How-To Geek

How To Harmonize Your Dual-Boot Setup for Windows and Ubuntu

YATRITRIVEDI   🐦 **@yatritrivedi**
UPDATED JUNE 13, 2019, 5:46PM EDT

Looking for some harmony between Windows 7 and Ubuntu in your dual-boot setup? Here are a few ways you can make the tense OS situation a little more unified and copacetic.

Background

When we covered How to Choose a Partition Scheme for Your Linux PC, we noticed that some people were wondering how to use a third partition between Linux and Windows to act as a storage partition.

Why It's Difficult

As a few commenters pointed out, you can't use an NTFS-formatted partition for /home in Linux. That's because NTFS doesn't preserve all of the properties and permissions used by Linux, and Windows doesn't even read Linux file systems. You can readily see this if you view a folder that's hidden in Windows from within Linux, or a file that

Linux sees as hidden in Windows. What works for one doesn't work for the other. Furthermore, there isn't an incredibly clean way to move the Users folder in Windows without messing with things. This is why many people with nicer machines end up using virtualization software; it's easier than forcing the two to co-operate side-by-side.

Image from cellguru.co.cc, assumed fair use

A Work-Around

There isn't a way to run your /home directory from a FAT32 or NTFS partition, so your configuration files and scripts will have to remain there. What you can do is redirect the other commonly used folders like Documents, Downloads, Music, etc. to another partition, one that can be read by Windows. Then, you can add these folders to your Windows 7 Libraries and mark them as the default save location.

This isn't a proper workaround. Your program-associated configuration files and other user-related settings will not be in the same place for this setup. If you have to reinstall either OS, you will have to perform a separate backup of your user settings. That being said, however, most people are really just concerned about their documents, music, videos, and so forth. This solves that issue by pointing both OSs to look in the same place for them.

Linux has come a long way with regards to reading and writing NTFS, and since it's much better than FAT32 and tougher to configure this setup with, that's what we'll be covering in this guide.

Partition Scheme

For this to work, you'll want your hard drive set up in a way similar to this:

- Your Windows partition

- Your Linux partition

- A large partition (or second hard drive!) to store your files

- A small swap partition

For later convenience, when you format your storage partition to NTFS, add an easily recognizable label to it. It'll be easier to find a drive called "storage" or "media" than by counting partition numbers.

Notice that we don't have a separate /home partition this time around. Since the vast majority of your important/large files will be on a separate partition, this negates the need for that. You're welcome to use a separate /home partition to make backing up the Linux-side of things easier, just remember that you can't exceed four primary partitions per disk.

Update: You can use more than four primary partitions if you [use GPT instead of MBR](#).

Auto-Mount Your Storage Partition (Linux)

Since we're using NTFS, it's a good idea to specifically tell your system to mount your storage partition or disk in the same place every time you boot. To do this, we'll be editing the /etc/fstab system file, which is the file system table used by Linux, but first, we have some preparations to make. Open up terminal, and if this makes you nervous, just take a deep breath and relax. It'll be okay.

Prep Work

We need to install ntfs-3g, the driver Linux will use to read and write to NTFS. If you already have it installed, it'll tell you, so don't worry.

```
sudo apt-get install ntfs-3g
```

If you see "ntfs-3g is already the newest version" then you already have it installed, otherwise you'll see it work, so wait for it to finish its thing. Next, let's create the directory where your partition will mount. If you want the drive to appear in the "Places" menu by default, you'll use:

```
sudo mkdir /media/storage
```

If you don't want it to appear in "Places" and you want to manually browse to it for whatever reason, you can use this instead:

```
sudo mkdir /mnt/storage
```

This will create a "storage" directory in /media. You can change this to something else if you like, but be sure it does not have any spaces. Spaces will create a problem when we configure it to automatically mount in the next few steps.

fstab

Now, it's time to edit the fstab file. First, we'll create a backup, just in case anything happens.

```
sudo cp /etc/fstab /etc/fstab.backup
```

It'll prompt you for your password, so go ahead and enter it. If, for whatever reason, you need to restore the backup in the future, you would do this:

```
sudo cp /etc/fstab.backup /etc/fstab
```

