



# Backing Up Data

**E**very computer user knows that backing up data is vital. This is usually because every computer user has lost data at some point, perhaps because of a corrupted file or accidental deletion.

Some of the people behind Unix were highly aware of such occurrences, and built in several advanced and useful backup tools. These have been mirrored within Linux, with the result that creating and maintaining backups is easy.

In this chapter, we'll first look at what data should be backed up, and then explore two ways to make backups: using the Simple Backup utility and from the command line.

## What Data Should You Back Up?

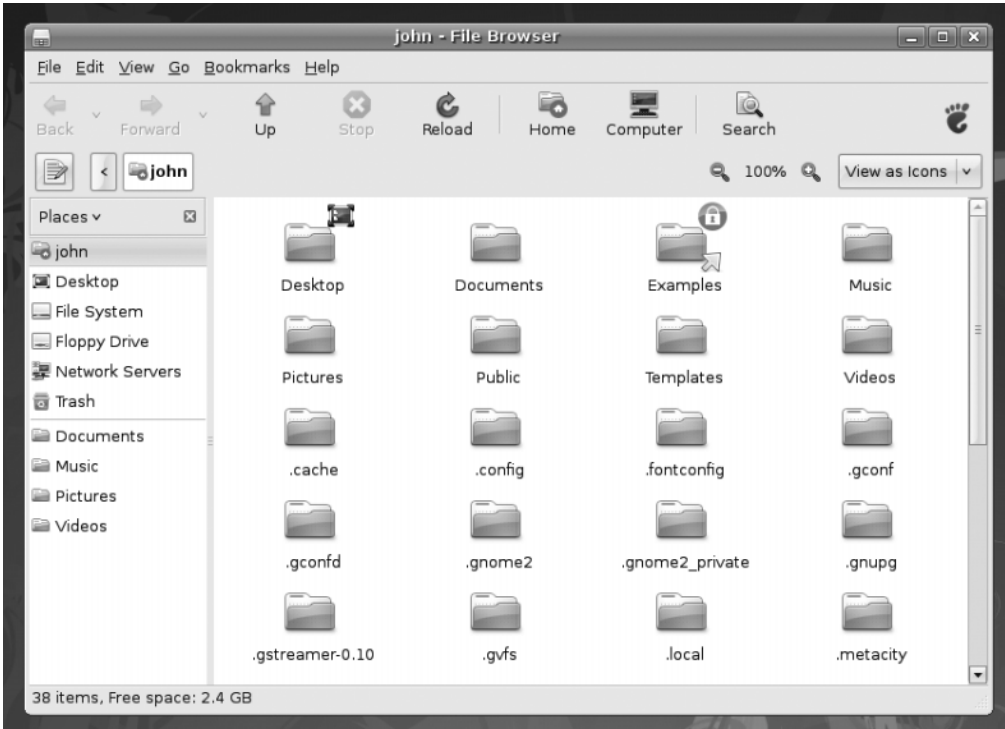
Data on your system can be classified into three broad types: program data, configuration data, and personal data. It's traditionally reasoned that backing up all types of data is inefficient, because it would mean backing up practically the entire hard disk. Because of this, you usually want to back up the latter two types of data: configuration and personal. The theory is that if your PC is hit by a hard-disk-wrecking disaster, you can easily reinstall the operating system from the CD or DVD. Restoring your system from backup is then simply a matter of ensuring the configuration files are back in place, so your applications work as you would like them to, and making sure that your personal data is once again made accessible.

Practically all the personal configuration data for programs you use every day, as well as your personal data, is stored in your `/home` directory (although the configuration files for software used system-wide are stored in the `/etc` directory). If you take a look in your `/home` directory, you might think that previous sentence is incorrect. On a freshly installed system, the directory appears largely empty. However, most, if not all, of the configuration files are hidden; their directory and filenames are preceded with a period (`.`), which means that Linux doesn't display them during a standard directory listing.

To view hidden files and folders in the Nautilus file manager, select View ► Show Hidden Files. This can be quite an eye-opener when you see the masses of data you didn't

even realize were there, as shown in the example in Figure 31-1. To view hidden files at the shell prompt, simply use the `-a` command option with the `ls` command:

```
ls -a
```



**Figure 31-1.** Most of the configuration files for programs are hidden—literally—in your `/home` directory.

The configuration files held in your `/home` directory relate solely to your user account. Any other users will have their own configuration files, entirely independent of yours. In this way, all users can have their own configuration settings for various applications, which can be backed up independently.

