

# Dual-booting Fedora and Windows

Seth Kenlon

For most people, running Fedora on their computer is a satisfying experience and a complete solution for what they need their computer to do. However, if for some reason you are obligated or compelled to have a Windows installed on your computer, you can safely have both Windows and Linux on one machine. They can't run at the same time, but it's entirely possible to reboot your machine and select to boot into either Windows or Fedora Linux. This is called "dual-booting". It's a good technique to learn, since even after you have graduated out of Windows entirely, you may wish to dual boot two or three different Linux distributions on one machine.

The Linux installer is one of the most robust and advanced installers of any Operating System. Its ability to recognize foreign file systems, to consolidate data, and to partition disks without losing data is unparalleled. The Fedora installation program is called Anaconda and manages to transform what could be a confusing series of commands into a user-friendly but powerful graphical interface.

You should, and absolutely must, backup your data before attempting an installation of a new operating system. Certainly, in most cases Linux will install without even touching your data, much less erasing it all, but you will find the whole process much less stressful if you have backed it all up. So, if you haven't backed up your data lately, please step away from this article and do it now.

## The Fedora Installer

If you're booting from a LiveCD, you can click on the *Install to Hard Drive* icon on the desktop. Otherwise, you'll

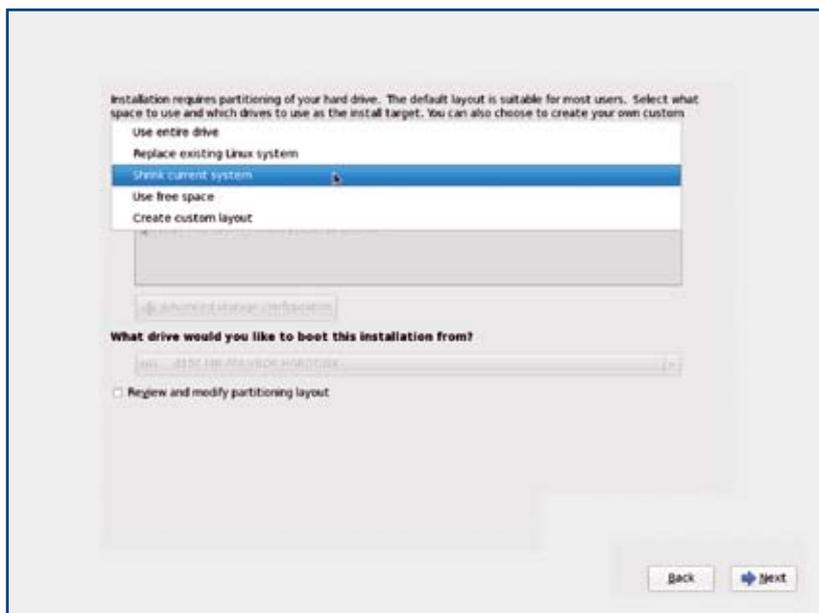
insert the Fedora installation disc and the install process will begin as soon as it boots.

Anaconda, the Fedora installer, will start after an initial option to check the installation disc for errors. If you have never installed Linux before, you may want to check the media for errors; this way if there are any problems while you install, you can at least reasonably rule out a badly burned disc. After you've done a few installs on different machines and feel more familiar with the process, you'll find yourself checking media less (although technically speaking, it never hurts to check).

After this step, an attractive graphical installer screen will appear. Some steps in Anaconda are self-explanatory, like setting your language preference and time zone. The important part is deciding where and how to install Linux.



**Figure 1.** The unglamorous but important media check prior to starting Anaconda



**Figure 2.** Anaconda earns its keep by doing all the hard work for you!

There are two typical scenarios when creating a dual boot system:

- You have one hard drive in your computer, and need to partition it so that your computer believes you actually have two hard drives; one for Linux and one for Windows.
- You have two hard drives in your computer and can use one for Linux and one for Windows (rare in a laptop but a real possibility on a desktop or server).

### Scenario One

In the first instance, Anaconda will most often be able to do the heavy lifting for you. When presented with the partitioning options, simply select *Shrink current system*.

