

The TRANSLATOR *package*  
Manual for version 1.01  
<http://bitbucket.org/rivanvx/translator>

Till Tantau, Vedran Miletić

June 4, 2010

Copyright 2007 by Till Tantau  
Copyright 2010 by Vedran Miletić

Permission is granted to copy, distribute and/or modify all files of this package under the terms of the GNU Public License, Version 2 or any later version published by the Free Software Foundation. A copy of the license is included in the section entitled GNU Public License.

Permission is also granted to distribute and/or modify all files of this package under the conditions of the LaTeX Project Public License, either version 1.3 of this license or (at your option) any later version. A copy of the license is included in the section entitled L<sup>A</sup>T<sub>E</sub>X Project Public License.

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Overview of the Package . . . . .	3
1.2	How to Read This Manual . . . . .	3
1.3	Contributing . . . . .	3
1.4	Getting Help . . . . .	3
<b>2</b>	<b>Installation</b>	<b>4</b>
2.1	Package and Driver Versions . . . . .	4
2.2	Installing Prebundled Packages . . . . .	4
2.3	Installation in a texmf Tree . . . . .	4
2.4	Updating the Installation . . . . .	5
<b>3</b>	<b>Basic Concepts</b>	<b>5</b>
3.1	Keys . . . . .	5
3.2	Language Names . . . . .	5
3.3	Language Paths . . . . .	5
3.4	Dictionaries . . . . .	6
<b>4</b>	<b>Usage</b>	<b>6</b>
4.1	Basic Usage . . . . .	6
4.2	Providing Translations . . . . .	6
4.3	Creating and Using Dictionaries . . . . .	7
4.4	Creating a User Dictionaries . . . . .	9
4.5	Translating Keys . . . . .	9
4.6	Language Path and Language Substitution . . . . .	10
4.7	Package Loading Process . . . . .	10
<b>5</b>	<b>Licenses</b>	<b>11</b>
5.1	The GNU General Public License, Version 2 . . . . .	11
5.1.1	Preamble . . . . .	11
5.1.2	Terms and Conditions For Copying, Distribution and Modification . . . . .	11
5.1.3	No Warranty . . . . .	14
5.2	The L <sup>A</sup> T <sub>E</sub> X Project Public License . . . . .	14
5.2.1	Preamble . . . . .	14
5.2.2	Definitions . . . . .	14
5.2.3	Conditions on Distribution and Modification . . . . .	15
5.2.4	No Warranty . . . . .	16
5.2.5	Maintenance of The Work . . . . .	17
5.2.6	Whether and How to Distribute Works under This License . . . . .	17
5.2.7	Choosing This License or Another License . . . . .	17
5.2.8	A Recommendation on Modification Without Distribution . . . . .	18
5.2.9	How to Use This License . . . . .	18
5.2.10	Derived Works That Are Not Replacements . . . . .	18
5.2.11	Important Recommendations . . . . .	19

# 1 Introduction

## 1.1 Overview of the Package

The translator package is a  $\text{\LaTeX}$  package that provides a flexible mechanism for translating individual words into different languages. For example, it can be used to translate a word like “figure” into, say, the German word “Abbildung”. Such a translation mechanism is useful when the author of some package would like to localize the package such that texts are correctly translated into the language preferred by the user. The translator package is *not* intended to be used to automatically translate more than a few words.

You may wonder whether the translator package is really necessary since there is the (very nice) **babel** package available for  $\text{\LaTeX}$ . This package already provides translations for words like “figure”. Unfortunately, the architecture of the babel package was designed in such a way that there is no way of adding translations of new words to the (very short) list of translations directly built into babel.

The translator package was specifically designed to allow an easy extension of the vocabulary. It is both possible to add new words that should be translated and translations of these words.

The translator package can be used together with babel. In this case, babel is used for language-specific things like special quotation marks and input shortcuts, while translator is used for the translation of words.

## 1.2 How to Read This Manual

This manual explains the commands of the translator package and its usage. The “public” commands and environments provided by the **translator** package are described throughout the text. In each such description, the described command, environment or option is printed in red. Text shown in green is optional and can be left out.

In the following documentation, the installation is explained first, followed by an overview of the basic concepts used. Then, we explain the usage of the package.

## 1.3 Contributing

Since this package is about internationalization, it needs input from people who can contribute translations to their native tongue.

In order to submit dictionaries, please do the following:

1. Read this manual and make sure you understand the basic concepts.
2. Find out whether the translations should be part of the translator package or part of another package. In general, submit translations and new keys to the translator project only if they are of public interest.  
For example, translations for keys like **figure** should be send to the translator project. Translations for keys that are part of a special package should be send to the author of the package.
3. If you are sure that the translations should go to the translator package, create a dictionary of the correct name (see this documentation once more).
4. Finally, submit the dictionary using the correct forum on the development site.