Under Ubuntu, you can back up both configuration data and personal files using Simple Backup, which can be downloaded from the Ubuntu software repositories.

Keep in mind that there's little point in making backups if you leave the resultant archive files on your hard disk. For full backup protection, the archives should be stored elsewhere, such as on an external hard disk, network mount, or CD/DVD-ROM. Consider using GNOME's CD/DVD Creator (click **Go** ► **CD/DVD Creator** on the menu of any open Nautilus window or **Places** ► **CD/DVD Creator** from the desktop).

## Using Simple Backup

Simple Backup is a series of programs that enable quick and easy backup and restoration of personal data, as well as system configuration files. Its output, which takes the form of backup directories containing an archive of the files, plus configuration data, can be written to your hard disk (or a network mount attached to it), or to a remote Internet location, such as an FTP server.

Simple Backup was created courtesy of the Google Summer of Code sponsorship scheme and was designed with the help of Ubuntu developers. To install Simple Backup, open Synaptic Package Manager (**System** ► **Administration**), and then search for `sbackup`. Click its entry in the list of results, mark it for installation, and click **Apply**. You'll then find entries for the backup and restoration components of Simple Backup on the **System** ► **Administration** menu.

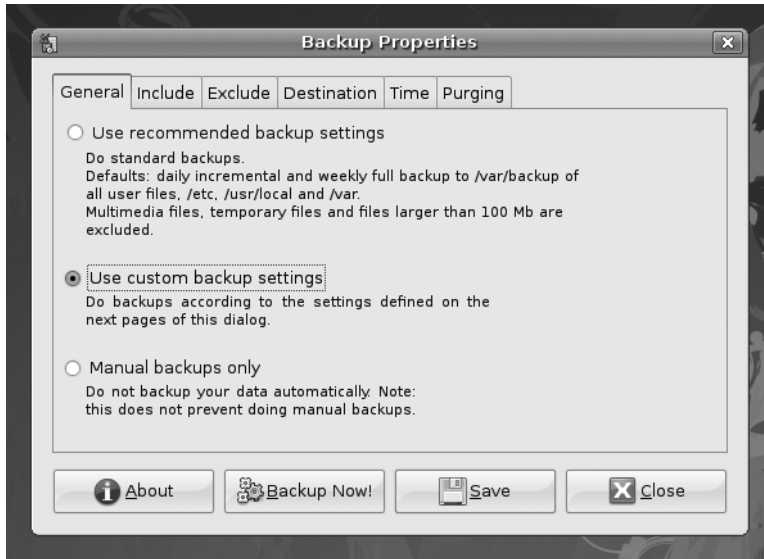
---

**Note** Simple Backup is in the Universe software repository. If you haven't already set up your online software repositories, follow the instructions in Chapter 8 to do so.

---

## Backing Up Data via Simple Backup

To configure a backup, select **System** ► **Administration** ► **Simple Backup Config**. You'll see the Backup Properties dialog box, as shown in Figure 31-2. Using this dialog box, you can choose the files that Simple Backup backs up, as well as when it does so. Once you've made your changes, click the **Save** button. This should be done before making a backup. If scheduled backups are set, it's sufficient to save the changes and quit the program. The backup jobs will take place automatically, in the background, at the set times.



**Figure 31-2.** Simple Backup can work automatically or with custom settings you specify.

As listed on the General tab of the Backup Properties dialog box, Simple Backup can operate in three different modes:

**Use recommended backup settings:** This is by far the best choice for fuss-free operation. Simple Backup will perform a daily backup of your /home directory, as well as the vital system data held in /etc, /usr/local, and /var. However, it will deliberately exclude any multimedia files (because of their large size), along with any temporary files and files of any type that exceed 100MB (again for size reasons). By default, the backup directory created is placed in /var/backup.

**Use custom backup settings:** This is effectively the same as the recommended backup, and includes the same list of file inclusions and exceptions, but you are able to edit the settings manually. For example, you might choose to include MP3 files, rather than excluding them. The custom backup option lets you alter where the eventual backup directory is saved and the time when the backup is made.

**Manual backups only:** This effectively deactivates Simple Backup, so that it no longer periodically backs up files. However, you can still click the Backup Now! button to manually perform a backup according to the settings on the other Backup Properties dialog box tabs.

