

KEVIN GODBY

# UBUNTU MANUAL: L<sup>A</sup>T<sub>E</sub>X FOR AUTHORS AND TRANSLATORS



# Contents

- 1 Introduction 5
- 2 The Basics 7
  - Getting Started 7
  - Organization of Files 7
- 3 Style Guide 9
  - Formatting Text 9
  - GUI Elements 9
  - Graphics 10
  - Other document elements 11
- 4 Notes for Translators 15
  - Document headings 15
  - Formatting commands 15
  - Lists 16
  - Author and editor notes 16
  - Glossary-related commands 16
  - Cross-referencing commands 16
  - Other document commands 17
- 5 Notes for Editors 19
  - Obsolete commands 19
- 6 Installing TeX Live 2009 21



# 1 Introduction

The **Ubuntu Manual** project has set out to produce a beginner's guide for Ubuntu, covering everything from installation to commonly used applications. The manual will be provided in PDF format in a variety of languages and will be freely available. To generate the PDFs, we use L<sup>A</sup>T<sub>E</sub>X.

L<sup>A</sup>T<sub>E</sub>X is a typesetting system that has been around for over twenty years. Since L<sup>A</sup>T<sub>E</sub>X is a large (and sometimes complex) system this guide will only cover what you need to know to write, edit, and translate the contents of the Ubuntu Manual.

If you have questions or run into any problems that this guide doesn't cover, feel free to email the Ubuntu Manual mailing list at [ubuntu-manual@lists.launchpad.net](mailto:ubuntu-manual@lists.launchpad.net).



## 2 The Basics

### Getting Started

As an author or editor, you will be modifying the Ubuntu Manual source files directly. The source files for  $\LaTeX$  have a `.tex` extension. You can edit the source files using your favorite text editor such as emacs, vim, or GEdit.  $\LaTeX$  code is similar to HTML in that most of the “code” is simply the text of the manual with a few formatting commands sprinkled in.

Translators should also familiarize themselves with the basics of  $\LaTeX$  formatting and read the special translator notes starting on page 4.

### Organization of Files

Each of the chapters of the manual has its own subdirectory:

Chapter	Directory name
Prologue	prologue
1. Installation	installation
2. Around Your Desktop	around-desktop
3. Default Applications	default-apps
4. Preferences and Hardware	prefs-hardware
5. Software and Packaging	software-packaging
6. System Maintenance	maintenance
7. The Command Line	command-line
8. Security	security
9. Troubleshooting	troubleshooting
10. Learning More About Linux	learning-more

Once you’ve selected a chapter that you’d like to help write or edit, you will find a `.tex` file in that chapter’s directory. Some chapters will have all of their text in that one file, while other chapters have split each section into its own file. If you see a bunch of `\input` commands in the file, then you will have to look in the appropriate `.tex` file for the text of that section.



# 3 Style Guide

## Formatting Text

### Punctuation

*Quotation marks* Quotation marks in  $\LaTeX$  are entered as `` and '' , *not* as ". Single quotation marks are entered as ` and '. Quotation marks for other languages are entered as their Unicode characters.

*Indicating sudden breaks* To indicate a sudden break in thought—like this—use an em dash. To enter an em dash in the manual, use the `\dash` command. This command will print an em dash without spaces—like this—for US English, but can be set to print an en dash with spaces – like this – for other languages such as UK English.

*Indicating a range* If you wish to indicate a range (such as: pages 37–40 or 2005–2007), use an en dash. An en dash is entered in  $\LaTeX$  as two hyphens: -. Do not put spaces before or after the en dash when used to indicate a range.

*Special characters* There are a few characters that  $\LaTeX$  considers special (used for its own syntax). To typeset these characters, precede the character with a backslash (`\`). The special characters are:

\$ % \_ { } & #

The above line was typed like this:

`\$ \% \_ \{ \} \& \#`

Finally, to type a backslash character, use the `\textbackslash` command.

### GUI Elements

*Menu items* To give a sequence of menu items that should be selected, use the `\menu` and `\then` commands. For example:

To open the Calculator application, click **Applications** • **Accessories** • **Calculator**.

is typeset by:

To open the Calculator application, click `\menu{Applications\then Accessories\then Calculator}`.

Use the `\menu` command for single menu items as well:

Pull down the **File** menu and then click **Quit**.

