

L^AT_EX for Ubuntu Manual Authors and Translators

Kevin Godby

22–23 February 2010

Contents

Introduction

\LaTeX for Authors and Editors

\LaTeX for Translators

For More Information

What is the Ubuntu Manual project about?

Getting Started with Ubuntu 10.04 aims to be a complete beginner's manual for Ubuntu, featuring comprehensive guides, how-tos and information on anything one needs to know after first installing Ubuntu.

Designed to be as user-friendly and easy-to-follow as possible, it will provide the first point of reference to any Ubuntu newcomer with lots of information in one easy-to-access PDF. Plus, every six months there will be a new revision released to coincide with each new release of Ubuntu.

What is L^AT_EX?

L^AT_EX is the typesetting system we're using to create the manual's PDF. It's similar to HTML in that the text is marked up with special codes to denote the style of the text, but unlike HTML it's also a complete programming language.

As authors, editors, and translators, you won't have to worry about the programming side, just the markup side.

What does \LaTeX code look like?

All of the \LaTeX code is stored in `.tex` files. These files are regular text files that may be edited with your favorite text editor (e.g., vim, emacs, gedit).

The markup commands in \LaTeX begin with a backslash (`\`). For example,

```
My name is \textbf{Kevin}.
```

will be typeset as:

My name is **Kevin**.

What does \LaTeX code look like?

My name is `\textbf{Kevin}`.

The `\textbf` command tells \LaTeX to print the text in **boldface** type. The braces (`{}`) tell \LaTeX when to start and stop using bold text.

We avoid using these low-level commands, however, preferring instead to use semantic markup. Semantic markup means that, instead of telling \LaTeX how something should look, we tell it what something means.

Semantic Markup

For example, instead of typing:

Pull down the `\textbf{Help}` menu and click `\textbf{About}`.

we would say:

Pull down the `\menu{Help}` menu and click `\menu{About}`.

This ensures that all the menu items are formatted consistently throughout the manual and also allows us to modify the formatting of the menu items in one place, instead of editing each file.

We'll see a list of these semantic markup commands soon.

Special Characters in L^AT_EX

Most text can be typed just as you'd expect, and everything will work great. There are, however, a few characters that need special handling.

L^AT_EX knows the difference between opening and closing quotation marks. So instead of using the normal quotation marks ("), you will need to use two acute accents (“backticks”) for opening quotation marks, and two apostrophes for closing quotation marks:

```
Kevin said, ``Quotation marks are special.``
```

To interrupt a sentence, use the `\dash` command. This allows us to automatically use the appropriate dash based on the language. (US English uses an em dash without space—like this—while UK English uses an en dash with space – like this.)

More Special Characters in \LaTeX

There are a few more characters that \LaTeX considers special because it uses them in its programming language. They are:

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

If you want one of these characters to appear in the typeset document, put a backslash (`\`) in front of that character. The above line was typed like this:

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

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

Document Structure

Getting Started has two parts, over ten chapters, and a slew of sections and subsections. Each of these heading levels has its own \LaTeX command that formats the heading and automatically adds it to the table of contents.

```
\part           \chapter  
\section       \subsection  
\subsubsection
```

If you want to add a new section on how to install Frozen Bubble, for example, you would write:

```
\section{Frozen Bubble}
```

and \LaTeX would print the section heading and add it to the table of contents at the front of the manual.

Paragraphs and margin notes

To start a new paragraph in \LaTeX , just add a blank line. You don't need to indent the paragraphs as \LaTeX will take care of this for you.

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

```
\marginnote{This text will appear in the  
margin of the manual.}
```

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. \LaTeX 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.

GUI Elements

The instructions we write often contain the names of GUI elements such as menu items, buttons, check boxes, drop-down lists, etc. Each of these elements has its own special formatting. To simplify the formatting of these elements, we've created some new markup commands for them.

If you want to direct the user to choose a menu item, you would write:

Click `\menu{Applications\then Accessories\then Calculator}` to start the `\application{Calculator}` application.

which produces:

Click **Applications** ▶ **Accessories** ▶ **Calculator** to start the **Calculator** application.

GUI Elements

The list of current commands for GUI elements follows:

<code>\button</code>	<code>\tab</code>
<code>\dropdown</code>	<code>\checkbox</code>
<code>\window</code>	<code>\keystroke</code>
<code>\radiobutton</code>	<code>\textfield</code>
<code>\application</code>	<code>\commandlineapp</code>

There are a few antiquated commands used in the manual that should be updated if you see them:

Both `\menuitem` and `\nav` should be changed to `\menu`.

`\option` should be changed to `\checkbox`.

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.
  -- Aristotole, 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.
  -- Aristotole, Ethica Nicomachea II (c. 325 BC)
```

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

Cross-Referencing

You can refer the read to other chapters or sections for further information by using the `\chapl原因` and `\seclink` commands.

Please see `\chapl原因{ch:installation}` for more information on installing Ubuntu.

generates:

Please see **Chapter 1: Installation** for more information on installing Ubuntu.

The argument to the `\chapl原因` and `\seclink` is a label. The label is set using the `\label` command immediately after the `\chapter` or `\section` command.

Notes for Translators

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

In the following example, the red text should be translated, but the black text should be left as is:

Click `\menu{Applications\then Accessories\then Calculator}`
to start the `\application{Calculator}` application.

The labels in the `\chaplink`, `\seclink`, and `\ref` commands should never be translated.

Questions?