---

**Note** Simple Backup doesn't create a new backup each time it runs, because that would take too long. The first time it runs, a full backup is taken, but those created afterward are *incremental backups*, and only files that are new or that have changed are backed up. The backup directory created during the first run is given the file extension `.ful`, while the backup directories created after this have the extension `.inc`. As you might expect, if the original `.ful` backup directory can't be found, a new full backup will be created.

---

### Including Files and Folders in the Backup Job

Assuming that you've chosen a custom backup, and therefore are able to alter the backup settings, clicking the Include tab in the Backup Properties dialog box allows you to specifically define directories and files that you wish to include in the backup. Simply click the Add File or Add Directory button, and then browse to the relevant location (to add a directory, you'll need to click to open it before clicking the Open button).

Bear in mind that adding a directory does so recursively, which means that any directories contained within that directory are also backed up. For this reason, you don't need to specifically add your `/home/<username>` directory, because the entire `/home` directory is included in the backup by default. This means the backup will also include all other users' directories within `/home`, too.

### Excluding Files and Folders from the Backup Job

You have a wide range of choices when it comes to excluding files and folders from the list. Directories can be excluded based on their location. Files can be excluded based on location, type, or size.

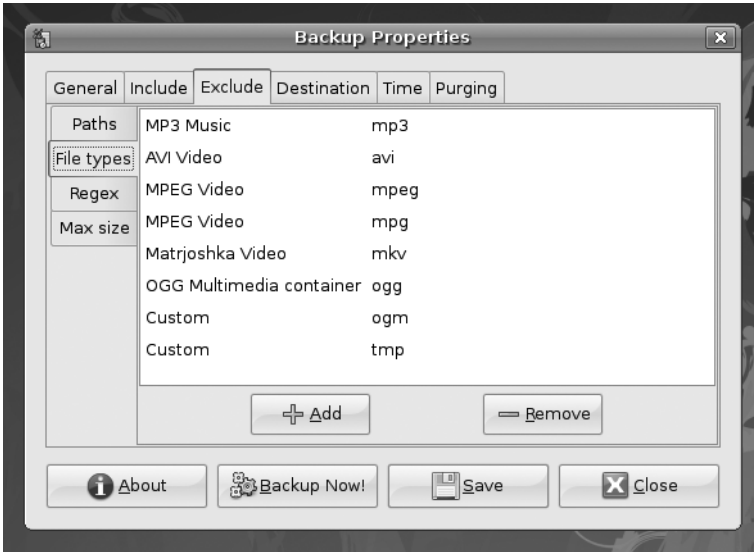
Clicking the Exclude tab in the Backup Properties dialog box reveals a set of side tabs on the left side of the program window, which allow you to exclude items from the backup as follows:

**Paths:** To exclude a specific file or folder, click this side tab. As with including files, click the Add File or Add Directory button, and then browse to the relevant location.

**File Types:** To exclude certain types of files, click this side tab, as shown in Figure 31-3. After clicking the Add button, you'll see that you can choose from a brief list of standard file types or filter by file extension (such as `mp3` for MP3 files or `zip` for compressed Zip files). If you want to back up your multimedia files, remove the corresponding file type entries from this list.

**Regex:** If you're competent at using regular expressions, as outlined in Chapter 15, you can use them to specify extremely precise rules by clicking this side tab.

**Max size:** Any files larger than the stated size on this tab aren't backed up. By removing the check next to the Do Not Backup line, you can deactivate this feature (although that could lead to massive backup files, which would take a long time to generate).



**Figure 31-3.** *Excluding certain types of large files will lead to far smaller backup files.*

## Changing the Backup Directory Destination

By default, the backup directory created by Simple Backup is stored in the `/var/backup` directory. By clicking the Destination tab in the Backup Properties dialog box, you can choose to save it in a different location on your hard disk. Of course, if you have a network mount, you can also opt to save it there. In most cases, we advise that you use `/var/backup` to store the newly created backup files, and copy the files to their permanent destinations later. You might even choose to do this periodically and automatically. By following the instructions in Chapter 32, which explain how to schedule tasks, you could set up a cron job to automatically copy the files to a network mount or removable storage device.

---

**Note** Remember that Simple Backup creates incremental backups, so you should copy *all* the backup directories and files within `/var/backup` to the external storage device, rather than just the latest one.

---

You can even transfer the backup directory across the Internet via SSH file transfer or the less secure FTP standard. To do so, simply enter the protocol, username, password, and URL in the following format:

```
sftp://username:password@mysite.com/remotedirectory
```

It's important to precede the address with the protocol you intend to use: `sftp://` for SSH or `ftp://` for FTP.

