

www.ubuntubuzz.com

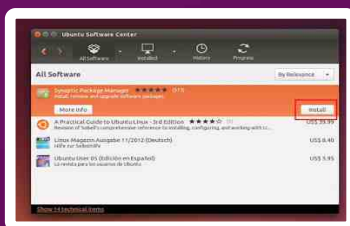
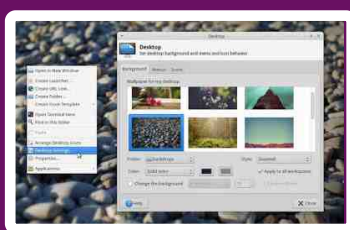
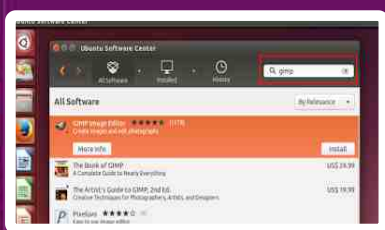


UbuntuBuzz Magazine #09

A Monthly Bundle from ubuntubuzz.com

December 2015

Ubuntu Desktop



Ubuntu Software Center Guide

Installing Ubuntu Apps As Easy As Installing Android Apps

LXDE Desktop

Beginner Guide To Tweak LXDE

KDE Customization

Beginner Guide To Tweak KDE

Configure apt-get To Work Behind A Proxy

Beginner Guide To Tweak KDE



About UbuntuBuzz Magazine Bundle

This magazine is a monthly collection of UbuntuBuzz.com articles. Each bundle issue will cover its previous month articles. This magazine is owned by Mahmudin Ashar as the UbuntuBuzz owner and done by Ade Malsasa Akbar as the writer.

Cover done with Scribus, contents done with LibreOffice, assembled with PDFSAM, in Ubuntu 12.04.

Fonts used are Ubuntu, FreeSans, FreeMono, Libre Baskerville, and Roboto.

Bundle #09 December contains articles from November 2015.

Table of Contents

BEGINNER KDE DESKTOP CUSTOMIZATION GUIDE.....	2
SYNAPTIC PACKAGE MANAGER BEGINNER GUIDE.....	8
GDEBI PACKAGE INSTALLER BEGINNER GUIDE.....	12
APT-RDEPENDS: VISUAL MAPPING FOR PACKAGE DEPENDENCIES.....	16
HOW TO INSTALL OFFICIAL SLACK CLIENT ON UBUNTU.....	20
LIST OF COMMANDS FOR PACKAGE DEPENDENCY CHECKING IN UBUNTU.....	22
CHANGE SCRIBUS DEFAULT FONT.....	33
BEGINNER XFCE DESKTOP CUSTOMIZATION GUIDE.....	34
BEGINNER LXDE DESKTOP CUSTOMIZATION GUIDE.....	40
CONFIGURING APT.CONF PROXY AUTHENTICATION FOR APT-GET.....	46
HOW TO ADD UBUNTU PPA MANUALLY.....	48
HOW TO INSTALL SCRIBUS 1.5 NG IN UBUNTU 14.04.....	51
ENABLE AUTOSAVE IN LIBREOFFICE.....	53
CHANGE LIBREOFFICE WRITER DEFAULT FONT.....	54
UBUNTU SOFTWARE CENTER BEGINNER GUIDE.....	55

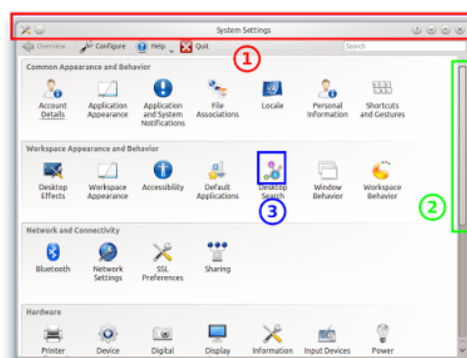
BEGINNER KDE DESKTOP CUSTOMIZATION GUIDE

Previously, we've written a [customization guide for LXDE](#). Now we write about KDE customization. KDE is the most complex desktop environment among which we've written about here. But don't worry it is a basic guide to customize KDE starting from changing wallpaper until installing new theme. We use KDE 4.8.5 on Kubuntu 12.04 as a basis. We hope it helps everyone begins KDE. Enjoy.

Concepts

1. KDE has four different themes: widget (toolkit) theme, Plasma theme, icon theme, and window decoration theme.
2. The main difference between KDE and another desktop is Plasma theme (a special theme for the panel and Plasmoid appearance).
3. Plasmoid is KDE term for desktop widget (such as clock, CPU monitor, can also be placed at panel).
4. Plasma theme is theme for Plasma, it means theme for your Plasmoid and panel.
5. Color theme is color settings for whole desktop i.e. change default silver-white-blue color based into blackened one or bumblebee-based color.
6. KDE customization is centralized in KDE System Settings. Basically no need to use Terminal.

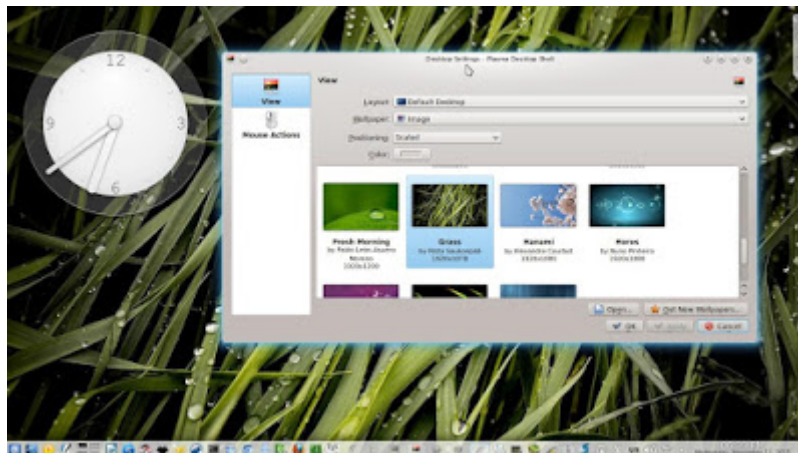
Window Anatomy



Like another desktop environment, KDE has three essential visual components:

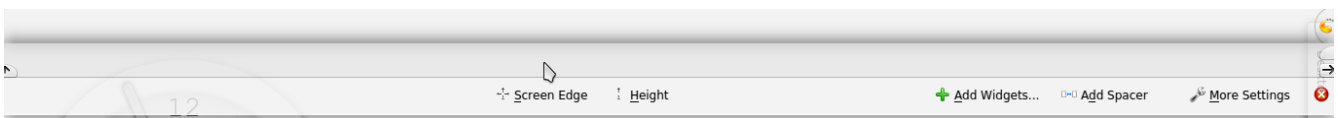
1. Red box: window decoration (or window border). Controlled by window manager of KDE (named KWin).
2. Green box: widget toolkit (i.e. scrollbar, button, progress bar). It is controlled by widget theme (i.e. QtCurve and GTK).
3. Blue box: icon theme.

Change Wallpaper



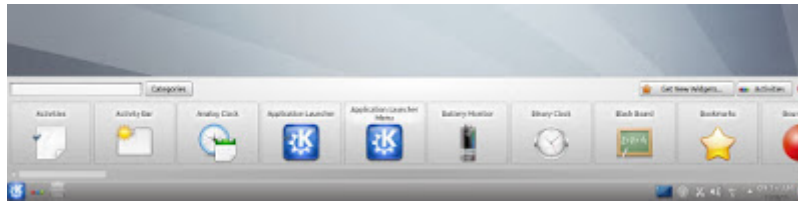
To change wallpaper, right-click on desktop > Default Desktop Settings > select a wallpaper. To insert your own picture, press Open button.

Create Panel



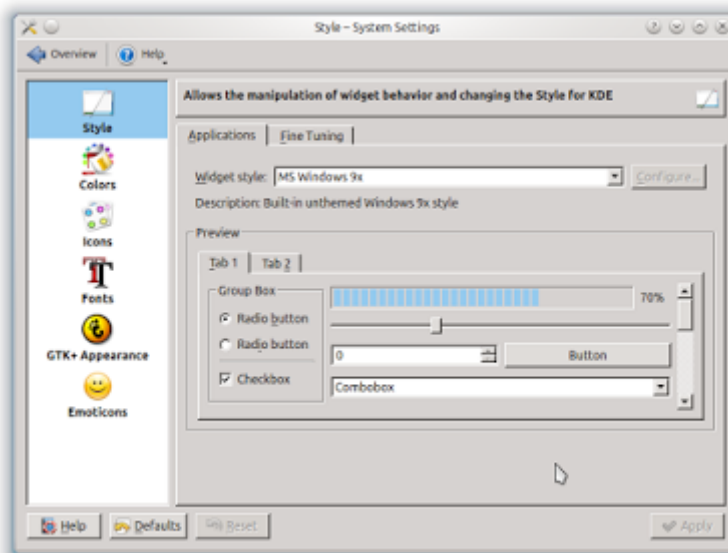
To create a new panel, right-click on desktop > Add Panel > Empty Panel. We can move its position by right-click panel > Panel Settings > hold click on Screen Edge button > drag it where on the desktop you want it to be.

Add Widget (Plasmoid)



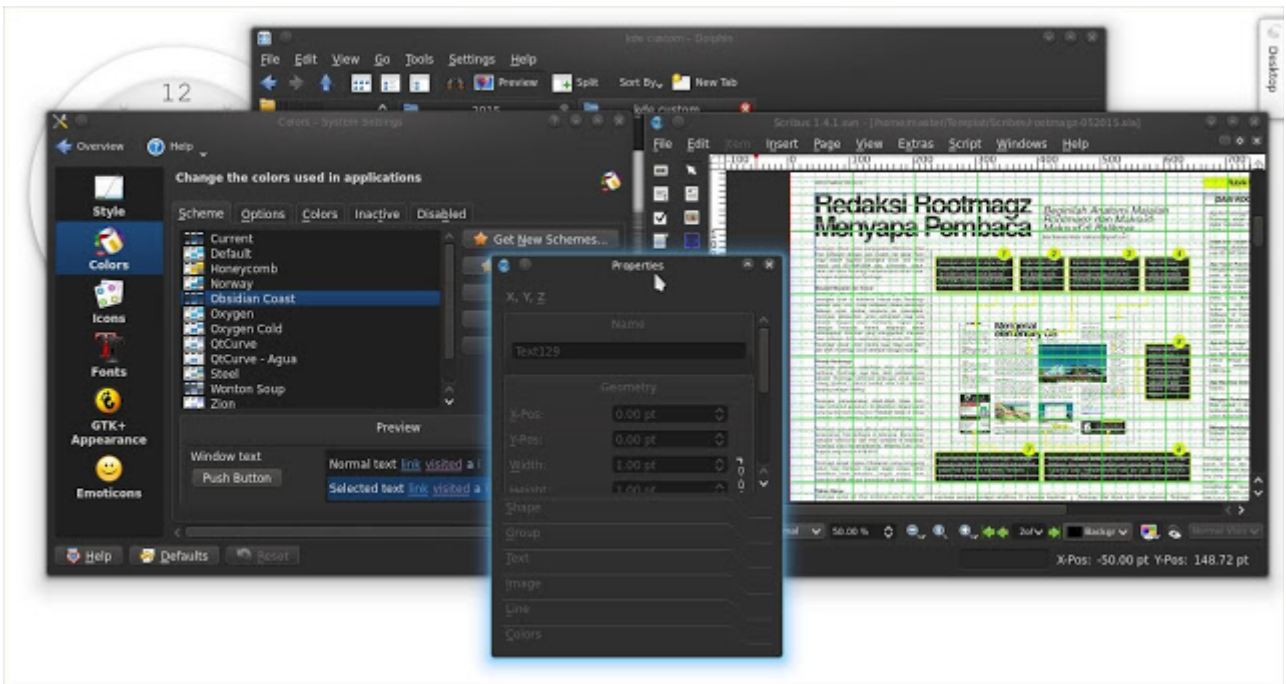
To add desktop widget (called Plasmoid, not to be confused with toolkit widget below), right-click on desktop > Add Widgets > widgets selection bar will appear > drag and drop one to desktop. To add widget into panel, just drag and drop it to panel.

Change Widget Theme

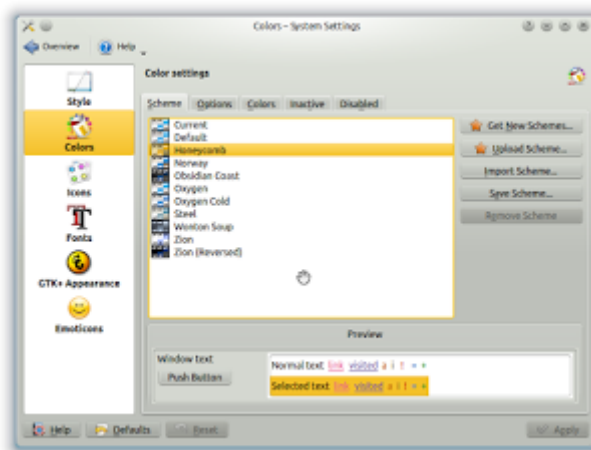


To change widget (toolkit) theme, open KDE System Settings > Application Appearance > Style > Widget Style: choose one. KDE default widget theme is Oxygen, but it gives some other options. Most noticeable among them are GTK+ Style and MS Windows 9x.

Change Color Theme



To change color theme, open System Settings > Application Appearance > Colors > select one scheme. Picture above depicts blackened KDE by just selecting Obsidian Coast color theme. Simply we don't need any additional theme to achieve this. Picture below depicts bumblebee color from Honeycomb color theme. We may modify every detail of a color or create a new scheme by using tabs left (Options - Colors - Inactive - Disabled).

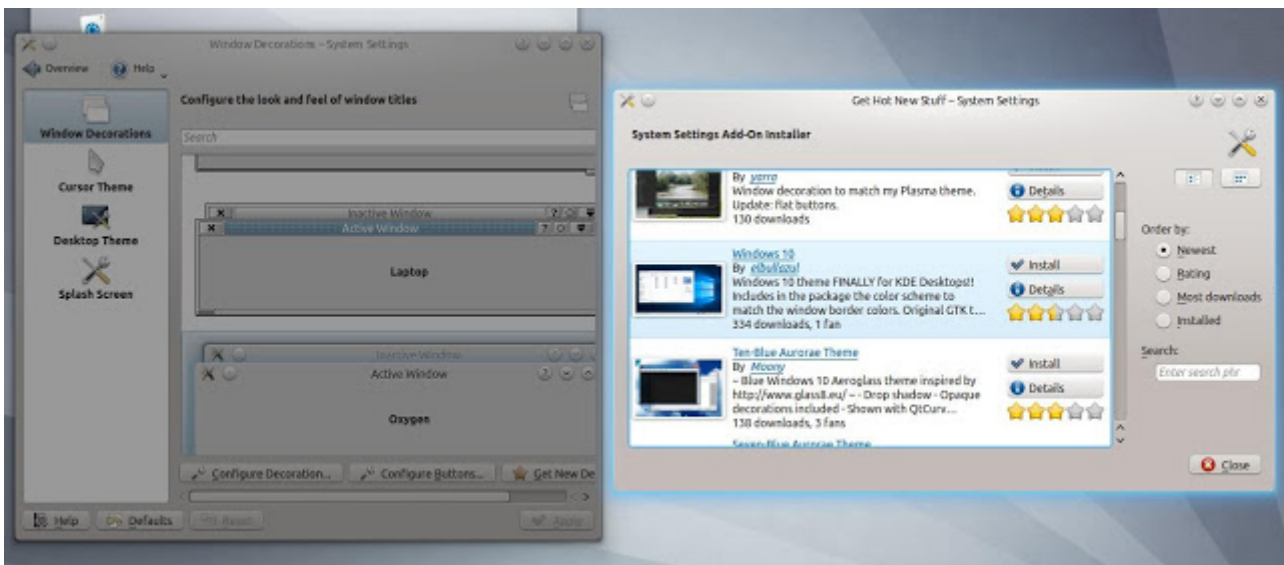


Change Icon Theme



To change icon theme, open System Settings > Application Appearance > Icons > select icon theme. On picture above, we enable Humanity icon after we've installed some icon themes.

