

# **texi2latex**

---

translating Texinfo to L<sup>A</sup>T<sub>E</sub>X  
version 0.9.4

**Torsten Bronger**

---

This manual is for texi2latex (version 0.9.4), which is a conversion tool from Texinfo to L<sup>A</sup>T<sub>E</sub>X.

Copyright © 2004, 2005 Torsten Bronger <[bronger@physik.rwth-aachen.de](mailto:bronger@physik.rwth-aachen.de)>.

This documentation is free software; you can redistribute it and/or modify it under the terms of the GFDL licence. Please see the ‘fdl.txt’ file of the texi2latex distribution for further information.

# Contents

<b>1</b>	<b>About texizlatex</b>	<b>5</b>
1.1	Why texizlatex? . . . . .	5
1.2	Limitations and bugs . . . . .	5
<b>2</b>	<b>Installation</b>	<b>7</b>
2.1	Prerequisites . . . . .	7
2.2	Installation using make . . . . .	7
2.3	Manual installation . . . . .	8
2.4	Xalan as XSLT processor . . . . .	8
<b>3</b>	<b>Usage</b>	<b>9</b>
3.1	Command line options . . . . .	10
3.2	Step-by-step conversion . . . . .	11
<b>4</b>	<b>Customisation</b>	<b>13</b>
4.1	Using L <sup>A</sup> T <sub>E</sub> X config files . . . . .	13
4.2	Overriding L <sup>A</sup> T <sub>E</sub> X hooks . . . . .	14
4.2.1	Some special inline hooks . . . . .	14
4.2.2	Further inline hooks . . . . .	15
4.2.3	Block hooks . . . . .	17
4.3	XSLT customisation layer . . . . .	20
<b>A</b>	<b>Why XSLT?</b>	<b>21</b>



# Chapter 1

## About texi2latex

The project's homepage can be found at Savannah (<http://savannah.nongnu.org/projects/texi2latex>). There you find the latest release.

I created vo.9 of texi2latex from 2004/12/24 to 2005/01/02. Unfortunately, my actual work and other Free Software projects make it impossible for me to continue with it. I'll try to fix any bugs you report though. If you want to improve texi2latex, contact me ([bronger@physik.rwth-aachen.de](mailto:bronger@physik.rwth-aachen.de)).

### 1.1 Why texi2latex?

The most important reason for using texi2latex is customisation. For people knowing L<sup>A</sup>T<sub>E</sub>X it is almost trivial to implement their own layout. Changing the default font face, for example, is a one-liner. The same is true for page margins and headers. Since texi2latex's output is plain L<sup>A</sup>T<sub>E</sub>X with only a few quirks, you can provide your own preamble to adjust everything to your needs. See chapter 4 [Customisation] on page 13, for more information.

Another advantage of texi2latex in comparison to makeinfo's T<sub>E</sub>X output format is that some things are difficult to implement in Plain T<sub>E</sub>X but are already available in L<sup>A</sup>T<sub>E</sub>X. Floats, for example, are not yet implemented in Texinfo, although the specification includes them. Their realisation with L<sup>A</sup>T<sub>E</sub>X is almost trivial. Other examples include full PDF support, full support of non-English languages, and decent line-breaking of URLs.

### 1.2 Limitations and bugs

The XML output format of makeinfo is almost but not fully mature. So if you want to use advanced features such as index mapping (@synindex), it is necessary to install a very fresh

version of Texinfo, whenever possible a CVS version. And even then some things will still be missing.

**Note:** texi<sub>2</sub>latex has its own version of ‘`texinfo.dtd`’. It is necessary that this is used instead of Texinfo’s CVS DTD. However, normally this is done automatically. See chapter 3 [Usage] on page 9, for how to invoke texi<sub>2</sub>latex without worrying about the correct DTD.

Other problems include:

- texi<sub>2</sub>latex doesn’t yet support `@verb`, `@flushleft` and `@flushright`, due to problems in `makeinfo`. But they are already included into the XSLT code, which means that they’ll work as soon as `makeinfo` is fixed.
- Overlong lines in multi-column tables are not broken yet.
- Two consecutive commas result in left double bottom quotes. This is a feature of T<sub>1</sub> encoded fonts, however, actually `tbrplent`<sup>1</sup> should prevent it.

The complete list of known deficiencies is included into the ‘TODO’ file of the texi<sub>2</sub>latex distribution.

