

Fixing gdebi. The Ugly Hack!

One of my favorite software installation tools (when using Aptitude) is a little number known as gdebi. It makes installing applications with a .deb a very painless and rapid process. On top of that, you can later click on the original .deb and use gdebi to uninstall it. Best of all, it'll resolve dependencies when that is a situation it can handle.

You don't have to mess around with anything – just click and install. It's lovely, small, and effective. It's just like a Linux application should be!

This is how the manual describes gdebi:

gdebi lets you install local deb packages resolving and installing its dependencies. apt does the same, but only for remote (http, ftp) located packages. It can also resolve build-depends of debian/control files.

Note the lack of excessive adjectives. It was obviously not written by me. I have oft sung the praises of gdebi and am personally the motivation for hundreds of folks installing it. Seriously... I've told countless people to install gdebi! It's just that awesome. I've probably been using gdebi since I first used a distro with the Aptitude package manager.



See? It's so lovely and simple.

Imagine my dismay when I discovered gdebi was broken!

I'd open gdebi, click on install, gdebi would close without asking me for a password, crash, and not install the software. This is an ugly, ugly hack to fix it. It's so very ugly – but it works.

Crack open your terminal with CTRL + ALT + T and enter the following:

```
[code]sudo nano /usr/share/applications/gdebi.desktop[/code]
```

Find this line:

```
[code]Terminal=false[/code]
```

And change it to:

```
[code]Terminal=true[/code]
```

Then save it. (CTRL + X, Y, and ENTER)

Now for the ugliness. This will fix the problem, but every time you use gdebi a terminal window will open up along with it. Fortunately, the terminal window will close itself after you're done. It's an ugly, ugly hack – but it does work when gdebi closes without installing the application.

As always, thanks for reading. Leave a comment below or look to your right where you can sign up to get notifications when new articles are published. If you're feeling energetic, go ahead and register so that you can write an article or two yourself! If you want to write an article without registering, you can do that too – just click here!

How To: Install 'gedit' With All The Bells and Whistles

There are many text editors out there, but gedit is a perennial favorite. There are also many plugins to extend gedit, and this is an easy way to install a bunch at once.

gedit, the default editor for the Gnome desktop environment, is a part of the GNOME Core Applications and is available in pretty much every distro. It also doesn't actually need a lot of additional dependencies, which helps make it useful for most any popular desktop environment.

You can use gedit for anything, from programming to writing markup for your website. You can use it as a plain text editor and there are many ways to extend it, to add functionality not included by default. There are plugins to highlight syntax, to auto-complete words, to auto-close brackets, etc. You can make it do all sorts of things you'd not expect from a plain text editor.

Given that disk space is absurdly cheap these days, I don't see any reason to not just go ahead and install as many plugins at once as I can. I may not use them all, but I'll use most of them and I can just not enable those that I don't want to use. So, how to install it all at once?

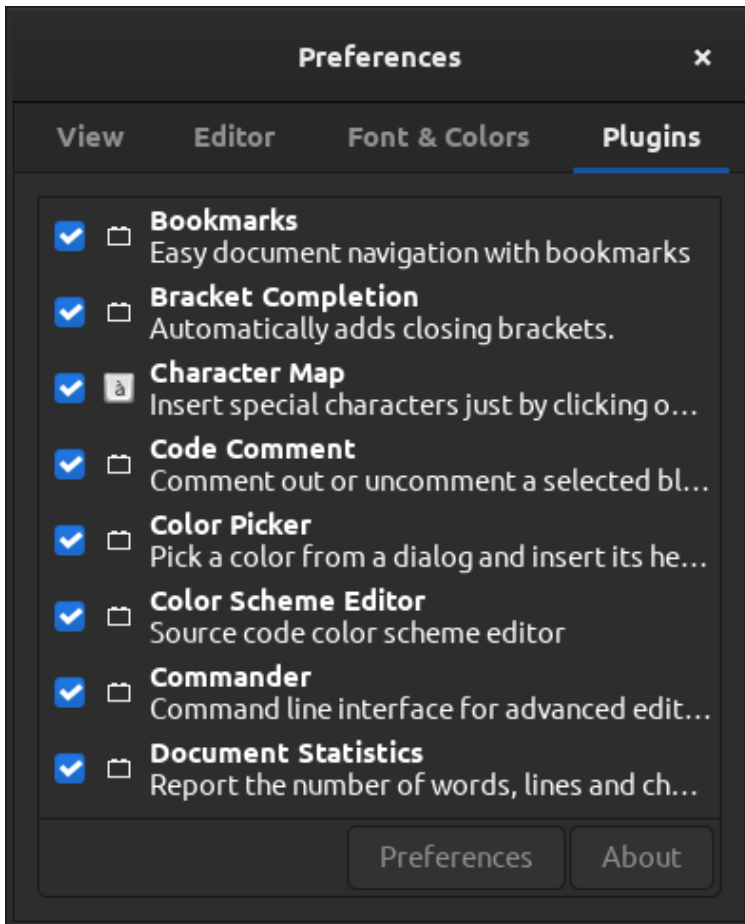
Crack open your terminal with CTRL + ALT + T and enter the following:

```
[code]sudo apt install gedit gedit-plugins-common gedit-plugin-*/code]
```

