Chapter 3

Troubleshooting Backup and Recovery

From time to time you might experience backup failures. It is vitally important that you determine the cause of the failure. Most often, the failure is due to worn or faulty media. Proceeding without determining the cause of a failure makes all your future backups suspect and defeats the purpose of backups.

This chapter contains the following sections:

- "Unreadable Backups" on page 69
- "Reading Media From Other Systems" on page 69
- "Restoring the Wrong Backup" on page 72
- "Errors Creating the Backup" on page 71
- "Testing for Bad Media" on page 73
- "Backup and Recovery Error Messages" on page 74

Unreadable Backups

The reasons a backup might be unreadable include:

- The data on the backup tape is corrupted due to age or media fault.
- The tape head is misaligned now, or was when the backup was made.
- The tape head is dirty now, or was when the backup was made.

Reading Media From Other Systems

You may not be able to read data created on another vendor's workstation, even if it was made using a standard utility, such as *tar* or *cpio*. One problem may be that the tape format is incompatible. Make sure the tape drive where the media originated is compatible with your drive.

If you are unable to verify that the drives are completely compatible, use *dd* to see if you can read the tape at the lowest possible level. Place the tape in the drive and enter the command:

mt blksize

The mt(1M) command with these options tells you the block size used to write the tape. Set the block size correspondingly (or larger) when you use *dd* to read the tape. For example, if the block size used was 1024 bytes, use the command:

dd if=/dev/tape of=/usr/tmp/*outfile* **bs=1024**

If *dd* can read the tape, it displays a count of the number of records it read in and wrote out. If *dd* cannot read the tape, make sure your drive is clean and in good working order. Test the drive with a tape you made on your system.

If you can read the tape with *dd*, and the tape was created using a standard utility, such as *tar* or *cpio*, you may be able to convert the data format with *dd*. Several conversions may help:

- swab–swap every pair of bytes
- sync–pad every input block to ibs
- block–convert ASCII to blocked ASCII
- unblock–convert blocked ASCII to ASCII
- noerror–do not stop processing on an error

The *dd* program can convert some completely different formats:

- ascii–convert EBCDIC to ASCII
- ebcdic–convert ASCII to EBCDIC
- ibm–slightly different map of ASCII to EBCDIC

Converting case of letters:

- lcase–map alphabetics to lowercase
- ucase–map alphabetics to uppercase

If the data was written on another vendor's system, you may be able to convert it using *dd*, then pipe the converted output to another utility to read it.

Many other vendors use byte-ordering that is the reverse of the order used by IRIX. If this is the case, you can swap them with the following command:

dd if=/dev/tape conv=swab of=/usr/tmp.O/*tapefile*

Then use the appropriate archiving utility to extract the information from */tmp/tapefile* (or whatever filename you choose). For example, use this command to extract information if the *tar* utility was used to make the tape on a byte-swapped system:

```
tar xvf /usr/tmp.O/tapefile .
```

Note that you could also pipe the *dd* output to another local or remote tape drive (if available) if you do not need or want to create a disk file.

Or you can use the no-swap tape device to read your files with the following *tar* command line:

```
tar xvf /dev/rmt/tps0d4ns
```

Of course, if your tape device is not configured on SCSI channel 4, the exact */dev/rmt* device name may be slightly different. For example, it could be */dev/rmt/tps0d3ns*.

It is good practice to preview the contents of a *tar* archive with the **t** keyword before extracting. If the tape contains a system file and was made with absolute pathnames, that system file on your system could be overwritten. For example, if the tape contains a kernel, */unix*, and you extract it, your own system kernel will be destroyed. The following command previews the above example archive:

tar tvf /tmp/*tarfile*

If you wish to extract such a tape on your system without overwriting your current files, use this command to force the extraction to use relative pathnames:

```
tar Rx
```

or the corresponding *bru* command:

```
bru -j
```

Errors Creating the Backup

If you see errors on the system console when trying to create a backup, some causes are:

- The tape is not locked in the drive. You may see an error message similar to this:

  ```
  /dev/nrtape rewind 1 failed:Resource temporarily unavailable
  ```

  Make sure the tape is locked in the drive properly. See your *Owner's Guide* if you do not know how to lock the tape in the drive.
- File permission problems. These are especially likely with file-oriented backup programs; make sure you have permission to access all the files in the hierarchy you are backing up.
- The drive requires cleaning and maintenance.
- Bad media; see "Testing for Bad Media" on page 73.

If you encounter problems creating backups, fixing the problem should be your top priority.

Restoring the Wrong Backup

If you accidentally restore the wrong backup, you should rebuild the system from backups. Unless you are very sure of what you are doing, you should not simply restore the correct backup version over the incorrect version. This is because the incorrect backup may have altered files that the correct backup won't restore.

In the worst possible case, you may have to reinstall the system, then apply backups to bring it to the desired state. Here are some basic steps to recovering a filesystem.