## 1.4 Getting Help

When you need help with the package, please do the following:

1. Read this manual, at least the part that has to do with your problem.
2. Consider rereading the manual, especially the part that has to do with your problem.
3. If that does not solve the problem, try having a look at the development page for translator (see the title of this document). Perhaps someone has already reported a similar problem and someone has found a solution.

4. On the website you will find numerous forums for getting help. There, you can write to help forums, file bug reports, join mailing lists, and so on.
5. Before you file a bug report, especially a bug report concerning the installation, make sure that this is really a bug. In particular, have a look at the `.log` file that results when you `TEX` your files. This `.log` file should show that all the right files are loaded from the right directories. Nearly all installation problems can be resolved by looking at the `.log` file.
6. *As a last resort* you can try to email us (authors). We do not mind getting emails, we simply get way too many of them. Because of this, we cannot guarantee that your emails will be answered timely or even at all. Your chances that your problem will be fixed are somewhat higher if you use the forums and mailing lists. (Naturally, we read the lists and answer questions when we have the time).
7. Please, do not phone us in our offices.

## 2 Installation

This package is distributed under the GPL license and under the L<sup>A</sup>T<sub>E</sub>X public license, see Sections 5.1 and 5.2. Typically, the package will already be installed on your system. Naturally, in this case you do not need to worry about the installation process at all and you can skip the rest of this section.

### 2.1 Package and Driver Versions

This documentation is part of version 1.01 of the translator package. To use it, you just need a reasonably up-to-date L<sup>A</sup>T<sub>E</sub>X-system, but the requirements are rather low. There are no special dependencies.

### 2.2 Installing Prebundled Packages

We do not create or manage prebundled packages of translator, but perhaps other people might be willing to do so. If you have a problem with installing such a bundle, you might wish to have a look at the page of whoever created the package.

### 2.3 Installation in a texmf Tree

For a permanent installation, you place the files of the the TRANSLATOR package in an appropriate `texmf` tree. When you ask T<sub>E</sub>X to use a certain class or package, it usually looks for the necessary files in so-called `texmf` trees. These trees are simply huge directories that contain these files. By default, T<sub>E</sub>X looks for files in three different `texmf` trees:

- The root `texmf` tree, which is usually located at `/usr/share/texmf/` or `c:\texmf\` or somewhere similar.
- The local `texmf` tree, which is usually located at `/usr/local/share/texmf/` or `c:\localtexmf\` or somewhere similar.
- Your personal `texmf` tree, which is usually located in your home directory at `~/texmf/` or `~/Library/texmf/`.

You should install the package either in the local tree or in your personal tree, depending on whether you have write access to the local tree. Installation in the root tree can cause problems, since an update of the whole T<sub>E</sub>X installation will replace this whole tree.

Inside whatever `texmf`-tree that you have chosen, create the sub-directory `texmf/tex/latex/translator` and put everything from the package in this directory.

Finally, you may need to run the program `texhash` to rebuild T<sub>E</sub>X's cache. In MikT<sub>E</sub>X, there is a menu option for this.

## 2.4 Updating the Installation

To update your installation from a previous version, all you need to do is to replace everything in the directory `texmf/tex/latex/translator` with the files of the new version. The easiest way to do this is to first delete the old version and then proceed as described above.

Sometimes, there are changes in the syntax of certain command from version to version. If things no longer work that used to work, you may wish to have a look at the release notes and at the change log.

## 3 Basic Concepts

### 3.1 Keys

The main purpose of the translator package is to provide translations for *keys*. Typically, a key is an English word like **Figure** and the German translation for this key is “Abbildung”.

For a concept like “figures” a single key typically is not enough:

1. It is sometimes necessary to translate a group of words like “Table of figures” as a whole. While these are three words in English, the German translation is just a single word: “Abbildungsverzeichnis”.
2. Uppercase and lowercase letters may cause problems. Suppose we provide a translation for the key **Figure**. Then what happens when we want to use this word in normal text, spelled with a lowercase first letter? We could use T<sub>E</sub>X’s functions to turn the translation into lowercase, but that would be wrong with the German translation “Abbildung”, which is always spelled with a capital letter.
3. Plurals may also cause problems. If we know the translation for “Figure”, that does not mean that we know the translation for “Figures” (which is “Abbildungen” in German).

Because of these problems, there are many keys for the single concept of “figures”: **Figure**, **figure**, **Figures**, and **figures**. The first key is used for translations of “figure” when used in a headline in singular. The last key is used for translations of “figure” when used in normal text in plural.

A key may contain spaces, so **Table of figures** is a permissible key.

Keys are normally English texts whose English translation is the same as the key, but this need not be the case. Theoretically, a key could be anything. However, since the key is used as a last fallback when no translation whatsoever is available, a key should be readable by itself.