Next, you need to find what the UUID of your storage partition is. The UUID stands for "universally unique identifier" and acts as a proper

serial number that will not change until the partition is reformatted.
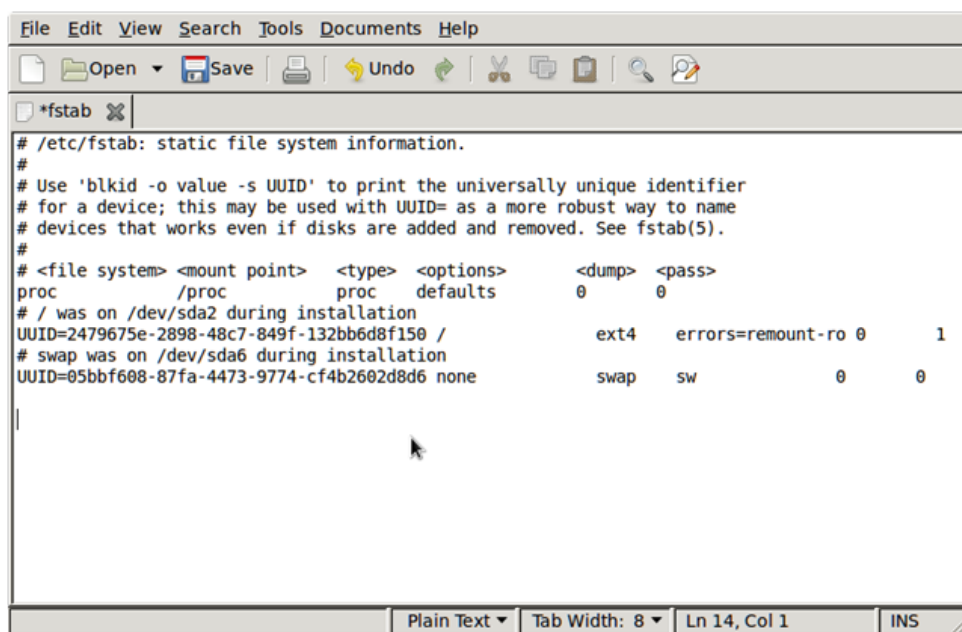Run the following command:

```
sudo blkid
```

Enter your password, and you'll see some output resembling this:

```
/dev/sda1: UUID="23A87DBF64597DF1" TYPE="ntfs"
/dev/sda2: UUID="2479675e-2898-48c7-849f-132bb6d8f150"
TYPE="ext4"
/dev/sda5: UUID="66E53AEC54455DB2" LABEL="storage"
TYPE="ntfs"
/dev/sda6: UUID="05bbf608-87fa-4473-9774-cf4b2602d8d6"
TYPE="swap"
```

Find the line that has the correct label to your storage partition
(makes things easy, doesn't it?) and copy the UUID.

```
gedit admin:///etc/fstab
```

You'll see gedit open, like so:

You may see an uglier theme on gedit than usual, but don't worry it. Add the following lines to the bottom of fstab, substituting your own UUID instead of mine:

```
# storage mount
UUID=66E53AEC54455DB2 /media/storage/   ntfs-3g auto,user,rw 0 0
```

The first line is a comment, indicated by the leading hash tag. The next line tells fstab to look for the partition with the specified UUID, mount it to /media/storage/, and to use the ntfs-3g driver. Furthermore, it makes sure that it automatically mounts at boot, makes it accessible by users (not just root), gives both read and write privileges, and skip file-system checks (you'll probably want to use Windows to do that). Lastly, double-check, and triple-check to make sure you didn't touch anything else, and that the UUID is correct.

```
File  Edit  View  Search  Tools  Documents  Help

  Open  ▾   Save      Undo        X            Q

 *fstab
# /etc/fstab: static file system information.
#
# Use 'blkid -o value -s UUID' to print the universally unique identifier
# for a device; this may be used with UUID= as a more robust way to name
# devices that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point>   <type>  <options>       <dump>  <pass>
proc            /proc           proc    defaults        0       0
# / was on /dev/sda2 during installation
UUID=2479675e-2898-48c7-849f-132bb6d8f150 /       ext4    errors=remount-ro 0       1
# swap was on /dev/sda6 during installation
UUID=05bbf608-87fa-4473-9774-cf4b2602d8d6 none    swap    sw              0       0

# storage mount
UUID=66E53AEC54455DB2 /media/storage/     ntfs-3g         auto,user,rw 0 0|



                          Plain Text ▾ | Tab Width: 8 ▾ | Ln 15, Col 73        INS
```