Install Window Decoration, Icon, & Plasma Theme



To install any theme in KDE, we just need to open System Settings, open a theme section (e.g. Window Decoration), and press Get New Theme. We can install theme directly from it (of course it needs internet access). As example, to install a new window decoration, we open System Settings > Workspace Appearance > Window Decoration > press Get

New Decorations > we selected a theme named "Windows 10" > select the installed decoration theme (Windows 10) > press Apply. Same there with another themes such as icon, desktop theme, wallpaper, and cursor theme. These awesome functionality source is <http://kde-look.org> where KDE downloads all themes from.

References

- [KDE-LOOK](#): all theme downloads: .
- [DeviantArt LinuxLounge](#): KDE themes from LinuxLounge in DA.
- [DeviantArt KDE Search](#): global search for KDE in DA.
- [DeviantArt KDE Users](#): KDE themes from KDE Users in DA.
- [Sourceforge Caledonia](#): Caledonia art suite for KDE.

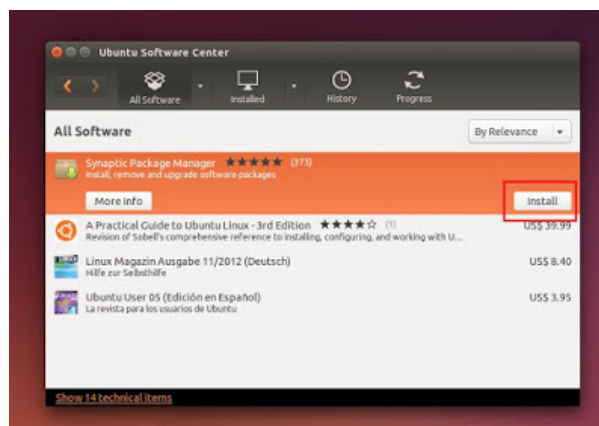
SYNAPTIC PACKAGE MANAGER BEGINNER GUIDE

At the old days, Ubuntu brought Synaptic Package Manager in every release until it completely replaced with Ubuntu Software Center at 11.10 Oneiric Ocelot. Synaptic and Ubuntu Software Center are basically same things (package managers) but different in features. Ubuntu Software Center (later called as USC) is easy to use for beginners. But for many reasons, somebody will need Synaptic again. For example, Synaptic has advanced settings and detailed information in every action it does (while USC hides almost all detailed information). So here is a guide to use Synaptic.

Concept

1. Synaptic shows list of applications in text only without icons. It is different with USC.
2. Synaptic asks user first what to do (selecting actions), then asks user again to Apply it (executing actions). It is the same concept with Gparted usage.
3. Synaptic shows detailed information for every piece of action or application. This will make new user a bit confused, but not for long time.
4. Synaptic asks for root password every user starts it.

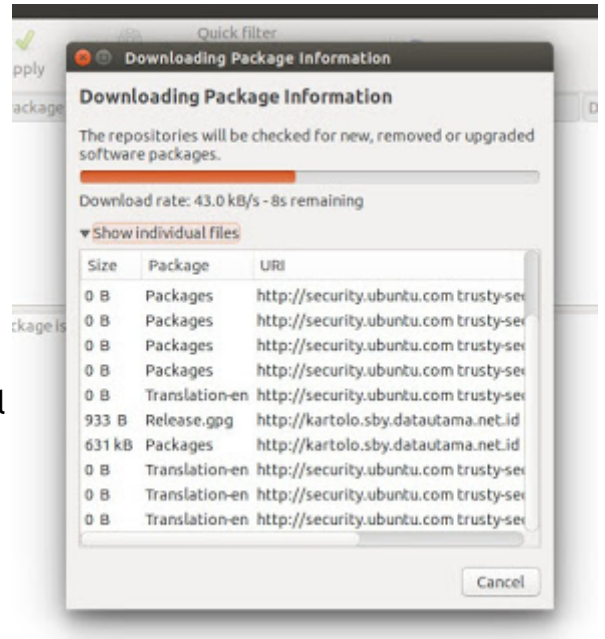
Install Synaptic



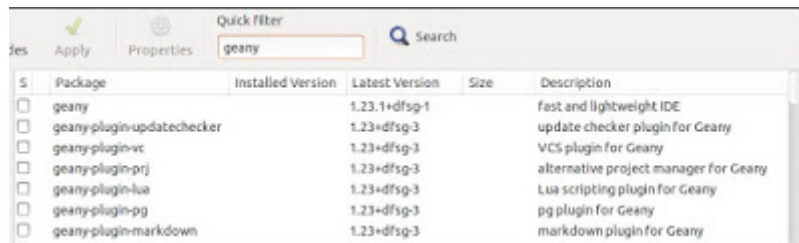
You must open Ubuntu Software Center > search for synaptic > press Install button. Of course, it's because nowadays Ubuntu have no Synaptic preinstalled anymore.

First Time Usage

We do "reload" by pressing Reload button. This will download repository indexes (map of server contents) about 15 MB in total. Same with Ubuntu Software Center. See picture above. Every line showed is index file. Here, Synaptic is better than USC because it shows details KB/s speed and every URL of individual file.

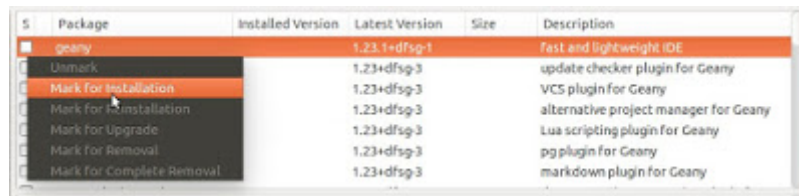


Search Application



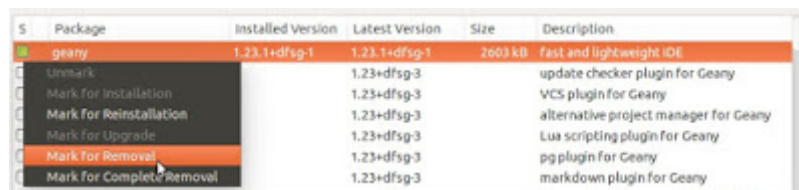
To search application, enter the name into search box. In this example, we search geany. See picture above. The result will be geany, geany-plugin-updatechecker, geany-plugin-vc, and so on. The application we want to install is usually having the most generic name (geany).

Install Application



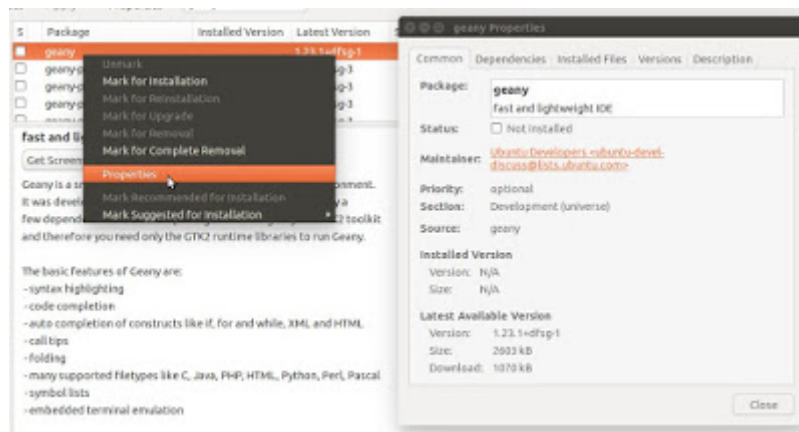
To install application, click white box in front of application name > Mark for Installation > press Apply button > let Synaptic installs the program.

Uninstall Application



To uninstall application, click green box in front of application name > Mark for Removal > press Apply button > let Synaptic removes it.

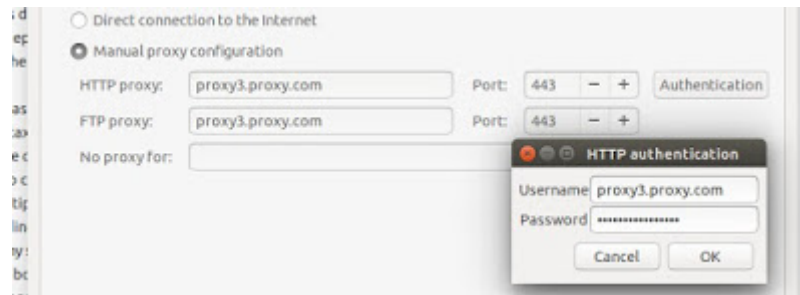
Show Application Details



To show detailed information of application, right-click application name entry (.e.g geany) > Properties. We can see common details, dependencies list, what files will be

installed along, version of application, and description.

Set Proxy

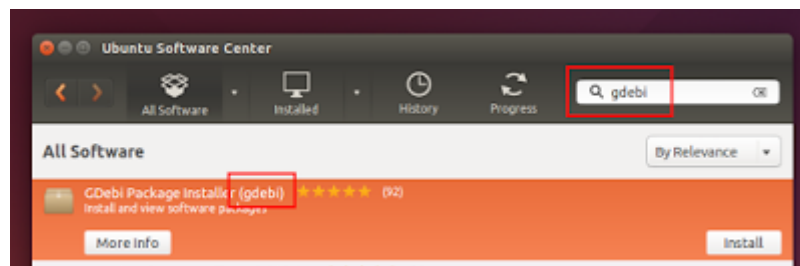


To set proxy and authentication, open menu Settings > Preferences > Network > check Manual proxy configuration > insert proxy URL and port > click Authentication button > insert username & password. Important: probably we will need to copy & paste username and password into its text boxes (at least in Synaptic 0.81.1 in Ubuntu 14.04 we need to do it).

GDEBI PACKAGE INSTALLER BEGINNER GUIDE

We've written a guide about Synaptic previously. Now we need to write about GDebi. GDebi is a package manager to install a single .deb package in Ubuntu. GDebi will help you resolve dependencies automatically by right-clicking on the package file. We will give two different examples to use GDebi here.

Install GDebi



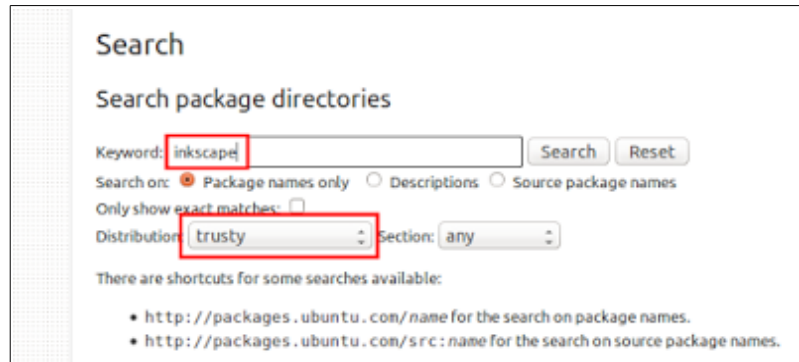
To use GDebi in Ubuntu, you must install it first. Use Ubuntu Software Center to do it.

First Example

This first example is suitable for a single package from packages.ubuntu.com. Most case is when you download a single package only without its dependency packages. To complete dependencies, you use GDebi to automatically resolve and download all dependencies.

1. Obtain A Package from packages.ubuntu.com

We are using Ubuntu 14.04 Trusty Tahr so we give example how to download package for trusty. If you are using 15.04 Vivid Vervet, change trusty with vivid and vice versa. We will download inkscape package (a vector drawing application).

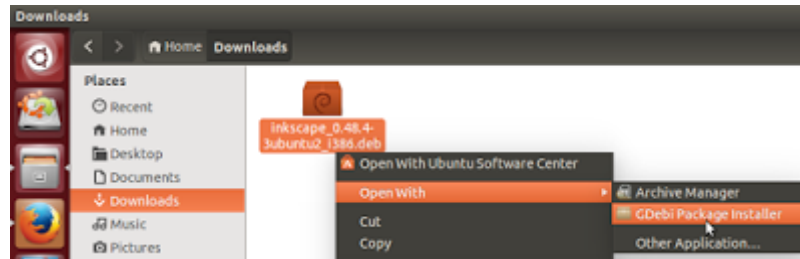


1. Open <http://packages.ubuntu.com>.
2. In the Search package directories section, select Distribution: trusty, then search inkscape. Don't make any changes except those.
3. On the Package hits section, you will notice trusty link with Inkscape description. Click that link.
4. On the Package: inkscape page, you will notice so many packages listed down. They are dependencies for inkscape package. See the Download table on the most bottom of page. There are amd64 and i386 choices. If you are using 32 bit then select i386 otherwise select amd64.
5. On the Download page, there are many links for inkscape package. So select one link. We choose North America mirror.
6. You've been downloaded inkscape package. In our example, the file name is **inkscape_0.48.4-3ubuntu2_i386.deb**. Usually Firefox saves the file on Downloads folder except you select the destination.

2. Install The Package & Its Dependencies

To demonstrate how to install a package (from Ubuntu official repository) with GDebi, we will install a single inkscape package. You must enable your internet connection to do

this. Or, you have a complete copy of Ubuntu 14.04 repository (or repository of your version of Ubuntu) in DVD or local HDD.



1. Open Nautilus file manager to the downloaded package.
2. Right-click package > Open With > GDebi Package Installer.
3. GDebi window will appear. You will probably see warning Same version is available in software channel. It means the inkscape package you invoke is installable from repository, don't need to use GDebi. Ignore it.
4. You will see dependencies number in GDebi for inkscape package. In our 14.04, the dependencies needed is three (Requires the installation of 3 packages).
5. Press Install Package button.
6. Let GDebi download and install all dependencies including inkscape package.
7. Installation finished. Go check for Inkscape in Ubuntu menu.