### 3.2 Language Names

The translator package uses names for languages that are different from the names used by other packages like `babel`. The reason for this is that the names used by `babel` are a bit of a mess, so Till decided to clean things up for the translator package. However, mappings from `babel` names to translator names are provided.

The names used by the translator package are the English names commonly used for these languages. Thus, the name for the English language is **English**, the name for German is **German**.

Variants of a language get their own name: The British version of English is called **BritishEnglish**, the US-version is called **AmericanEnglish**.

For German there is the special problem of pre-1998 as opposed to the current (not yet fixed) spelling. The language **German** reflects the current official spelling, but **German1997** refers to the spelling used in 1997.

### 3.3 Language Paths

When you request a translation for a key, the translator package will try to provide the translation for the current *language*. Examples of languages are German or English.

When the translator looks up the translation for the given key in the current language, it may fail to find a translation. In this case, the translator will try a fallback strategy: It keeps track of a *language path* and successively tries to find translations for each language on this path.

Language paths are not only useful for fallbacks. They are also used for situations where a language is a variant of another language. For example, when the translator looks for the translation for a key in Austrian, the

language path starts with Austrian, followed by German. Then, a dictionary for Austrian only needs to provide translations for those keys where Austrian differs from German.

### 3.4 Dictionaries

The translations of keys are typically provided by *dictionaries*. A dictionary contains the translations of a specific set of keys into a specific language. For example, a dictionary might contain the translations of the names of months into the language German. Another dictionary might contain the translations of the numbers into French.

## 4 Usage

### 4.1 Basic Usage

Here is a typical example of how to use the package:

```
\documentclass[german]{article}

\usepackage{babel}
\usepackage{some-package-that-uses-translator}

\begin{document}
...
\end{document}
```

As can be seen, things really happen behind the scenes, so, typically, you do not really need to do anything. It is the job of other package to load the translator package, to load the dictionaries and to request translations of keys.

### 4.2 Providing Translations

There are several commands to tell the translator package what the translation of a given key is. As said before, as a normal author you typically need not provide such translations explicitly, they are loaded automatically. However, there are two situations in which you need to provide translations:

1. You do not like the existing translation and you would like to provide a new one.
2. You are writing a dictionary.

You provide a translation using one of the following commands:

**\newtranslation**[*<options>*]{*<key>*}{*<translation>*}

This command defines the translation of *<key>* to be *<translation>* in the language specified by the *<options>*.

You can only use this command if the translation is really “new” in the sense that no translation for the keys has yet been given for the language. If there is already a translation, an error message will be printed.

The following *<options>* may be given:

- **to=<language>** This options tells the translator, that the translation *<translation>* of *<keys>* applies to the language *<language>*.

Inside a dictionary file (see Section 4.3), this option is set automatically to the language of the dictionary.

*Example:* `\newtranslation[to=German]{figure}{Abbildung}`

*Example:* `\newtranslation[to=German]{Figures}{Abbildungen}`

**\renewtranslation**[*<options>*]{*<key>*}{*<translation>*}

This command works like `\newtranslation`, only it will redefine an existing translation.

**\providetranslation**[*<options>*]{*<key>*}{*<translation>*}

This command works like `\newtranslation`, but no error message will be printed if the translation already exists. In this case, the existing translation is not changed.

This command should be used by dictionary authors since their translations should not overrule any translations given by document authors or other dictionary authors.

**\deftranslation**[*<options>*]{*<key>*}{*<translation>*}

This command defines the translation “no matter what”. An existing translation will be overwritten.

This command should typically be used by document authors to install their preferred translations.

*Example:* `\deftranslation[to=German]{figure}{Figur}`

Here is an example where a translation is provided by a document author:

```
\documentclass[ngerman]{article}

\usepackage{babel}
\usepackage{some-package-that-uses-translator}

\deftranslation[to=German]{Sketch of proof}{Beweisskizze}

\begin{document}
...
\end{document}
```

### 4.3 Creating and Using Dictionaries

Two kinds of people will create *dictionaries*: First, package authors will create dictionaries containing translations for the (new) keys used in the package. Second, document authors can create their own private dictionaries that overrule settings from other dictionaries or that provide missing translations.

There is not only one dictionary per language. Rather, many different dictionaries may be used by `TRANSLATOR` when it tries to find a translation. This makes it easy to add new translations: Instead of having to change translator’s main dictionaries (which involves, among other things, the release of a new version of the `translator` package), package authors can just add a new dictionary containing just the keys needed for the package.