So, what's going on there? You can string together applications with aptitude and this is installing gedit first, a pack of common plugins, and then every other plugin that uses the 'gedit-plugin-*' format and is in the repositories.

If you don't already know, the asterisk is known as a wildcard. A wildcard basically means, "any character." So, foo* is anything from fool to foolish and foob* is anything from foobar to foob-gibberish183742 or whatever. It's unlike the question mark, which only matches one character, in that it means any and all characters.

It should be noted that this only installs the plugins. You still can't use them until you enable them. To select them, you need to first open 'gedit' (which will almost certainly be called "Text Editor" in your application menu) and click on preferences where you can navigate to the right-most tab and enable them. It looks like this:



Select plugins 'til you're satisfied!

If you did this properly, you should now have a bunch of plugins enabled and not have to hunt them down, one by one, trying to find and install them manually. Is this a bit of overkill? Perhaps, but disk space is cheap and the entire thing takes up less than 19 additional MB on my system.

Like always, thanks for reading. Scroll up and look right. Toss your name and email address in there and you'll get notified of new articles. I promise, I won't spam you. You ain't even gotta use your real name!

How To: Check Your Logs For Errors

When you notice a problem with your Linux box, you're possibly only noticing the symptoms and not actually seeing the reason. The reasons are often made clear in your error logs. This is how to read them.

Back in the early 1900s (when horses, trains, and your feet were the primary mode of travel), it required the terminal to really deal with your error logs. You'd ask for help on a forum and they'd want you to run something like:

```
[code]grep -i "Feb 10" /var/log/*.log | sort | uniq -c | sort -n[/code]
```

It worked and it worked well, especially if you knew what you were looking for. If you weren't, then you'd be scrolling through tons of text in the terminal with few visual cues or no easy way to sort through the data.

