /dev/tape/CART rew
```

9. Restore either the full file system or, if you are doing a partial restore, restore the files and/or directories of the file system that are needed (in this case, `/dev/dsk/src`).

There are two methods of restoring file systems. This step does not discuss the interactive method in detail, but it is highly effective and very easy to use. It is invoked by using the `-i` option. For a good explanation of how to interact with the interactive shell interface to select files and directory contents for restoration, see the `restore(8)` man page.

The `-f` option addresses the logical device on which the dump tape is mounted.

To invoke the interactive method at this point, type the following command line:

```
sn5111# /etc/restore -i -f /dev/tape/CART
```

The other method of restoring a file system follows for doing a full or a partial file system restore.

To do a full file system restore :

The `-r` option invokes a full file system restore (this example is for a square tape). You may want to run the `restore` command as a background process by appending an `&` symbol to the end of the command line.

```
sn5111# /etc/restore -r -f /dev/rss00
```

To do a partial file system restore :

Two examples are given. One example restores the `/src/uts/Nmakefile` file to the `/src` file system. The other example restores the `/src/uts/cl/sys` subdirectory and all of its contents to the `/src` file system. You may want to run the `restore` command as a background process by appending an `&` symbol to the end of the command line.

The `-x` option invokes a partial file system restore. You should list the files or directories that you want extracted from tape as the last arguments of the command line. You should specify the path name for each file you want to restore relative to the topmost directory of the file system in which it resides.

In the following example, the `Nmakefile` file is restored. The file's full path name in the `/src` file system is `/src/uts/Nmakefile`. The file's path name is relative to the topmost directory of the `/src` file system; that is, relative to `/src`, it is `/uts/Nmakefile`.

```
sn5111# /etc/restore -x -f /dev/tape/CART /uts/Nmakefile
```

The following example restores the `/src/uts/c1/sys` directory and all of its files and subdirectories and their contents:

```
sn5111# /etc/restore -x -f /dev/tape/CART /uts/c1/sys
```

10. Unmount the file system when the restore has completed, as follows:

```
sn5111# /etc/umount /dev/dsk/src
```

11. Remount the restored file system on its normal mount point and check the file system, as in the following example:

```
sn5111# /etc/fsck /dev/dsk/src
```

If the file system was unmounted cleanly, this step is optional.

12. Mount the restored file system on its normal mount point. The file system is now ready for users to access. The `/src` file system usually is mounted on the `/usr` file system, as follows:

```
sn5111# /etc/mount /dev/dsk/src /usr/src
```

The file system restoration of `/src` is now complete.

Procedure 15: Restoring a partial file system by using `tpdaemon`

Assumptions

For this procedure, it is assumed that the `tpdaemon` is up and that all tape hardware (devices, controllers, and so on) are configured to be up and available to the user. In the following example, the restore is done from a square (CART) tape. That square tape was written as unlabeled (`-1 n1`), and it had a volume name of `JON1`.

A partial file system restore is being performed. This means that only a file or a directory or some subset of the file system is restored to the logical device. The rest of the file system remains untouched.

The file system restored in the example is the same file system backed up (dumped) in the backup procedure (`/src`).

As with the `dump` command, you should execute the `restore` command on as quiet a system as possible. Tell the user who owns the files being restored to stay out of that directory until the restore is completed.

In the backup (`dump`) example, `/dev/dsk/src` was dumped using the following command:

```
sn5111# /etc/dump -t 0 -u -g CART -v JON1 -l nl /dev/dsk/src
```

When doing a restore, the value for the `-l` (label) option and the `-v` (volume) option must match the label and volume that you used on the `/etc/dump` command.

Procedure 15(a): Partial file system restore

The following are the steps for performing a partial file system restore:

1. Determine which tape device group that you plan to use for your restore. This will be the same as the preceding backup (see step 1 of the backup procedure).
2. Verify that the tape(s) written from the preceding backup are set to prevent tape contents from being overwritten (see step 2 of the backup procedure).
3. Change directories to the mount point directory on which the file system is mounted. Because you are restoring files in `/usr/src`, enter the following command:

```
sn5111# cd /usr/src
```

4. Restore the files and/or directories of the file system that are needed. In this case, the `/dev/dsk/src` file system is used.

There are two methods of doing file system restores. This procedure does not discuss the interactive method in detail, but it is highly effective and very easy to use. To invoke it, use the `-i` option. The `restore(8)` man page gives a good explanation of how to interact with the interactive shell interface to select files and directory contents for restoration.

To invoke the interactive method, type the following command line:

