

## About Ubuntu

Ubuntu is a community developed Linux-based operating system that is perfect for laptops, desktops and servers. Whether you use it at home, at school or at work Ubuntu contains all the applications you'll ever need, from word processing and email applications, to web server software and programming tools.

Ubuntu is an African word meaning "Humanity to others", or "I am what I am because of who we all are". The Ubuntu distribution brings the spirit of Ubuntu to the software world.



Ubuntu  
[www.ubuntu.com](http://www.ubuntu.com)



Ubuntu and Ubuntu logo are trademarks of Canonical, Ltd.

## Ubuntu Promise

Ubuntu will always be free of charge, including enterprise releases and security updates.

Ubuntu comes with full commercial support from Canonical and hundreds of companies around the world.

Ubuntu includes the very best translations and accessibility infrastructure that the free software community has to offer.

Ubuntu CDs contain only free software applications; we encourage you to use free and open source software, improve it and pass it on.

\*\*\*

The current version of Ubuntu, 9.10, was released on October 29, 2009

**Fultus™**



Published by Fultus Corporation  
[www.fultus.com](http://www.fultus.com)



Ubuntu  
[www.ubuntu.com](http://www.ubuntu.com)



**ubuntu**  
linux for human beings

# Ubuntu 9.10

## Packaging Guide



By Ubuntu Documentation Project





# Ubuntu 9.10

## Ubuntu Packaging Guide



*Fultus™ Books*



## Ubuntu 9.10

# Ubuntu Packaging Guide

ISBN-10: 1-59682-174-4

ISBN-13: 978-1-59682-174-3

Copyright © 2009 Canonical Ltd. and members of the Ubuntu Documentation Project

Cover design and book layout by Fultus Corporation



*Published by Fultus Corporation*

Publisher Web: *www.fultus.com*

Linbrary - Linux Library: *www.linbrary.com*

Online Bookstore: *store.fultus.com*

email: *production@fultus.com*



This material may only be distributed subject to the terms and conditions set forth in the Creative Commons  
ShareAlike 3.0 License (CC-BY-SA),  
(the latest version is presently available at <http://creativecommons.org/licenses/by-sa/3.0/legalcode>).

Ubuntu, Canonical and Ubuntu logo are trademarks or registered trademarks of Canonical Ltd., Inc., in the U.S. and other countries. All product names and services identified throughout this manual are trademarks or registered trademarks of their respective companies.

The author and publisher have made every effort in the preparation of this book to ensure the accuracy of the information. However, the information contained in this book is offered without warranty, either express or implied. Neither the author nor the publisher nor any dealer or distributor will be held liable for any damages caused or alleged to be caused either directly or indirectly by this book.

---

# Table of Contents

<b>Credits and License</b> .....	<b>5</b>
<b>Chapter 1. Introduction</b> .....	<b>6</b>
1.1. Where to Begin .....	6
1.2. Prerequisites .....	6
<b>Chapter 2 Getting Started</b> .....	<b>8</b>
2.1. Binary and Source Packages.....	8
2.2. Packaging Tools .....	8
2.3. The Personal Builder: pbuilder .....	9
<b>Chapter 3. Basic Packaging</b> .....	<b>11</b>
3.1. Packaging from Scratch.....	11
3.1.1. changelog.....	14
3.1.2. control .....	15
3.1.3. copyright.....	17
3.1.4. rules .....	18
3.1.5. Building the Package Locally (Binary Only) .....	20
3.1.6. Building the Package (Source and Binary) .....	21
3.1.7. Requirements for New Ubuntu Packages.....	22
3.1.7.1. Packaging .....	22
3.1.7.2. Other .....	22
3.2. Packaging with CDBS .....	22
3.3. Packaging Python modules with CDBS .....	26
3.4. Advanced Packaging.....	26
3.4.1. Creating More Than One Binary Package .....	26
3.5. Packaging Shared Libraries .....	27
3.5.1. Shared Libraries.....	27
3.5.1.1. Where a symbol comes from .....	27
3.6. Common Mistakes .....	29
3.6.1. dh_make Example Files.....	29
3.6.2. Abusing dh_installdirs .dirs Files .....	29
3.6.3. Changing the Original Tarball.....	30
3.6.3.1. Solutions .....	31
3.6.3.2. Tips.....	32
3.6.4. Copyright Information .....	33
3.6.4.1. Example 1: GPL .....	34