Dictionaries are named according to the following rule: The name of the dictionary must start with its *kind*. The kind tells translator which kind of keys the dictionary contains. For example, the dictionaries of the kind `translator-months-dictionary` contain keys like `January` (note that this is a key, not a translation). Following the kind, the name of a dictionary must have a dash. Then comes the language for which the dictionary file provides translations. Finally, the file name must end with `.dict`.

To continue the example of the month dictionary, for the German language the dictionary is called

`translator-months-dictionary-German.dict`

Its contents is the following:

```
\ProvidesDictionary{translator-months-dictionary}{German}

\providetranslation{January}{Januar}
\providetranslation{February}{Februar}
\providetranslation{March}{M\''arz}
\providetranslation{April}{April}
\providetranslation{May}{Mai}
\providetranslation{June}{Juni}
\providetranslation{July}{Juli}
\providetranslation{August}{August}
\providetranslation{September}{September}
\providetranslation{October}{Oktober}
\providetranslation{November}{November}
\providetranslation{December}{Dezember}
```

Note that the `\providetranslation` command does not need the option `[to=German]`. Inside a dictionary file `TRANSLATOR` will always set the default translation language to the language provided by the dictionary. However, you can still specify the language, if you prefer.

The `\ProvidesDictionary` command currently only prints a message in the log-files.

**`\ProvidesDictionary`**`{<kind>}{<language>}[<version>]}`

This command currently only prints a message in the log-files. The format is the same as for L<sup>A</sup>T<sub>E</sub>X's `\ProvidesPackage` command.

Dictionaries are stored in a decentralized manner: A special dictionary for a package will typically be stored somewhere in the vicinity of the package. For this reasons, `TRANSLATOR` needs to be told which *kinds* of dictionaries should be loaded and which *languages* should be used. This is accomplished using the following two commands:

**`\usedictionary`**`{<kind>}`

This command tells the `translator` package, that at the beginning of the document it should load all dictionaries of kind `<kind>` for the languages used in the document. Note that the dictionaries are not loaded immediately, but only at the beginning of the document.

If no dictionary of the given *kind* exists for one of the language, nothing bad happens.

Invocations of this command accumulate, that is, you can call it multiple times for different dictionaries.

**`\uselanguage`**`{<list of languages>}`

This command tells the `translator` package that it should load the dictionaries for all languages in the `<list of languages>`. The dictionaries are loaded at the beginning of the document.

Here is an example of how all of this works: Suppose you wish to create a new package for drawing, say, chess boards. Let us call this package `chess`. In the file `chess.sty` we could now write the following:

```
// This is chess.sty

\RequirePackage{translator}
\usedictionary{chess}

...

\newcommand\MoveKnight[2]{%
  ...
  \translate{knight}
  ...
}
```

Now we create dictionaries like the following:

```
// This is chess-German.dict
\ProvidesDictionary{chess}{German}

\providetranslation{chess}{Schach}
\providetranslation{knight}{Springer}
\providetranslation{bishop}{L\"aufer}
...

and

// This is chess-English.dict
\ProvidesDictionary{chess}{English}

\providetranslation{chess}{chass}
\providetranslation{knight}{knight}
\providetranslation{bishop}{bishop}
...
```



Here are a few things to note:

- The package `chess.sty` does not use the command `\uselanguage`. After all, the package does not know (or care) about the language used in the final document. It only needs to tell the translator package that it will use the dictionary `chess`.
- You may wonder why we need an English dictionary. After all, the keys themselves are the ultimate fallbacks if no other translation is available. The answer to this question is that, first of all, English should be treated like any other language. Second, there are some situations in which there is a “better” English translation than the key itself. An example is explained next.
- The keys we chose may not be optimal. What happens, if some other package, perhaps on medieval architecture, also needs translations of knights and bishops. However, in this different context, the translations of knight and bishop are totally different, namely `Ritter` and `Bischof`.

Thus, it might be a good idea to add something to the key to make it clear that the “chess bishop” is meant:

```
// This is chess-German.dict
\providetranslation{knight (chess)}{Springer}
\providetranslation{bishop (chess)}{L\"aufer}

// This is chess-English.dict
\providetranslation{knight (chess)}{knight}
\providetranslation{bishop (chess)}{bishop}
```

## 4.4 Creating a User Dictionaries

There are two ways of creating a personal set of translations. First, you can simply add commands like

```
\deftranslation[to=German]{figure}{Figur}
```

to your personal macro files.

Second, you can create a personal dictionary file as follows: In your document you say

```
\documentclass[ngerman]{article}

\usepackage{translator}
\usedictionary{my-personal-dictionary}
```

and then you create the following file somewhere where  $\text{\TeX}$  can find it:

```
// This is file my-personal-dictionary-German.dict
\ProvidesDictionary{my-personal-dictionary}{German}

\deftranslation{figure}{Figur}
```

## 4.5 Translating Keys

Once the dictionaries and languages have been setup, you can translate keys using the following commands:

`\translate`[ $\langle options \rangle$ ]{ $\langle key \rangle$ }

This command will insert the translation of the  $\langle key \rangle$  at the current position into the text. The command is robust.