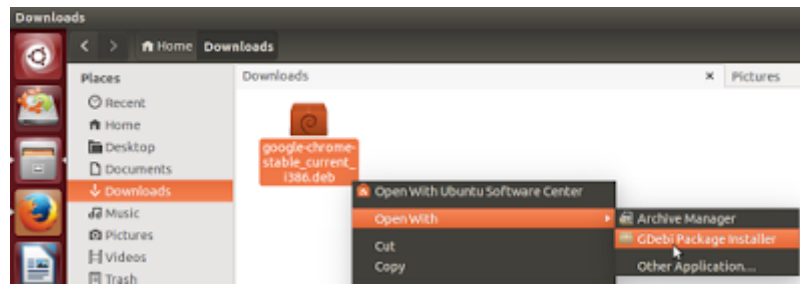
Second Example

This second example is suitable for a single package which is not available in Ubuntu official repository. GDebi is far better to install this kind of package, compared to Synaptic and USC. We take Google Chrome package as example.

1. Obtain The Package

1. Go to <https://www.google.com/chrome/browser/desktop>.
2. Download the .deb package.
3. In our example, the file name is **google-chrome-stable_current_i386.deb**.

2. Install The Package & Its Dependencies



1. Open Nautilus file manager to the package downloaded.
2. Right-click package file > Open With > GDebi Package Installer.
3. GDebi will appear. Press Install Package button.
4. Let GDebi download all dependencies and install them all.
5. Go check for Google Chrome on Ubuntu menu.

APT-RDEPENDS: VISUAL MAPPING FOR PACKAGE DEPENDENCIES

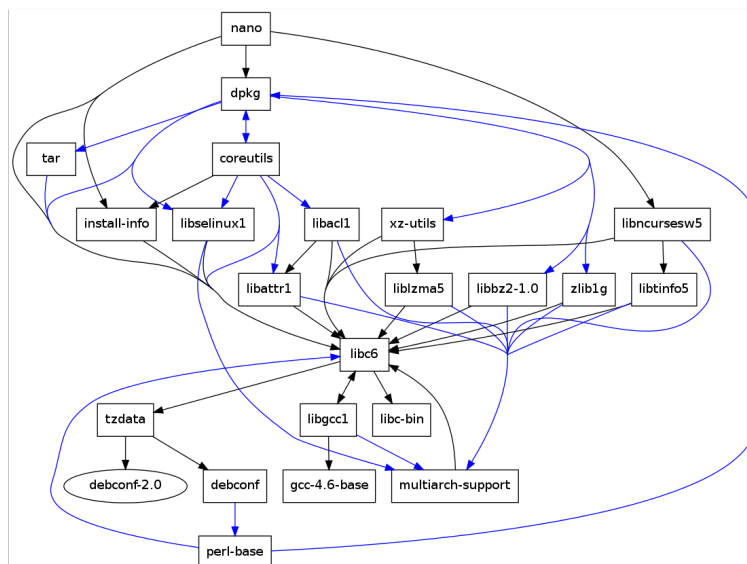
apt-rdepends is a great tool to list complete dependencies of a package. Beside of that, apt-rdepends has capability to produce graph-compatible output. In other words, user can use apt-rdepends to produce dependencies list then use dot to create dependencies map in an image file (whether it is PNG or else). In this article, we will create maps for some Ubuntu packages separately.

Preparation

```
sudo apt-get install apt-rdepends dot
```

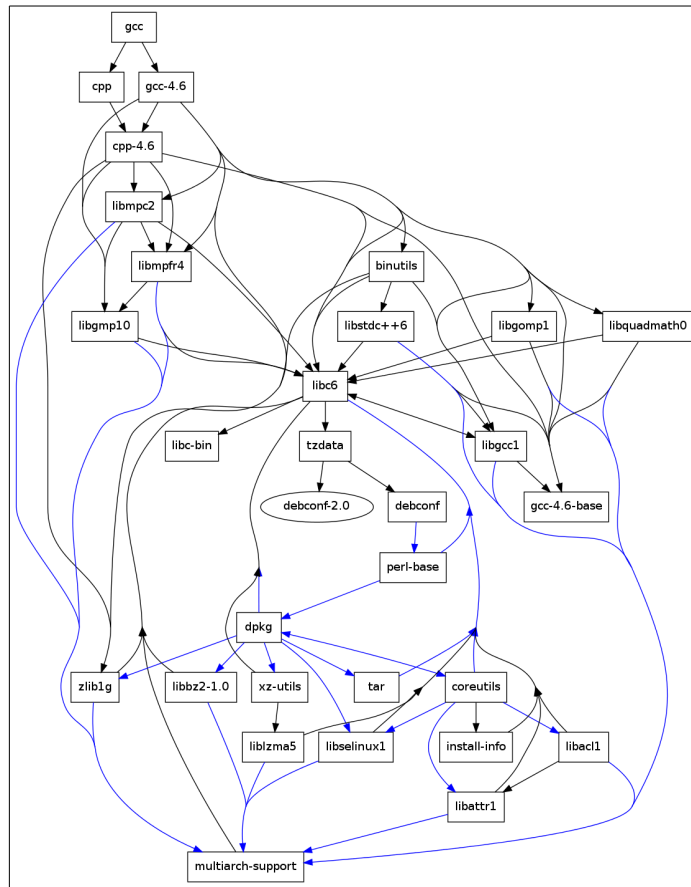
Mapping nano

```
apt-rdepends --dotty nano | dot -Tpng > dependency-map.png
```



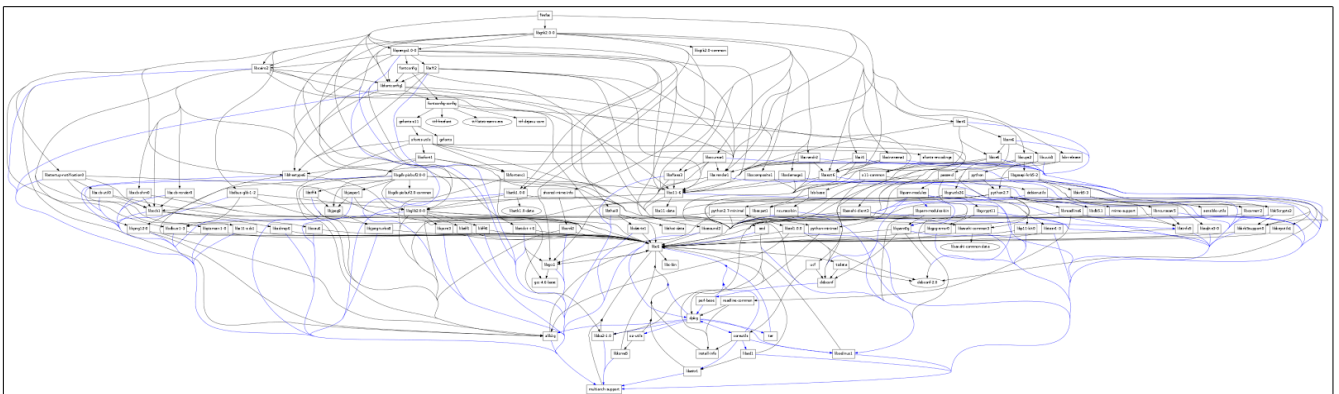
Mapping gcc

```
apt-rdepends --dotty nano | dot -Tpng > dependency-map.png
```



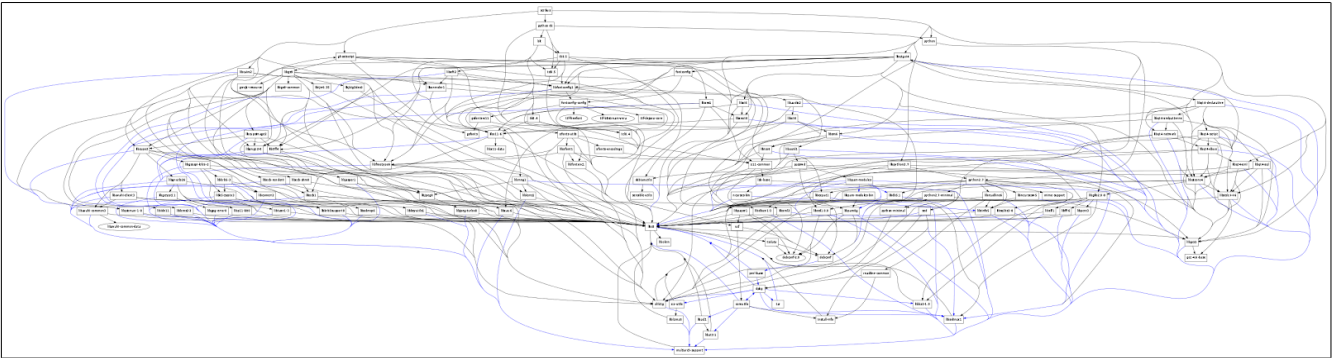
Mapping firefox

```
apt-rdepends --dotty firefox | dot -Tpng > dependency-map.png
```



Mapping scribus

```
apt-rdepends --dotty scribus | dot -Tpng > dependency-map.png
```



Map Explanation

- **Blue:** pre-dependes.
- **Black:** depends.
- Both blue and black lines are symbols of dependency. For example, in the nano graph, nano package has black arrow to dpkg package. It means nano depends to dpkg. In simpler words, nano needs dpkg. But on the same graph, dpkg package has blue arrow to tar package. In simpler words, it means dpkg needs tar, but when installation happens, tar must be installed and configured completely before dpkg installed. For more explanation about pre-dependes you can read Debian FAQ https://www.debian.org/doc/manuals/debian-faq/ch-pkg_basics.en.html.

Commands Explanation

- `apt-rdepends --dotty <package>`: resolve dependencies recursively and output the list in dot-readable text format.
- `| dot -Tpng > dependency-map.png`: receive dependency list output from another program and produce the graph in PNG file.

Notes

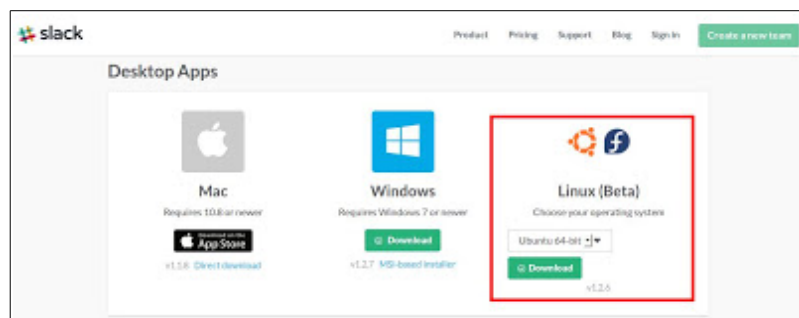
To save into SVG (so the image file will be scalable), use `-Tsvg` instead `-Tpng`.

HOW TO INSTALL OFFICIAL SLACK CLIENT ON UBUNTU

A few months ago [we wrote about ScudCloud](#), an unofficial Slack.com client for Linux. Now Slack has its own official client for Linux. At this time, we will use Slack Desktop latest version 1.2.6. It is available for Ubuntu and Fedora. We will tell how to install it.

Download Slack

Go to <http://slack.com/downloads> and select Ubuntu (either 32 or 64 bit) and press Download button. You will download DEB file named slack-desktop-1.2.6-i386.deb or slack-desktop-1.2.6-amd64.deb. Save it in Downloads in Home directory.



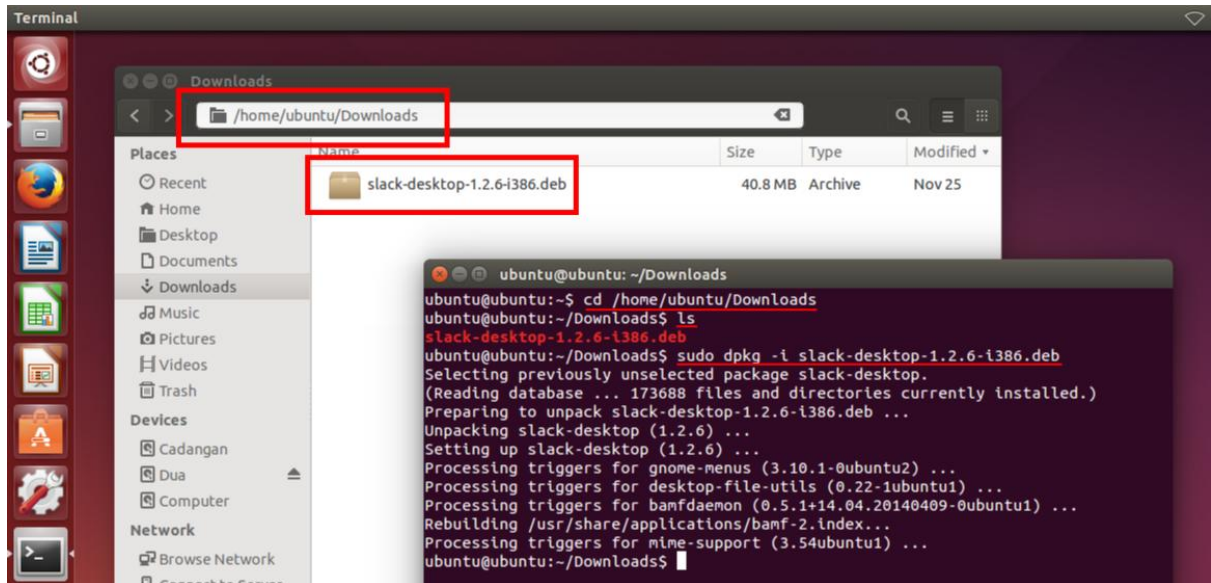
Install Slack

In Terminal, go inside ~/Downloads directory and then perform command either command:

For 32 bit file: `sudo dpkg -i slack-desktop-1.2.6-i386.deb`

For 64 bit file: `sudo dpkg -i slack-desktop-1.2.6-amd64.deb`

Note: in Ubuntu 14.04, we don't need any external dependency so Slack Desktop installed successfully without downloading any other package. The package installation can be done without internet.



LIST OF COMMANDS FOR PACKAGE DEPENDENCY CHECKING IN UBUNTU

Every new user will experience dealing with package dependency in Ubuntu. This list is a shortcut to help you get what dependencies you want in faster way. You can choose any command you need from below. We list variety of 5 commands here dpkg, dpkg-deb, apt-rdepends, apt-cache, and apt-get. We use Ubuntu 12.04 32 bit for this article. If you are looking for package dependency searching in Ubuntu then this article is for you.

Prerequisites

- You must install apt-rdepends first because it is not built-in included in Ubuntu. Install it with `sudo apt-get install rdepends`.
- You must at least ever once doing `sudo apt-get update`. It needs internet access. It builds complete APT Database for your Ubuntu. If you have no APT Database, no command apt-cache, apt-rdepends, or apt-get can be performed completely.

About APT Cache Terminology

In simplified words, APT Cache is a term to explain both APT Database and APT Package Files. APT Database is stored at `/var/lib/apt/lists/`. It contains plain text files stored from `sudo apt-get update` a user performs. APT Package Files are stored in `/var/cache/apt/archives/`. It contains DEB files stored from any `sudo apt-get install` command a user performs. apt-cache command works for these two types of cache.