This will literally take all your Windows data, including your system and personal data, and consolidate it into a reasonable space on the hard drive. With the space that is left over, it will do all the laborious work for you, including partitioning, formatting it to a Linux file system,

defining mount points, creating swap space, and installing Fedora.

### Troubleshooting

To be fair, there are obvious variables here. The success of this operation will depend greatly on how much free space you actually have on the hard drive. If you have a 120 GB hard drive in your machine and you are using 118 of it, it is highly doubtful that Anaconda will be able to do much for you. In this case, it will be up to you to clear some data off your drive by perhaps archiving some of it to DVD-R or an external drive. It's some extra work, but be honest – you've probably been meaning to do that anyway!

Another potential cause for failure is a corrupted file system or extreme fragmentation of data on the Windows partition. This is a problem with your Windows installation, so for safety, Anaconda will not attempt to repair such a file system although it will allow you to do so. You'll want to fix this because even if you haven't noticed problems in Windows, this could mean the health of your data is at risk. Of course, you've backed it all up, so there's no need to panic.

Since the problem is with your Windows data, you'll need to do the repair from within Windows itself. Specifically, you can use a command called `chkdsk` to have Windows check and repair itself so that Linux can then safely re-size it without the risk of damaging your data.

To use `chkdsk`, reboot out of Anaconda into Windows. Once Windows has booted and you've logged in, click on the *Start* menu and select *Run*. In the *Run* field, type "cmd" (without the quotes) and hit *Return*. A command line will appear, at which point you should enter:

```
chkdsk c: /f
```

You will need to reboot the machine for that command to actually run. After it has rebooted, you will of course need to reboot again, this time to boot back into the Fedora installer.

Once the drive has been successfully partitioned by Anaconda, the new partition will be prepared for a Li-



**Figure 3.** It may be a command line, but don't let it fool you, it's still just Windows

nux install. You will be given fair warning that the new partition is about to be formatted, and shortly thereafter the partition will have been initialized and ready to proceed.

### Scenario One, Version Two

Some users, in anticipation of an upcoming Linux install, might partition their hard drive in advance. There are many tools for this; there are off-the-shelf solutions like Partition Manager or Partition Magic, or even Disk Utilities (if dual booting with Mac OS), and there are also free tools that any sys admin will know and love, like Gparted LiveCD, and System Rescue. However it happened, if you have a spare partition already on your drive, then you can choose to use the extra partition for your Linux install by simply choosing *Use free space* at the Partition setup screen.

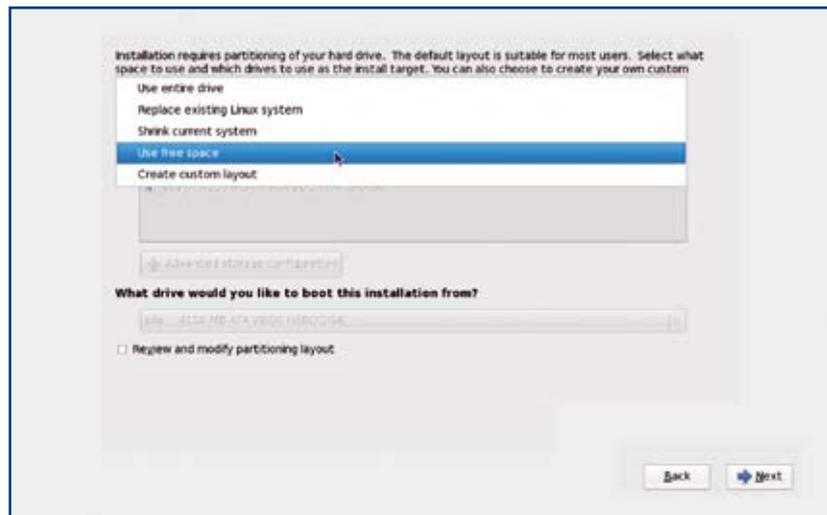
### Scenario Two

In the second scenario, you have a second hard drive in your computer that you can dedicate entirely to Linux. This process is simple. During the Partition setup, choose *Use entire disk* and then select which hard drive you wish to use from the checkbox list on the same screen.