The translation process of  $\langle key \rangle$  works as follows: TRANSLATOR iterates over all languages on the current *language path* (see Section 4.6). For each language on the path, TRANSLATOR checks whether a translation is available for the  $\langle key \rangle$ . For the first language for which this is the case, the translation is used. If there is no translation available for any language on the path, the  $\langle key \rangle$  itself is used as the translation.

*Example:* `\caption{\translate{Figure}~2.}`

The following options may be given:

- **to**= $\langle language \rangle$  This option overrides the language path setting and installs  $\langle language \rangle$  as the target language(s) for which TRANSLATOR tries to find a translation.

**\translatelet**[ $\langle options \rangle$ ]{ $\langle macro \rangle$ }{ $\langle key \rangle$ }

This command works like the **\translate** command, only it will not insert the translation into the text, but will set the macro  $\langle macro \rangle$  to the translation found by the **\translate** command.

*Example:* **\translatelet\localfigure{figure}**

## 4.6 Language Path and Language Substitution

**\languagepath**{ $\langle language path \rangle$ }

This command sets the language path that is searched when TRANSLATOR looks for a key.

The default value of the language path is **\language,English**. The **\language** is the standard  $\TeX$  macro that expands to the current language. Typically, this is exactly what you want and there is no real need to change this default language path.

There is a problem with the names used in the macro **\language**. These names, like **ngerman**, are not the ones used by TRANSLATOR and we somehow have to tell the translator about aliases for cryptic language names like **ngerman**. This is done using the following command:

**\languagealias**{ $\langle name \rangle$ }{ $\langle language list \rangle$ }

This command tells the translator that the language  $\langle name \rangle$  should be replaced by the language in the  $\langle language list \rangle$ .

*Example:* **\languagealias{ngerman}{German}**

*Example:* **\languagealias{german}{German1997,German}**

For the languages used by the babel package, the aliases are automatically setup, so you typically do not need to call either **\languagepath** or **\languagealias**.

## 4.7 Package Loading Process

The translator package is loaded “in stages”:

1. First, some package or the document author requests the the translator package is loaded.
2. The translator package allows options like **ngerman** to be given. These options cause the necessary aliases and the correct translator languages to be requested.
3. During the preamble, packages and the document author request creating dictionary kinds and certain languages to be used. There requests are protocolled.
4. At the beginning of the document (**\begin{document}**) the requested dictionary-language-pairs are loaded.

The first thing that needs to be done is to load the package. Typically, this is done automatically by some other package, but you may wish to include it directly:

**\usepackage**[ $\langle options \rangle$ ]{**translator**}

When you load the package, you can specify (multiple) babel languages as  $\langle options \rangle$ . The effect of giving such an option is the following: It causes the translator package to call **\uselanguage** for the appropriate translation of the babel language names to translator’s language names. It also causes **\languagealias** to be called for the languages.

## 5 Licenses

### 5.1 The GNU General Public License, Version 2

#### 5.1.1 Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software—to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation’s software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author’s protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors’ reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone’s free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

#### 5.1.2 Terms and Conditions For Copying, Distribution and Modification

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The “Program”, below, refers to any such program or work, and a “work based on the Program” means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term “modification”.) Each licensee is addressed as “you”.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program’s source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:
  - (a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
  - (b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
  - (c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:
  - (a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
  - (b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
  - (c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsubsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.
5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.
6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.
7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.
9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.  
  
Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.
10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

### 5.1.3 No Warranty

10. Because the program is licensed free of charge, there is no warranty for the program, to the extent permitted by applicable law. Except when otherwise stated in writing the copyright holders and/or other parties provide the program “as is” without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The entire risk as to the quality and performance of the program is with you. Should the program prove defective, you assume the cost of all necessary servicing, repair or correction.
11. In no event unless required by applicable law or agreed to in writing will any copyright holder, or any other party who may modify and/or redistribute the program as permitted above, be liable to you for damages, including any general, special, incidental or consequential damages arising out of the use or inability to use the program (including but not limited to loss of data or data being rendered inaccurate or losses sustained by you or third parties or a failure of the program to operate with any other programs), even if such holder or other party has been advised of the possibility of such damages.

## 5.2 The L<sup>A</sup>T<sub>E</sub>X Project Public License

*LPPL Version 1.3c 2008-05-04*

### Copyright 1999, 2002–2008 L<sup>A</sup>T<sub>E</sub>X3 Project

Everyone is allowed to distribute verbatim copies of this license document, but modification of it is not allowed.