*Other gui elements* There are commands for other GUI elements as well:

GUI element	Command	Appearance
Button	<code>button</code>	<b>Cancel</b>
Tab	<code>\tab</code>	<b>Advanced</b>
Drop-down list	<code>\dropdown</code>	<b>Country</b>
Checkbox	<code>\checkbox</code>	<b>Remember my password</b>
Window title	<code>\window</code>	<b>Preferences</b>
Keyboard key(s)	<code>\keystroke</code>	Ctrl+Q
Radio button	<code>\radiobutton</code>	<b>Single click to open items</b>
Text box	<code>\textfield</code>	<b>Full name</b>

*Application names* The names of applications should be typeset with the `\application` command. This will add the application to the index automatically. For command-line-based applications, use the `\commandlineapp` command instead.

*User input* When quoting what a user should type, use the `\userinput` command. If you're just having the user press a key on the keyboard, use the `\keystroke` command instead. For example:

Type `\userinput{Hello, world!}` and press `\keystroke{Enter}`.

is typeset as:

Type **Hello, world!** and press Enter.

*Cross-referencing* In a manual of this size, it's often helpful to point the reader to another chapter or section for more information on a related topic. There are few commands that can help you do that. To cross-reference another chapter, use the `\chaplink` command. Similarly, to cross-reference a section of a chapter, use the `\seclink` command. The `\chaplink` command will insert the text "Chapter *(X)*: *(Chapter Title)*" and link it to the beginning of that chapter. The `\seclink` command will insert the name of the section and link it to that section.

## Graphics

If a screenshot should be added at some point, make a note of it using the `\screenshotTODO` command. Provide a description of what the screenshot

should depict. This will add a warning notice in the PDF and will add that screenshot to the to do list.



Figure 3.1: Firefox web browser window

## Other document elements

*Paragraphs* To start a new paragraph in L<sup>A</sup>T<sub>E</sub>X, just add a blank line. You don't need to indent the paragraphs as L<sup>A</sup>T<sub>E</sub>X will take care of this for you.

*Margin notes* The manual uses margin notes to provide definitions, tips, and pointers to more information. To add a margin note, use the `\marginnote` command:

```
\marginnote{Margin note text}
```

*Comments* If you want to add a note to yourself (or others) in the `.tex` file, just type a percent sign (%) followed by your note. L<sup>A</sup>T<sub>E</sub>X will ignore everything on the line after the percent sign.

```
This text will appear in the PDF. % But this text won't!
```

Remember, if you want a percent sign to actually appear in the PDF, you'll need to precede it with a backslash:

```
Linux users are 50\% smarter than non-Linux users.
```

*Terminal commands* There are also special commands for typesetting text that appears or is entered into a terminal. An example will illustrate the commands:

```
The \commandlineapp{fortune} program works like this:
\begin{terminal}
\prompt \userinput{fortune}
What we have to learn to do we learn by doing.
-- Aristotle, Ethica Nicomachea II (c. 325 BC)
\end{terminal}
```

The above generates the following output:

The **fortune** program works like this:

```
$ fortune
What we have to learn to do we learn by doing.
-- Aristotle, Ethica Nicomachea II (c. 325 BC)
```

The `\prompt` command will print a BASH-style user prompt (`$`). The `\rootprompt` command will print a BASH-style root prompt (`#`). The `\userinput` command should contain any text that the user types in.

While the `terminal` environment is handy for multiple lines of terminal input/output, if you want to put terminal text inline with your paragraph text, you can use the `\userinput` and `\code` commands.

*Warnings and notices for advanced instructions* While the instructions in this manual should be as safe as possible, we sometimes mention commands or programs that can potentially do damage if not used correctly. To draw attention to these rare circumstances, you can put a note in the `\warning` command. This will set the text off so it's more eye-catching.

Similarly, there is a `\advanced` command for notes to advanced users.

*Lists* There are two types of lists that we use in the manual: numbered lists and bulleted lists. Both lists work the same way, they just have different names.

```
\begin{itemize}
  \item First list item
  \item Second list item
  \item Third list item
\end{itemize}
```

produces:

- ▶ First list item
- ▶ Second list item
- ▶ Third list item

```
\begin{enumerate}
  \item First list item
  \item Second list item
  \item Third list item
\end{enumerate}
```

produces:

1. First list item
2. Second list item
3. Third list item



## 4 Notes for Translators

TeX commands are preceded by a backslash character (\). While the command names should not be translated, their arguments sometimes should be.

In the following list of commands, the **red** text should be translated and the black text should not be translated.

