

# Editing Files that Belong to Root

*(Use to copy rxtx files to system folders)*

*Making system-wide changes graphically*

*Most new Linux users just want to be able to click and drag and drop. So one of these commands should be your new best friends...*

*If you're using Ubuntu (Gnome), press Alt-F2 and type*

`gksudo nautilus`

*If you're using Kubuntu (KDE), press Alt-F2 and type*

`kdesu konqueror`

*If you're using Xubuntu (XFCE), press Alt-F2 and type*

`gksudo thunar`

## If you have experience with Linux but are new to Ubuntu

If you're already familiar with Linux and just wonder why Ubuntu/Kubuntu uses `sudo` instead of `root`, this page isn't for you. You're really looking for <https://help.ubuntu.com/community/RootSudo>. It has a full explanation of why Ubuntu uses `sudo` instead of `root` and what the advantages and disadvantages of that choice are.

## If you are new to Linux and Ubuntu

This page is for people who are new to Linux *and* Ubuntu/Kubuntu and want to know what the deal is with file permissions.

## What you may be used to

My guess is you're probably a typical ex-Windows user (as I was), used to running as Administrator and being able to edit pretty much any file you want—even system files, even the registry—at will, very little stopping you... maybe the occasional "read-only" file that you can just make read/write by checking a little box in the Properties dialogue.

You may also never have installed and configured an operating system before (especially if you bought a computer with Windows preinstalled for you).

## The need to edit system files

Some in the Linux camp will say "you shouldn't have to edit such-and-such a file." That's simply not true. Most Linux users aren't just users—they have to install and configure stuff themselves (as Dell or

HP won't do it for them), usually configuration files and system files that normal users don't have to touch (say, in order to adjust the possible screen resolutions or install new software). You *do* have to edit these files from time to time. It's in your best security interests to have to go out of your way to edit them, but you should still be able to occasionally edit, move, copy, and rename files that only an Administrator (or Root) can modify.

## Frequently edited system files

In Linux, the only files a user can modify are the ones in her home directory. For example, if the user's name is Bronte, the files she can modify live in the /home/bronte/ folder. Anything outside the /home/bronte folder she will not be able to modify. Some of the top files new installers need to modify are:

/etc/fstab (making hard drives and partitions viewable as folders)

/boot/grub/menu.lst (controlling the boot menu that lets you choose which operating system to boot to)

/etc/apt/sources.list (the list of sources from which you can easily install software)

/etc/X11/xorg.conf (screen resolution options, among other things)

and all of these files need Root privileges to modify.

## Why do people recommend the command-line?

If you're like a lot of new Linux users, you probably have not yet embraced the command-line. Some people may have told you to "just"

```
sudo rm -r /usr/share/icons/themename
```

as if "rm" really means anything to you yet. The fact of the matter is that people recommend command lines because they're easier instructions to give than "Click on that button that looks like a purple arrow. A new dialogue will pop up with three choices. Pick the left-most choice."

## Making system-wide changes graphically

Most new Linux users just want to be able to click and drag and drop. So one of these commands should be your new best friends...

If you're using **Ubuntu (Gnome)**, press Alt-F2 and type

```
gksudo nautilus
```

If you're using **Kubuntu (KDE)**, press Alt-F2 and type

```
kdesu konqueror
```

If you're using **Xubuntu (XFCE)**, press Alt-F2 and type

```
gksudo thunar
```

## What does *sudo* do?

What do these commands do? They allow you to generally operate as a regular user but have a temporary window *within* your user account to operate as Root (just within the Nautilus or Konqueror window you've opened). Once you close the Root window, it's just as if you're a user again. You don't have to log out of user, log in as Root, make changes, log out of Root, and log back in as user again. You can be Root and user *simultaneously*.

Mac OS X users may be familiar with this security model (Mac uses sudo, believe it or not!). As a Mac user, you operate as a user. When you want to install something, a dialogue box comes up asking for your administrator (sudo/root) password. Once it's installed, you're a user again.

So how does this work? Let's say you want to edit the `/etc/apt/sources.list` file. If you want to do it command-line style, you'd

```
sudo nano /etc/apt/sources.list
```

and then use Nano to edit it. If you do it the graphical way we've been talking about, you'd run the *gksudo nautilus* or *kdesu konqueror* command and a file browser window will pop up (with Root/administrator privileges just in that window). You'd browse to `/etc/apt` and find the file called `sources.list`. You'd click to open it, edit it, close it, and close the browser window, and it would be done.

## Final recommendations

If you find that (especially right after the initial Ubuntu/Kubuntu install) that you're making a lot of changes as Root, create a launcher with the command *gksudo nautilus* or *kdesu konqueror*. Then, when you click on that launcher, you can type in your password, make root changes, and then close the window and be a user again.

**Warning:** Always make backup copies of any configuration files you modify *before* modifying them.