#### 5.2.1 Preamble

The L<sup>A</sup>T<sub>E</sub>X Project Public License (LPPL) is the primary license under which the L<sup>A</sup>T<sub>E</sub>X kernel and the base L<sup>A</sup>T<sub>E</sub>X packages are distributed.

You may use this license for any work of which you hold the copyright and which you wish to distribute. This license may be particularly suitable if your work is T<sub>E</sub>X-related (such as a L<sup>A</sup>T<sub>E</sub>X package), but it is written in such a way that you can use it even if your work is unrelated to T<sub>E</sub>X.

The section ‘WHETHER AND HOW TO DISTRIBUTE WORKS UNDER THIS LICENSE’, below, gives instructions, examples, and recommendations for authors who are considering distributing their works under this license.

This license gives conditions under which a work may be distributed and modified, as well as conditions under which modified versions of that work may be distributed.

We, the L<sup>A</sup>T<sub>E</sub>X3 Project, believe that the conditions below give you the freedom to make and distribute modified versions of your work that conform with whatever technical specifications you wish while maintaining the availability, integrity, and reliability of that work. If you do not see how to achieve your goal while meeting these conditions, then read the document ‘`cfgguide.tex`’ and ‘`modguide.tex`’ in the base L<sup>A</sup>T<sub>E</sub>X distribution for suggestions.

#### 5.2.2 Definitions

In this license document the following terms are used:

**Work** Any work being distributed under this License.

**Derived Work** Any work that under any applicable law is derived from the Work.

**Modification** Any procedure that produces a Derived Work under any applicable law – for example, the production of a file containing an original file associated with the Work or a significant portion of such a file, either verbatim or with modifications and/or translated into another language.

**Modify** To apply any procedure that produces a Derived Work under any applicable law.

**Distribution** Making copies of the Work available from one person to another, in whole or in part. Distribution includes (but is not limited to) making any electronic components of the Work accessible by file transfer protocols such as FTP or HTTP or by shared file systems such as Sun's Network File System (NFS).

**Compiled Work** A version of the Work that has been processed into a form where it is directly usable on a computer system. This processing may include using installation facilities provided by the Work, transformations of the Work, copying of components of the Work, or other activities. Note that modification of any installation facilities provided by the Work constitutes modification of the Work.

**Current Maintainer** A person or persons nominated as such within the Work. If there is no such explicit nomination then it is the 'Copyright Holder' under any applicable law.

**Base Interpreter** A program or process that is normally needed for running or interpreting a part or the whole of the Work.

A Base Interpreter may depend on external components but these are not considered part of the Base Interpreter provided that each external component clearly identifies itself whenever it is used interactively. Unless explicitly specified when applying the license to the Work, the only applicable Base Interpreter is a 'L<sup>A</sup>T<sub>E</sub>X-Format' or in the case of files belonging to the 'L<sup>A</sup>T<sub>E</sub>X-format' a program implementing the 'T<sub>E</sub>X language'.

### 5.2.3 Conditions on Distribution and Modification

1. Activities other than distribution and/or modification of the Work are not covered by this license; they are outside its scope. In particular, the act of running the Work is not restricted and no requirements are made concerning any offers of support for the Work.
2. You may distribute a complete, unmodified copy of the Work as you received it. Distribution of only part of the Work is considered modification of the Work, and no right to distribute such a Derived Work may be assumed under the terms of this clause.
3. You may distribute a Compiled Work that has been generated from a complete, unmodified copy of the Work as distributed under Clause 2 above, as long as that Compiled Work is distributed in such a way that the recipients may install the Compiled Work on their system exactly as it would have been installed if they generated a Compiled Work directly from the Work.
4. If you are the Current Maintainer of the Work, you may, without restriction, modify the Work, thus creating a Derived Work. You may also distribute the Derived Work without restriction, including Compiled Works generated from the Derived Work. Derived Works distributed in this manner by the Current Maintainer are considered to be updated versions of the Work.
5. If you are not the Current Maintainer of the Work, you may modify your copy of the Work, thus creating a Derived Work based on the Work, and compile this Derived Work, thus creating a Compiled Work based on the Derived Work.
6. If you are not the Current Maintainer of the Work, you may distribute a Derived Work provided the following conditions are met for every component of the Work unless that component clearly states in the copyright notice that it is exempt from that condition. Only the Current Maintainer is allowed to add such statements of exemption to a component of the Work.
  - (a) If a component of this Derived Work can be a direct replacement for a component of the Work when that component is used with the Base Interpreter, then, wherever this component of the Work identifies itself to the user when used interactively with that Base Interpreter, the replacement component of this Derived Work clearly and unambiguously identifies itself as a modified version of this component to the user when used interactively with that Base Interpreter.
  - (b) Every component of the Derived Work contains prominent notices detailing the nature of the changes to that component, or a prominent reference to another file that is distributed as part of the Derived Work and that contains a complete and accurate log of the changes.

