

Optimizing Your System

part



in this part

Chapter 1
Hacking the Installation

Chapter 2
Making Ubuntu Usable

Chapter 3
Configuring Devices

COPYRIGHTED MATERIAL

Hacking the Installation

This chapter explores options for installing and configuring devices. Where you choose to install Ubuntu, which variation you install, and what options you select will impact the system usability.

Before You Begin

Before you install the operating system, be sure to create a backup of anything you want to keep. Copy all data off of the system. You can save it to a CD-ROM, copy it to a spare computer, or physically change hard drives—the method does not matter. Do not keep sensitive data on the same system, even if it is kept on a different hard drive or in a separate partition. If you accidentally format or repartition a working hard drive that contains data you wanted to keep, then the data will be gone.



Warning This chapter deals with drive partitioning, formatting, and installing operating systems. If you play with a working system, there is a serious risk of accidentally deleting your working configuration.

Drive device identifiers can be confusing—the label `/dev/sda1` looks a lot like `/dev/sda2` and `/dev/hda1`. Before every partition, format, and copy, be sure to *triple-check* the device identifier! When you make a mistake, there will be no going back.

Selecting a Distribution

Ubuntu is a Linux distribution based on Debian Linux. Different Linux distributions target different functional niches. The goal of Ubuntu is to bring Linux into the desktop workspace. To do this, it needs to provide a stable user interface, plenty of office tools, drivers for a myriad of peripherals, and still be user-friendly. Although different groups manage nearly every open source project, Canonical Ltd. provides a central point for development and support. Canonical, along with the Ubuntu community, can answer most of your technical (and not so technical) questions.

chapter

1

in this chapter

- Before you begin
- Selecting a distribution
- Installing the server or workstation
- Using a USB drive
- Upgrading Ubuntu

Which Distribution Is Right for You?

Different Linux distributions fill specific needs. For example, although RedHat started life as a unifying distribution, it primarily supported English applications. SuSE was a popular internationalized distribution. Many distributions were maintained by modifying other distributions. For example, Kheops is a French version of RedHat, and the Beowulf clustered computing environment is based on RedHat.

Although RedHat has seeded many different distributions, it is not alone. Debian Linux is another distribution with a significant following. As with RedHat, Debian has been used to spawn many different niche distributions. Although Ubuntu is based on Debian, it is also seeding other distributions.

Different distributions of the Linux operating system are sometimes called *flavors*. There are over 370 different supported flavors of Linux, each with a different focus. You can see the listing of official distributions at www.linux.org.

Ubuntu is the basis for a variety of Linux distributions—most only differ in the user interface, although some do include specific software configurations. The basic Ubuntu distribution uses the Gnome desktop and is geared toward desktop or server systems. Other distributions based on Ubuntu include:

- **Kubuntu**—a variation of Ubuntu with the K Desktop Environment (KDE).
- **Xubuntu**—a variation of Ubuntu with the Xfce Desktop Environment.
- **Edubuntu**—a modified version of Ubuntu that is loaded with educational applications.
- **nUbuntu**—a modified version of Ubuntu with a security-testing focus.

In all cases, it is possible to switch from one installed version to another. For example, you can install Ubuntu, add in KDE, and remove Gnome and you'll have an environment that looks like Kubuntu. To convert an Ubuntu installation to Kubuntu requires changing the desktop, office applications (OpenOffice to KOffice), and swapping other tools. Instead of modifying one distribution to look like another, you should just start with the right distribution.

Note



Changing desktop environments is just the beginning. You will need to replace every Gnome application with its KDE equivalent.

To give you an example of the complexity, here's how to change from Gnome to KDE:

1. Install KDE.

```
sudo apt-get install kubuntu-desktop kde-core
```

This requires about 360 MB of disk space. The installation will ask if you want Gnome (gdm) or KDE (kdm) as the default desktop.

2. Log out. This gets you out of the active Gnome desktop.
3. On the login page, click Options (bottom left corner).
4. Select the Sessions menu item.
5. Select KDE from the Sessions menu and use Change Session to accept it (see Figure 1-1).
6. Log in using KDE.
7. If you no longer need Gnome, you have the option to remove it by removing every Gnome package on the system.

```
dpkg --get-selections '*gnome*' | awk '{print $1}' | \
xargs sudo apt-get remove
```

Tip

Many Gnome applications only need the Gnome libraries to run. If you keep both desktops on the same system, then you can use many of the applications under the same desktop.



FIGURE 1-1: The login menu after adding KDE to Ubuntu

Installing the Server or Workstation

All Ubuntu versions are designed to require only one CD-ROM for installing the system. This reduces the need for swapping disks during the installation. Unfortunately, one CD-ROM cannot hold everything needed for a complete environment. To resolve this issue, Dapper Drake 6.06 LTS has three different types of initial install CD-ROMs that address different system needs.

- **Desktop**—The Desktop ISO-9660 (ISO) image provides a Live CD. This can be used to test-drive the operating system or install a desktop or workstation system. The installation includes the Gnome graphical environment and user-oriented tools, including office applications, multimedia players, and games.
- **Alternate**—Similar to the Desktop image, this image installs the desktop version of Ubuntu, but it does not use a graphical installer. This is a very desirable option when the graphics or mouse does not work correctly in the Desktop CD-ROM. This ISO is intended for use when there are additional configuration steps required beyond the default install.
- **Server**—This minimal install image has no graphical desktop. It is ideal for servers and headless (without monitor) systems. The ISO image includes server software such as a Secure Shell server, web server, and mail server, but none are installed by default (see Chapter 12).



Note

The names for the CD-ROMs do not exactly match the functionality. The names were chosen to avoid confusion with previous Ubuntu releases. (If they called the Desktop CD-ROM *Install*, people might not realize it was also a live CD.) Better names might be Live CD with Desktop Install OEM and Text Desktops, and Server and Minimal System Configuration. But then again, these are pretty long names, so we'll stick with Desktop, Alternate, and Server.

In addition to these three CD-ROM images, an Ubuntu DVD image is available. The DVD ISO contains everything found on all three of the CD-ROM images, including the Live operating system.

The different CD-ROMs are used to install different types of systems. Table 1-1 lists the different options available.

Table 1-1 CD-ROM Options for Ubuntu Dapper Drake 6.06 LTS

<i>Desktop CD-ROM</i>	
Start or install Ubuntu	Start the Live CD and allow graphical install.
Start Ubuntu in safe graphics mode	Start the Live CD with a more standardized graphics setting.
Check CD for defects	Analyze the CD-ROM image to ensure it was created correctly.
Memory test	Test the system's RAM.
Boot from first hard disk	Allows booting the system from the hard drive.

Table 1-1 *Continued*

Alternate CD-ROM	
Install in text mode	Perform a text mode installation and the installed system does not use the GUI.
Install in OEM mode	Text mode installation with the user account <i>oem</i> and includes the GUI.
Install a server	Minimal installation.
Check CD for defects	Analyze the CD-ROM image to ensure it was created correctly.
Rescue a broken system	Mounts the system and creates a command shell.
Memory test	Test the system's RAM.
Boot from first hard disk	Allows booting the system from the hard drive.
Server CD-ROM	
Install to the hard drive	Install a minimal configuration.
Install a LAMP server	Install a server with Linux, Apache, MySQL, PHP, Perl, and Python.
Rescue a broken system	Mounts the system and creates a command shell.
Memory test	Test the system's RAM.
Boot from first-hard disk	Allows booting the system from the hard drive.