## Ubuntu 9.10

---

3.6.4.2. Example 2: GPL with different GPL derived files from a different author.....	35
3.6.4.3. General Advice.....	36
3.7. Reference Packages.....	36
<b>Chapter 4. Patch Systems .....</b>	<b>37</b>
4.1. General Tips .....	37
4.2. Patching Without a Patch System .....	38
4.2.1. Example 1 .....	40
4.2.2. Example 2 .....	40
4.3. CDBS with Simple Patchsys (Example Package: pmount) .....	40
4.4. dpatch.....	41
4.5. quilt (Example Package: xterm).....	42
4.6. Patching Other People's Packages.....	42
<b>Chapter 5. KDE.....</b>	<b>43</b>
5.1. Essential Packaging Bits.....	43
5.1.1. KDE Manpages.....	44
<b>Chapter 6. Supplementary Files.....</b>	<b>45</b>
6.1. .desktop Files.....	45
6.2. Man Pages.....	47
6.2.1. Using POD .....	47
6.2.2. Basic template .....	48
6.2.3. Some formatting options.....	48
6.2.4. Necessary packaging changes .....	49
<b>Chapter 7. Recipes .....</b>	<b>50</b>
7.1. Updating an Ubuntu Package.....	50
7.1.1. Ingredients .....	50
7.1.2. Method.....	51
7.1.3. For Bonus Points... ..	52
7.2. Creating a Debdiff .....	52
7.2.1. Ingredients .....	52
7.2.2. Method.....	53
7.3. Creating and Using a debian/watch File.....	54
7.3.1. Ingredients .....	55
7.3.2. Method.....	55
7.3.3. Notes .....	57
<b>Appendix A.....</b>	<b>58</b>
A.1. Additional Resources .....	58
A.2. Example files .....	60
A.3. List of Debhelper Tools.....	61
<b>Ubuntu 9.10 Official Documentation Collection.....</b>	<b>63</b>

## Credits and License

The documents on this website are maintained by the Ubuntu documentation team<sup>1</sup>. For a list of contributors, see the contributors page<sup>2</sup>.

Unless stated otherwise on the front page of the document, the documents are made available under the Creative Commons ShareAlike 3.0 License (CC-BY-SA).

You are free to modify, extend, and improve the Ubuntu documentation source code under the terms of this license. All derivative works must be released under this license.

An explanation<sup>3</sup> and copy<sup>4</sup> of the license are available.

---

<sup>1</sup> <https://wiki.ubuntu.com/DocumentationTeam>

<sup>2</sup> <https://help.ubuntu.com/libs/C/contributors.html>

<sup>3</sup> <http://creativecommons.org/licenses/by-sa/3.0/>

<sup>4</sup> <http://creativecommons.org/licenses/by-sa/3.0/legalcode>

# Chapter 1.

## Introduction

Welcome to the *Ubuntu* Packaging Guide! This guide is primarily addressed to those who would like to make and maintain *Ubuntu* packages. Although many of the concepts in this guide could be used to make binary packages for personal use, it is designed for those people wanting to distribute their packages to and for others. While it is also written with the *Ubuntu* Linux distribution in mind, it should also be useful for any Debian-based distribution.

There are several reasons you might want to learn how to package for *Ubuntu*. First, building and fixing *Ubuntu* packages is a great way to contribute to the *Ubuntu* community. It is also a good way to learn how *Ubuntu* and the applications you have installed work. Maybe you want to install a package that is not in the *Ubuntu* repositories. Hopefully after you have completed this guide you will have the tools and knowledge you need to do all of these things.