- (c) No information in the Derived Work implies that any persons, including (but not limited to) the authors of the original version of the Work, provide any support, including (but not limited to) the reporting and handling of errors, to recipients of the Derived Work unless those persons have stated explicitly that they do provide such support for the Derived Work.
- (d) You distribute at least one of the following with the Derived Work:
  - i. A complete, unmodified copy of the Work; if your distribution of a modified component is made by offering access to copy the modified component from a designated place, then offering equivalent access to copy the Work from the same or some similar place meets this condition, even though third parties are not compelled to copy the Work along with the modified component;
  - ii. Information that is sufficient to obtain a complete, unmodified copy of the Work.
- 7. If you are not the Current Maintainer of the Work, you may distribute a Compiled Work generated from a Derived Work, as long as the Derived Work is distributed to all recipients of the Compiled Work, and as long as the conditions of Clause 6, above, are met with regard to the Derived Work.
- 8. The conditions above are not intended to prohibit, and hence do not apply to, the modification, by any method, of any component so that it becomes identical to an updated version of that component of the Work as it is distributed by the Current Maintainer under Clause 4, above.
- 9. Distribution of the Work or any Derived Work in an alternative format, where the Work or that Derived Work (in whole or in part) is then produced by applying some process to that format, does not relax or nullify any sections of this license as they pertain to the results of applying that process.
- 10. (a) A Derived Work may be distributed under a different license provided that license itself honors the conditions listed in Clause 6 above, in regard to the Work, though it does not have to honor the rest of the conditions in this license.
- (b) If a Derived Work is distributed under a different license, that Derived Work must provide sufficient documentation as part of itself to allow each recipient of that Derived Work to honor the restrictions in Clause 6 above, concerning changes from the Work.
- 11. This license places no restrictions on works that are unrelated to the Work, nor does this license place any restrictions on aggregating such works with the Work by any means.
- 12. Nothing in this license is intended to, or may be used to, prevent complete compliance by all parties with all applicable laws.

#### **5.2.4 No Warranty**

There is no warranty for the Work. Except when otherwise stated in writing, the Copyright Holder provides the Work ‘as is’, without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The entire risk as to the quality and performance of the Work is with you. Should the Work prove defective, you assume the cost of all necessary servicing, repair, or correction.

In no event unless required by applicable law or agreed to in writing will The Copyright Holder, or any author named in the components of the Work, or any other party who may distribute and/or modify the Work as permitted above, be liable to you for damages, including any general, special, incidental or consequential damages arising out of any use of the Work or out of inability to use the Work (including, but not limited to, loss of data, data being rendered inaccurate, or losses sustained by anyone as a result of any failure of the Work to operate with any other programs), even if the Copyright Holder or said author or said other party has been advised of the possibility of such damages.



### 5.2.5 Maintenance of The Work

The Work has the status ‘author-maintained’ if the Copyright Holder explicitly and prominently states near the primary copyright notice in the Work that the Work can only be maintained by the Copyright Holder or simply that it is ‘author-maintained’.

The Work has the status ‘maintained’ if there is a Current Maintainer who has indicated in the Work that they are willing to receive error reports for the Work (for example, by supplying a valid e-mail address). It is not required for the Current Maintainer to acknowledge or act upon these error reports.

The Work changes from status ‘maintained’ to ‘unmaintained’ if there is no Current Maintainer, or the person stated to be Current Maintainer of the work cannot be reached through the indicated means of communication for a period of six months, and there are no other significant signs of active maintenance.

You can become the Current Maintainer of the Work by agreement with any existing Current Maintainer to take over this role.

If the Work is unmaintained, you can become the Current Maintainer of the Work through the following steps:

1. Make a reasonable attempt to trace the Current Maintainer (and the Copyright Holder, if the two differ) through the means of an Internet or similar search.
2. If this search is successful, then enquire whether the Work is still maintained.
  - (a) If it is being maintained, then ask the Current Maintainer to update their communication data within one month.
  - (b) If the search is unsuccessful or no action to resume active maintenance is taken by the Current Maintainer, then announce within the pertinent community your intention to take over maintenance. (If the Work is a L<sup>A</sup>T<sub>E</sub>X work, this could be done, for example, by posting to `comp.text.tex`.)
3.
  - (a) If the Current Maintainer is reachable and agrees to pass maintenance of the Work to you, then this takes effect immediately upon announcement.
  - (b) If the Current Maintainer is not reachable and the Copyright Holder agrees that maintenance of the Work be passed to you, then this takes effect immediately upon announcement.
4. If you make an ‘intention announcement’ as described in 2b above and after three months your intention is challenged neither by the Current Maintainer nor by the Copyright Holder nor by other people, then you may arrange for the Work to be changed so as to name you as the (new) Current Maintainer.
5. If the previously unreachable Current Maintainer becomes reachable once more within three months of a change completed under the terms of 3b or 4, then that Current Maintainer must become or remain the Current Maintainer upon request provided they then update their communication data within one month.