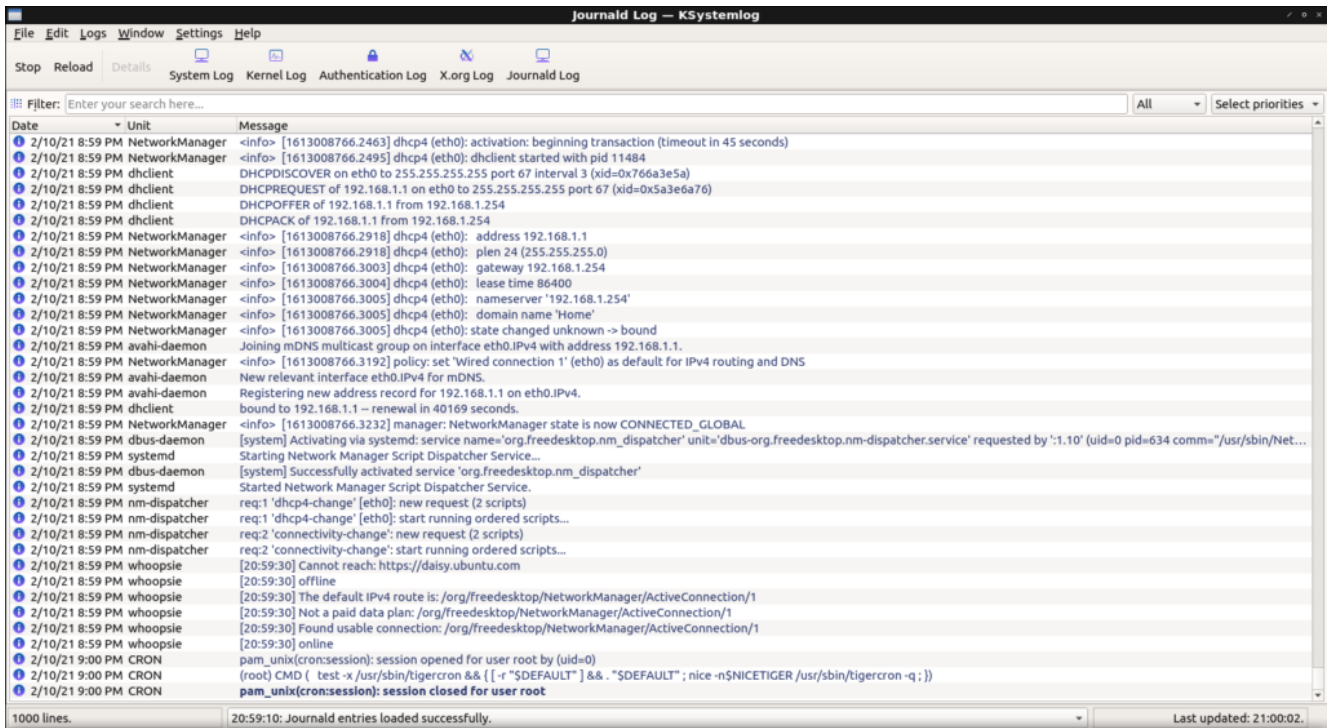
Well, enter KSystemLog. KSystemLog's man page describes it as thus:

Description: system log viewer

KSystemLog show all logs of your system, grouped by General (Default system log, Authentication, Kernel, X.org...), and optional Services (Apache, Cups, etc, ...). It includes many features to read nicely your log files:

- * Colorize log lines depending on their severities*
- * Tabbed view to allow displaying several logs at the same time*
- * Auto display new lines logged*
- * Detailed information for each log lines*

And it looks like this:



KSystemlog UI – not pictured, fancy colors

As you can see from the image, this is a handy GUI method to view your various system logs. If you look at the image above, you'll see that one of the lines is a darker blue than the previous lines. This is because it was a new log entry, an event that happened after I'd opened the application.

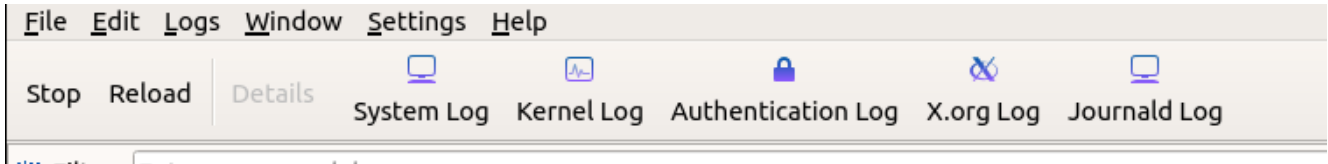
So, in other words, you can open KSystemLog and then open an application that's giving you trouble and see what errors are thrown. You can also examine the entire list of errors and events.

Not only that, but the events in the logs are color-coded. This provides easy visual cues as the events range in color from a light gray (normal) to a bright red for emergency-level events. You can also apply filters and search for specific events.