Configuring Dual Boot

Dual-boot systems were very popular during the late 1990s and early 2000. Since different operating systems are incompatible, users would boot into the appropriate system to run native applications. Today, dual-boot systems are not as common. Computers are relatively inexpensive so it is easier to have separate Windows and Linux computers, and many options exist for exchanging files and data between systems (see Chapter 6). In addition, virtual machines such as VMware, Qemu, and Xen enable you to run native applications within a window, so there is rarely a need to dual-boot.

Some users still have a need for a dual-boot system. Many games, for example, are more responsive under the native operating system and outside of a virtual machine. If you need a dual-boot system, there are a few configuration steps:

1. Partition the disk for multiple operating systems. The easiest way is to just create one partition that does not use the entire disk. If you have multiple disks, then each disk can contain a different operating system.
2. If you will be using a Windows system, install it on the allocated partition. (You can use the Windows partitioner to create the first partition.) Be sure to install Windows *first* since Windows has a bad habit of disabling boot loaders during its installation.

3. After you have installed the first operating system, use any of the Ubuntu install methods to install Ubuntu.
 - Do *not* select the entire disk for installation (unless you are installing on a separate drive).
 - Use the partitioner to create a new partition for Ubuntu—do not modify the existing partition.

The Ubuntu installer is smart enough to identify other operating systems and add them to the boot menu for dual-booting. This enables you to easily dual-boot between Ubuntu, Windows, BSD, and other operating systems. On PowerPC systems, you can dual-boot between Ubuntu and Mac OS 9 or Mac OS X without a problem.

Tip



Configuring a dual-boot system is relatively easy. However, configuring a multi-boot computer with three or more operating systems can add complexity to the boot menu. I recommend installing Ubuntu last, since its boot manager installation will automatically detect other operating systems and label them properly.

Using the Desktop CD-ROM

The Desktop CD-ROM installation in Dapper Drake is different than previous versions of Ubuntu. By default, the installer starts a graphical, Live CD (see Figure 1-2). This can be used for system recovery, debugging, or browsing the Web.

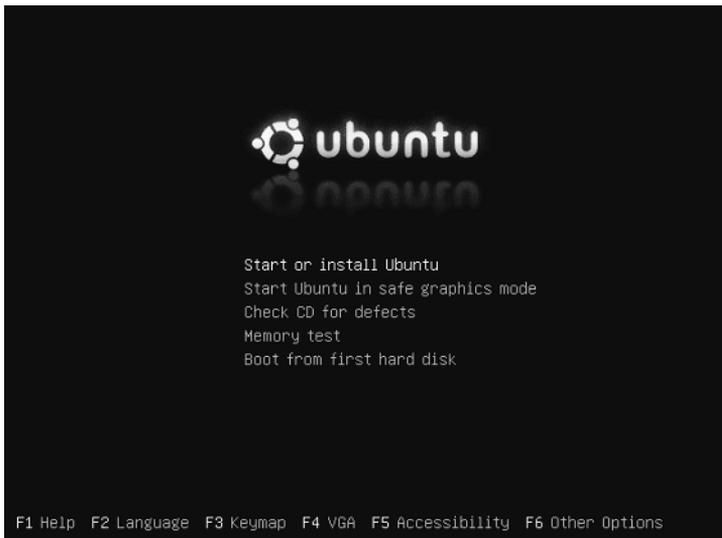


FIGURE 1-2: The Desktop CD-ROM boot selection menu

**Note**

The Desktop CD-ROM boot selection gives you 30 seconds to make a decision before it boots the graphical Live CD. If you want to select a different option, be sure to watch it boot (don't walk away) and make a menu selection. Pressing any key while on the menu will stop the 30-second timer.

When the CD-ROM boots, you will see a graphical desktop. On the desktop is an Install icon that can be used for installing the file system (see Figure 1-3). The same option exists on the menu under System → Administration → Install.

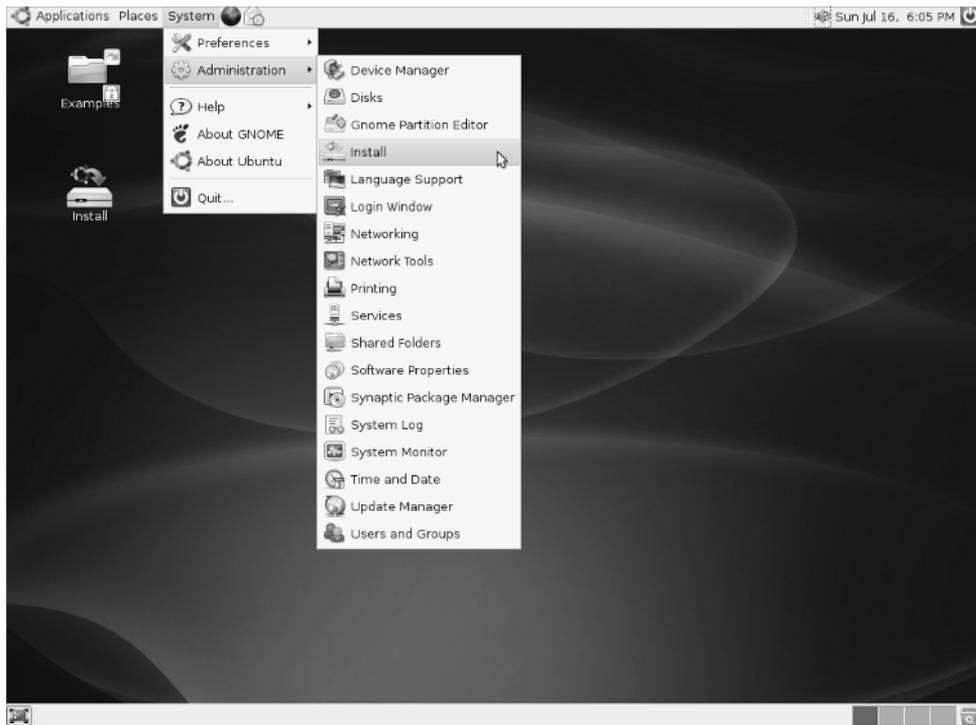


FIGURE 1-3: The Desktop CD-ROM and install options

If something goes wrong during the installation, you only have a few options for debugging the problem. After the graphical desktop appears, you can press `Ctrl+Alt+F1` through `Ctrl+Alt+F4` to provide command line prompts. `Ctrl+Alt+F7` returns to the graphical display. Otherwise, you may want to consider using the Alternate CD-ROM image for installation.

**Note**

Pressing `Ctrl+Alt` with `F1` through `F4` keys takes you out of the graphical desktop. Once out of the graphical mode, you don't need to use `Ctrl`. Simply pressing `Alt+F1` through `Alt+F8` will switch between terminals. This is because `Alt+F1` through `Alt+F12` are keyboard signals used by the desktop; `Ctrl` with `Alt` is used to distinguish between desktop and console requests.