1. dpkg: Show Package Information

Synopsis: `dpkg -I <package> <directory>`

Example: `dpkg -I uget_1.8.0-1_i386.deb`

This command will show control file contents including dependencies stated inside the package. The output will be like this.

```

new debian package, version 2.0.
 size 193578 bytes: control archive= 1955 bytes.
    948 bytes,    23 lines    control
   2618 bytes,   34 lines    md5sums
    185 bytes,    7 lines    * postinst      #!/bin/sh
    160 bytes,    5 lines    * postrm        #!/bin/sh
Package: uget
Version: 1.8.0-1
Architecture: i386
Maintainer: Ubuntu Developers <ubuntu-devel-
discuss@lists.ubuntu.com>
Original-Maintainer: Elías Alejandro Año Mendoza
<ealmdz@gmail.com>
Installed-Size: 736
Depends: libc6 (>= 2.4), libcairo2 (>= 1.2.4), libcurl3 (>=
7.16.2-1), libglib2.0-0 (>= 2.20.0), libgstreamer0.10-0 (>=
0.10.0), libgtk2.0-0 (>= 2.24.0), libnotify4 (>= 0.7.0),
libpango1.0-0 (>= 1.14.0)
Section: net
Priority: optional
Homepage: http://urlget.sourceforge.net/
Description: easy-to-use download manager written in GTK+2
 Uget (formerly urlgfe) is a simple, lightweight and easy-to-use
 download manager.
It provides the following features:
 * Resume downloads.
 * Queue downloads.
 * Classify downloads in categories.
 * Mozilla Firefox integration (through Flashgot plugin).
 * Clipboard monitoring.
 * Import downloads import from HTML files.
 * Batch download.
.

```

It also can be launched from the command line.

2. dpkg-deb: Extracting DEB Package

Synopsis: `dpkg-deb --raw-extract <package> <directory>`

Example: `dpkg-deb --raw-extract uget_1.8.0-1_i386.deb /tmp`

This command will extract DEB file named above into /tmp directory. This will produce two directories named DEBIAN and usr. What you need is to read a file named control inside DEBIAN directory. Do command `cat control` inside DEBIAN then it produces output exactly like this.

```
Package: uget
Version: 1.8.0-1
Architecture: i386
Maintainer: Ubuntu Developers <ubuntu-devel-
discuss@lists.ubuntu.com>
Original-Maintainer: Elías Alejandro Año Mendoza
<ealmdz@gmail.com>
Installed-Size: 736
Depends: libc6 (>= 2.4), libcairo2 (>= 1.2.4), libcurl3 (>=
7.16.2-1), libglib2.0-0 (>= 2.20.0), libgstreamer0.10-0 (>=
0.10.0), libgtk2.0-0 (>= 2.24.0), libnotify4 (>= 0.7.0),
libpango1.0-0 (>= 1.14.0)
Section: net
Priority: optional
Homepage: http://urlget.sourceforge.net/
Description: easy-to-use download manager written in GTK+2
Uget (formerly urlgfe) is a simple, lightweight and easy-to-use
download manager.
It provides the following features:
* Resume downloads.
* Queue downloads.
* Classify downloads in categories.
* Mozilla Firefox integration (through Flashgot plugin).
* Clipboard monitoring.
* Import downloads import from HTML files.
* Batch download.
```

It also can be launched from the command line.

The dependencies information shows up after Depends flag. So, the dependencies of this uget package are libc6 (>= 2.4), libcairo2 (>= 1.2.4), libcurl3 (>= 7.16.2-1), libglib2.0-0 (>= 2.20.0), libgstreamer0.10-0 (>= 0.10.0), libgtk2.0-0 (>= 2.24.0), libnotify4 (>= 0.7.0), libpango1.0-0 (>= 1.14.0).

3. dpkg-deb: Show Depends Field Contents of A Package Directly

Synopsis: dpkg-deb --field <package> <field>

Example: dpkg-deb --field uget_1.8.0-1_i386.deb Depends

Output: libc6 (>= 2.4), libcairo2 (>= 1.2.4), libcurl3 (>= 7.16.2-1), libglib2.0-0 (>= 2.20.0), libgstreamer0.10-0 (>= 0.10.0), libgtk2.0-0 (>= 2.24.0), libnotify4 (>= 0.7.0), libpango1.0-0 (>= 1.14.0)

This command sums up what 2nd command above did, into just one single step. This command pulls directly Depends field from control file inside the DEBIAN directory. For frequent working with package, this command is more suitable than the previous. We see the output is exactly the same.

4. apt-cache: Show Package Information

Synopsis: apt-cache show <package>

Example: apt-cache show uget

This command shows up control file content inside a package that are already inside APT database. You must override <package> field with package name, not file name. It is similar with dpkg -l command above except apt-cache show will take information from APT Database not from a single DEB file.

5. apt-cache: Show Package More Information

Synopsis: apt-cache showpkg <package>

Example: apt-cache showpkg uget

This command shows up detailed information similar with show command before. But showpkg shows where a package belongs to in APT Database. Especially in which database file it belongs. The output will be like this.

```
Package: uget
Versions:
2.0-0ubuntu0~precise
(/var/lib/apt/lists/ppa.launchpad.net_plushuang-tw_uget-
stable_ubuntu_dists_precise_main_binary-i386_Packages)
  Description Language:
      File:
/var/lib/apt/lists/ppa.launchpad.net_plushuang-tw_uget-
stable_ubuntu_dists_precise_main_binary-i386_Packages
      MD5: df1c64f6fd949d38e1bd89de60a8af32
  Description Language: en
      File:
/var/lib/apt/lists/ppa.launchpad.net_plushuang-tw_uget-
stable_ubuntu_dists_precise_main_i18n_Translation-en
      MD5: df1c64f6fd949d38e1bd89de60a8af32
```

Reverse Depends:

Dependencies:

```
2.0-0ubuntu0~precise - libappindicator3-1 (2 0.2.92) libc6 (2
2.15) libcairo2 (2 1.2.4) libcurl3 (2 7.16.2-1) libgdk-pixbuf2.0-0
(2 2.22.0) libglib2.0-0 (2 2.28.0) libgstreamer0.10-0 (2 0.10.0)
libgtk-3-0 (2 3.0.0) libnotify4 (2 0.7.0) libpango1.0-0 (2 1.14.0)
libssl1.0.0 (2 1.0.0) aria2 (0 (null))
```

Provides:

```
2.0-0ubuntu0~precise -
```

Reverse Provides:

So, based on showpkg command, the dependencies of uget package are

```
libappindicator3-1 (2 0.2.92) libc6 (2 2.15) libcairo2 (2 1.2.4) libcurl3 (2 7.16.2-1) libgdk-
pixbuf2.0-0 (2 2.22.0) libglib2.0-0 (2 2.28.0) libgstreamer0.10-0 (2 0.10.0) libgtk-3-0 (2
3.0.0) libnotify4 (2 0.7.0) libpango1.0-0 (2 1.14.0) libssl1.0.0 (2 1.0.0) aria2 (0 (null)).
```

6. apt-rdepends: Show Complete Dependencies Recursively

Synopsis: apt-rdepends <package>

Example: apt-rdepends nano

This command will track and show complete dependencies list of a package. This will list dependencies recursively so it will produce very long output. This command includes whole dependencies, it doesn't exclude installed dependencies. For the sake of output simplicity, we use nano instead of uget as example. Output below are exactly all dependencies of nano package in our Ubuntu 12.04.

```
Reading package lists... Done          PreDepends: tar (>= 1.23)
Building dependency tree              PreDepends: xz-utils
Reading state information... Done      PreDepends: zlib1g (>= 1:1.1.4)
nano                                   coreutils
  Depends: dpkg (>= 1.15.4)            Depends: dpkg (>= 1.15.4)
  Depends: install-info                Depends: install-info
  Depends: libc6 (>= 2.11)              PreDepends: libacl1 (>= 2.2.51-
  Depends: libncursesw5 (>=           5)
5.7+20100313)                          PreDepends: libattr1 (>=
dpkg                                    1:2.4.46-5)
  PreDepends: coreutils (>= 5.93-    PreDepends: libc6 (>= 2.15)
1)                                       PreDepends: libselinux1 (>=
  PreDepends: libbz2-1.0                1.32)
  PreDepends: libc6 (>= 2.11)          install-info
  PreDepends: libselinux1 (>=         Depends: libc6 (>= 2.4)
1.32)                                   libc6
```

```

Depends: libc-bin (= 2.15-0ubuntu10)
Depends: libgcc1
Depends: tzdata
libc-bin
libgcc1
  Depends: gcc-4.6-base (= 4.6.3-1ubuntu5)
  Depends: libc6 (>= 2.2.4)
  PreDepends: multiarch-support
gcc-4.6-base
multiarch-support
  Depends: libc6 (>= 2.13-0ubuntu6)
tzdata
  Depends: debconf (>= 0.5)
  Depends: debconf-2.0
debconf
  PreDepends: perl-base (>= 5.6.1-4)
perl-base
  PreDepends: dpkg (>= 1.14.20)
  PreDepends: libc6 (>= 2.11)
debconf-2.0
libacl1
  Depends: libattr1 (>= 2.4.46-3)
  Depends: libc6 (>= 2.4)
  PreDepends: multiarch-support
libattr1
  Depends: libc6 (>= 2.4)
  PreDepends: multiarch-support
libselinux1
  Depends: libc6 (>= 2.8)
  PreDepends: multiarch-support
libbz2-1.0
  Depends: libc6 (>= 2.4)
  PreDepends: multiarch-support
tar
  PreDepends: libc6 (>= 2.8)
xz-utils
  Depends: libc6 (>= 2.7)
  Depends: liblzma5 (>= 5.1.1alpha+20110809)
liblzma5
  Depends: libc6 (>= 2.4)
  PreDepends: multiarch-support
zlib1g
  Depends: libc6 (>= 2.4)
  PreDepends: multiarch-support
libncursesw5
  Depends: libc6 (>= 2.4)
  Depends: libtinfo5 (= 5.9-4)
  PreDepends: multiarch-support
libtinfo5
  Depends: libc6 (>= 2.4)
  PreDepends: multiarch-support

```

7. apt-rdepends -p <package>

Synopsis: apt-rdepends --print-state <package>

Example: apt-rdepends --print-state nano

This command including --print-state option gives every output status whether it is Installed or NotInstalled. The result is exactly same with normal command, but including status. It is useful to begin learning dependency for offline installation. We choose nano as example to avoid too long output.

```

Reading package lists... Done
Building dependency tree
Reading state information... Done
fping
  Depends: libc6 (>= 2.7)
[Installed]
libc6
  Depends: libc-bin (= 2.15-0ubuntu10) [Installed]
  Depends: libgcc1 [Installed]
  Depends: tzdata [Installed]
  Depends: libc-bin
  Depends: libgcc1
  Depends: gcc-4.6-base (= 4.6.3-

```



```

1ubuntu5) [Installed]
  Depends: libc6 (>= 2.2.4)
[Installed]
  PreDepends: multiarch-support
[Installed]
gcc-4.6-base
multiarch-support
  Depends: libc6 (>= 2.13-
0ubuntu6) [Installed]
tzdata
  Depends: debconf (>= 0.5)
[Installed]
  Depends: debconf-2.0
[NotInstalled]
debconf
  PreDepends: perl-base (>=
5.6.1-4) [Installed]
perl-base
  PreDepends: dpkg (>= 1.14.20)
[Installed]
  PreDepends: libc6 (>= 2.11)
[Installed]
dpkg
  PreDepends: coreutils (>= 5.93-
1) [Installed]
  PreDepends: libbz2-1.0
[Installed]
  PreDepends: libc6 (>= 2.11)
[Installed]
  PreDepends: libselinux1 (>=
1.32) [Installed]
  PreDepends: tar (>= 1.23)
[Installed]
  PreDepends: xz-utils
[Installed]
  PreDepends: zlib1g (>= 1:1.1.4)
[Installed]
coreutils
  Depends: dpkg (>= 1.15.4)
[Installed]
  Depends: install-info
[Installed]
  PreDepends: libacl1 (>= 2.2.51-
5) [Installed]
  PreDepends: libattr1 (>=
1:2.4.46-5) [Installed]
  PreDepends: libc6 (>= 2.15)
[Installed]
  PreDepends: libselinux1 (>=
1.32) [Installed]
install-info
  Depends: libc6 (>= 2.4)
[Installed]
libacl1
  Depends: libattr1 (>= 2.4.46-3)
[Installed]
  Depends: libc6 (>= 2.4)
[Installed]
  PreDepends: multiarch-support
[Installed]
libattr1
  Depends: libc6 (>= 2.4)
[Installed]
  PreDepends: multiarch-support
[Installed]
libselinux1
  Depends: libc6 (>= 2.8)
[Installed]
  PreDepends: multiarch-support
[Installed]
libbz2-1.0
  Depends: libc6 (>= 2.4)
[Installed]
  PreDepends: multiarch-support
[Installed]
tar
  PreDepends: libc6 (>= 2.8)
[Installed]
xz-utils
  Depends: libc6 (>= 2.7)
[Installed]
  Depends: liblzma5 (>=
5.1.1alpha+20110809) [Installed]
liblzma5
  Depends: libc6 (>= 2.4)
[Installed]
  PreDepends: multiarch-support
[Installed]
zlib1g
  Depends: libc6 (>= 2.4)
[Installed]
  PreDepends: multiarch-support
[Installed]
debconf-2.0

```

8. apt-rdepends: Show NotInstalled Dependencies

Synopsis: `apt-rdepends -p --state-show=NotInstalled --state-follow=NotInstalled <package>`

Example: `apt-rdepends -p --state-show=NotInstalled --state-follow=NotInstalled xchat`

This command is the clearest of `apt-rdepends` when we want to know exactly what dependencies for a package to be downloaded for one Ubuntu system. This command is same with `apt-get --print-uris` below except this `apt-rdepends` doesn't print URL. This command firstly list the complete dependencies of package, then filter only the packages `NotInstalled` to show. It discards all packages has status `Installed`. It is basically the same when user see dependencies list every `sudo apt-get install` performed. So, with this command you will able to note the dependencies and go download them one by one then go back to install them completely. It will be same with `apt-get install`. Output from our example above will be like this.

```
Reading package lists... Done
Building dependency tree
Reading state information... Done
xchat
  Depends: libsexy2 (>= 0.1.8) [NotInstalled]
  Depends: xchat-common (= 2.8.8-3ubuntu12) [NotInstalled]
libsexy2
xchat-common
```

So, if we want to install `xchat` on our Ubuntu 12.04, we just need to download `libsexy2` and `xchat-common` packages then install them manually with `dpkg`.