A change in the Current Maintainer does not, of itself, alter the fact that the Work is distributed under the LPPL license.

If you become the Current Maintainer of the Work, you should immediately provide, within the Work, a prominent and unambiguous statement of your status as Current Maintainer. You should also announce your new status to the same pertinent community as in 2b above.

### 5.2.6 Whether and How to Distribute Works under This License

This section contains important instructions, examples, and recommendations for authors who are considering distributing their works under this license. These authors are addressed as ‘you’ in this section.

### 5.2.7 Choosing This License or Another License

If for any part of your work you want or need to use *distribution* conditions that differ significantly from those in this license, then do not refer to this license anywhere in your work but, instead, distribute your work under a different license. You may use the text of this license as a model for your own license, but your license should not refer to the LPPL or otherwise give the impression that your work is distributed under the LPPL.

The document ‘`modguide.tex`’ in the base L<sup>A</sup>T<sub>E</sub>X distribution explains the motivation behind the conditions of this license. It explains, for example, why distributing L<sup>A</sup>T<sub>E</sub>X under the GNU General Public License (GPL) was considered inappropriate. Even if your work is unrelated to L<sup>A</sup>T<sub>E</sub>X, the discussion in ‘`modguide.tex`’ may still be relevant, and authors intending to distribute their works under any license are encouraged to read it.

### 5.2.8 A Recommendation on Modification Without Distribution

It is wise never to modify a component of the Work, even for your own personal use, without also meeting the above conditions for distributing the modified component. While you might intend that such modifications will never be distributed, often this will happen by accident – you may forget that you have modified that component; or it may not occur to you when allowing others to access the modified version that you are thus distributing it and violating the conditions of this license in ways that could have legal implications and, worse, cause problems for the community. It is therefore usually in your best interest to keep your copy of the Work identical with the public one. Many works provide ways to control the behavior of that work without altering any of its licensed components.

### 5.2.9 How to Use This License

To use this license, place in each of the components of your work both an explicit copyright notice including your name and the year the work was authored and/or last substantially modified. Include also a statement that the distribution and/or modification of that component is constrained by the conditions in this license. Here is an example of such a notice and statement:

```
%% pig.dtx
%% Copyright 2005 M. Y. Name
%
% This work may be distributed and/or modified under the
% conditions of the LaTeX Project Public License, either version 1.3
% of this license or (at your option) any later version.
% The latest version of this license is in
%   http://www.latex-project.org/lppl.txt
% and version 1.3 or later is part of all distributions of LaTeX
% version 2005/12/01 or later.
%
% This work has the LPPL maintenance status ‘maintained’.
%
% The Current Maintainer of this work is M. Y. Name.
%
% This work consists of the files pig.dtx and pig.ins
% and the derived file pig.sty.
```

Given such a notice and statement in a file, the conditions given in this license document would apply, with the ‘Work’ referring to the three files ‘`pig.dtx`’, ‘`pig.ins`’, and ‘`pig.sty`’ (the last being generated from ‘`pig.dtx`’ using ‘`pig.ins`’), the ‘Base Interpreter’ referring to any ‘L<sup>A</sup>T<sub>E</sub>X-Format’, and both ‘Copyright Holder’ and ‘Current Maintainer’ referring to the person ‘M. Y. Name’.

If you do not want the Maintenance section of LPPL to apply to your Work, change ‘maintained’ above into ‘author-maintained’. However, we recommend that you use ‘maintained’ as the Maintenance section was added in order to ensure that your Work remains useful to the community even when you can no longer maintain and support it yourself.

### 5.2.10 Derived Works That Are Not Replacements

Several clauses of the LPPL specify means to provide reliability and stability for the user community. They therefore concern themselves with the case that a Derived Work is intended to be used as a (compatible or incompatible) replacement of the original Work. If this is not the case (e.g., if a few lines of code are reused for a completely different task), then clauses 6b and 6d shall not apply.

### 5.2.11 Important Recommendations

**Defining What Constitutes the Work** The LPPL requires that distributions of the Work contain all the files of the Work. It is therefore important that you provide a way for the licensee to determine which files constitute the Work. This could, for example, be achieved by explicitly listing all the files of the Work near the copyright notice of each file or by using a line such as:

```
% This work consists of all files listed in manifest.txt.
```

in that place. In the absence of an unequivocal list it might be impossible for the licensee to determine what is considered by you to comprise the Work and, in such a case, the licensee would be entitled to make reasonable conjectures as to which files comprise the Work.