```
sn5111# /etc/restore -i -V JON1 -l nl -g CART
```

A description of the other method of performing file system restoration follows.

Two examples follow. One example restores the `/src/uts/Nmakefile` file to the `/src` file system. The other example restores the `/src/uts/c1/sys` subdirectory and all of its contents to the `/src` file system.

The `-x` option invokes a partial file system restore. It also is used for a full file system restore if the `-r` option fails. You should list the files or directories that you want extracted from tape as the last arguments of the command line. You must specify the path name for each file you want to restore relative to the topmost directory of the file system in which it resides.

In the following example, the `Nmakefile` file is restored. The file's full path name in the `/src` file system is `/src/uts/Nmakefile`. The file's path name relative to the topmost directory of the `/src` file system (that is, relative to `/src`) is `/uts/Nmakefile`.

```
sn5111# /etc/restore -x -V JON1 -l nl -g CART /uts/Nmakefile
```

The following example restores the `/src/uts/c1/sys` directory, including all of its files and subdirectories and their contents:

```
sn5111# /etc/restore -x -V JON1 -l nl -g CART /uts/c1/sys
```

5. If no `-g` (tape type group name) option is supplied on the `/etc/restore` command to specify a particular kind of tape device to reserve, `restore` will use the default device group name set by the `DEV_DGN` argument in the `/etc/config/tapeconfig` file. By default, when your system arrives, the `DEV_DGN` argument is set to round (TAPE) tapes, as shown in the following excerpt from the `/etc/config/tapeconfig` file:

```
#
# default device group name
# Specify a decimal number
# This must be one of the device types specified in the CNT
# configuration
#
DEF_DGN          TAPE
```

To change this default, change your tape configuration either by using the menu system `Configure System ==> Tape Configuration` menu or by editing the `/etc/config/tapeconfig` file.

Because the `/etc/restore` command also defaults to `DEF_DGN` tapes, you can specify other tape types by using the `-g` option.

6. The tape mount request initiated by the `restore` command causes the system to inform you that operator messages exist. The following message repeats on the console until you open up a window to reply to operator messages:

There are operator messages that require attention

To open up a window to reply to the tape mount operator messages, type the following command:

```
sn5111# /usr/lib/msg/oper
```

Your entire screen now shows a display that looks something like the following:

```
Command: msgd Page: 1 [delay 10] Thu Oct 7 17:09:02 1993

Msg #   Time   System Messages
=====  =====  =====

1      17:07   TM122 - mount tape JON1(nl) on a CART device for jones
                14, () or reply cancel / device name
:: display truncated:
Enter '?' for help.
>
```

At this point, you can respond in one of three ways:

- Mount the tape in the device
- Specify a different device on which you want to mount the tape
- Cancel the request

In the preceding example, assume that an unlabeled CART tape was mounted in a CART drive. This causes another message to appear that will require a response.

7. Respond to the next operator message by specifying the volume serial number (VSN) of the tape.

In this case, the message number was 2 and the volume serial number was JON1, so `/usr/lib/msg/rep 2 JON1` was typed at the `>` prompt, as follows:

```
Command: msgd Page: 1 [delay 10] Thu Oct 7 17:11:10 1993
Msg #   Time   System Messages
=====
1    17:07   TM122 - mount tape JON1(nl) on a CART device for jones
                14, () or reply cancel / device name
2    17:10   TM011 - enter vsn for tape on device CART:: display truncated:
> /usr/lib/msg/rep 2 JON1
```

All messages related to your tape job have now disappeared.

8. Exit the operator window, and return to the system prompt by typing `exit` at the `>` prompt, as follows:

```
Command: msgd Page: 1 [delay 10] Mon Oct 11 14:02:23 1993
Msg #   Time   System Messages
=====
::: display truncated:
Enter '?' for help.
> exit
sn5111#
```

The partial file system restoration of `/src` is now complete.

Note: In some cases (you must specify tape block size, for example) it may be necessary to perform the tape daemon interface manually. The following manual example does the same thing as the preceding "automatic" restore example:

```
sn5111# rsv CART
sn5111# tpmnt -l nl -r in -n -v JON1 -g CART -p /tmp/dumpfile
sn5111# /etc/restore -r -f /tmp/dumpfile
sn5111# rls -a
```