The Need for Speed

The Desktop installation CD-ROM provides a Live CD for exploring the operating system without performing an installation. It can also be used to access an existing system and perform repairs or recovery. However, the Live CD is not the fastest of systems. On a fast computer (for example, 2 GHz with a 40x CD-ROM drive) it can still take three minutes to go from boot to Live CD desktop. This can seem like an eternity if you just need to fix one text file on a critical server.

If you require a Live CD for system repairs, or for using Linux without a hard drive, consider an alternative system. Knoppix, Gnoppix, and DSL (Damn Small Linux) are designed for speed. Each is built for a fast start time when booting from a single CD-ROM.

Sometimes graphics are not even a concern. If you just need a command prompt to repair a system, consider the Ubuntu Server or Alternate CD-ROM images. Both contain a repair option that will allow you to access the local system and make quick fixes. And if you really need a prompt fast, select any of the installation options on the Server or Alternate images and press Alt+F2. This will give you a prompt where you can mount the hard drive and perform repairs quickly.

Using the Alternate CD-ROM

The Alternate CD-ROM image enables you to install a desktop image with graphics disabled, or an OEM-configurable system (see Figure 1-4). It also allows for a server install and can be used to upgrade systems that lack network access.

Text Mode Install

The text mode install and OEM installation both create user workstations, but they have very different configurations. The text mode system lacks the graphical installer, but everything else is present. This is ideal for computers with limited resources or low RAM.

OEM Installation

The OEM installation mode creates the graphical desktop, but creates the user account `oem` with the password `oem`. This account can be used to customize the system. After the install, you can run the `oem-config` script if you need to change any of the original installation responses, or `oem-config-prepare` to remove the temporary `oem` account and enable configuration prompting during the next boot (for end-user configuration). This is an ideal choice for installers who know that they will be customizing the system and removing the installation account before the system is passed to someone else.

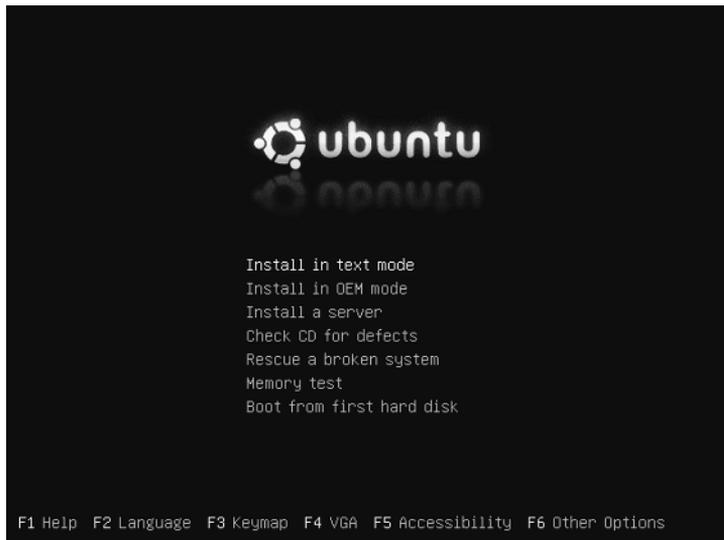


FIGURE 1-4: The Alternate CD-ROM boot selection menu



Warning

During the OEM installation, the Alternate CD-ROM prompts for a password but not a user name. It is not until after the installation completes that you are told the account name for the password is `oem`. Both the account and password are removed when `oem-config-prepare` is used.

Networkless Upgrades and Repairs

The Alternate CD-ROM contains all of the necessary packages for upgrading a Breezy Badger system (Ubuntu 5.x) to Dapper Drake (6.06). This means that the CD-ROM can be used to perform upgrades when network access is unavailable.

Unlike the Desktop CD-ROM, the Alternate CD-ROM cannot be used to run a live graphical system. But, it does have a rescue mode for repairing a non-functioning operating system.

Using the Server CD-ROM

While the Alternate CD-ROM is focused on OEM customizations, the Server CD-ROM focuses on network services (see Figure 1-5). The two main install options are a basic server and a LAMP server. The basic server provides a minimal system image. The LAMP server adds a LAMP application stack to the server image.



Note

LAMP stands for Linux, Apache, MySQL, and PHP/Perl/Python. This is a common configuration for a web server with server-side applications and database support.

Debugging problems with the Server and Alternate installations is much easier than diagnosing the Desktop CD-ROM. At any time during the installation, you can press Alt+F4 and see the current installation progress. If the system hangs, you can tell which subsystem caused the problem. Alt+F2 provides a command prompt, and Alt+F1 returns to the user-friendly installation screen.



FIGURE 1-5: The Server CD-ROM boot selection menu

Using the Smart Boot Manager

One of my computers is so old that it does not support booting from the CD-ROM drive. However, all was not lost! On all the Ubuntu installation CD-ROMs is a small disk image: `install/sbm.bin`. This is the Smart Boot Manager, one of the best-kept secrets on the installation CD-ROMs. This is an image made for a floppy disk. To create the disk, use:

```
dd if=sbm.bin of=/dev/fd0
```

If you boot off of this floppy disk, then you will see a menu that includes booting from the hard drive or CD-ROM. Using this disk, you should be able to boot off any of the installation CD-ROMs. Unfortunately, SBM does not support booting off USB or Firewire devices.

The server installation can also be used to set up a system very quickly. Although the Desktop CD-ROM does allow installing a graphical desktop, the installer is very slow. In contrast, the Server CD-ROM uses a text installer and is very quick. After installing the basic text-mode operating system, you can install additional packages using `apt-get` (see Chapter 4). For example, you may want to install the Gnome desktop using `sudo apt-get install ubuntu-desktop`. This creates the same system as the Desktop CD-ROM, but is much faster than booting the Live Desktop CD and performing the graphical installation. All the necessary files are found on the Server CD-ROM.

Changing Options

The Desktop, Alternate, and Server ISO images are bootable and include a CD-ROM tester (for making sure the CD-ROM was created correctly) and a RAM tester. The System and Alternate CD-ROMs include a recovery shell for debugging an installed system, while the Desktop CD-ROM includes a Live CD system that can also be used for repairing the local host. There is also an option for specifying kernel boot parameters. Some common parameters include:

- Configure a RAM disk. The default is 1 GB for the Desktop ISO and 16 MB for the Alternate and Server ISOs. For example, the Server ISO uses:

```
ramdisk_size=16384
```

- Specify an alternate root disk. The default specifies the RAM disk (`/dev/ram`) but for debugging a system, you can specify a hard drive such as `/dev/hda`.

```
root=/dev/ram
```

```
or
```

```
root=/dev/hda
```

- The Advanced Configuration and Power Interface (ACPI) support on some hardware can cause the installer to fail. ACPI support can be explicitly disabled using `acpi=off`.
- Similar to ACPI, some PCMCIA chipsets (particularly on older motherboards and some Dell systems) can cause the installer's auto detection to hang. The `start_pcmcia=off` boot option disables PCMCIA, allowing you to bypass this type of problem.

At the initial installation menu, you can press F6 to see the current options and make changes. Pressing F1 shows you other kernel options that are common remedies when the system fails to install.