9. `apt-cache`: Show Only Important Dependencies

Synopsis: `apt-cache depends --recurse --no-recommends --no-suggests --no-conflicts --no-breaks --no-replaces --no-enhances <package>`

Example: `apt-cache depends --recurse --no-recommends --no-suggests --no-conflicts --no-breaks --no-replaces --no-enhances ed`

This command will track and show all dependencies but discards unneeded dependencies because of `--no-*` options. Important means only `Depends` and `Pre-depends` statuses will be printed. We choose `ed` package here to avoid long output. This command is practically same with next command.

```

ed
  Depends: libc6
  |Depends: dpkg
  Depends: install-info
libc6
  Depends: libc-bin
  Depends: libgcc1
  Depends: tzdata
dpkg
  PreDepends: libbz2-1.0
  PreDepends: libc6
  PreDepends: libselinux1
  PreDepends: zlib1g
  PreDepends: coreutils
  PreDepends: tar
  PreDepends: xz-utils
install-info
  Depends: libc6
libc-bin
libgcc1
  Depends: gcc-4.6-base
  Depends: libc6
  PreDepends: multiarch-support
tzdata
  |Depends: debconf
  Depends: <debconf-2.0>
  cdebconf
  debconf
libbz2-1.0
  Depends: libc6
  PreDepends: multiarch-support
libselinux1
  Depends: libc6
  PreDepends: multiarch-support
zlib1g
  Depends: libc6
  PreDepends: multiarch-support
coreutils
  |Depends: dpkg
  Depends: install-info
  PreDepends: libacl1
  PreDepends: libattr1
  PreDepends: libc6
  PreDepends: libselinux1
  tar
  PreDepends: libc6
  xz-utils
  Depends: libc6
  Depends: liblzma5
  gcc-4.6-base
  multiarch-support
  Depends: libc6
  debconf
  PreDepends: perl-base
  cdebconf
  Depends: libc6
  Depends: libdebconf-indebconf4
  Depends: libnewt0.52
  Depends: libslang2
  Depends: libtextwrap1
  Depends: debconf
  Depends: dpkg
  libacl1
  Depends: libattr1
  Depends: libc6
  PreDepends: multiarch-support
  libattr1
  Depends: libc6
  PreDepends: multiarch-support
  liblzma5
  Depends: libc6
  PreDepends: multiarch-support
  perl-base
  PreDepends: libc6
  PreDepends: dpkg
  libdebconf-indebconf4
  Depends: libc6
  libnewt0.52
  Depends: libc6
  Depends: libslang2
  libslang2
  Depends: libc6
  PreDepends: multiarch-support
  libtextwrap1
  Depends: libc6
  <debconf-2.0>

```

10. apt-cache depends --recurse --important <package>

Synopsis: apt-cache depends --recurse -i <package>

Example: apt-cache depends --recurse -i ed

This command is shorten form of the previous command. The output is same with the previous.

11. apt-cache depends --recurse --important --installed <package>

Synopsis: apt-cache depends --important --installed <package>

Example: apt-cache depends --important --installed ed

This command is same with previous except this prints only installed dependencies. It is useful, but is not useful when you are looking for dependencies that must be downloaded. The output will be shorter like this.

```
ed                PreDepends: multiarch-support
                 Depends: libc6
                 |Depends: dpkg
                 Depends: install-info
libc6             PreDepends: multiarch-support
                 Depends: libc-bin
                 Depends: libgcc1
                 Depends: tzdata
dpkg             PreDepends: libbz2-1.0
                 PreDepends: libc6
                 PreDepends: libselinux1
                 PreDepends: zlib1g
                 PreDepends: coreutils
                 PreDepends: tar
                 PreDepends: xz-utils
install-info     Depends: libc6
libc-bin
libgcc1          Depends: gcc-4.6-base
                 Depends: libc6
                 PreDepends: multiarch-support
tzdata           |Depends: debconf
                 cdebconf
                 debconf
libbz2-1.0       Depends: libc6
                 PreDepends: multiarch-support
libselinux1      Depends: libc6
                 PreDepends: multiarch-support
                 libattr1
                 PreDepends: multiarch-support
                 zlib1g
                 Depends: libc6
                 PreDepends: multiarch-support
                 coreutils
                 Depends: install-info
                 PreDepends: libacl1
                 PreDepends: libattr1
                 PreDepends: libc6
                 PreDepends: libselinux1
tar              PreDepends: libc6
xz-utils         Depends: libc6
                 Depends: liblzma5
gcc-4.6-base    multiarch-support
                 Depends: libc6
debconf          PreDepends: perl-base
cdebconf         Depends: libc6
                 Depends: libnewt0.52
                 Depends: libslang2
                 Depends: debconf
                 Depends: dpkg
libacl1          Depends: libattr1
                 Depends: libc6
                 PreDepends: multiarch-support
libattr1
```

```

Depends: libc6
PreDepends: multiarch-support
liblzma5
Depends: libc6
PreDepends: multiarch-support
perl-base
PreDepends: libc6

PreDepends: dpkg
libnewt0.52
Depends: libc6
Depends: libslang2
libslang2
Depends: libc6
PreDepends: multiarch-support

```

12. apt-get: List All Dependencies URL

Synopsis: `sudo apt-get install --yes --print-uris <package> | grep \' | cut --delimiter=\' --field=2`

Example: `sudo apt-get install --yes --print-uris xchat | grep \' | cut --delimiter=\' --field=2`

This command is actually a combo for three different commands (`apt-get install`, `grep`, and `cut`). This firstly will print exactly all complete dependencies URLs plus many other additional information, then take only the URL lines, then discards any other additional information except the URLs. So, we will have a complete list of URLs to be downloaded separately. Note them into a paper, download manually on another computer, take it back to offline computer, and finally install them with `dpkg`. This is an emulation for complete `apt-get install` sessions when user doesn't have direct internet connection. The output for this command is like this.

```

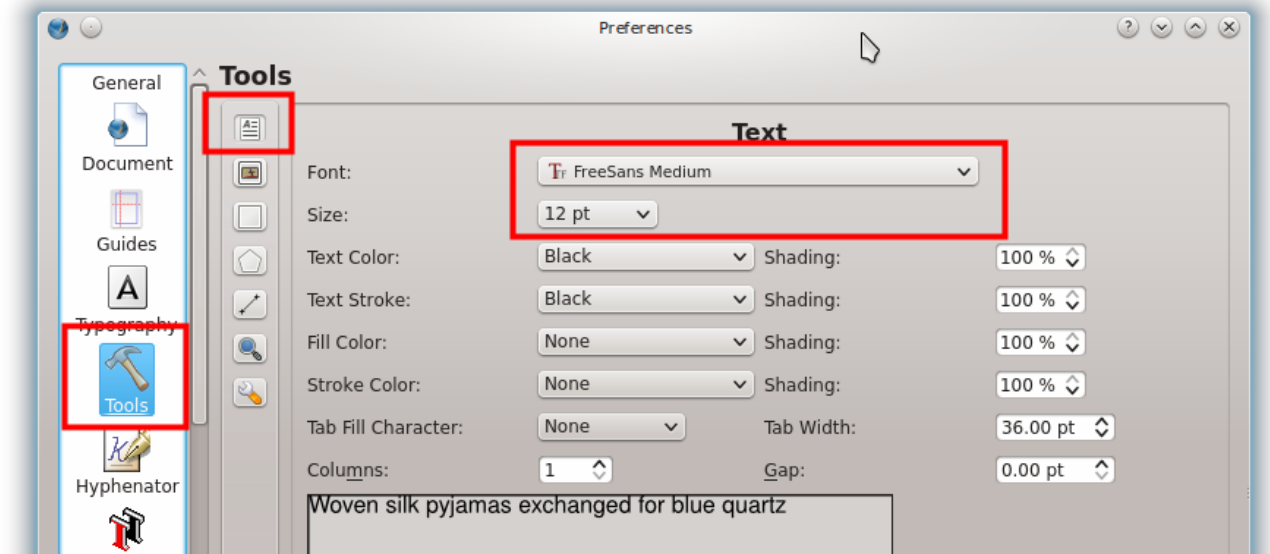
http://kambing.ui.ac.id/ubuntu/pool/main/libs/libsexy/libsexy2_0.1
.11-2build2_i386.deb
http://kambing.ui.ac.id/ubuntu/pool/universe/x/xchat/xchat-
common_2.8.8-3ubuntu12_all.deb
http://kambing.ui.ac.id/ubuntu/pool/universe/x/xchat/xchat_2.8.8-
3ubuntu12_i386.deb

```

Note: our Ubuntu 12.04 sets the repository to Kambing repository in Indonesia. So the URL will be like above. Same thing will happen on your system.

CHANGE SCRIBUS DEFAULT FONT

In Scribus, when we use Text Frame, it will choose one font by default. Sometimes, we need to change this default font. To do it, go to menu File > Preferences > Tools > Text Frame Properties > change the default font. In this example, we choose FreeSans Medium 12pt. So, every new frame in new document will have this font.



Note: additional options above such as Text Color can be adjusted for more specific needs. For example, select Text Color green if you want all text later to be green. Press Defaults button to reset any option you change to default.

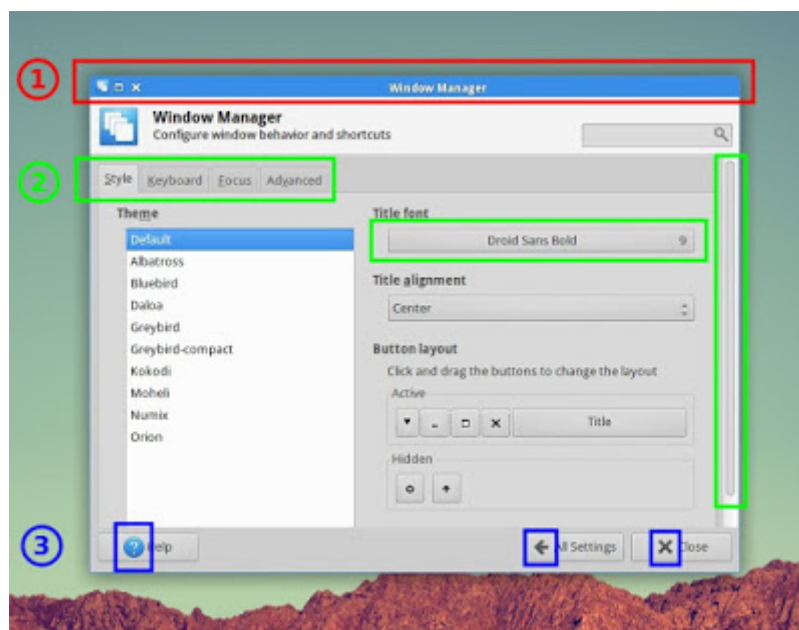
BEGINNER XFCE DESKTOP CUSTOMIZATION GUIDE

Our latest desktop customization guide was Cinnamon. Now, we're moving into XFCE. This guide can be applied to customize Xubuntu, Manjaro XFCE, openSUSE XFCE Spin, or any Linux distro using XFCE as the desktop. We cover how to install desktop theme, add new panel, add applet, and we start them from concepts. Enjoy.

Concepts

1. XFCE has 3 kinds of theme: GTK Theme, XFWM4 Theme, and icon theme.
2. Installing a kind of theme needs enabling the theme.
3. XFCE customization is centralized in XFCE System Settings.
4. XFWM4 (XFCE Window Manager 4) is XFCE Window Manager.

Window Anatomy



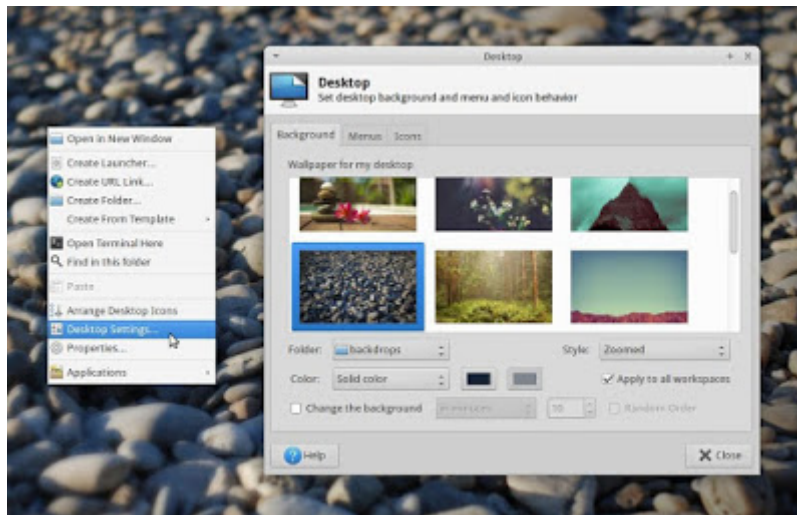
1. Red Box: window decoration (sometimes called window border), controlled by window theme.
2. Green Boxes: GTK components. It means toolkit components such as tab, button,

scrollbar, etc. They are controlled by GTK Theme.

3. Blue Boxes: icon components. Controlled by icon theme.

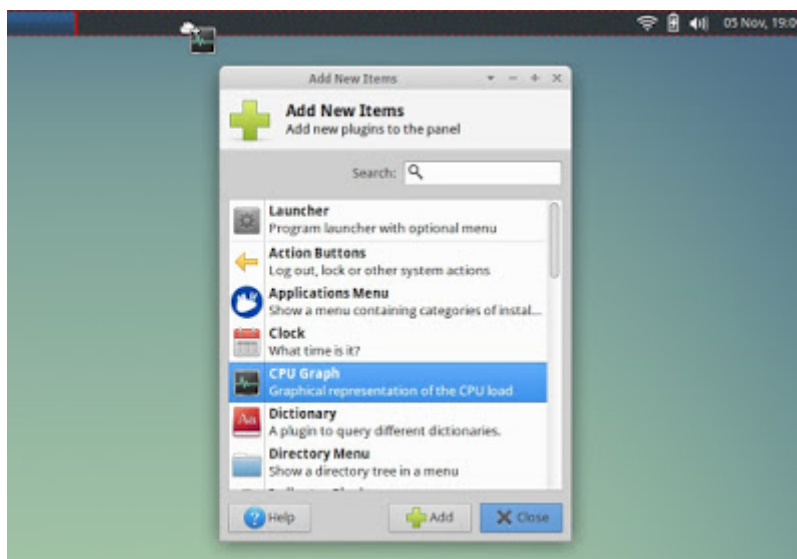
Basically, we custom XFCE desktop by changing these three themes.

Change Wallpaper



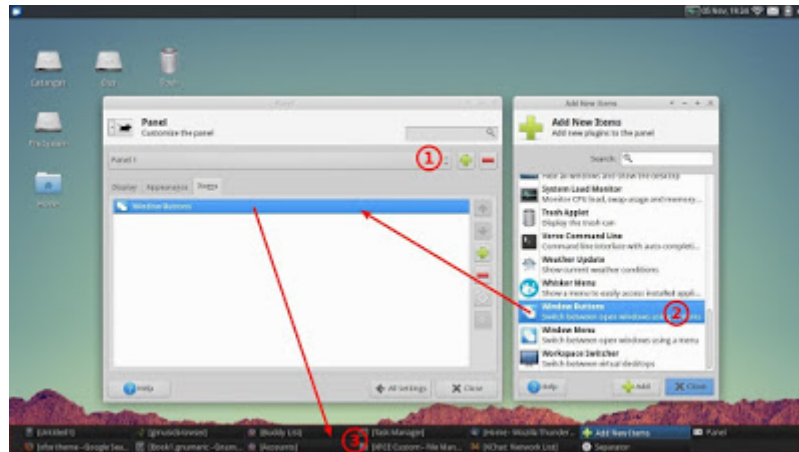
To change wallpaper, right-click on desktop > Desktop Settings > select a wallpaper.