## Changing the Time Period Between Backups

Clicking the Time tab in the Backup Properties dialog box lets you set the frequency of the backup. You can opt to back up hourly, daily, weekly, or monthly. You can also set the exact time of the backup if necessary. For example, you could set a backup to take place every week on a Tuesday at 12.30 p.m. Simply select the interval period from the Do Backups drop-down list, and then select from the Day of Month, Day of Week, Hour, and Minute lists, as necessary. Simple Backup uses the system scheduler, *cron* (discussed in more detail in Chapter 32). The use of *cron* means that Simple Backup doesn't need to be running all the time for the backup to take place. Simple Backup is started and stopped automatically in the background as needed.

You can also elect to perform a full backup after a certain number of days have passed (up to 1,000). A full backup means that Simple Backup creates a new complete backup, rather than incremental ones.

## Purging Old Backup Files

By clicking the Purging tab in the Backup Properties dialog box, you can opt to automatically delete old backup directories. This can save on storage space. Purging can be done either by specifying a cutoff date, so that any backup archive older than the specified number of days is deleted, or it can be done logarithmically. This means that the program keeps just one backup out of the many that might be created in a week, month, and so on. All others are deleted. For obvious reasons, you should use the purging option with care!

## Restoring Data via Simple Backup

If the worst happens and you need to restore any number of files from the backup, you can click System ► Administration ► Simple Backup Restore.

The first step is to select the location of the backup directories. If the backups aren't contained in `/var/backup`, click Use Custom, and either type the path into the field or click the file browse button and locate the backup directories. Then click the Available Backups drop-down list to choose a backup directory from which to restore. The directory names contain the dates and times the backups were made, and it makes sense to choose the latest (unless you want to revert to an older version).

Once the backup has been selected, the files that the backup archive contains will be displayed. Each directory will have a small triangle to its left, which you can click to expand the directory and show its contents.

After you've found the file(s) or directories you want to restore, highlight them, and then click the Restore button. But beware, because this will rewrite the files and directories to their original locations—files or directories already there with matching filenames will be overwritten!

If you want to restore the files to a different location, click the Restore As button, and then choose a folder.

---

**Caution** Restored files and directories are owned by root. This is because Simple Backup runs with superuser powers. Therefore, one of the first things you'll have to do is use the `chown` command, preceded by `sudo`, to change the ownership and group of the file to what they were originally. See Chapter 14 for more details about file ownership and how the `chown` command works.

---

## Making Backups from the Command Line

Although Simple Backup allows the uninitiated to make quick backups, the `tar` program is preferred by Linux old-timers. This creates `.tar` files and is one of the original carry-overs from Unix. `tar` stands for Tape ARchive and refers to backing up data to a magnetic tape backup device. Although `tar` files are designed for backup, they've also become a standard method of transferring files across the Internet, particularly with regard to source files or other installation programs.

A `tar` file is simply a collection of files bundled into one. By default, the `tar` file isn't compressed, although additional software can be used to compress it. `tar` files aren't very sophisticated compared to modern archive file formats. They're not encrypted, for example, but this can also be one of their advantages.

---

**Note** Linux comes with a couple more backup commands, which you might choose to use. They are `cpio` and `pax`. Both aim to improve on `tar` in various ways, but neither is broadly supported at the moment. `cpio` is installed by default under Ubuntu, and `pax` can be found via the Synaptic Package Manager. Examine their man pages for more details.

---

### Creating tar Files

Perhaps unsurprisingly, `tar` files are created at the console using the `tar` command. Usually, all that's needed is to specify a source directory and a filename, which can be done like so:

```
tar -cf mybackup.tar /home/keir/
```

This will create a backup called `mybackup.tar` based on the contents of the `/home/keir/` directory. The `tar` command is automatically recursive so, in this example, it will delve into all subdirectories beneath `/home/keir`. The `-c` command option tells `tar` you're going to create an archive, and the `-f` option indicates that the filename for the archive will immediately follow. If you don't use the `-f` option, `tar` will send its output to standard output, which means that it will display the contents of the archive on the screen.