Installing a Minimal System

Sometimes you want to start with a minimal configuration and add packages as needed. This is usually the case for hardware that has limited disk space, little RAM, or a slow CPU. Minimal systems are also desirable for mission critical and Internet accessible servers, where unnecessary applications may consume critical resources or add security risks.

The Server CD-ROM provides the simplest minimal installation option. The basic configuration does not install any additional software packages and uses less than 300 MB of disk space. The Alternate CD-ROM does provide a basic install but does not enable many of the packages—these packages are placed on the system but not turned on. Table 1-1 (shown earlier) shows the available boot-menu options and the type of installation.

In both cases, unnecessary packages can be removed. For example, the Alsa sound driver can be uninstalled, freeing 200 KB of disk space. The command to list all installed packages is:

```
dpkg -l | more
```

If you want to see what files are included in the package, use `dpkg -L packagename`. For example:

```
dpkg -L alsa-base | more
```

Many packages have dependent packages, so removals are not always simple. To identify conflicts before removing a package use:

```
sudo apt-get -s remove alsa-base
```

The `-s` option says to simulate the removal of the `alsa-base` package—it does all of the safety checks and lists all dependencies and conflicts without actually doing the removal. If there are no conflicts, then you can remove the `-s` and perform the actual removal. You can replace `alsa-base` with any of the packages installed on the system. You can also list multiple packages on the `apt-get` command line.

Warning



The `apt-get` program tries to not break dependencies, so removing one package may remove a dependent package. Be careful: if you select the wrong dependent package, you can end up removing critical parts of the operating system. For example, the package `perl-base` cannot be removed without removing the `console-data` package. Removing `perl-base` and `console-data` will also automatically select and remove `cron`, `debconf`, LVM support, `python`, `wget`, and dozens of other system packages. Even if you have no plans to program in Perl, removing it will cripple your system. Use the `-s` option before doing the removal to check if there will be undesirable consequences.

Installing Over the Network

Although installing from a CD-ROM can be convenient, it does not scale well when you need to manage dozens or hundreds of systems. In addition, systems without CD-ROM drives need an option for installing the operating system. Ubuntu provides a bare-minimum boot image for installing Ubuntu over the network. There are different versions of the mini image based on the desired architecture. For example:

```
http://archive.ubuntu.com/ubuntu/dists/dapper/main/installer-i386/current/images/netboot/
```

This directory contains the Dapper Drake mini image for the i386. There are similar directories for the AMD64 (`install-amd64`), PowerPC (`install-powerpc`), and Sparc

(`install-sparc`) platforms. Similarly, there are directories for Breezy, Hoary, and other Ubuntu releases. Each of the directories contains similar pre-configured boot images.

- **boot.img.gz**—A compressed, 8 MB image of a bootable installer.
- **mini.iso**—The `boot.img` file ready for burning to a CD-ROM.
- **netboot.tar.gz**—The `boot.img` contents, ready for installing over the network.
- **pxelinux.0**—The Pre-boot eXecution Environment (PXE) for network installation. This requires a DHCP and TFTP server.
- **pxelinux.cfg**—A directory required for PXE installations.
- **ubuntu-installer**—A directory required for PXE and TFTP installations.

**Note**

If you only have a few systems or are installing on a home system, then you do not need the PXE option—use the `boot.img` or `mini.iso` instead. If you plan to update more than a few dozen systems or are already using DHCP to configure operating systems, then PXE is a good option.

To use the mini images, simply copy the image onto a device. For example, to use an external hard drive for installing Ubuntu, use:

```
zcat boot.img.gz | dd of=/dev/hdb
```

This command uncompresses the image and copies it to the external drive (`/dev/hdb`). This works for most external media.

To boot the network installer from the CD-ROM, just burn the `mini.iso` image to the CD-ROM. This can either be done two ways. From the Ubuntu desktop, you can right-click the ISO and select Write to Disc from the menu (see Figure 1-6), or you can burn it from the command-line using the `cdrecord` command.

```
cdrecord dev=/dev/hdc blank=fast mini.iso
```

The result from all of these different boot options is a disk (or CD-ROM or PXE configuration) that can install Ubuntu over the network.

Installing on a Mac

In general, installing Dapper Drake on a Macintosh, like an iMac or PowerBook, is no more complicated than installing on a PC. In fact, the menus and display look identical. However, there are a few caveats.

With respect to the installation ISO images, don't use the PowerPC Desktop CD-ROM. For the Dapper Drake 6.06 LTS release, the PowerPC Desktop CD-ROM is larger than a CD-ROM. Most CD-ROMs hold 650 MB or 700 MB, but the PowerPC image requires 701 MB. Unless you have a DVD writer or uncommon media such as an 800 MB CD-R, you will not be able to burn this CD-ROM. The 6.06.1 version of the PowerPC Desktop CD-ROM corrected this problem by reducing the ISO to just less than 700 MB. However, I have not been able to boot *any* of my PowerPC systems with this ISO.



FIGURE 1-6: The Write to Disc menu option for ISO images

When installing Ubuntu on a Mac, consider using the Alternate or Server CD-ROM images, or perform a network install (see *Installing Over the Network*). The PowerPC version of the Alternate, Server, and `mini.iso` CD-ROM images are all bootable and functional. To boot off of any of these installation disks:

1. Burn the selected ISO to a CD-ROM. You can do this on a Mac, Windows, or Linux system.
2. Insert the CD-ROM into the Mac.
3. As you power-on the Mac, hold down the C key. This will boot the CD-ROM. Alternately, you can power-on while holding down the Option key. This enables you to select the boot drive, including the CD-ROM.

Although you can install Ubuntu to most hard drives, you cannot install it to a USB drive. Although most of the installation process will work, the installation of the Yaboot boot loader will fail since it does not support booting from USB devices.

Using a USB Drive

The ubiquitous USB flash memory drives (also called *thumb drives*) have effectively replaced floppy disks. They are smaller, less fragile, and store much more data. Because they are convenient, they can be used to kick off an installation, repair a damaged system, run a stand-alone operating system, or simply share files.

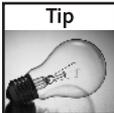
Formatting a USB Drive

USB drives support two basic formats: floppy drive and hard drive. A USB floppy drive consists of one large formatted drive. In contrast USB hard drives contain partition tables and one or more formatted partitions. If you purchased a thumb drive and never formatted it, then it is most likely configured as a USB hard drive with one large partition.

**Warning**

Before formatting or partitioning any device, be sure the device is not mounted! Use the `mount` command (without any parameters) to see if it is mounted, and then use `umount` to unmount any partitions. For example, to unmount `/dev/sdc1` mounted at `/media/usbdrive`, you can use `sudo umount /dev/sdc1` or `sudo umount /media/usbdrive`.

Thumb drives are usually partitioned just like a regular hard drive. Commands such as `fdisk` and `cfdisk` can easily modify the drive partitions, and `mkfs` can be used to format a partition. Besides capacity, speed is a significant difference between thumb drives and hard drives. When you change the partition table on a flash drive or format a partition, wait a few seconds before removing the drive; otherwise, some data may be buffered and not yet transferred.