The drives will be listed by their Linux names, such as sda, sdb, sdc, and so on, not by their Windows names. You should be able to tell which drive is which from their size, or their manufacturer, or, if all else fails, then by which bus on your system they are connected; sda would be the first IDE bus, sdb the second, and so on. Select only the drive you wish Linux to use. This will format the entire drive, set up the file system and mount points, and install Fedora.

### The Bootloader

If you were to install Linux now, you'd have a computer with two operating systems on it. However, there would be no way to choose which OS to boot into when you turned on the computer. What your system needs before you commit the install is a menu, shown at boot time, that would allow you to choose whether you'd like to boot into Windows or into Fedora.



**Figure 4.** Using existing free space on a drive will use partitions that you have already created on your hard drive

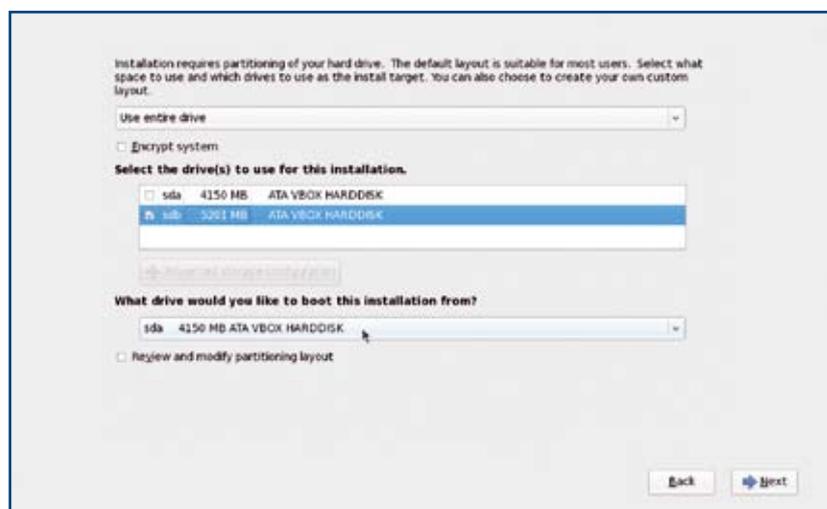
Fortunately, there is just such a menu that will be installed along with Fedora. It is known as GRUB and it is a robust menu that you can configure to suit your needs. Anaconda will automatically detect that you have a Windows partition on your computer, and will let you choose how you want the GRUB menu to appear and what options you are given.

Most of this will be transparent to you but it's good to be aware of what's going on in the background.

The most typical place to install GRUB is the *Master Boot Record* or *MBR*. This is a small blank partition at the very start of your hard drive, reserved for boot loaders and similar boot information. Allowing GRUB to in-

stall here will make choosing which OS to boot into simple and easy for you.

There is rarely any reason not to overwrite your current MBR with the new GRUB menu. Choosing to not install GRUB would make getting your computer to boot into the Fedora partition extremely difficult (in fact, a near impossibility for most users). However, if you installed Fedora on a secondary hard drive in your computer, and your BIOS supports changing the boot sequence, then you could opt to forego the GRUB installation and simply choose between your boot volumes manually. This requires some familiarity and comfort with BIOS manipulation, so is not recommended to the casual computer user. Letting