Please report bugs at the Savannah tracker system, and send your questions to the mailing list. Both can be found on texi<sub>2</sub>latex’s homepage. The mailing list is also available at [gmane.org](http://gmane.org).

---

<sup>1</sup>`tbrplent` originally is a part of `tbook` (<http://tbookdtd.sourceforge.net>). It is responsible for replacing UTF-8 sequences with L<sup>A</sup>T<sub>E</sub>X representations.

# Chapter 2

## Installation

texi2latex is a very simple package, so installation should be very simple, too, (almost) no matter which operating system you use.

### 2.1 Prerequisites

For building texi2latex, a C++ compiler is necessary. Well, actually you only use it for building a supporting program called tbrplent.

In order to use texi2latex, you need makeinfo from the standard Texinfo distribution, a modern and complete L<sup>A</sup>T<sub>E</sub>X, and an XSLT processor.

makeinfo should be clear.

As far as L<sup>A</sup>T<sub>E</sub>X is concerned, it's important to have all packages that your document includes. This is hyperref, hypbmsec, listings, float, caption2, longtable, booktabs, and fancybox. Most of them should have been shipped with your L<sup>A</sup>T<sub>E</sub>X distribution. If not, you get them at CTAN (<http://www.ctan.org/>).

It is highly probable though that you must install an XSLT 1.0 processor. The best one in my opinion is Saxon (<http://sf.net/projects/saxon>). **Attention:** Don't download and install version 8.x but version 6.5.3.

### 2.2 Installation using make

This way will certainly only work on Unix, Linux, and the like.

Open the 'Makefile' with your favourite text editor and adjust the paths to your needs. The file is self-explaining. Then type

```
make
make install      (as root)
```

## 2.3 Manual installation

The manual installation I describe here is not a full substitute of the `make` call. You should regard it as an emergency procedure if using `make` is not possible for you. In this case, you will have to call the conversion programs manually, see section 3.2 [Step-by-step conversion] on page 11.

1. Compile the C++ file `tbrplent.cc` in the directory `tbrplent/`. Move the resulting executable to somewhere in your `PATH`.

`tbrplent` needs the file `tbents.txt`. You can move `tbents.txt` to an arbitrary directory, however, `tbrplent` must find it. There are two possibilities:

- (a) You define the environment variable `TBLIBDIR` that points to that directory. This variable must end in a `/` (or `\` on Windows).
- (b) Compile the path to `tbents.txt` into `tbrplent` by calling

```
c++ -O2 -DTBLIBDIR="/directory/of/tbents.txt/" tbrplent.cc
```

Here, too, you have to include the trailing `/`.

2. Move the directory `xslt/` with its complete contents to where you want to have it. It's totally unimportant where it is, but you must specify its path when you invoke Saxon, see section 3.2 [Step-by-step conversion] on page 11.

## 2.4 Xalan as XSLT processor

I recommend Saxon as XSLT processor, but XSLT 1.0 is a well-defined standard, so there are alternatives. The most serious one is Xalan-C++ (<http://xml.apache.org/xalan-c/>). If you use the shell script provided with `texi2latex`, you just have to (un)comment the proper lines to activate Xalan instead of Saxon. However, in my tests Xalan was slightly slower, although it's written in C++. (Saxon is a Java application.)



# Chapter 3

## Usage

In the following, I describe the use of `texi2latex` with the shell script of the same name. Probably it only works on Linux. It's very simple though. If you can adapt it to other systems, I'll be glad to hear about it. If the shell script doesn't work for you, see section [3.2](#) [Step-by-step conversion] on page [11](#).

Let's assume your original Texinfo document is stored in the file `'mymanual.texi'`. Then you can convert it to its  $\LaTeX$  equivalent `'mymanual.ltx'` by saying

```
texi2latex mymanual.texi
```

(you may omit the extension). By the way, the resulting file extension `'ltx'` may be unusual, but it avoids problems with an existing `'mymanual.tex'` (maybe from a Texinfo run). However, if you want to use `texi2latex` and `texi2dvi` parallelly, you have to delete `'mymanual.aux'` and `'mymanual.toc'` before you switch to the respective other program.

Note that `'mymanual.ltx'` is ready to be processed by `latex` as well as by `pdflatex`. For the latter, the abilities of  $\LaTeX$ 's `hyperref` package are extensively used.