### 1.1. Where to Begin

If you are completely new to Debian-based packaging then you will want to read this guide completely through, paying special attention to the section called *Prerequisites*, and the section *Basic Packaging*. People who are experienced with Debian-based packaging will find the section *Ubuntu Packaging* most helpful.

### 1.2. Prerequisites

This guide assumes that the reader has a reasonable knowledge of building and installing software from source on Linux distributions. The guide also uses the Command Line Interface (CLI) throughout, so you should be comfortable using a terminal. You should be able to at least use the following:

- **make:** GNU Make is a very important software building tool. It is used to transform a complex compilation task into a trivial one. It is important that you know how to use it, because we will store most of the information about the packaging process in a Makefile. Documentation is available at the *GNU*<sup>1</sup> website.

---

<sup>1</sup> <http://www.gnu.org/software/make/manual/make.html>

---

- **./configure:** This script is included in almost all Linux source, especially for software written in compiled languages such as C and C++. It is used to generate a Makefile (file used by `make`) that is properly configured for your system. Standard Debian packaging tools use it, so it is important that you know what the `configure` script does. Information on `./configure` can be found in the `make` documentation.
- **Apt/Dpkg:** Beyond the basic use of installing programs, `apt` and `dpkg` have many features that are useful for packaging.
  - **apt-cache dump** - lists every package in the cache. This command is especially helpful in combination with a `grep` pipe such as `apt-cache dump | grep foo` to search for packages whose names or dependencies include `foo`.
  - **apt-cache policy** - lists the repositories (`main/restricted/universe/multiverse`) in which a package exists.
  - **apt-cache show** - displays information about a binary package.
  - **apt-cache showsrc** - displays information about a source package.
  - **apt-cache rdepends** - shows reverse dependencies for a package (which packages require the queried one).
  - **dpkg -S** - lists the binary package to which a particular file belongs.
  - **dpkg -l** - lists currently installed packages. This is similar to `apt-cache dump` but for installed packages.
  - **dpkg -c** - lists the contents of a binary package. It is useful for ensuring that files are installed to the right places.
  - **dpkg -f** - shows the control file for a binary package. It is useful for ensuring that the dependencies are correct.
  - **grep-dctrl** - searches for specialized information in packages. It is a specific use of the `grep` package (but not installed by default).
- **diff and patch:**
  - The `diff` program can be used to compare two files and to make patches. A typical example might be `diff -ruN file.old file.new > file.diff`. This command will create a `diff` (recursively if directories are used) that shows the changes, or *delta*, between the two files.
  - The `patch` program is used to apply a patch (usually created by `diff` or another similar program) to a file or directory. To apply the patch created above, we can invoke `patch -p0 < file.diff`. The option `-p` tells `patch` how much it should strip from the paths for the file names in the patch. The option `-p0` means to strip nothing, or leave the path intact.

# Ubuntu 9.10

## Official Documentation Collection

Title	Author	Edition	ISBN- 10	ISBN- 13
Ubuntu 9.10 Installation Guide	Ubuntu Documentation Project	paperback	1-59682-171-X	978-1-59682-171-2
		eBook (pdf)	1-59682-175-2	978-1-59682-175-0
Ubuntu 9.10 Desktop Guide	Ubuntu Documentation Project	paperback	1-59682-172-8	978-1-59682-172-9
		eBook (pdf)	1-59682-176-0	978-1-59682-176-7
Ubuntu 9.10 Server Guide	Ubuntu Documentation Project	paperback	1-59682-173-6	978-1-59682-173-6
		eBook (pdf)	1-59682-177-9	978-1-59682-177-4
Ubuntu 9.10 Packaging Guide	Ubuntu Documentation Project	paperback	1-59682-174-4	978-1-59682-174-3
		eBook (pdf)	1-59682-178-7	978-1-59682-178-1
<i><a href="http://www.linlibrary.com">http://www.linlibrary.com</a></i>				