If you used incremental backups, such as from *backup* or *bru*:

1. Make a complete backup of the current state of the filesystem. If you successfully recover the filesystem, you will not need this particular backup. But if there is a problem, you may need to return to the current, though undesirable, state.
2. Start with the first complete backup of the filesystem that was made prior to the backup that you want to have when you're finished. Restore this complete backup.
3. Apply the series of incremental backups until you reach the desired (correct) backup.

If you accidentally restored the wrong file-oriented backup (such as a *tar* or *cpio* archive):

1. Make a complete backup of the affected filesystem or directory hierarchy. You may need this not only as protection against an unforeseen problem, but to fill any gaps in your backups.
2. Bring the system to the condition it was in just before you applied the wrong backup.

   If you use an incremental backup scheme, follow steps 2 and 3 above (recovering from the wrong incremental backup).

   If you use only utilities such as *tar* and *cpio* for backups, use what backups you have to get the system to the desired state.
3. Once the system is as close as possible to the correct state, restore the correct backup. You are finished. If the system is in the desired state, skip the remaining steps.

   If you cannot bring the system to the state it was in just before you applied the wrong backup, continue with the next series of steps.
4. If you cannot manage to bring the system to the correct state (where it was just before you restored the wrong backup), get it as close as possible.
5. Make a backup of this interim state.
6. Compare the current interim state with the backup you made at the outset of this process (with the incorrect backup applied) and with the backup you wish to restore. Note which files changed, which were added and removed, and which files remain unchanged in the process of bringing the system to the desired state.

   Using these notes, manually extract the correct versions of the files from the various tapes.

Testing for Bad Media

Even the best media can go bad over time. Symptoms are:

- Data appears to load onto the tape correctly, but the backup fails verification tests. (This is a good reason to always verify backups immediately after you make them.)

  Another tape is then able to back up the data successfully and pass verification tests.
- Data retrieved from the tape is corrupted, while the same data loaded onto a different tape is retrieved without problems.

- The backup media device driver (such as the SCSI tape driver) displays errors on the system console when trying to access the tape.
- You are unable to write information onto the tape.

If errors occur when you try to write information on a tape, make sure the tape is not simply write-protected. Be sure you are using the correct length and density tape for your drive.

Make sure that your drive is clean and that tape heads are aligned properly. It is especially important to check tape head alignment if a series of formerly good tapes suddenly appear to go bad.

Once you are satisfied that a tape is bad, mark it as a bad tape and discard it. Be sure to mark it "bad" to prevent someone else from accidentally using it.

Backup and Recovery Error Messages

Following are some of the possible error messages you may see that indicate problems with a backup or recovery.

```
unix: dks0d1s0: Process [tar] ran out of disk space
```

This error, or similar errors reporting a shortage of disk space, may occur if you are backing up data to a disk partition that does not have enough free space left to contain the data to be backed up.

Such errors may likewise occur in data restores if the data being recovered does not fit on the destination disk partition. Note that if you are uncompressing data that was compressed for backup, the uncompressed data could easily require twice as much space as the compressed data.

You may wish to add disk space, reclaim disk space, repartition existing disk space (see "IRIX Admin: Disks and Filesystems"), or redesign your backup procedure, for example, to use data compression (see "Saving Files Using Data Compression" on page 20).

```
unix: ec0: no carrier: check Ethernet cable

unix: NFS write error 151 on host garfield

unix: NFS2 getattr failed for server some.host.name: Timed out
```

These and similar network errors only represent a problem if you are using network resources (for example, a remote tape or disk drive) in your backup or recovery procedure. If this is the case, reestablish proper network connections (see "IRIX Admin: Networking and Mail") and either verify that your backup or recovery was successful or reinitiate it.

```
unix: Tape 3: Hardware error, Non-recoverable

mediad: Could not access device /dev/rmt/tps0d6nr, Device busy

unix: Tape 3: requires cleaning

unix: Tape 3: Unrecoverable media error

unix: NOTICE: SCSI tape #0, 6 had 1 successful retried commands (0% of
r/w)

unix: NOTICE: SCSI tape #0,7 Incompatible media when reading
```

These are all examples of tape access errors. Depending on whether you were trying to back up or recover data, the system encountered a problem writing or reading the tape. Be sure there is a tape in the drive indicated in the error message, and that it is not set on write-protect if you are attempting a backup. (Also, tape drives should be periodically cleaned according to manufacturer instructions.)

If these are not the problem, test the tape for read and/or write capabilities using one or more of the backup and recover utilities. Note that a media error can occur anywhere on a tape; to verify the tape, write and read the entire tape. You can also select "Run Confidence Tests" from the System toolchest and double-click on the Tape Drive test.

If you have any doubts about the quality of the tape you're using (for example, it is getting old), copy it to a new tape (if it still has good data) and discard it. If you are using a tape drive that you have not used before, verify that the tape type is compatible with the new drive. Run the mt(1) command to reset the tape drive. Run the hinv(1M) command to determine if the tape drive is recognized by the system.

A “device already in use” or “device busy” error probably means that some other program was using the tape drive when you tried to access it.