When you're ready, click save and then reboot. Don't skip the reboot, as it's necessary for the next step as well as to make sure things work.
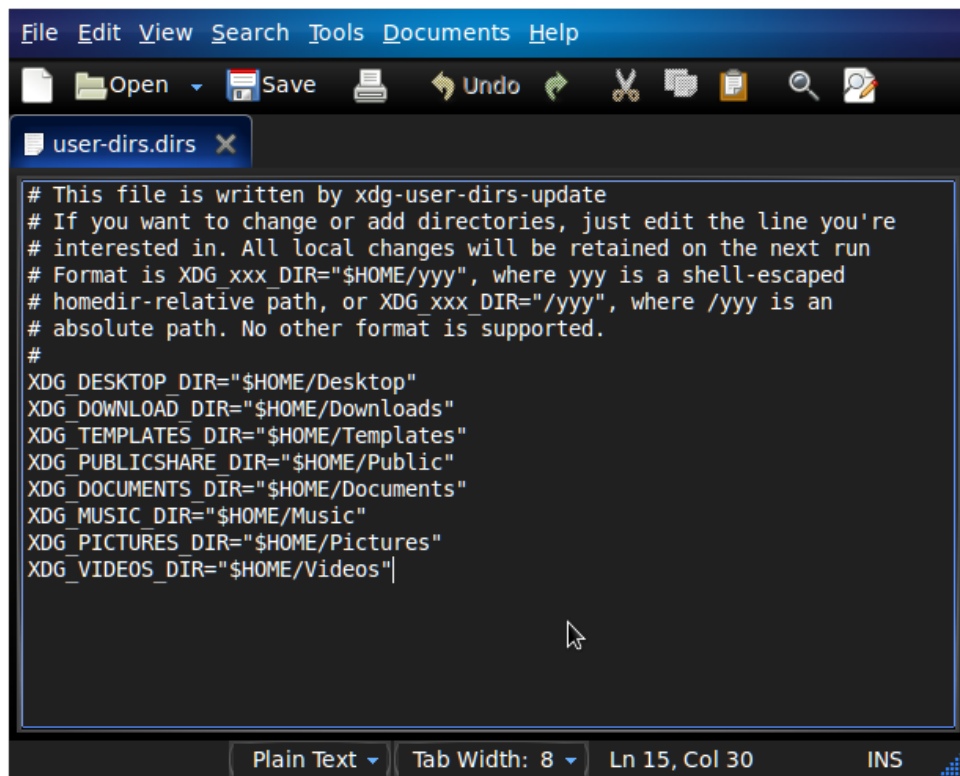
You should be able to boot into Ubuntu as if nothing happened, but you'll notice that you've got "storage" (or whatever you named it) under the Places menu now! If not, check to make sure you got fstab correct. See above to restore fstab from your backup, if you need to.

Configure Your Subfolders (Linux)

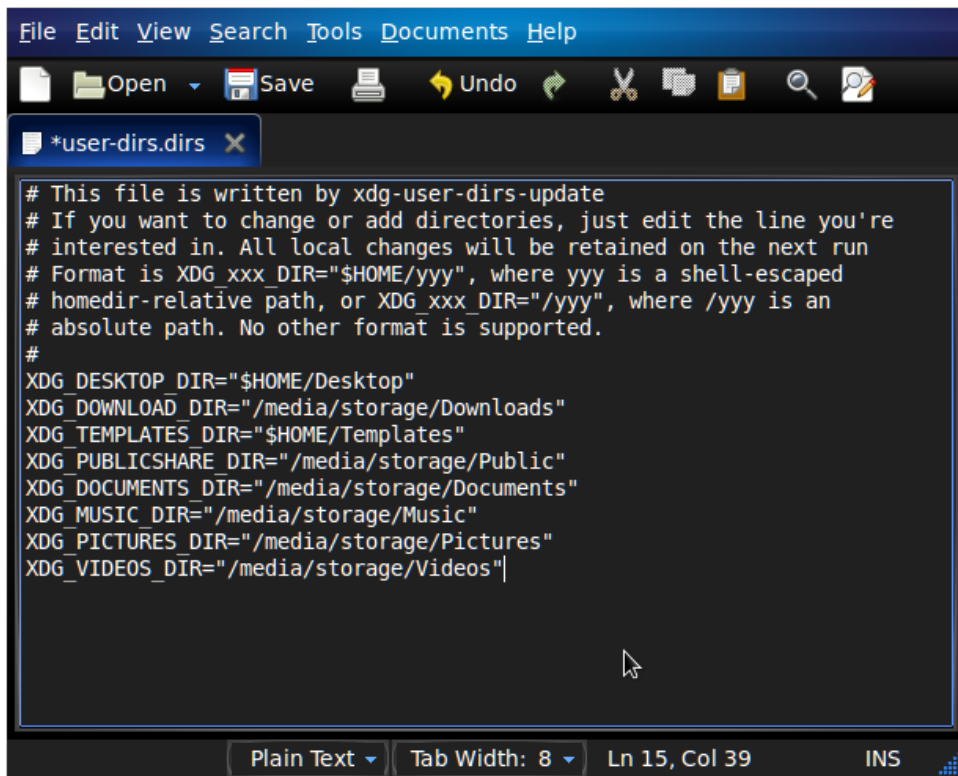Open up terminal and enter the following command:

```
gedit .config/user-dirs.dirs
```