**Tip**

When writing to a thumb drive, I usually run the `sync` command (`sudo sync`). This flushes all cached data to the disk. When the command returns, it is safe to remove the drive.

When you use the `fdisk` or `cfdisk` command on a thumb drive, you configure it as a USB hard drive. However, you can also configure it as a USB floppy drive. This requires formatting the device *without* partitioning it. For example, to make an ext2-formatted USB floppy drive on my 64 MB USB thumb drive (`/dev/sdc`), I can use:

```
$ sudo mkfs /dev/sdc
mke2fs 1.38 (30-Jun-2005)
/dev/sdc is entire device, not just one partition!
Proceed anyway? (y,n) y
Filesystem label=
OS type: Linux
Block size=1024 (log=0)
Fragment size=1024 (log=0)
16128 inodes, 64512 blocks
3225 blocks (5.00%) reserved for the super user
First data block=1
8 block groups
```

```
8192 blocks per group, 8192 fragments per group
2016 inodes per group
Superblock backups stored on blocks:
    8193, 24577, 40961, 57345
```

```
Writing inode tables: done
Writing superblocks and filesystem accounting information: done
```

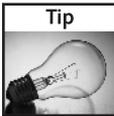
```
This filesystem will be automatically checked every 22 mounts or
180 days, whichever comes first. Use tune2fs -c or -i to override.
$ sudo sync
```

**Warning**

When you first plug in a USB hard drive, all the partitions will appear and automatically mount. However, to create a USB floppy drive, be sure to unmount all partitions and then format the main device (for example, `/dev/sda` or `/dev/sdc`) and not a partition (for example, `/dev/sda2` or `/dev/sdc1`). You will need to disconnect and reconnect the device after you format it in order to remove any stale device partition identifiers.

Sharing Files with a USB Drive

The simplest and most common use for a USB drive is to share files between systems. Dapper supports most USB drives. Simply plugging the drive into the USB port will automatically mount the drive. From there, you can access it as you would access any mounted partition.

**Tip**

Many thumb drives have a light to indicate that the drive is being accessed. Even if the drive is not mounted, do not unplug the drive until the light indicates all activity has stopped.

Linux, Windows, Mac, and most other systems support FAT file systems. In order to share files with other users, consider formatting the drive with `mkdosfs`. For example:

1. Install the `dosfstools` package if `mkdosfs` is not already installed.
2. Unmount the drive (for example, `/dev/sda1`) if it is currently mounted.
3. Format the drive using either FAT16 or FAT32.

```
sudo apt-get install dosfstools
mkdosfs -F 16 /dev/sda1 # format FAT16
mkdosfs -F 32 /dev/sda1 # format FAT32
```

**Tip**

If you want to create a FAT-formatted USB floppy drive, then use the `-I` option. For example:

```
sudo mkdosfs -I -F 32 /dev/sda.
```

If you do not mind restricting file sharing to Linux-only systems, then you can format the drive using an ext2 or ext3 file system using any of the following commands:

```
mkfs /dev/sda1          # default format is ext2
mkfs -t ext2 /dev/sda1 # explicitly format type as ext2
mkfs -t ext3 /dev/sda1 # explicitly format type as ext3
mkfs.ext2 /dev/sda1    # directly call format ext2
mkfs.ext3 /dev/sda1    # directly call format ext3
```

Booting from a USB Drive

Beyond file sharing, USB drives can be used as bootable devices. If your computer supports booting from a USB drive, then this is a great option for developing a portable operating system, emergency recovery disk, or installing the OS on other computers.

Although most systems today support USB drives, the ability to boot from a USB thumb drive is not consistent. Even if you create a bootable USB drive, your BIOS may still prevent you from booting from it. It seems like every computer has a different way to change BIOS settings. Generally, you power on the computer and press a key before the operating system boots. The key may be F1, F2, F10, Del, Esc, or some other key or combination of keys. It all depends on your computer's BIOS. When you get into the BIOS, there is usually a set of menus, including one for the boot order. If you can boot from a USB device, this is where you will set it. However, every computer is different, and you may need to have the USB drive plugged in when you power-on before seeing any options for booting from it.



Warning

Making changes to your BIOS can seriously screw up your computer. Be careful!

Different USB Devices

Even if your computer supports booting from a USB device, it may not support all of the different USB configurations. In general, thumb drives can be configured one of three ways:

- **Small USB floppy drives**—Thumb drives configured as USB floppy devices (that is, no partitions) with a capacity of 256 MB or less are widely supported. If your computer cannot boot this configuration, then the chance of your computer booting any configuration is very slim.
- **Large USB floppy drives**—These are USB floppy devices with capacities greater than 256 MB. My own tests used two different 1 GB thumb drives and a 250 GB USB hard drive.
- **USB hard drives**—In my experience, this is the least-supported bootable configuration. I only have one computer that was able to boot from a partitioned USB hard drive.

Changing between a USB hard drive or USB floppy drive is as simple as formatting the base device or using `fdisk` and formatting a partition. However, converting a large USB floppy device into a small USB floppy device is not direct.

1. Use `dd` to create a file that is as big as the drive you want to create. For example, to create a 32 MB USB drive, start with a 32 MB file:

```
dd if=/dev/zero of=usbfloppy.img bs=32M count=1
```

2. Treat this file as the base device. For example, you can format it and mount it.

```
mkfs usbfloppy.img
sudo mkdir /mnt/usb
sudo mount -o loop usbfloppy.img /mnt/usb
```

3. When you are all done configuring the USB floppy drive image, unmount it and copy it to the real USB device (for example, `/dev/sda`). This will make the real USB device appear as a smaller USB floppy device.

```
sudo umount /mnt/usb
dd if=usbfloppy.img of=/dev/sda
```

The 10-Step Boot Configuration

Creating a bootable USB thumb drive requires 10 basic steps:

1. Unmount the drive. When you plug a USB drive into the computer, Ubuntu immediately mounts it. You need to unmount it before you can partition or format it.
 - a. Use the `mount` command to list the current mount points and identify the USB thumb drive. Be aware that the device name will likely be different for you. In this example, the device is `/dev/sda1` and the drive label is NEAL.

```
$ mount
/dev/hda1 on / type ext3 (rw,errors=remount-ro)
proc on /proc type proc (rw)
/sys on /sys type sysfs (rw)
varrun on /var/run type tmpfs (rw)
varlock on /var/lock type tmpfs (rw)
procbususb on /proc/bus/usb type usbfs (rw)
udev on /dev type tmpfs (rw)
devpts on /dev/pts type devpts (rw,gid=5,mode=620)
devshm on /dev/shm type tmpfs (rw)
lrm on /lib/modules/2.6.15-26-686/volatile type tmpfs (rw)
/dev/sda1 on /media/NEAL type vfat (rw,nodev,quiet,umask=077)
```

- b. Use the `umount` command to free the device:

```
sudo umount /dev/sda1
```

2. Blank, or zero, the USB device. This is needed because previous configurations could leave residues that will interfere with future configurations. The simplest way to zero a device is to use `dd`. Keep in mind, large drives (even 1 GB thumb drives) may take a long time to zero. Fortunately, you usually only need to zero the first few sectors.

```
dd if=/dev/zero of=/dev/sda          # format all of /dev/sda
dd if=/dev/zero of=/dev/sda count=2048 # format the first 2048 sectors
```

Use the `sync` command (`sudo sync`) to make sure all data is written. After zeroing the device, unplug it and plug it back in. This will remove any stale device partitions. Ubuntu will not mount a blank device, but it will create a device handle for it.