2/10/21 8:59 PM	kgiii-desktop	nm-dispatcher	req:2 'connectivity-change': new request (2 scripts)
2/10/21 8:59 PM	kgiii-desktop	nm-dispatcher	req:2 'connectivity-change': start running ordered scripts...
2/10/21 8:59 PM	kgiii-desktop	whoopsie[734]	[20:59:30] Cannot reach: https://daisy.ubuntu.com
2/10/21 8:59 PM	kgiii-desktop	whoopsie[734]	[20:59:30] offline
2/10/21 8:59 PM	kgiii-desktop	whoopsie[734]	[20:59:30] The default IPv4 route is: /org/freedesktop/NetworkManager/ActiveConnection/1
2/10/21 8:59 PM	kgiii-desktop	whoopsie[734]	[20:59:30] Not a paid data plan: /org/freedesktop/NetworkManager/ActiveConnection/1
2/10/21 8:59 PM	kgiii-desktop	whoopsie[734]	[20:59:30] Found usable connection: /org/freedesktop/NetworkManager/ActiveConnection/1

Note the color change for errors.

KSystemLog is a KDE application but pulls in very few dependencies. It loads rapidly and will pick up all installed log files by default. In my case, it finds these:



Currently available logs on this particular system.

Chances are pretty good that it is in your default repositories. If you're using a Debian derivative then you can install it by opening the terminal and entering:

```
[code]sudo apt install ksystemlog -y[/code]
```

Anyhow, there you have it. Adjust the above command for your distro's package manager (or go dig it out if you're manually installing packages). You can take the information from the logged errors and use them as search terms to help.

It's absolutely amazing how many errors can be resolved by just taking the exact error message and tossing it verbatim into a search engine of your choice. I've used this method to troubleshoot so many times that I think it's an asset we should all have in our toolbox.

As always, go ahead and subscribe to get notifications when new articles are published. If you want, you can go ahead and register and start helping. If you don't want to maintain your own site and want to write articles, that can be arranged. Thanks for reading!

Install the non-Snap Version of Chrome/Chromium

So many questions are answered by telling people to skip the snap version and to install the .deb/PPA versions. This is how you do that for Google Chrome and Chromium. Chrome and Chromium are browsers from Google and Chrome is based on Chromium but has more features inside it – and isn't open source. Chromium is open source, and many other browsers are based on it.

This, like many of my articles, is going to be fairly limited in scope. It's only really useful if you're using Debian derivatives like Ubuntu, ElementaryOS, or Mint. In fact, part of it will only apply if you've enabled PPAs on your computer.

PPA, if you're curious, stands for Personal Package Archive. Rather than beat a dead horse, I'll just suggest that you visit this site to learn more about them. It should be noted that PPAs were initially meant for testing and many consider them to be a security risk because you're essentially letting someone else decide what gets installed on your system without the oversight of the official project. Anyone can make a PPA and advertise it.

Anyhow, back to the topic at hand. The process is different for both of them and so we'll go ahead and get started with Chrome.

Chrome:

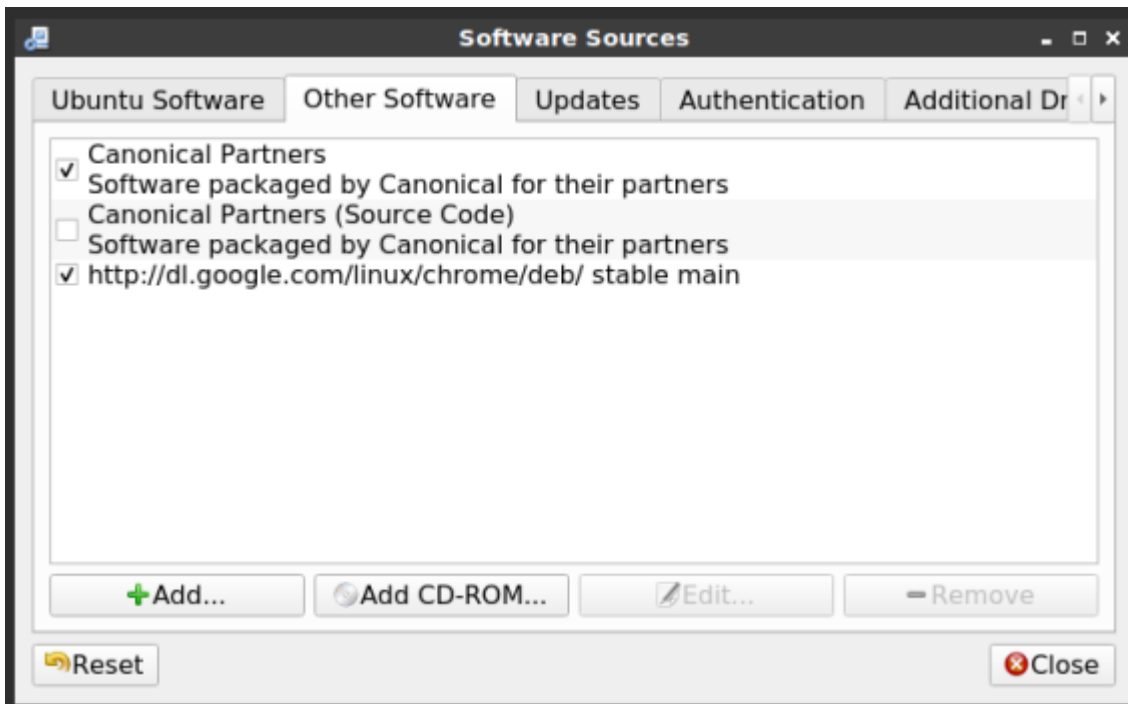
Chrome is going to be the easiest. Crack open your terminal with CTRL + ALT + T and enter the following:

```
[code]cd Downloads  
wget  
https://dl.google.com/linux/direct/google-chrome-stable\_curren
```

```
t_amd64.deb
```

```
sudo apt install ./google-chrome-  
stable_current_amd64.deb[/code]
```