**Figure 5.** Using an entire hard drive for a nice, big Linux install



**Figure 6.** The GRUB menu allows you to choose what OS you want to boot into

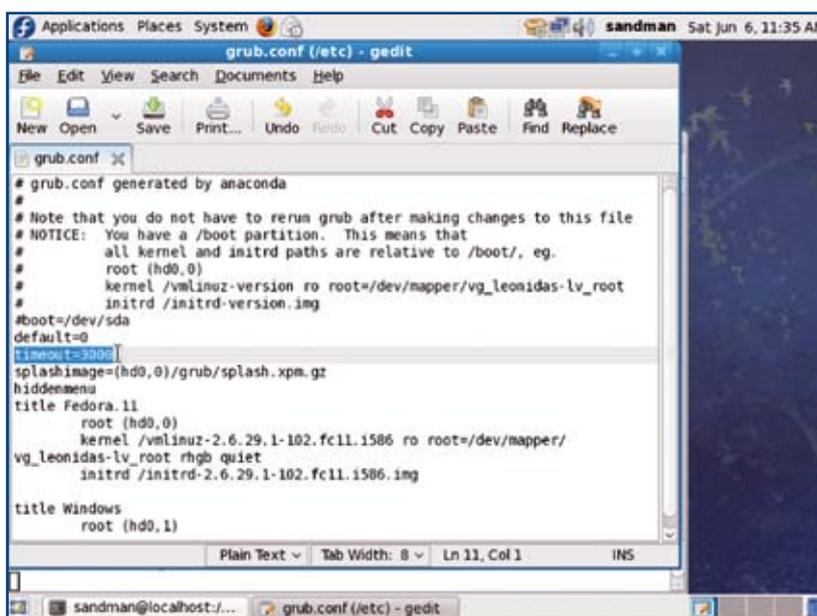
Anaconda do what it does best is generally a good policy.

## Configuring GRUB (optional)

You now have a “dual boot” system; that is, it can boot either into Fedora Linux or Windows. The more you use Linux, the less you will probably find yourself booting into Windows. But for the time being, this is your system, and since Linux is now controlling even the boot loader, you’ll find that a newfound flexibility and con-

figurability starts from the moment you press the power button.

For instance, by default the GRUB menu gives you an attractive and practical menu allowing you to choose whether to boot into Windows or Fedora Linux. If, however, you are booting into the same OS 90% of the time that you turn on your computer, an extra menu to navigate through becomes an extra and laborious step. Since it’s Linux-based, it can, of course, be modified to your liking.



**Figure 7.** Editing your GRUB preferences is easy, and helps you custom design your system

Modifying the GRUB menu must be done from Linux, so the first step is to boot into your Fedora system. Once you’ve booted into Linux, you will be asked to provide user information if this is the first time you’ve booted since installing Fedora. If so, create a user account for yourself, and in no time you’ll be staring at your default GNOME desktop.

Once there, go to *Applications* -> *System Tools* -> *Terminal*. Once your *Terminal* is open, type:

```
su -c `gedit /etc/grub.conf`
```

Enter your root password (which you created during installation) when prompted.

This will open, with administrator (or “root”) privileges, your GRUB configuration file. There are three lines scattered throughout this small text document that are important for basic modification of the GRUB menu, so scan through the document and find them. They are:

```
default= <some number>
timeout= <some number>
hiddenmenu
```

The first, `default`, defines which Operating System will be booted if nothing else is done during the initial boot sequence. Counting always starts at 0 in Linux, so `default=0` refers to whichever OS is on the first partition. If that is Windows, then if you turn the computer on and walk away, Windows will be on your screen when you come back. If Fedora was installed on the second partition, then a `default=1` would make Fedora the default OS.

The second option, `timeout`, means that the GRUB menu will be on screen for a certain number of seconds before it continues to boot to the default OS. To stop it from booting as usual, you simply hit any key while GRUB is on your screen during boot up. After that, the countdown stops and you can deliberate over which OS you really want to boot into, or leave GRUB on your screen all day, or quickly boot into one or the other – whatever you choose. You can give this timeout more or less of a delay, too. If

you prefer to have GRUB wait a whole 15 seconds or 3 minutes, simply input that time, in seconds. If you want it to wait only three seconds (anything less gets pretty tricky to catch, but it's up to you), it will do that, too. All you need to do is type in how many seconds you want GRUB to wait, and then save the document. Next time you boot, the change will be implemented.

In some cases, you might not want other users to be disturbed by an unfamiliar GRUB menu at all. This is especially the case if you are sharing a computer with family or friends who are used to turning on the computer and ending up in Windows without having to interact with the computer at all. This is the function of the `hiddenmenu` option a little further down in the `grub.conf` file.