If you typed in a command like the preceding example, you would see this message:

---

```
Removing leading '/' from member names.
```

---

This means that the folders and files added to the archive will all have the initial forward slash removed from their paths. So, rather than store a file in the archive as this:

```
/home/keir/Mail/file1
```

the file will be stored as follows:

```
home/keir/Mail/file1
```

The difference between the two forms concerns us when the files are extracted from the archive. If the files had the initial slash, tar would restore this particular file to `/home/keir/Mail/file1`. If there were already a file of that name in that location, it would be overwritten. With the leading slash removed, tar will create a new directory wherever you choose to restore the archive. In this example, it will create a new directory called `home`, and then a directory called `keir` within that, and so on.

Because of the potential of accidentally overwriting data by specifying absolute paths in this way, a better way of backing up a directory is simply to change into its parent and specify it without a full path:

```
cd /home/  
tar -cf mybackup.tar keir
```

When this particular archive is restored, it will simply create a new folder called `keir` wherever it's restored.

## Compressing tar Archives

You can also compress the archive from within tar, although it actually calls in outside help from either `bzip2` or `gzip`, depending on which you specify. In theory, `bzip2` should achieve a better compression ratio than `gzip`, but the difference isn't always significant.

To create a tar archive compressed using `bzip2`, the following should do the trick:

```
tar -cjf mybackup.tar.bz2 keir
```

This will create a compressed backup from the directory `keir`. The `-j` command option passes the output from tar to the `bzip2` program, although this is done in the background. Notice the change in the backup filename extension to indicate that this is a `bzip2` compressed archive.

The following command will create an archive compressed with the older gzip compression:

```
tar -czf mybackup.tar.gz keir
```

This uses the `-z` command option to pass the output to gzip. This time, the filename shows it's a gzip compressed archive, so you can correctly identify it in the future.

## Extracting Files from a tar Archive

Extracting files using tar is as easy as creating them:

```
tar -xf mybackup.tar
```

The `-x` option tells tar to extract the files from the `mybackup.tar` archive.

Extracting compressed archives is simply a matter of adding the `-j` or `-z` option to the `-x` option:

```
tar -xjf mybackup.tar.bz2
```

---

**Note** Technically speaking, tar doesn't require the preceding hyphen before its command options. However, it's a good idea to use the hyphen anyway, so you won't forget to use it with other commands in the future.

---

## Viewing tar Archive Information

To view the contents of a tar archive without actually restoring the files, use the `-t` option:

```
tar -tf mybackup.tar | less
```

This example adds a pipe into less at the end, because the listing of files probably will be large and scroll off the screen. Just add the `-j` or `-z` option if the tar archive is also compressed.

In addition, you can add the `-v` (for verbose) option to all stages of making, extracting, and viewing an archive to see more information (chiefly the files that are being archived or extracted).

Typing `-vv` provides even more information:

```
tar -cvvf mybackup.tar keir
```

This will create an archive and also show a complete directory listing as the files and folders are added, including permissions.

## Saving the File to a CD-R/RW

Once the tar file has been created, the problem of where to store it arises. As we mentioned earlier, storing backup data on the same hard disk as the data it was created to back up is foolish, since any problem that might affect the hard disk might also affect the archive. You could end up losing both sets of data!

If the archive is less than 700MB, it should be possible to store it on a CD-R or CD-RW. To do this from the command line, first the file must be turned into an ISO image, and then it must be burned.

---

**Note** Remember that the Ubuntu desktop has a very capable CD/DVD burning tool that can save the hassle of working at the command-line prompt. To access it, click Places ► CD/DVD Creator. Simply drag the backup file(s) onto the Nautilus window, and then click the Write to Disc button.

---

To turn a backup archive into an ISO image, use the `genisoimage` command:

```
genisoimage -o backup.iso mybackup.tar.bz2
```

You can then burn the ISO image to a CD by using the `wodim` command. Before using this command, you must determine which device name your CD writer uses.

Typically, the device name `/dev/cdrom` is already associated with your CD drive. This name is just a link to the actual location of the drive. You can run Device Manager (see Chapter 8 for more information about Device Manager) and locate your CD writer in the hardware list to find its precise device name. In the example in Figure 31-4, the precise device name of the CD writer is `/dev/hdb`. Therefore, either `/dev/cdrom` or `/dev/hdb` will do the trick, as long as there is only one CD drive in the machine. If you have more than one CD drive, you should use the precise device name to specify the correct CD writer.