Tada! You're done. When you finish it up and start Chrome, you'll get to set it as the default browser and stuff like that. On top of that, it will automatically update with the system – as it adds its own repository.



Note, the new repository contains the beta version as well.

Chromium:

Chromium is a bit more complicated. For this one, you'll need to add a PPA first. Once again, open your terminal with CTRL + ALT + T.

Now, let's add the PPA.

```
[code]sudo add-apt-repository ppa:saiarcot895/chromium-  
beta[/code]
```

Yes, I know it says beta. Also, any modern distro should do this for you, but you may need to first update the database of installable packages.

```
[code]sudo apt update[/code]
```

Once that's done, or if you don't need to do so, you can go ahead and install Chromium.

```
[code]sudo apt install chromium-browser[/code]
```

Then, much like with Chrome, you will have a regularly updated version of Chromium at your disposal. Like always, go ahead and subscribe to be notified of new articles. I promise, I won't send you any spam or give your email addresses away.

Ubuntu Restricted Extras

As I trawl the 'net, I see some confusion about the Ubuntu Restricted Extras package. This post is meant to clear up any confusion. Given the nature of the beast, I assume this is also valid for Ubuntu derivatives – but I've not explored all of them personally (and there are many).

So, what's in the package? It's a package with more than one thing in it and will also download (if allowed) more data along the way. This is what's inside:

```
gstreamer0.10-ffmpeg  
gstreamer0.10-fluendo-mp3  
gstreamer0.10-pitfdll  
gstreamer0.10-plugins-bad  
gstreamer0.10-plugins-ugly  
gstreamer0.10-plugins-bad-multiverse  
gstreamer0.10-plugins-ugly-multiverse  
icedtea6-plugin  
libavcodec-extra-52
```

```
libmp4v2-0  
ttf-mscorefonts-installer  
unrar
```

Basically, it's a bunch of fonts, codecs (so that you can play patent-encumbered media files), and the ability to open .rar compressed files and use Java applets in the browser. It's some handy stuff, but because it's all non-free Ubuntu doesn't include it by default. Indeed, you won't even find it unless you enable the 'multiverse' repository, like so:

```
[code]sudo add-apt-repository multiverse[/code]
```