If the word `hiddenmenu` precedes the list of operating systems on the machine in `grub.conf`, then unless you hit a key during GRUB, you won't actually see GRUB at all. You'll only see a subtle message about the system getting ready to boot in a few seconds, but to most users on a dual booting machine, this just looks like part of the normal Windows boot process. They never notice that there's a new line of text for a few seconds (a three second timeout is enough time for you to catch it and boot into Linux, but not enough time for a normal user to notice that anything new).

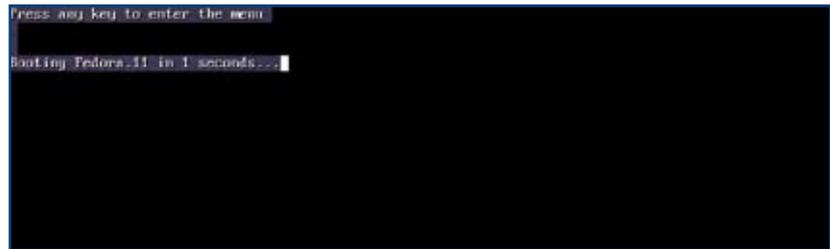
If you decide the `hiddenmenu` option is not something you want, all you need do is put a `#` symbol in front of it, like so:

```
# hiddenmenu
```

This, in Linux code, makes that line a mere comment; a sort of reminder to yourself that the option exists, but that it is not being implemented at the moment. If you do this, and reboot, you'll see that the menu is no longer hidden.

## Integrating the Systems

While you may be able to boot only into one system at a time, that doesn't mean that you'll only be able to access the data on one or the other only by rebooting. With Li-



**Figure 8.** A stealth GRUB menu gives the user the option to boot into a different system, sometimes without them even knowing it

nux, it is possible to mount the partition that contains Windows into any directory, and use much of the data that exists on your Windows drive whilst still in Linux.

The process is simple and may even be somewhat automated for you already. Since Linux can read FAT and NTFS drives (the formats that Windows uses for its drives), you may find that your Windows hard drive appears on your GNOME desktop or in your KDE device notifier as an available volume without you having to do anything at all. To mount the volume, simply double click on the icon and it will mount and open in your file manager. Navigate through the maze that is the Windows method of storing your data, and you can look at your photos or listen to your music or edit documents right off your Windows drive without ever rebooting.

If Fedora didn't automatically detect the availability of your Windows

drive, then all that is required are a few extra steps on your part. It is easiest in the *Terminal*, so open one up via *Applications -> System Tools -> Terminal*. Once that's open, issue this command:

```
su -c 'mkdir /media/windows'
```

This creates an empty folder in your `/media` folder, which is the default location for external drives, CDs, DVDs, and other forms of media, to be mounted.

Next, type:

```
su -c 'mount /dev/sda2 /media/ \
windows'
```

This mounts the partition containing Windows, in the example above it would be `/dev/sda2` but it may be slightly different on your system, into the folder `/media/windows`. To see the folder in your graphical file manager, simply go to the *Places*



**Figure 9.** From within Linux, you can read and write to your Windows data

menu and select *Computer*. That's the very basis of your file system, so navigate from *Computer* to the */media* folder, and then to the */windows* directory. You'll see all the Windows files there and can use whatever is compatible with Linux applications.

If Fedora isn't doing this automatically for you and you find it helpful to have your Windows volume available at all times, you can add it as an entry into your file system table. This is located in a file called *fstab* in the */etc* directory (same place as *grub.conf*; notice that */etc* contains a lot of configuration files so it is a good place to know about).

To add it, open *fstab* as root:

```
su -c 'gedit /etc/fstab'
```

You'll see lines of text like this:

```
tmpfs /dev/shm tmpfs ↵
defaults sysfs /sys ↵
sysfs defaults
proc /proc proc defaults
```

What you'll want to add is:

```
/dev/sda2 /media/windows/ ↵
auto defaults 0 0
```

And then save the file. You probably get the rough idea of what you entered; the location of the volume to be mounted (in this example, */dev/sda2*), the loca-

tion that you will be able to read the volume within your Linux file system (*/media/windows*), the file system type ("auto" lets Linux figure it out), and mount options and frequency of logs. You may need to customize the */dev/sda2* depending on where your Windows volume really is, and you're free to mount it anywhere you like although either */media* or */mnt* are the most often used parent folders for mounting external drives.