### Document headings

- ▶ \title{*book title*}
- ▶ \author{*book authors*}
- ▶ \part{*part heading*}
- ▶ \chapter{*chapter heading*}
- ▶ \section{*section heading*}
- ▶ \subsection{*subsection heading*}
- ▶ \subsubsection{*subsubsection heading*}
- ▶ \paragraph{*paragraph heading*}
- ▶ \subparagraph{*subparagraph heading*}

### Formatting commands

- ▶ \marginnote{*margin note text*}
- ▶ \textbf{*bold text*}
- ▶ \textit{*italic text*}
- ▶ \emph{*italic text*}
- ▶ \smallcaps{*acronym*}
- ▶ \application{*application name*}
- ▶ \commandlineapp{*command-line application name*}
- ▶ \menu{*menu name* \then *submenu name* \then *menu item*}
- ▶ \button{*button name*}
- ▶ \checkbox{*checkbox name*}
- ▶ \tab{*tab name*}
- ▶ \dropdown{*drop-down list name*}
- ▶ \window{*window name*}
- ▶ \textfield{*text box name*}
- ▶ \keystroke{*Keyboard key names separated by +*}
- ▶ \userinput{*stuff the user types*}
- ▶ \code{*terminal output*}
- ▶ \begin{*terminal*} *output from the terminal* \end{*terminal*}
- ▶ \warning{*warning text*}

- ▶ `\advanced{advanced usage text}`
- ▶ `\screenshot{filename}{label}{caption}`

The following formatting commands should *not* be translated:

- ▶ `\dash`
- ▶ `\url{URL}`
- ▶ `\prompt`
- ▶ `\rootprompt`

## Lists

The only part of lists that need to be translated is the text of the list items themselves.

```
\begin{itemize} or \begin{enumerate}
\item list item text
\item another list item
\end{itemize} or \begin{enumerate}
```

## Author and editor notes

Do not translate any of the following commands:

- ▶ `\todo{notes to appear in PDF margin}`
- ▶ `\screenshotTODO{description of screenshot}`
- ▶ `\begin{comment} notes to authors/editors \end{command}`

## Glossary-related commands

- ▶ `\newglossaryentry{keyword}{name={term},description={definition}}`
- ▶ `\gls{keyword}`
- ▶ `\glspl{keyword}`
- ▶ `\Gls{keyword}`
- ▶ `\Glspl{keyword}`

## Cross-referencing commands

Do not translate any of these commands:

- ▶ `\label{label}`
- ▶ `\ref{label}`
- ▶ `\pageref{label}`
- ▶ `\nameref{label}`
- ▶ `\chapl原因{label}`
- ▶ `\seclink{label}`

## Other document commands

Do not translate any of the following:

- ▶ `\frontcover`
- ▶ `\mainmatter`
- ▶ `\appendix`
- ▶ `\backmatter`
- ▶ `\providecommand{⟨command name⟩}{⟨command definition⟩}`
- ▶ `\documentclass[⟨options⟩]{⟨document type⟩}`
- ▶ `\include{⟨file⟩}`
- ▶ `\begin{⟨environment⟩}`
- ▶ `\end{⟨environment⟩}`
- ▶ `\printglossaries`
- ▶ `\printindex`
- ▶ `\LoadLicenseFile`



# 5 Notes for Editors

## Obsolete commands

There are a few commands that were used in the beginning of the project that have since been replaced. If you encounter any of these obsolete commands, please replace them with the updated command.

`\menuitem` This command should be replaced by the `\menu` command.

`\nav` This command should be replaced by the `\menu` command.

`--` This em dash command should be replaced by the `\dash` command.



## 6 Installing T<sub>E</sub>X Live 2009

We're using a number of features that require the latest version of T<sub>E</sub>X Live (2009). Unfortunately, the version of T<sub>E</sub>X Live that comes with Ubuntu 9.10 is T<sub>E</sub>X Live 2007. To install T<sub>E</sub>X Live 2009, follow these steps:

1. Uninstall all the Ubuntu TeX Live packages: `sudo apt-get remove texlive-*`
2. Download the TeX Live 2009 install script: <http://mirror.ctan.org/systems/texlive/tlnet/install-tl-unx.tar.gz>
3. Unpack the tarball: `tar -zxvf install-tl-unx.tar.gz`
4. Change to the newly-unpacked directory: `cd install-tl-*`
5. Run the script: `sudo ./install-tl`
6. Select where you'd like to install everything, and any other options you prefer. I highly recommend enabling the "create symlinks to standard directories" option.
7. To compile a translation, run: `make ubuntu-manual-<lang>.pdf` where *<lang>* is the language code (see the `po/` directory for a list of supported languages).

Note that not all languages are supported yet (especially those requiring non-Latin scripts).

If you have any questions or run into any problems, feel free to contact me. I'm usually in IRC as "godbyk" and can, of course, be contacted via email.