3. If you are making a USB hard drive, then partition the device.
4. Format the partitions. If you are making a USB floppy drive, then format the base device. For USB hard drives, format each of the partitions.
5. Mount the partition.
6. Copy files to the partition.
7. Place the kernel and boot files on the partition.
8. Configure the boot menus and options.
9. Use the `sync` command (`sudo sync`) to make sure all data is written and then unmount the partition.
10. Install the boot manager.

Now the device should be bootable. The next few sections show different ways to do these 10 steps.

Kicking Off the Network Install with a USB Drive

USB drives can be used to simplify system installs. For example, if the computer can boot from a USB drive then you can use it to kick off a network install.



Note

The preconfigured network boot image, `boot.img`, is very small—only 8 MB. It should work on all USB drives.

Configuring the thumb drive for use as a network installation system requires some simple steps:

1. Plug in the USB drive. If it mounts, unmount it.
2. Download the boot image. There is a different boot image for every platform. Be sure to retrieve the correct one.

```
wget http://archive.ubuntu.com/ubuntu/dists/\
dapper/main/installer-i386/current/images/netboot/boot.img.gz
```

3. The boot image is pre-configured as a USB floppy drive. Copy the image onto the thumb drive. Be sure to specify the base device (for example, `/dev/sda`) and not any existing partitions (for example, `/dev/sda1`).

```
zcat boot.img.gz > /dev/sda
```

Now you are ready to boot off the thumb drive and the operating system will be installed over the network.

Every PC that I tested with Boot from USB support was able to run the default network installer: `boot.img.gz`. However, since USB support is not consistent, this may not necessarily work on your hardware. If you cannot get it to boot, then make sure your BIOS is configured to boot from the USB drive, that it boots from the USB before booting from other devices, and that the USB drive is connected to the system. If you have multiple USB devices connected, remove all but the bootable thumb drive.

Using the Boot Image with Files

The `boot.img.gz` image is a self-contained file system and only uses 8 MB of disk space. If you have a bigger thumb drive (for example, 64 MB or 512 MB), then you can copy diagnostic tools or other stuff onto the drive.

In order to create a bootable USB drive, you will need a boot loader. The choices are GRUB or SYSLINUX. There are significant tradeoffs here. GRUB is the default boot loader used when Ubuntu is installed. However, GRUB requires knowing the drive identifier (for example, `/dev/sda1`). Since you may plug-in and remove USB devices, the identifier may change, breaking the boot loader's configuration. SYSLINUX does not use a static drive identifier, but is limited to supporting FAT12 or FAT16 drives. Since USB devices are expected to be portable, use SYSLINUX:

```
sudo apt-get install syslinux mtools
```

The main steps require you to format the drive as FAT16 and use `syslinux` to make it bootable.

1. Become root.
2. Unmount the USB drive, if it is already mounted.
3. Format the drive as a FAT16 USB floppy drive (in this example, `/dev/sdc`) and mount it.

```
mkdosfs -I -F 16 /dev/sdc
sync
mkdir /mnt/usb
mount -o loop /dev/sdc /mnt/usb
```

4. Mount the `boot.img` file. You will use this to provide the boot files.

```
zcat boot.img.gz > boot.img
mkdir /mnt/img
mount -o loop boot.img /mnt/img
```

5. Copy the files over to the USB drive. This can take a few minutes.

```
sudo bash # become root, run these commands as root
(cd /mnt/img ; tar -cf - *) | (cd /mnt/usb ; tar -xvf -)
sync
```

6. Set up the files for a bootable disk. This is done by copying over the SYSLINUX configuration files for an ISO image (`isolinux.cfg`) to a configuration file for a FAT16 system (`syslinux.cfg`).

```
mv /mnt/usb/isolinux.cfg /mnt/usb/syslinux.cfg
rm /mnt/usb/isolinux.bin
sync
```

7. Unmount the drive and make it bootable by installing the boot loader.

```
umount /mnt/usb
syslinux /dev/sdc
sync
eject /dev/sdc
exit # leave the root shell
```

Now you can boot from the USB drive in order to install the operating system.

Installing a Full File System

The Holy Grail for USB hacking is the ability to boot a standalone operating system from a thumb drive. It is not fast, it may not have much space, but it sure is cool. Given a large enough USB drive and a computer that can boot from the USB port, you can configure a thumb drive as a standalone operating system.



Warning

There are many different methods discussed in online forums for configuring a USB drive as a bootable system. Unfortunately, most of the instructions are either incomplete or very complicated. Even if you follow these steps exactly, you may still be unable to boot from the USB device. The instructions listed here worked consistently for me, but I spent days trying to make it work.

There are two configurations for making a bootable file system: a huge USB floppy drive or a large USB hard drive. In both of these examples, I will use the Ubuntu Desktop Live CD as the bootable device.

Using the Live CD from a USB Floppy Drive

Converting the Live CD to a bootable USB floppy drive requires at least a 1 GB thumb drive.

1. Become root. This is done for convenience since nearly every command must be done as root.


```
sudo bash
```
2. Unmount and blank the thumb drive. (See the section “The 10-Step Boot Configuration” for directions.)

3. Format the disk as one big FAT16 drive. The `-I` parameter to `mkdosfs` says to format the entire device. In this example, the USB drive is `/dev/sdc`.

```
mkdosfs -I -F 16 /dev/sdc
sync
```

Tip


FAT16 only supports drives up to 2 GB. If you have a larger USB drive, then you will need to use the hack found in the Different USB Devices section to convert a large USB drive into a smaller one.

4. Mount the Live CD and the USB drive.

```
mkdir /mnt/usb
mkdir /mnt/img
mount -o loop ubuntu-6.06-desktop-i386.iso /mnt/img/
mount /dev/sdc /mnt/usb
```

5. Copy over the files. This can take 20 minutes or longer. Go watch TV or have lunch. Also, ignore the errors about symbolic links since FAT16 does not support them.

```
(cd /mnt/img ; tar -cf - *) | (cd /mnt/usb ; tar -xvf -)
sync
```

6. Set up the files for a bootable disk. Since SYSLINUX does not support subdirectories for kernel files, you need to move these to the top directory on the USB drive.

```
# move the kernel files and memory tester
mv /mnt/usb/casper/vmlinuz /mnt/usb/vmlinuz
mv /mnt/usb/casper/initrd.gz /mnt/usb/initrd.gz
mv /mnt/usb/install/mt86plus /mnt/usb/mt86plus
# move boot files to top of the drive
mv /mnt/usb/isolinux/* /mnt/usb/
mv /mnt/usb/isolinux.cfg /mnt/usb/syslinux.cfg
rm /mnt/usb/isolinux.bin
# Optional: Delete Windows tools and ISO files to free space
rm -rf /mnt/usb/start.* /mnt/usb/autorun.inf
rm /mnt/usb/bin /mnt/usb/programs
rm -rf /mnt/usb/isolinux
# All done
sync
```

7. Edit the `/mnt/usb/syslinux.cfg` file and correct the kernel paths. Remove the paths `/casper/` and `/install/` wherever you see them. This is because Step 6 moved the files to the root of the USB drive. There should be eight occurrences of `/casper/` and one for `/install/`. After you write your changes, run `sync`.