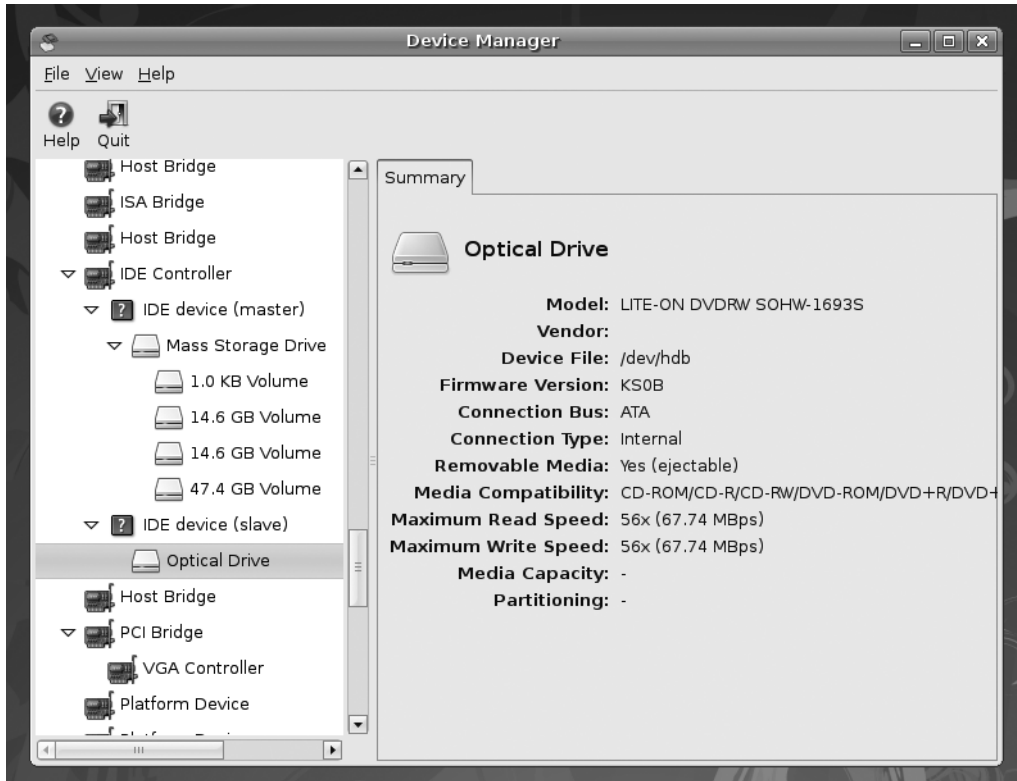
Specify this device name using the `dev=` command option. You must also set the speed at which the burn is to take place using the `speed=` option. This is the writing speed of the drive, which is usually mentioned in the drive's documentation. Put together, to burn the backup image, enter a command in this format:

```
wodim dev=<device name> speed=<speed of your drive> mybackup.iso
```

On a test system, this took the following form:

```
wodim dev=/dev/hdb speed=24 mybackup.iso
```

You'll need to eject and then reinsert the newly burned disc before you can examine its contents.



**Figure 31-4.** Use the Device Manager to find the precise device name of your CD writer.

## Saving the File to a DVD

If the backup file is larger than 700MB, you might want to burn the backup file to DVD, using a DVD-R/RW drive. You can use the `growisofs` command for this. You don't need to create an ISO file for this command, because one is created automatically in the background. Instead, you simply specify the file(s) you want to back up.

Apart from this difference, `growisofs` works along the same lines as `wodim`, and the following command should do the trick:

```
growisofs -Z /dev/dvd -R -J mybackup.tar.bz2
```

Typically, the device name for the DVD writer is linked from `/dev/dvd`, but you can use the Device Manager to get the precise device name.

If the DVD disc has any space left, you can simply add your next backup to the disc, alongside the older file. To do this, use the `-M` command option, instead of `-Z`:

```
growisofs -M /dev/dvd -R -J mybackup2.tar.bz2
```

In both cases, you'll need to eject the DVD and then reinsert it in order to read its contents.

---

**Note** Remember that you can use the `man` command to learn about any commands, including those detailed here. In the case of `wodim` and `growisofs` in particular, this is worthwhile, because they are very powerful pieces of software.

---

## Summary

In this chapter, we looked at making backups. First, you saw where in the Linux file system your personal files and other vital data are stored. Then we looked at how the Simple Backup tool can be used to back up system configuration and personal data. You next learned how to use `tar`, `bzip2`, and `gzip` at the command line to back up any kind of data. Finally, you learned how to burn CDs and DVDs of your backup files at the command line.

In the next chapter, we'll look at how tasks can be scheduled to occur at various times under Ubuntu.