This is the file where your "special" folders in your home directory are defined.

You can edit this to your liking. In place of where you see "$HOME/Downloads" you would put in an absolute folder location, like "/media/storage/Downloads". Go ahead and create those folders, or whatever folders you'd like to call them, and put the path down for each of these. Here's what the finished edit should look like:

Click save, and we're done the crux of the configuration. You may need to reboot for these changes to take effect, but you can just boot into Windows to finish out the process in the next section.

Basically, now when you browse and put files in your "Downloads" folder, they'll actually go to your storage drive's "Downloads" folder. Anything in your home folder itself will stay in /home/yourusername/, not on your storage drive. A few of the folders, like "Desktop" and "Templates," probably won't benefit from this treatment, either. Templates are rarely used, the desktop usually gets cluttered with shortcuts and the like, and the Windows desktop isn't elegantly redirected, unfortunately.

Configure Your Subfolders (Windows)

Boot into Windows, and you'll see that there's another partition called "storage" under "My Computer." Windows 7 has the beautiful Libraries feature built-in, so take a look at our article "Understanding the Libraries Feature in Windows 7," and you'll see step-by-step directions on how to add your new storage folders to your libraries.
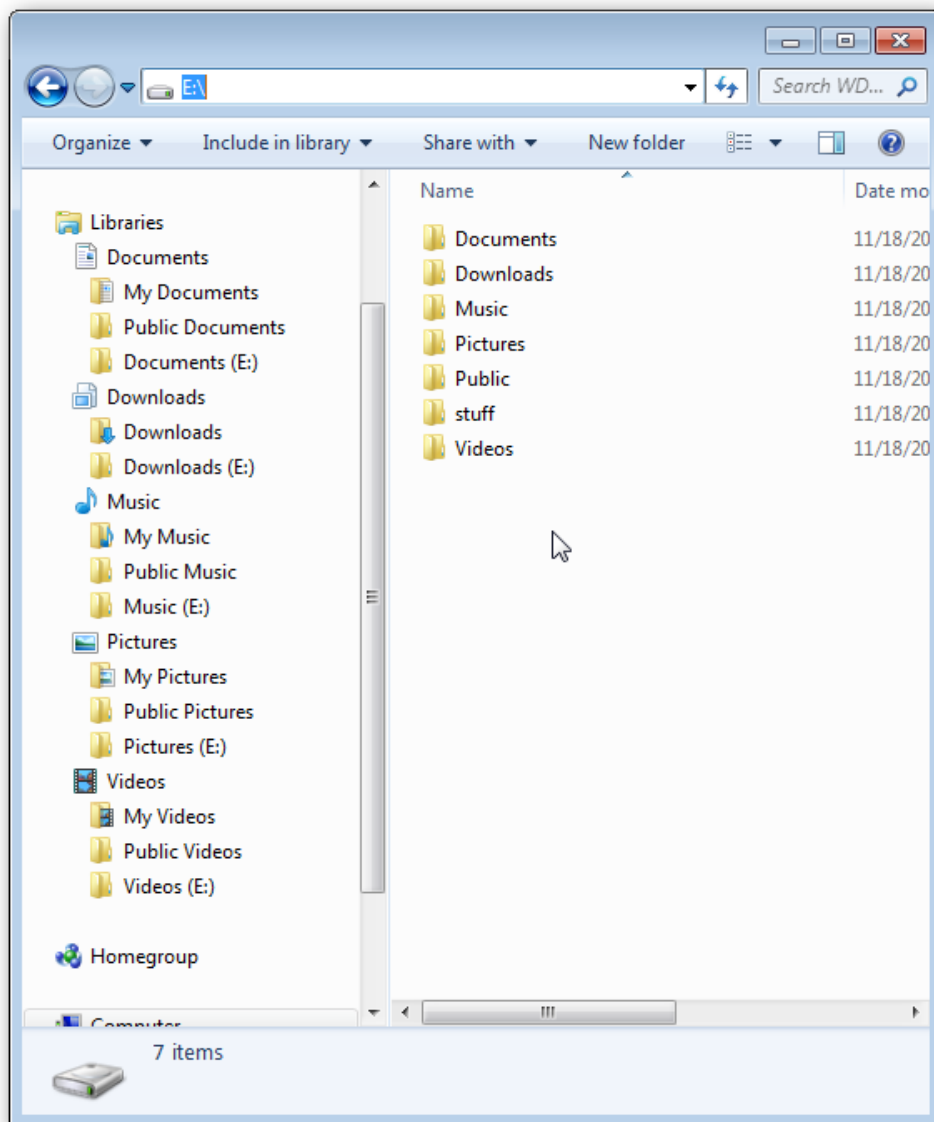
As you can see, my storage drive folders are a part of my libraries. My storage drive letter is E: because my network share is at D:. Also, take a look at our "Change the Default Save Folder for Windows 7 Libraries…" article so that when you stick things in your libraries, they automatically get saved to your new storage folders as well.
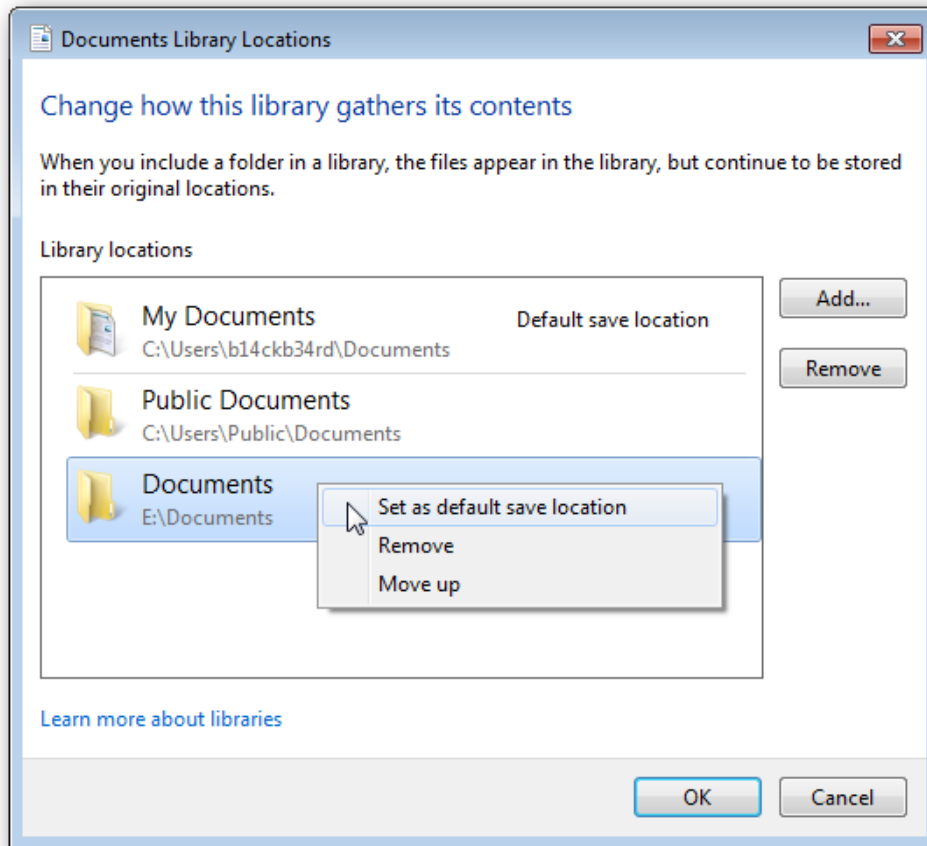
The last thing you'll have to change is the default "Downloads" directory in your preferred web browser(s), which can point to your "Downloads" library. All done!

It's also worth mentioning that if you have some know-how, you could even do this with a remotely shared drive on your network, though it may prove to be too slow for actual use. A better idea is to turn your storage partition into a shared drive that can be accessed by other computers in your network.

While there is no perfect solution to the problem of a unified shared drive for dual-booters, this layout works quite elegantly. Most people mainly care that their downloaded files, documents, and media files can be accessed very easily regardless of which OS they're using, and you'll see that this scheme does that fairly well. If you have any tips, or perhaps a better setup, please share them!

How-To Geek is where you turn when you want experts to explain technology. Since we launched in 2006, our articles have been read more than 1 billion times.