Also note that plain  $\LaTeX$  and `pdf $\LaTeX$`  can only deal with certain image formats. Plain  $\LaTeX$  usually only can include EPS files. `pdf $\LaTeX$`  can deal with PDF, JPEG, and PNG. So it may be necessary to convert images.

When you call `latex` or `pdflatex` on `'mymanual.ltx'`, a file `'makeindex.bat'` is generated in the current directory. This is a shell script that should work on both Linux and Windows. If you call it, all indices of your document are created. The program `makeindex` – usually included in modern  $\TeX$  distributions – is needed for this.

### 3.1 Command line options

The most important option to `texi2latex` was already explained – it’s the name of the Texinfo file, of course.

But you can add further options. They are given in an ‘*name=value*’ form.<sup>1</sup> So far, three options are defined:

‘`document-class`’

Default: `book`

The document class of the resulting L<sup>A</sup>T<sub>E</sub>X file. Please bear in mind that Texinfo documents have chapters. Because of this and other things you should know what you’re doing when you use this option.

‘`global-options`’

Default: *empty*

Global options (enclosed by ‘`[...]`’) given to the document class in the resulting L<sup>A</sup>T<sub>E</sub>X file. For example,

```
texi2latex mymanual global-options=12pt
```

adds ‘`12pt`’ to the `\documentclass` macro and thus switches the document font size to 12 pt.

‘`ignore-tabs`’

Default: `no`

If set to ‘`yes`’, all tabulator characters in `@verbatim` environments are simply ignored. This behaviour could be interesting if you have very many `@verbatim` in your document, but no tabs used. The XSLT algorithm that handles tabs is rather costly, so this speeds the conversion up a bit. However, don’t expect miracles: The original Texinfo manual with 27 `@verbatim`s is processed faster by only 2%.

‘`syntax-highlighting`’

Default: `no`

If set to ‘`yes`’, syntax highlighting for code displays is activated. Since Texinfo only supports `@list` so far, this option affects only Lisp code. Another constraint is that syntax highlighting is applied to displays only if they don’t contain other formatting commands, e.g. `@r{...}` or something like that.

---

<sup>1</sup>With Windows’ command line interpreter, you have to enclose every name-value pair with double quotes.

Technical note: These options are so-called “XSLT parameters”.

## 3.2 Step-by-step conversion

This awkward way is only necessary if you have installed `texi2latex` without `make`, as in section 2.3 [Manual installation] on page 8.

Let’s assume your original Texinfo document is stored in the file ‘`mymanual.texi`’.

1. Create ‘`mymanual.xml`’ by invoking

```
makeinfo --xml mymanual.texi
```

2. Edit the file ‘`mymanual.xml`’. In the second line, the URL to ‘`texinfo.dtd`’ must be changed to the file path to your local ‘`texinfo.dtd`’. For example, on my system this is ‘`/home/bronger/xml/texi2latex/xslt/texinfo.dtd`’.
3. Convert the XML file to  $\LaTeX$  with

```
saxon mymanual.xml /path/to/texi2latex.xsl | \  
/path/to/tbrplent > mymanual.ltx
```

Here you can also give the command line options, see section 3.1 [Command line options] on the facing page.

4. Call  $\LaTeX$  or `pdf $\LaTeX$`  to get a DVI or PDF file, respectively:

```
latex mymanual.ltx  
or  
pdflatex mymanual.ltx
```

5. If your document contains indices, the  $\LaTeX$  document that `texi2latex` produces will write a file called ‘`makeindex.bat`’ in the current directory. Depending on your system and shell configuration, you can call it with one of the following lines:

```
makeindex.bat  
./makeindex.bat  
bash makeindex.bat
```

It will call `makeindex` with all appropriate parameters to create the indices. You will have to call  $\LaTeX$  again for including them into the document.



# Chapter 4

## Customisation

Customisation is a vital feature of `texi2latex`. Therefore, it provides more than one way to accomplish that. I'll explain them from flexible to very flexible, but at the same time from easy to difficult.

### 4.1 Using $\LaTeX$ config files

The simplest method to customise `texi2latex`'s result is to create a file `'mymanual-t2l.cfg'` in the same directory as `'mymanual.texi'`. It is read by  $\LaTeX$  when the document is processed, and it is interpreted as last part of the  $\LaTeX$  preamble. Thus, it should contain  $\LaTeX$  preamble macros that affect the global layout. For example,

```
\usepackage{mathptmx,courier}
\usepackage[scaled]{helvet}
```

sets Times as the default font, Helvetica for sans-serif excerpts, and Courier for fixed-width excerpts. For further information, consult any good  $\LaTeX$  book or manual.