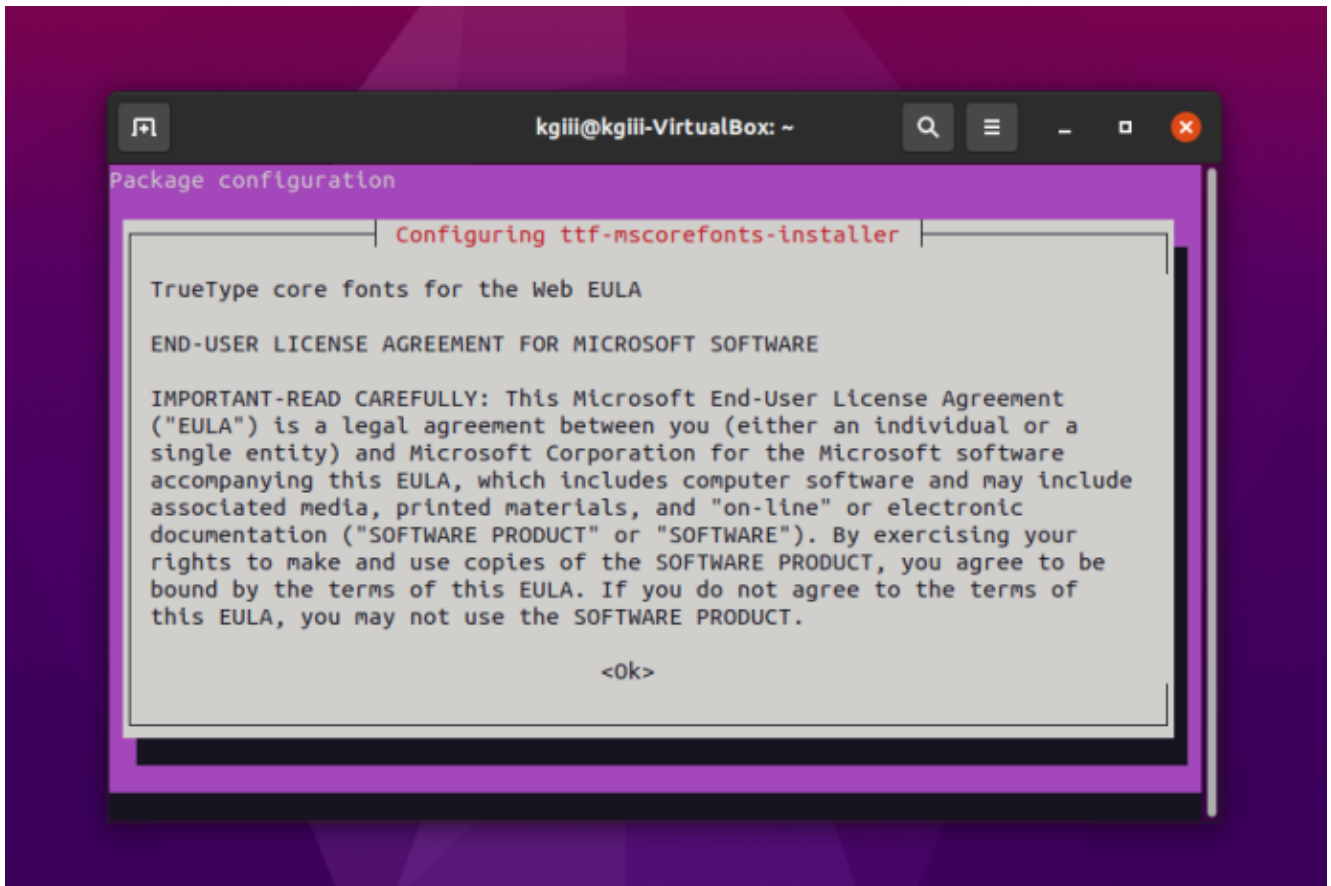
Then, you'll need to make sure that your system knows (though some more modern releases will do this automatically) about the new software choices.

```
[code]sudo apt update[/code]
```

Follow that up with:

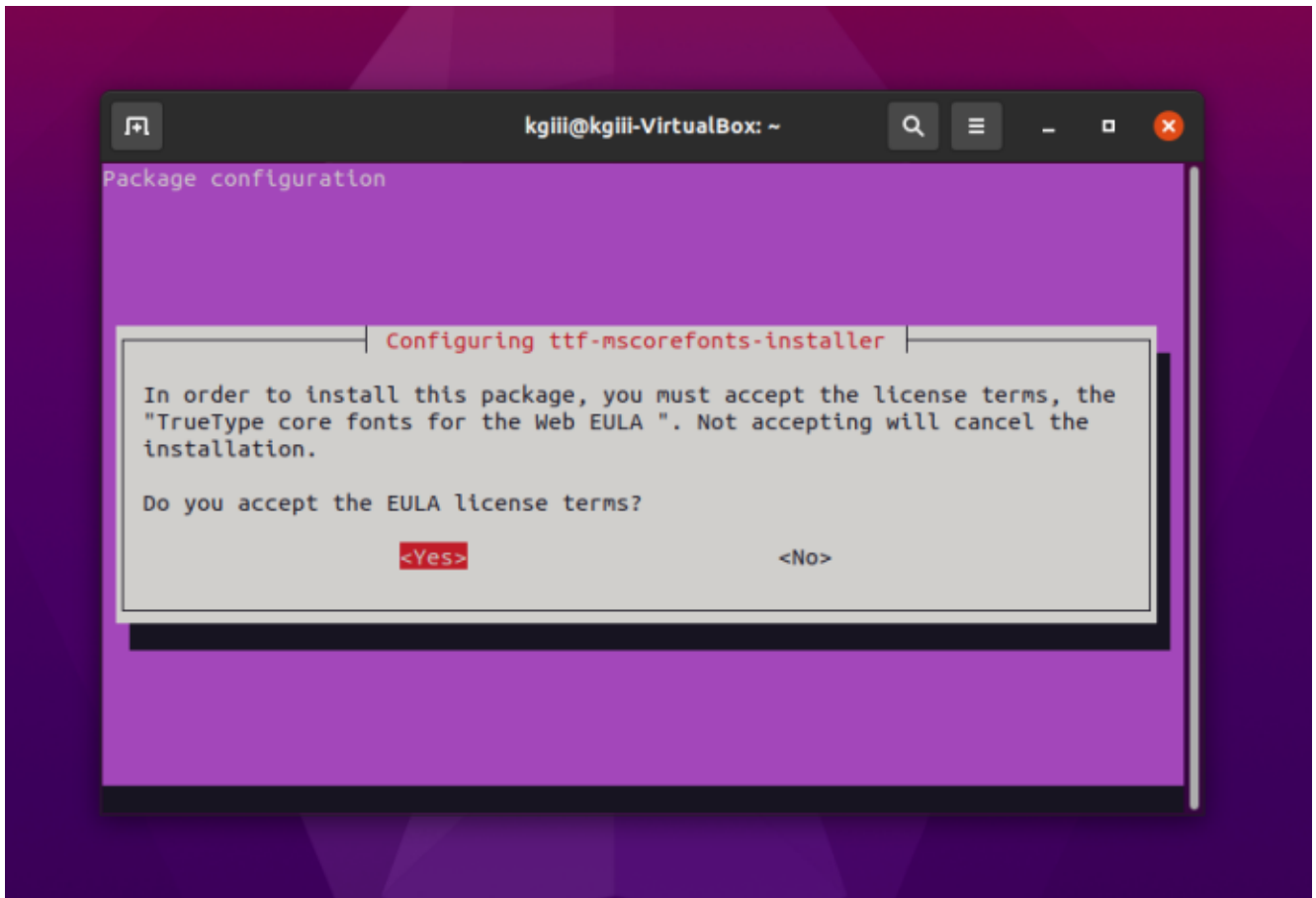
```
[code]sudo apt install ubuntu-restricted-extras[/code]
```

That's going to download a bunch of stuff and show a screen that trips up a lot of newer users. There's no obvious way to use a mouse and, indeed, you have to use your keyboard to agree to the user agreement. It looks like this:



Yes, you have to agree to proceed.

It's okay. Don't give up! You can use your TAB (or arrows) key to make selections and use the ENTER button to submit your answers. You'll do it again on the next screen.



Use the tab or the arrow keys and the enter to agree.

At that point, it'll download some more files with fonts in them and extract them to the right directories so that you can use them. You'll also be able to use the other included files/applications, though they are not 'free' in the sense of the license agreement (which is the reason they're not included by default and you have to jump through hoops to install them).