8. Unmount the drive and make it bootable.

```
umount /mnt/usb
syslinux /dev/sdc
sync
eject /dev/sdc
exit # leave the root shell
```

This USB thumb drive should now be bootable! You can run the Ubuntu Live operating system or install from this USB thumb drive.

For customization, you can change the boot menu by editing the `/mnt/usb/syslinux.cfg` file and modify the kernels.

Using the Live CD from a USB Hard Drive

Converting the Live CD to a USB hard drive is much more complicated. First, many computers that support booting from USB devices do not support this configuration. Even if the basic configuration is supported, there may be BIOS restrictions on the disk's layout. Second, the boot loader needs to support partitions. Finally, the USB drive's identifier cannot change after installation. This final issue is the main reason that I do not use GRUB or LILO as the boot loader.

Hard drives are defined by a combination of heads, sectors, and cylinders. Although heads and cylinders used to match physical drive heads and platters, this is no longer the case. In general, each sector contains 512 bytes of data, and each cylinder contains one set of heads \times sectors. However, when booting from a USB hard drive, many BIOS manufacturers assume 64 heads and 32 sectors per cylinder. This is the configuration used by USB ZIP drives. If you use a different configuration, it may not boot. In addition, the first partition must not be larger than 1023 cylinders.

Although the `syslinux` command only supports FAT12 and FAT16 file systems, the `syslinux` package includes `extlinux`, which support ext2 and ext3 file systems. For this example, we will use `extlinux` as the boot loader with an ext2 file system on the bootable partition.

There are 10 steps to configure a bootable operating system on the USB drive:

1. Become root.

```
sudo bash
```

2. Unmount the drive.

3. Use `fdisk` or `cfdisk` to partition the drive (for example, `/dev/sdc`). Be sure to specify 64 heads and 32 sectors. The last cylinder of the first partition must not be larger than 1023. If you have additional disk space after allocating the first partition, then you can allocate additional partitions.

```
# fdisk -H 64 -S 32 /dev/sdc
Command (m for help): d
No partition is defined yet!

Command (m for help): n
Command action
   e   extended
   p   primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-983, default 1): 1
Last cylinder or +size or +sizeM or +sizeK (1-983, default 983): 983

Command (m for help): a
```

```

Partition number (1-4): 1

Command (m for help): p

Disk /dev/sdc: 1030 MB, 1030750208 bytes
64 heads, 32 sectors/track, 983 cylinders
Units = cylinders of 2048 * 512 = 1048576 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sdc1    *           1           983     1006576   83  Linux

Command (m for help): w
The partition table has been altered!

```

Warning

The partition must be marked as “active,” otherwise you will not be able to boot from it.



4. Format the partition as an ext2 file system.

```
mkfs /dev/sdc1
```

5. Mount the Live CD and the USB drive.

```

mkdir /mnt/usb
mkdir /mnt/img
mount -o loop ubuntu-6.06-desktop-i386.iso /mnt/img/
mount /dev/sdc /mnt/usb

```

6. Copy over the files. As mentioned in the previous section, this can take 20 minutes or longer.

```
(cd /mnt/img ; tar -cf - *) | (cd /mnt/usb ; tar -xvf -)
sync
```

7. Create the boot files. Unlike `syslinux`, the boot files for `extlinux` can be located in a directory. In this case, we will reuse the `casper` directory since it already contains the kernel files.

```

cp /mnt/usb/isolinux/* /mnt/usb/casper/
rm /mnt/usb/casper/isolinux.bin
mv /mnt/usb/casper/isolinux.cfg /mnt/usb/casper/extlinux.conf
sync

```

Note

The extension for the boot configuration file is `.conf`, and not `.cfg`.



8. *Do not unmount* the drive yet! Making it bootable with `extlinux` requires the mounted directory containing the `extlinux.conf` file.

```

extlinux -z /mnt/usb/casper
sync

```

9. Copy over the boot loader. There is a file missing from the `syslinux` binary package but available in the source package. This file is called `mbr.bin` and is a master boot record containing the boot loader. Download the source package:

```
apt-get source syslinux
```

This creates a directory, such as `syslinux-3.11/`. In this directory is the missing file. Install it on the drive using:

```
cat mbr.bin > /dev/sdc
sync
```



If you are configuring a file image instead of the actual drive, then this `cat` command will truncate your file. Instead, use `dd if=mbr.bin of=usbdrive.img bs=1 notrunc` to install the master boot record to your USB drive image file (in this case, `usbdrive.img`).

10. Now, unmount the drive and boot from it.

If all goes well, you should have a working, bootable USB thumb drive. This drive can also be used as a bare-bones recovery and repair system.

As an alternative configuration, you can format the drive with FAT16 and use `syslinux` to make the partition bootable. In this case, you will also need to copy the boot files to the top of the partition and edit the `syslinux.cfg` file as described in the previous section.

Booting Variations

I used a variety of computers for testing the USB boot process. Every computer acted differently to different boot configurations.

- Every computer with Boot from USB support was able to boot the original `boot.img` file.
- Most computers were able to boot the Ubuntu Live operating system when my 1 GB thumb drive was formatted as a USB floppy drive. However, one computer gave a generic boot error message.
- Only one computer could boot the USB hard drive with the `ext2` file system. Using a real USB hard drive or thumb drive did not make any difference. In addition, specifying the USB ZIP configuration was the only way to make the hard drive configuration work.

Depending on the configuration variation and hardware that you use, you may see some well-known errors.

- **Blank screen**—If all you see is a blank screen with a blinking cursor, then something definitely did not work. This happens when the boot loader fails. It could be due to failing to install the boot loader properly, or it could be a BIOS problem. Try rebuilding the USB drive in case you missed a step. Also, try booting the USB drive on a different computer. If it works on one computer and not on another, then it is a BIOS problem. But if it fails everywhere, then it is probably the boot loader.

- **“PCI: Cannot allocate resource region...”**—This indicates a BIOS problem. You may be able to boot using additional kernel parameters to bypass the PCI errors, for example:
`live noapic nolapic pci=noacpi acpi=off`

However, you may not be able to get past this. Check if there is a BIOS upgrade available for your computer.

- **“cdrom: open failed”**—This error is generated by the `initrd.gz` file because Ubuntu’s installer wants to run from a CD-ROM. You can either choose a different installer image, or edit this `initrd.gz` file to disable the failure. To edit `initrd.gz`:

1. Extract the file. The `initrd.gz` file is actually a compressed archive containing all of the executables, libraries, and configuration files needed during boot.

```
mkdir extract
cd extract
zcat /mnt/usb/casper/initrd.gz | cpio -i
```



Note

If you have ever pressed Alt+F2 from the installation menu and noticed that there was a very minimal operating system with few commands available, this is it. The minimal operating system files are stored in the `initrd.gz` archive.