#### The global config file

If a file called `'texi2latex.cfg'` is in a search path of  $\TeX$  (including the current directory), this is interpreted like the document-specific configuration file from above. It has lower precedence though, i.e. any re-definitions in the document-specific file override the global one.

This global config file contains settings for all `texi2latex` runs. It's a good place for page size and margins settings, but there's no reason not to include font definition commands as well.

## 4.2 Overriding L<sup>A</sup>T<sub>E</sub>X hooks

You can override every L<sup>A</sup>T<sub>E</sub>X inline hook in a config file (see section 4.1 [Using LaTeX config files] on the page before) with

```
\renewcommand{\nameofHook}[1]{new definition}
```

In the new definition, every #1 is replaced with the respective text to be formatted. Note that it is unnecessary to enclose the whole new definition with braces in order to keep re-definitions local. This is done in the L<sup>A</sup>T<sub>E</sub>X source anyway.

Let's assume you wanted to change the appearance of URLs. You prefer the Texinfo way of printing them with a fixed-width font. Then the following line does the trick:

```
\renewcommand{\urlHook}[1]{\texttt{#1}}
```

Put this line in the file 'mymanual-t21.cfg' in the same directory as 'mymanual.texi'. That's it.

### 4.2.1 Some special inline hooks

I explain the following inline hooks separately from the rest (see section 4.2.2 [Further inline hooks] on the facing page) because they are good illustrations for how the whole thing works. They are in no way more important than the rest.

`\tabletermHook` [Macro]

Default: #1

The term of a two-columns table. With “term”, the first column of the table is meant. In the Texinfo source, it's the text after the @item.

`\sampHook` [Macro]

Default: `\lqHook\texttt{#1}\rnqHook`

This is an example of the texi2latex-specific macros `\lqHook`, `\rqHook`, `\lnqHook`, `\rnqHook`. They insert left and right quotation marks of the global document language. The second pair represents the “nested” variants for inner quotations or restrained markup. For example, if the language is American English (the default), these four commands result in “, ”, ‘, and ’, respectively.

`\abbrevwordHook` [Macro]

Default: #1

The abbreviation in an @abbr macro. It's the first argument in Texinfo.

`\abbrevdescHook` [Macro]

Default: ‘ (#1)’

The meaning description in an `@abbr` macro. It’s the second argument in Texinfo.

`\acronymwordHook` [Macro]

Default: `\mbox{#1}`

The acronym in an `@acronym` macro. It’s the first argument in Texinfo.

`\acronymdescHook` [Macro]

Default: `\footnote{#1}`

The meaning description in an `@acronym` macro. It’s the second argument in Texinfo.

`\urlHook` [Macro]

Default: `#1`

This hook influences three Texinfo macros: `@indicateurl`, `@url` (deprecated), and the URL part of `@uref`. This is one of the rare cases where `texizlatex` differs from Texinfo’s behaviour: Texinfo prints URLs in `typewriter` text, but `texizlatex` uses the standard font.

It doesn’t affect email addresses though. This is done with `\emailHook`, see section [4.2.2 \[Further inline hooks\]](#) on the current page.

### 4.2.2 Further inline hooks

The remaining bunch of hooks is just enumerated in the following list. Their definition is pretty straightforward. As a rule of thumb, the hook `\nameHook` defines the markup of the Texinfo macro `@name`.

`\codeHook` [Macro]

Default: `\mbox{\ttfamily #1}`

`\kbdHook` [Macro]

Default: `\mbox{\ttfamily\slshape #1}`

`\keyHook` [Macro]

Default: `{\fboxsep2pt\fbox{\sffamily\footnotesize #1}}`

If you want to have neat key buttons with 3D effect, write

```
\usepackage{keystroke}
\renewcommand{\keyHook}[1]{\keystroke{\footnotesize\sffamily #1}}
```

in your L<sup>A</sup>T<sub>E</sub>X config file.