Add Applet



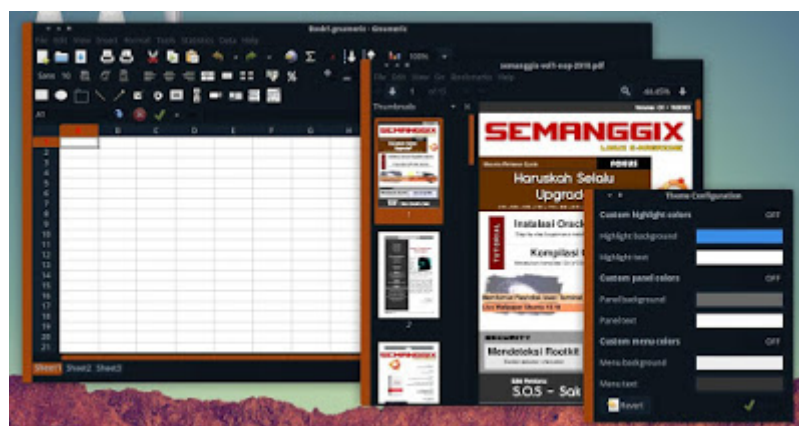
To add an applet into panel, right-click panel > Panel > Add New Items > drag an applet into panel.

New Panel



To add a new panel, open System Settings > Panel > press plus button (1) > a panel created > press plus button (below the first plus) > add an applet to newly created panel (e.g. Window Buttons) > new panel finished. You may need to custom the panel size and width too. Do it on the same window, from Display and Appearance tab.

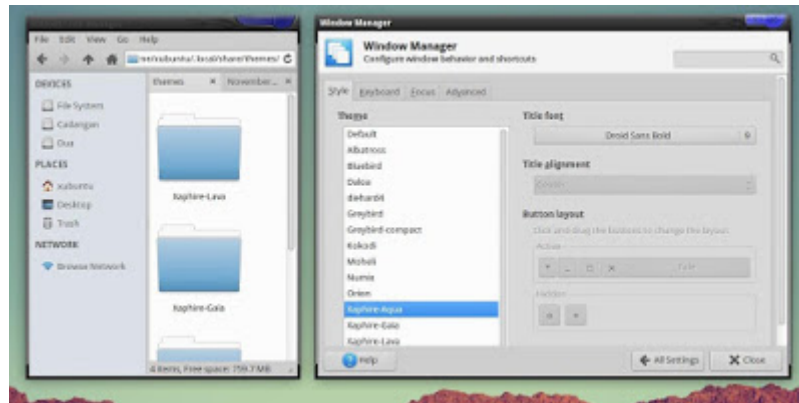
Install XFCE GTK Theme



To install a GTK theme, create `~/.local/share/themes` folder. Copy your GTK theme folder there. Ensure the path to be like this `~/.local/share/themes/<your_theme>/gtk-2.0/gtkrc`. Finally you must enable the GTK theme by System Settings > Appearance > Style > choose your installed GTK theme. If you install theme named Diehard, you will choose Diehard here. We're using [Diehard 4 GTK](#) for this time.

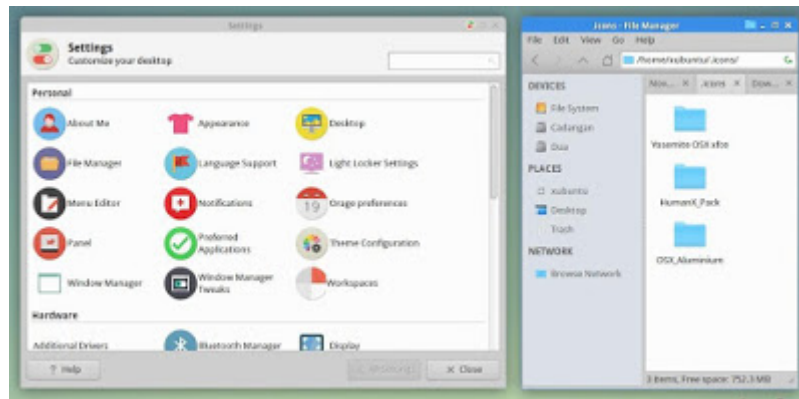
You don't need Terminal to do it. Of course, if you download theme as a .tar.gz archive you must extract the content first somewhere. You just need the GTK theme folder.

Install XFCE Window Decoration Theme



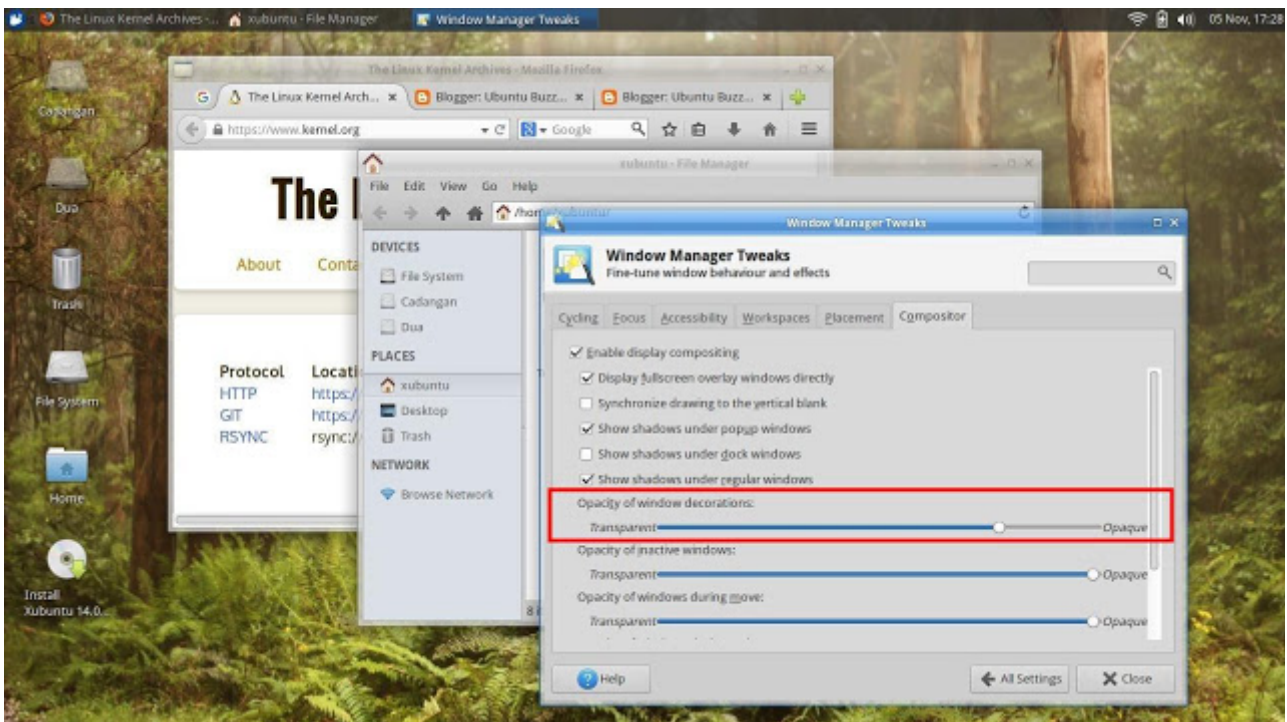
To install window decoration theme (XFWM4 theme), make sure `~/.local/share/themes` is exist. Copy your XFWM theme there. Ensure the complete path will be like this `~/.local/share/themes/<your_theme>/xfwm4`. Finally you must enable the theme from System Settings > Window Manager > Style > choose your theme. All window borders will automatically changed into your theme. In this example, we're using [Xaphire XFWM4 Theme](#).

Install Icon Theme



To install icon theme, create `~/icons`. Then copy your theme folder there. Finally, you must activate the icon too from System Settings > Appearance > Icons > select the newly installed icon. In this example, we're using [Yosemite OS X icons](#).

Translucent Window Decoration



Remember Vista Aero? You can have it on XFCE. To have translucent window borders, open System Settings > Window Manager Tweaks > Compositor > Opacity of window

decorations > slide the slider to make them translucent. See picture above.

References

- [DeviantArt XFCE](#): mainly XFCE themes.
- [Xfce-Look](#): all downloads about XFCE customization.

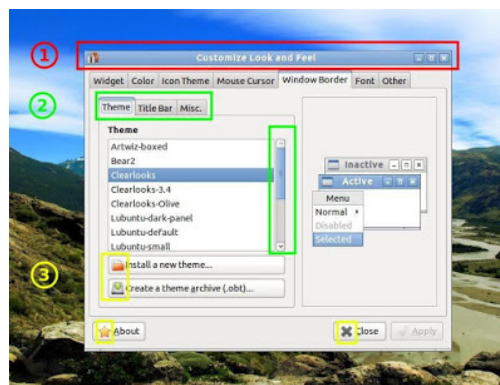
BEGINNER LXDE DESKTOP CUSTOMIZATION GUIDE

We have just written about [XFCE tweaking](#) yesterday. Now we need to explain briefly how to tweak LXDE. LXDE is different because it makes use of Openbox as window manager. Customizing LXDE is customizing Openbox too. Fortunately, LXDE can make use of GTK and Openbox themes available everywhere. We use Ubuntu 14.04 to demonstrate this guide. We hope this basic guide may help anyone uses LXDE. Enjoy.

Concepts

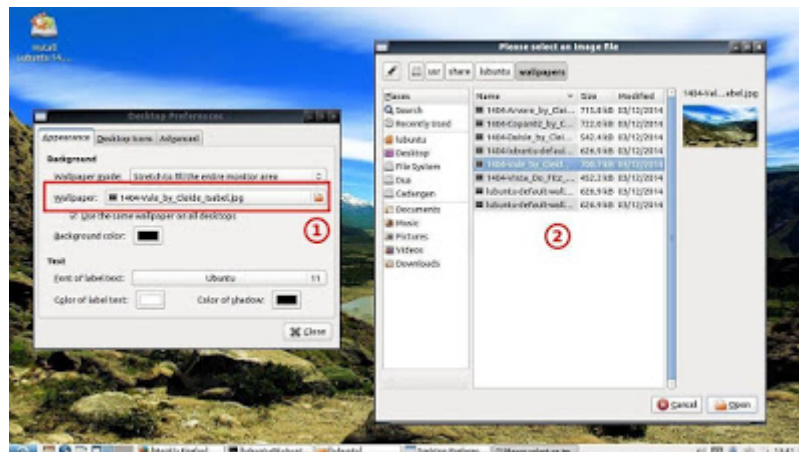
1. LXDE desktop consists of four components: window manager (Openbox), toolkit (GTK+), icon set, and panel.
2. LXDE customization (in Ubuntu) is centralized in LXAppearance program.
3. Openbox theme folder is ~/.theme. The structure should be ~/.theme/<theme_name>/openbox-3.
4. GTK theme folder is ~/.theme. The structure should be ~/.theme/<theme_name>/gtk-2.0.
5. Icon theme folder is ~/.icons.

Window Anatomy



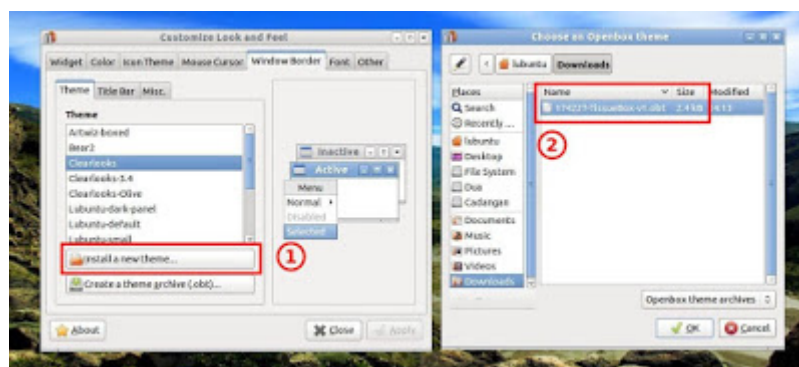
1. Red Box: window decoration (or window border). Controlled by window manager named Openbox.
2. Green Boxes: toolkit components (such as button, scrollbar, tab, etc.). Controlled by GTK theme.
3. Yellow Boxes: icon components.

Change Wallpaper

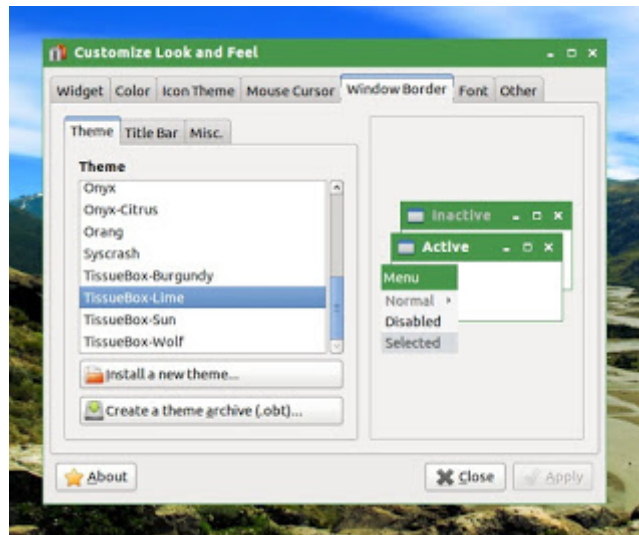


To change LXDE wallpaper, right-click on desktop > Desktop Preferences > click on Wallpaper (1) > select a wallpaper image > Open. LXDE has no wallpaper thumbnails view like GNOME or KDE yet.

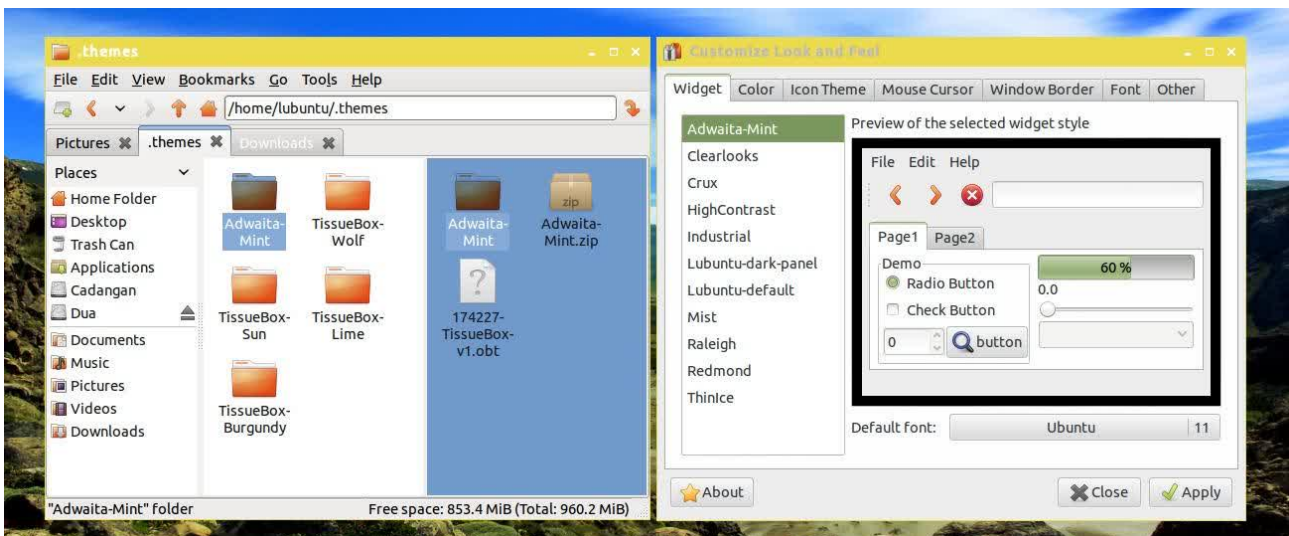
Install Window Theme



LXDE uses Openbox as the window manager. So if we want to change window border, we just need to install Openbox theme. To install Openbox theme (.obt file), open menu > Preferences > Customize Look and Feel > Window Border > Install a new theme ... > choose an OBT file > press Apply. The borders will change immediately.