Next time you boot into Linux, a volume containing all your Windows data should be mounted and ready to use.

## Creating a Single Boot System

Living on a dual boot system is something most Linux users have to do at some point. At first, Windows will probably be a safety net for you; sometimes work needs to get done fast and you simply won't have the time to learn how to do it on Linux at just that moment. However, if you keep learning new tricks on Linux, you eventually will find that you simply have no further need for Windows.

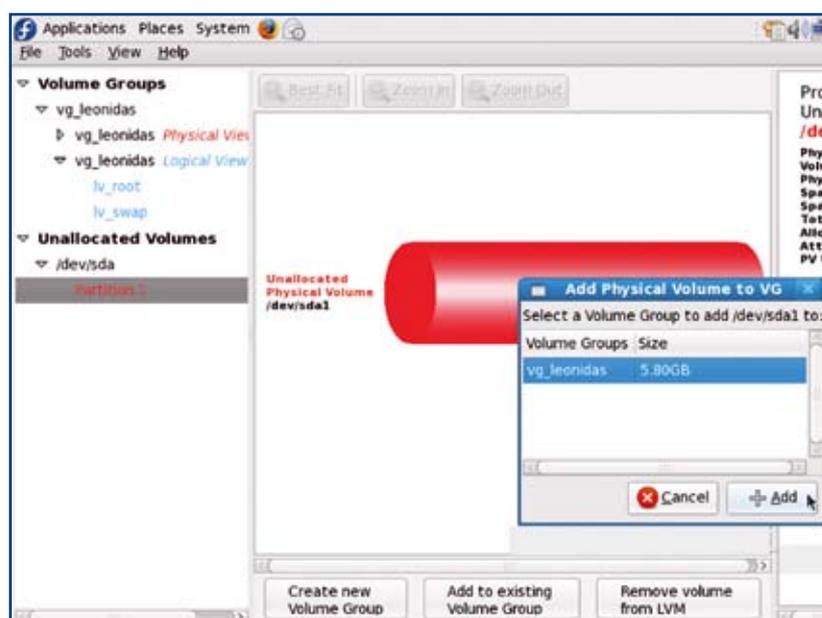
At some point, however, there will come a time when you are ready to reclaim your former Windows partition and use it for Linux. Happily, Fedora uses something called Logical Volumes and makes adding storage space to it quite easy.

To take the space formerly used by your Windows install and add it to your Linux system, go to *System* -> *Administration* -> *Logical Volume Management*. This shows your hard drive partitions as physical entities, that is, an individual partition or a drive. Linux doesn't really care whether its data is stored on one drive or two drives or 100 drives, so physical volumes have little meaning to your Fedora system. What really matters are the logical groupings of your drives. As is, your Windows partition or drive is not a part of the Linux system; Linux is not allowed to borrow space on your Windows drive for spare storage, it is not allowed to touch your data at all (unless you specifically mount the Windows volume yourself, of course). However, if you initialize the Windows drive (thereby erasing Windows and reformatting the volume), and add it to the logical grouping that Linux recognizes as part of its available file tree, then you will have all that extra space for Fedora.

To do this, you first must find out what drive Windows is on. This will be fairly obvious in the *Logical Volume Management* window, since it clearly identifies all the partitions that are used by Linux as part of the logical volume group (although do take care not to erase your GRUB menu in the MBR). You can also type in the word "mount" into a Terminal, and if the Windows partition is mounted then it will show its device label in the output of mount.

Once you've identified the Windows partition, go to *Tools* -> *Initialize Block Device* and type in the path of the partition you wish to initialize, such as */dev/sda2*. The partition will be initialized but not yet added to the volume group; to add it to the volume group, select the partition in the left column and click the *Add To Existing Volume Group* button at the bottom of the window. A new dialog box appears, asking you which Logical Volume Group you wish to add the partition to; you should have only one Logical Volume Group, so select it and click *Add*.

Now you have all that once wasted space as part of your Fedora Linux system, and may safely bid farewell to Windows. ●



**Figure 10.** Adding drives to Fedora's Logical Volume is easy