2. Edit the file `scripts/casper`. In the `mountrout()` function, comment out the following lines:

```
# for i in 0 1 2 3 4 5 6 7 8 9 a b c d e f 10 11 12 13; do
#     live_image=$(find_cd)
#     if [ "${live_image}" ]; then
#         break
#     fi
#     sleep 1
# done
# if [ "$?" -gt 0 ]; then
#     panic "Unable to find a CD-ROM containing a live file system"
# fi
```

3. Right below the section you commented out, insert these lines:

```
mount -t vfat -o ro /dev/sda $mountpoint
live_image=$mountpoint/casper/filesystem.squashfs
```

This example assumes that you are using a USB floppy drive configuration (`/dev/sda`). If you are using a USB hard drive configuration, then replace `/dev/sda` with `/dev/sda1`.



Warning

The device identifier must match the identifier found by the booted thumb drive. This is not necessarily the same device identifier that you are currently using to configure the thumb drive.

4. Repackage the `initrd.gz` file and put it on the USB drive:

```
find . | cpio -o --format='newc' | \
gzip -9 > /mnt/usb/casper/initrd.gz
```

- **Root not found**—There are a variety of errors to indicate that the root partition was not available during boot. This is usually caused when the USB drive is still initializing or transferring data and is not ready for the root partition to be mounted. You can fix this by extracting the `initrd.gz` file and editing the `conf/initramfs.conf` file. Add in a `WAIT` line to delay mounting by 15 seconds, giving the USB time to initialize, configure, and transfer data.

```
WAIT=15
```

Now you can repackage the `initrd.gz` file and boot with it.

The hack for editing the `initrd.gz` file can also be used to add commands to the basic boot image. For example, you can add an editor or diagnostic tools to the `/bin` directory. However, be sure that the commands that you add do not use shared libraries. For example, `file /usr/bin/vim.basic` shows a dynamically linked executable, so you cannot use it unless you also include all of the dependent libraries. In contrast, any executable identified as statically linked is good to go! If you are compiling programs (see Chapter 4 for installing the compilation environment), then you can use `gcc -static` to generate statically linked executables.

Upgrading Ubuntu

People who already use Ubuntu have the option to upgrade rather than reinstall. Ubuntu follows a strict upgrade path between major revisions; you should not just upgrade straight from Hoary to Dapper. The upgrade path follows the major Ubuntu releases:

- Warty Warthog (Warty, version 4) ⇄ Hoary Hedgehog (Hoary, version 5.04)
- Hoary ⇄ Breezy Badger (Breezy, 5.10)
- Breezy ⇄ Dapper Drake (Dapper, 6.06 LTS)
- Dapper ⇄ Edgy Eft (Edgy, 6.10)

Note



Although you could upgrade directly from Warty or Hoary to Dapper, this is likely to cause problems. Each upgrade assumes you are upgrading from the previous version. Skipping a version may break this assumption and cause upgrade problems.

To perform the upgrade:

1. Make sure all packages are currently up-to-date:

```
sudo apt-get update
sudo apt-get upgrade
```

2. Edit the file `/etc/apt/source.list`. This file tells the `apt-get` command where to retrieve the software (see Chapter 4). Replace all instances of the current version with the new version. For example, when upgrading from Breezy to Dapper, change lines such as:

```
deb http://us.archive.ubuntu.com/ubuntu breezy main restricted
deb-src http://us.archive.ubuntu.com/ubuntu breezy main restricted
```

to

```
deb http://us.archive.ubuntu.com/ubuntu dapper main restricted
deb-src http://us.archive.ubuntu.com/ubuntu dapper main restricted
```

3. Perform the upgrade:

```
sudo apt-get dist-upgrade
```



When you start the upgrade, there is no going back. Attempting to stop the upgrade will likely screw up the system, and a power outage during the upgrade can be disastrous. Be sure to make a backup before beginning the upgrade.

Determining the Version

Upgrading gets complicated when Ubuntu users refer to the operating system by names while the operating system reports numeric versions. The question becomes: How can you tell which version of Ubuntu is in use?

One approach is to use the graphical desktop. On the menu bar, System ⇄ About Ubuntu displays the version number and common name. Unfortunately, this is not an option for text-only systems such as the Ubuntu Server. This is also not practical for automated systems.

Another approach is to look at the current `/etc/apt/source.list` file. Assuming nobody has drastically modified the file, the common name for the operating system should be listed on the `deb` installation lines.

A better option is the `lsb_release` command. This command displays distribution specific information from the Linux Standard Base.

```
$ lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:   Ubuntu 6.06 LTS
Release:       6.06
Codename:      dapper
```

Upgrading Issues with Ubuntu

Ubuntu upgrades are not always painless. (I have not yet had a simple upgrade.) Although upgrading from a new Hoary install (with no additions) to Breezy to Dapper works well, you

are unlikely to be running a new installation of Hoary or Breezy. Customizations lead to upgrade complications. For example:

- **Custom system files**—Customizing files, such as `/etc/gdm/gdb.conf` (see Chapter 9), will prompt you to resolve installation conflicts. You can either overwrite or keep the old file, but you cannot merge changes.
- **Proprietary drivers**—Binary and custom drivers, ranging from the Macromedia Flash player to wireless network support, may break. You will need to uninstall and reinstall the software.
- **Shared Libraries**—Different versions of Ubuntu use different linked libraries. For example, Dapper uses newer libraries than Breezy. Code that is compiled for one set of libraries may break under the new system; be prepared to recompile as needed.

The time required to do an upgrade can be another significant issue. An upgrade usually takes at least three times longer than a clean install. This is because the upgrade checks files before modifying the system. While a 2 GHz computer may install in 15 minutes and upgrade in under an hour, a slower computer can take many hours. My 550 MHz iMac upgraded over the network from Breezy to Dapper in just less than 4 hours. The same computer did a network install of Dapper in less than 30 minutes.



Warning Be prepared to devote time to upgrading. Because you may be prompted occasionally to resolve conflicts, you cannot walk away and expect the upgrade to finish without your intervention. If the upgrade takes two hours, you should be near the computer for two hours. After the upgrade has been completed, you may need to spend additional time fixing broken drivers and recompiling software. (Be sure to stock up on coffee and order in for lunch.)

Even though Hoary, Breezy, and Dapper are all versions of Ubuntu, they are all major releases. They should be treated as different operating systems. Just as the upgrade path from Windows 2000 to Windows XP is not recommended, I don't recommend the upgrade path between major Ubuntu revisions. Instead, back up your files, inventory the software you need, and perform a clean install. After the install, restore your personal files and add in your software. This is faster and less painful than debugging Ubuntu after an upgrade failure.

Summary

The initial Ubuntu configuration will determine the ease and flexibility available when modifying the operating system. A right decision at the beginning can make everything else easier. The questions addressed in this chapter include:

- Do you upgrade or reinstall?
- Do you want a desktop, server, or custom installation?
- Should you install from a floppy disk, CD-ROM, USB, or across the network?
- If you upgrade, what are some problems you may run into?

Chapter 2 covers the post-installation environment and discusses things that you might want to change after you first log in. In Chapter 3, you'll learn how to configure the different types of devices and peripherals that you may want to use with your system.