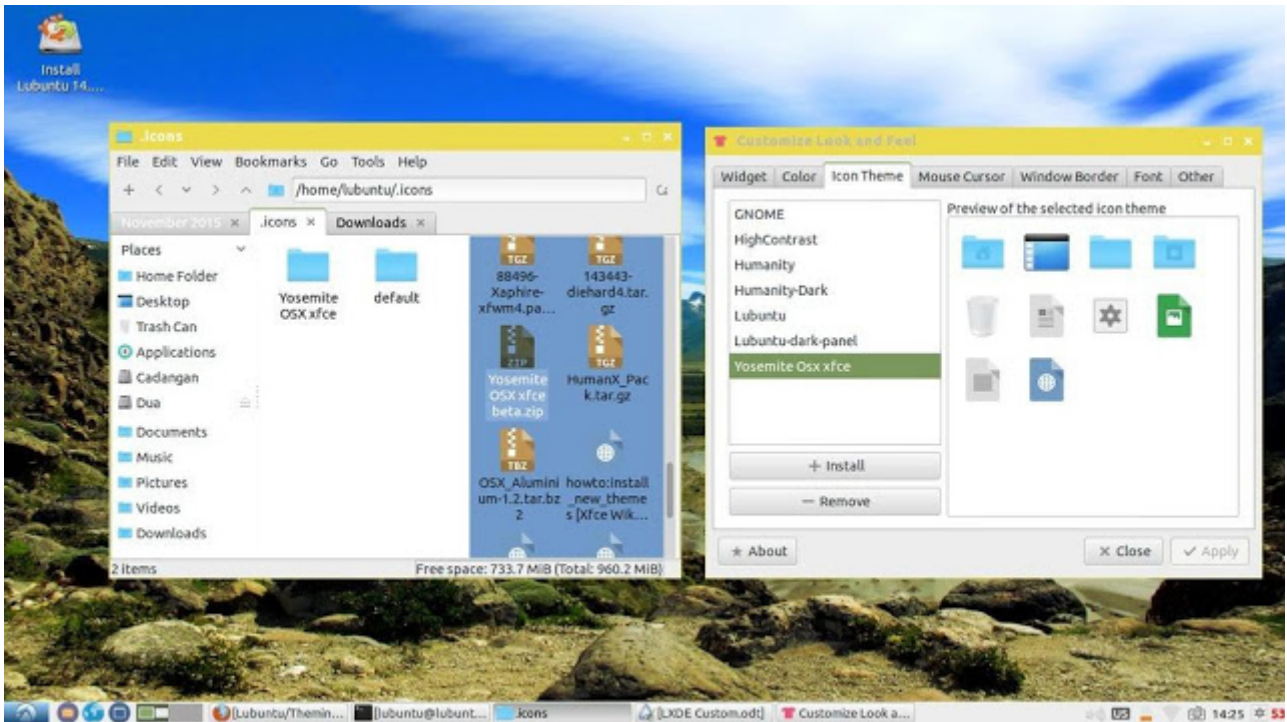
Install GTK Theme



As information, GTK 2.0 theme (like in the XFCE yesterday) can be installed in LXDE. To install GTK theme in LXDE, copy theme folder (i.e. Adwaita Mint like picture above) into

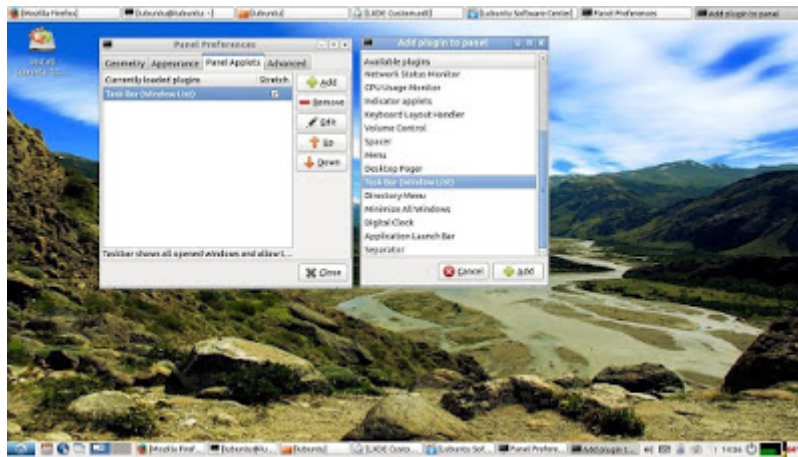
~/themes folder. Finally, enable GTK theme (Adwaita Mint) in the menu > Preferences > Customize Look and Feel > Widget by click and Apply.

Install Icon Theme



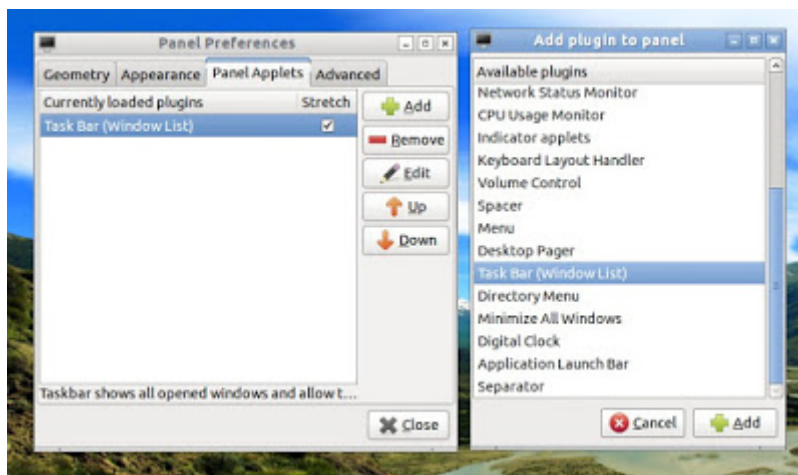
Icon theme for GNOME is installable in LXDE. For example, we can install icon set Yosemite XFCE from our previous tutorial in LXDE. To install icon, open menu > Preferences > Customize Look and Feel > Icon Theme > press Install button > select an icon archive file. Finally select installed icon to enable it. See picture above. Note: we can copy icon set folder (i.e. Yosemite OSX XFCE) into ~/icons folder.

Add Panel



To add a panel in LXDE, right-click on a panel > Create New Panel > a panel created. This is a plain panel. We need to add it some applets.

Add Applet



To add applet into panel, right-click panel > Panel Settings > Panel Preferences window > Panel Applets tab > press Add button > select applet.

References

- [Openbox Wiki Howto](#): official Openbox guide.
- [Openbox Wiki Theme](#): Openbox wiki theming.
- [BOX Look](#): download Openbox themes.

- [GNOME Look](#): download icons and GTK themes.
- [Openbox Themes Download](#): official Openbox themes.
- [DeviantArt Openbox](#): search Openbox artwork.
- [Lubuntu Wiki](#): LXDE theming guide from Ubuntu Official.

CONFIGURING APT.CONF PROXY AUTHENTICATION FOR APT-GET

Those who use Ubuntu in internet-restricted area such as college or office, probably will need proxy configuration for their apt-get command. Especially in the college, when every student given username and password for campus proxy to connect to the internet. Actually we just need to configure proxy or with/without authentication in apt.conf file. Here we go.

Assumptions

Let assume we have these credential information:

- Proxy server: proxy.server.edu
- Port: 443
- Username: myusername@mycollege.edu
- Password: mypassword

How To Configure

1. Open Terminal.
2. Perform command: `sudo gedit /etc/apt/apt.conf`
3. Type the configuration text below.
4. Save Gedit and exit.
5. Test command `sudo apt-get update` when you are in the proxified area.

How To Revert Configuration

1. Open Terminal.
2. Perform command: `sudo gedit /etc/apt/apt.conf`
3. Type a "#" sign without quotes in every beginning of line exactly before Acquire.

4. Save Gedit and exit.
5. Test command `sudo apt-get update` when you are not in proxyfied area.

Configuration Text

Given above assumptions, so our configuration will be like these.

```
Acquire::http::proxy  
"http://myusername@mycollege.edu:mypassword@proxy.college.edu:443"
```

HOW TO ADD UBUNTU PPA MANUALLY

Usually we can add a new PPA repository by adding it into Software Sources (software-properties-gtk) either from Ubuntu Software Center or Synaptic. But actually we can add same PPA manually. This manual method will help user to easily remove unused PPA later.

Search for Launchpad PPA Page

First, search for a PPA. For example, we can type shutter ppa on Google if we want Shutter application PPA. We must find an URL like this <https://launchpad.net/~shutter/+archive/ubuntu/ppa> and we must see the page contains "Technical details about this PPA" section. Otherwise do search again.

Shutter Team: "Shutter Team" team - Mozilla Firefox

Canonical Group Ltd (GB) <https://launchpad.net/~shutter/+archive/ubuntu/ppa>

"Shutter Team" team

Overview Code Bugs Blueprints Translations Answers

PPA for Shutter Team

PPA description

This is the official Shutter repository - it is recommended to use this to keep your Shutter easily updated.

For instructions on how to add the repository, see (<http://shutter-project.org/faq-help/ppa-installation-guide>).

Adding this PPA to your system

You can update your system with unsupported packages from this untrusted PPA by adding `ppa:shutter/ppa` to your system's Software Sources. ([Read about installing](#))

[Technical details about this PPA](#)

This PPA can be added to your system manually by copying the lines below and adding them to your system's software sources.

Display sources.list entries for:

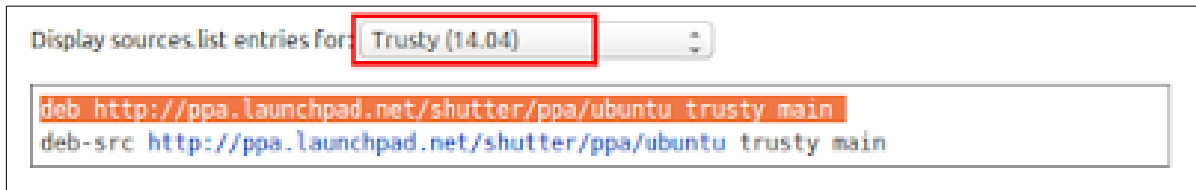
```
deb http://ppa.launchpad.net/shutter/ppa/ubuntu YOUR_UBUNTU_VERSION_HERE main
deb-src http://ppa.launchpad.net/shutter/ppa/ubuntu YOUR_UBUNTU_VERSION_HERE main
```

Signing key:

PPA status
Activity
0 updates added

Select Ubuntu Version

At Technical details section, select Ubuntu version. For example, we use 14.04 so we select Trusty 14.04. This selection is helpful so we just need to copy the text.



Copy PPA Source List

After selecting, we will find text lines like this. These are PPA URLs or Ubuntu calls them sources list. Notice that deb line is for binary package, and deb-src line is for source code package. We install binary package so copy the deb line only. Copy only the first line.

```
deb http://ppa.launchpad.net/shutter/ppa/ubuntu trusty main
deb-src http://ppa.launchpad.net/shutter/ppa/ubuntu trusty main
```

Paste PPA into Your sources.list File

Perform command `sudo gedit /etc/apt/sources.list` on the Terminal. Gedit text editor will appear to open our Ubuntu sources.list. Paste text into the new line. Our sources.list fill content will be like this. Notice the last line.

```
deb http://archive.ubuntu.com/ubuntu/ trusty main restricted
multiverse universe
deb http://security.ubuntu.com/ubuntu/ trusty-security main
restricted
deb http://archive.ubuntu.com/ubuntu/ trusty-updates main
restricted
deb http://ppa.launchpad.net/shutter/ppa/ubuntu trusty main
```

Reload

After adding a new repository (in this case, PPA), do command `sudo apt-get update`. This will reload the new repository so apt-get later can install from it.

Install

If the PPA is Shutter, then we will install shutter package. The command is `sudo apt-get install shutter`. Otherwise, instead using Terminal we can install it with Ubuntu Software Center or Synaptic.

HOW TO INSTALL SCRIBUS 1.5 NG IN UBUNTU 14.04

Scribus 1.5 NG has released both in upstream (developer source code) and in downstream (i.e. PPA repository). Scribus 1.5 for Ubuntu is available for 14.04, 15.04, 15.10, and 16.04. Notice that it doesn't provide 14.10. Although 1.5 is a testing version (a preview for the next 1.6 release), probably you want to test the new features or its stability on an experiment environment (i.e. live session). This guide needs you to open Terminal (Ctrl+Alt+T) and perform three commands. You will need internet connection to do it.

Add PPA

```
sudo apt-add-repository ppa:scribus/ppa
```

Reload

```
sudo apt-get update
```

Install

```
sudo apt-get install scribus-ng
```

Notes

- You can manually add repository by following our previous article.

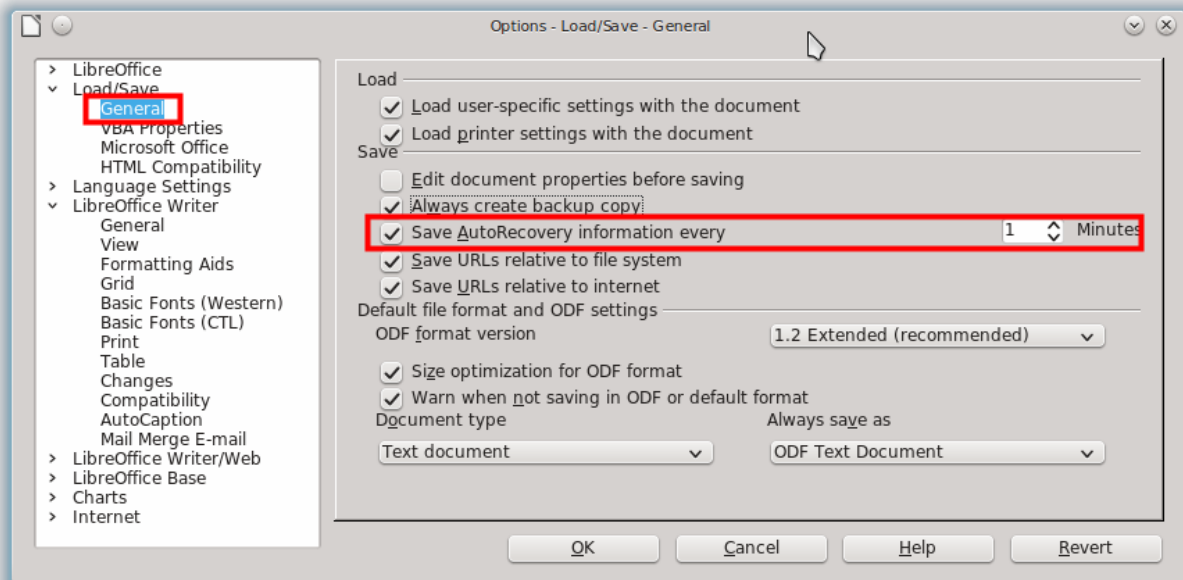
- You can install this Scribus 1.5 with Synaptic or Ubuntu Software Center after doing step Reload if you want to avoid Terminal usage.
- In Ubuntu 14.04, you will need to download totally about 60 MB packages.

Reference

<https://launchpad.net/~scribus/+archive/ubuntu/ppa>

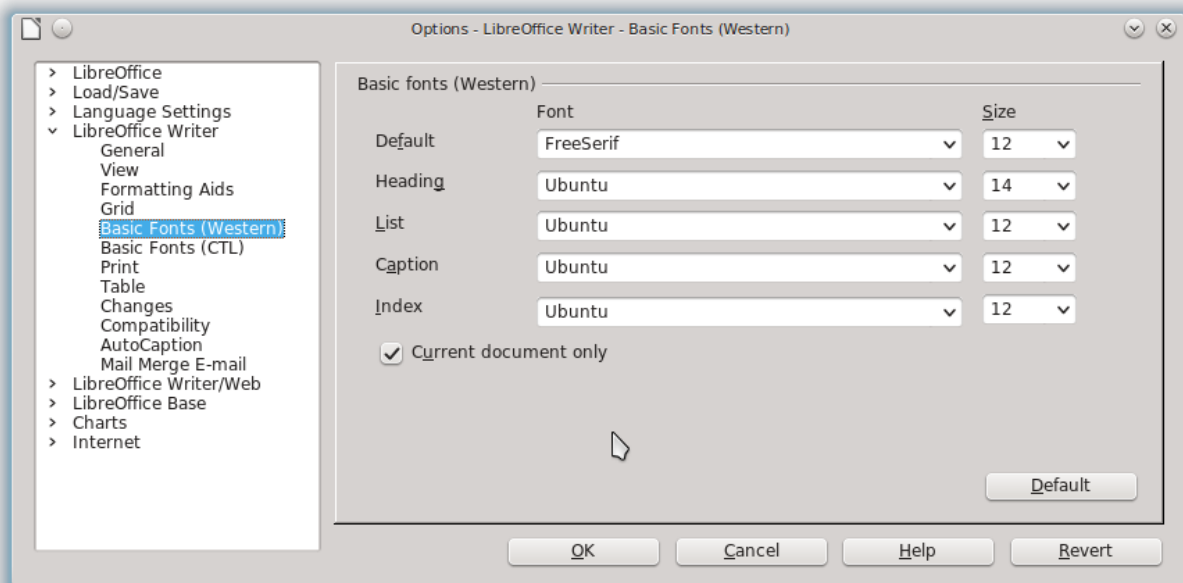
ENABLE AUTOSAVE IN LIBREOFFICE

To enable autosave in LibreOffice, go to menu Tools > Options > Load/Save > General > check Save AutoRecovery information every option > give a time value. For example, we give it 1 minute so LibreOffice will do autosave every 1 minute. You can also check Always create backup copy option so LibreOffice will always save hidden file as a copy of your current document. That kind of file is very useful after the computer experienced power failure or any other disaster. See picture below.



CHANGE LIBREOFFICE WRITER DEFAULT FONT

When user writes long document such as book, it is easier if they can change whole headings' fonts by few clicks only. Actually, some important parts in LibreOffice Writer document such as default font, heading, list, caption, and index have their own setting. So it is possible to change just one or whole types. To change default font in Writer, open menu Tools > Options > LibreOffice Writer > Basic Fonts (Western) > choose what type font you need to change. See picture below.



For example, if you want to change all headings on document to be FreeSerif 12pt, so choose on Heading field "FreeSerif" and insert "12" value in Size field. Note: if you want these changes apply into every new document later, don't check Current document only option. If you check it, then the changes only apply on current document.

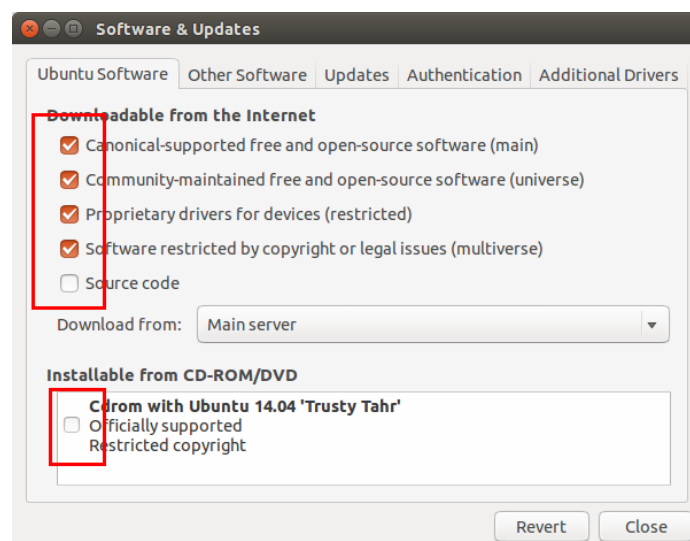
UBUNTU SOFTWARE CENTER BEGINNER GUIDE

Add/remove program system is the core functionality in every operating system. In Ubuntu nowadays, this functionality is handled by Ubuntu Software Center (will mentioned as USC). This is a beginner guide about using USC, introducing you basically how to do add/remove program in Ubuntu. USC resembles Google Play Store for Android user or App Store for Apple OS X and iOS user. Enjoy.

Concepts

1. Ubuntu Software Center is a program to help user download and install software from Ubuntu repository server.
2. Ubuntu Software Center shows applications with friendly appearance with icon and description.
3. Ubuntu Software Center simplify all process to install, just like Google Play Store in Android.
4. Ubuntu Software Center is technically a package manager front-end which replaced Ubuntu's old one Synaptic Package Manager. USC is simpler in usability than Synaptic.

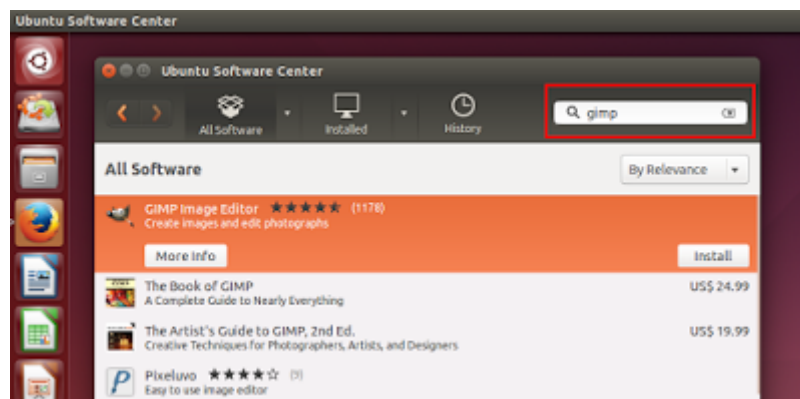
First Time Usage



Open Ubuntu Software Center > Edit > Software Sources > check all 4 repository components (see picture above) > uncheck Cdrom with Ubuntu... option (see picture above) > press Close button > let USC downloads index files. It will need approximately 15 MB download.

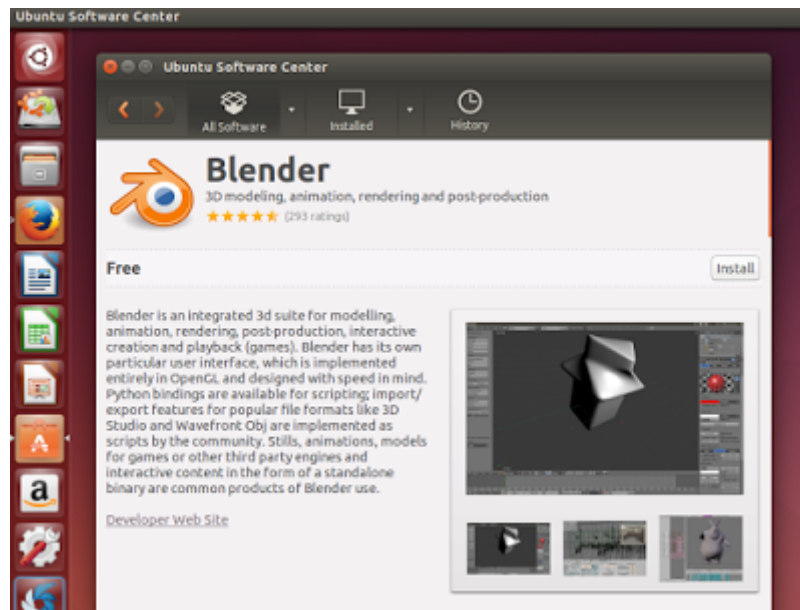
This action is called "Reload". At every first time usage of USC, you will need to do "reload" at least once. Reload means setting correct repository and downloading index of repository being used. Every time repository setting changed, you must do reload once. It is not upgrading any application, nor installing any application. It's just downloading index files (the map of server contents). Once you've done reload, you can install any application you want in USC.

Search Application



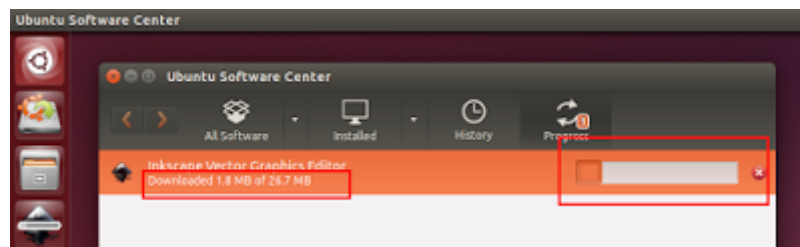
After you've done reloading, you can search for any application. Type on the search box at top-left corner or browse manually by category. In this example, we show you GIMP searching resulting in GIMP program, GIMP books, and some other related programs.

Show Application Details



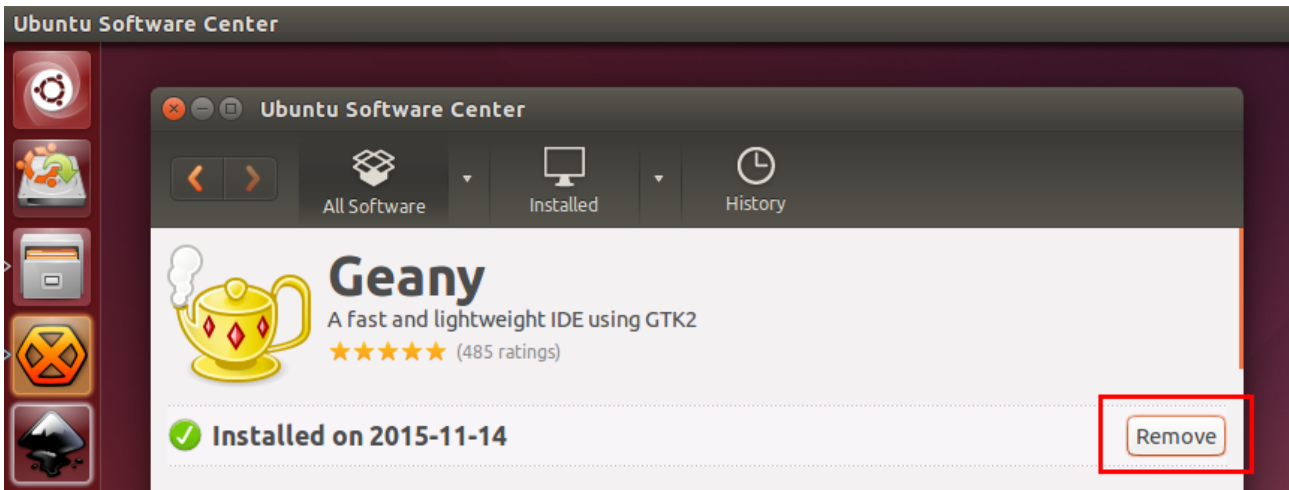
To show application details, click on More Info button on every application entry. You can see program description, screenshots, rating, technical information, and additionally comments by USC users worldwide.

Install Application



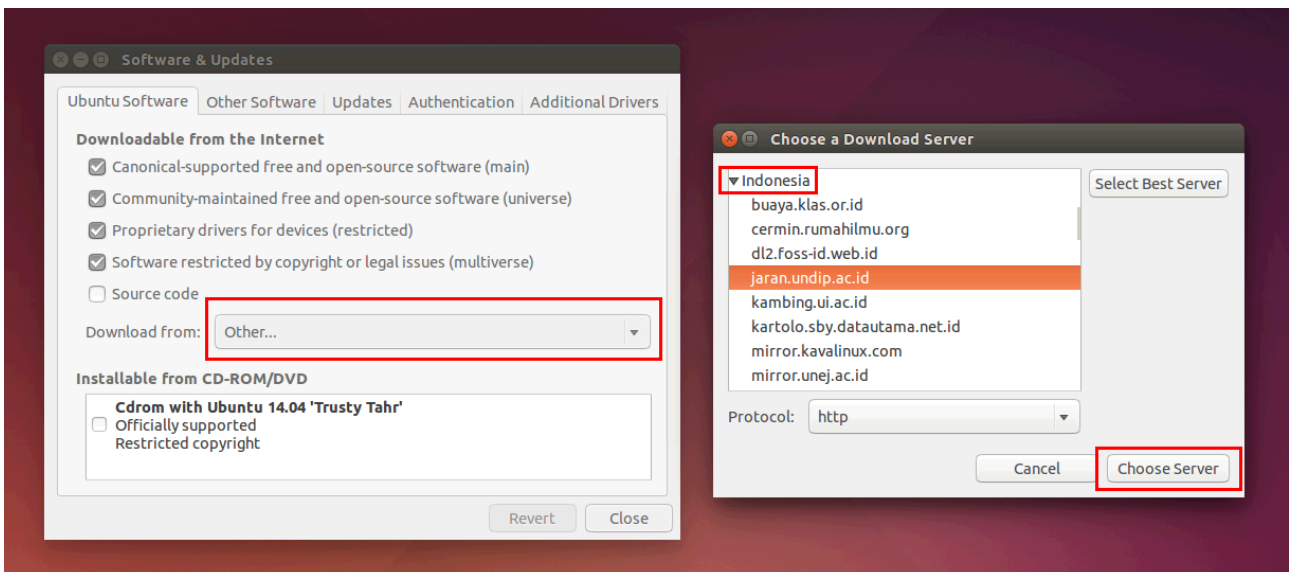
To install application, search first (i.e. inkscape) > click Install button > wait for downloading progress > finish. You can see your installed program on Ubuntu menu.

Uninstall Application



To uninstall an application (i.e. Geany), search for its name > click Remove button. This action doesn't need internet connection.

Change Repository



To change repository (e.g. select local repository on your country), open menu Edit > Software Sources > Download From: Other > select a country (i.e. Indonesia) > select one repository server selection (i.e. jaran.undip.ac.id) > Choose Server > close Software & Updates. USC will automatically reload your new repository choice by downloading index files once again. It needs approximately 15 MB to download.