<code>\verbHook</code>	[Macro]
Default: <code>\mbox{\ttfamily #1}</code>	
<code>\varHook</code>	[Macro]
Default: <code>\mbox{\slshape #1}</code>	
<code>\envHook</code>	[Macro]
Default: <code>\mbox{\ttfamily #1}</code>	
<code>\fileHook</code>	[Macro]
Default: <code>\lnqHook\texttt{#1}\rnqHook</code>	
<code>\commandHook</code>	[Macro]
Default: <code>\mbox{\ttfamily #1}</code>	
<code>\optionHook</code>	[Macro]
Default: <code>\lnqHook\mbox{\ttfamily #1}\rnqHook</code>	
<code>\dfnHook</code>	[Macro]
Default: <code>\emph{#1}</code>	
<code>\citeHook</code>	[Macro]
Default: <code>\textit{#1}</code>	
<code>\emailHook</code>	[Macro]
Default: <code>#1</code>	
<code>\emphHook</code>	[Macro]
Default: <code>\emph{#1}</code>	
<code>\strongHook</code>	[Macro]
Default: <code>\textbf{#1}</code>	



<code>\scHook</code>	[Macro]
Default: <code>\textsc{#1}</code>	
<code>\slantedHook</code>	[Macro]
Default: <code>\textsl{#1}</code>	
<code>\sansserifHook</code>	[Macro]
Default: <code>\textsf{#1}</code>	
<code>\iHook</code>	[Macro]
Default: <code>\textit{#1}</code>	
<code>\bHook</code>	[Macro]
Default: <code>\textbf{#1}</code>	
<code>\tHook</code>	[Macro]
Default: <code>\texttt{#1}</code>	
<code>\rHook</code>	[Macro]
Default: <code>\textrm{#1}</code>	
<code>\dmnHook</code>	[Macro]
Default: <code>\, \mbox{#1}</code>	
<code>\mathHook</code>	[Macro]
Default: <code>\ensuremath{#1}</code>	
<code>\footnoteHook</code>	[Macro]
Default: <code>\footnote{#1}</code>	

### 4.2.3 Block hooks

It is also possible to override L<sup>A</sup>T<sub>E</sub>X *environments* that are responsible for Texinfo's block environments such as `@examples` and `@displays`.

As you can see in the following list, the default values for most block hooks derive from the L<sup>A</sup>T<sub>E</sub>X environment `{display}`. See `preamble.mod.xsl` for its definition. It assures proper indentation in a nested environment (such as an enumeration). Moreover, it allows for additional indentation via its optional parameter in `[...]`. The default indentation is `\standardmargin` (equal to `\parindent`).

If you want to override the block hook *name*, say

```
\renewenvironment{nameHook}{opening commands}%
  {closing commands}
```

It is advisable, though not necessary, to use the `{display}` environment somewhere in the new definition.

As an example, let's assume you wanted to give all `@examples` a grey background. You achieve this by adding this to your  $\text{\LaTeX}$  config file:

```
\usepackage{color, framed}
\definecolor{shadecolor}{gray}{0.75}
\renewenvironment{exampleHook}{%
  \begin{shaded}\begin{display}\ttfamily}%
  {\end{display}\end{shaded}}
```

You can also re-define the `{display}` environment itself to change the layout of most block types. However, you must be careful with that. See its definition in `'preamble.mod.xsl'` for further information.

### List of all block element hooks

`quotationHook`

Default: `{\begin{quote}}{\end{quote}}`

`copyingHook`

Default: `{\begin{quote}}{\end{quote}}`

`verbatimHook`

Default: `{\begin{display}[Opt]\ttfamily}{\end{display}}`

`exampleHook`

Default: `{\begin{display}\ttfamily}{\end{display}}`

`lispHook`

Default: `{\begin{display}\ttfamily}{\end{display}}`

`displayHook`

Default: `{\begin{display}\relax}{\end{display}}`

`formatHook`

Default: `{\begin{display}[Opt]}{\end{display}}`

smallexampleHook

Default:

```
{\begin{small}\begin{exampleHook}}%
{\end{exampleHook}\end{small}}
```

smalldisplayHook

Default:

```
{\begin{small}\begin{displayHook}}%
{\end{displayHook}\end{small}}
```

smallformatHook

Default:

```
{\begin{small}\begin{formatHook}}%
{\end{formatHook}\end{small}}
```

smalllispHook

Default:

```
{\begin{small}\begin{lispHook}}%
{\end{lispHook}\end{small}}
```

flushleftHook

Default: {\begin{flushleft}}{\end{flushleft}}

flushrightHook

Default: {\begin{flushright}}{\end{flushright}}

groupHook

Default: {\begin{samepage}}{\end{samepage}}

cartoucheHook

Default:

```
{\begin{center}\shadowbox\bgroup
  \hbox to \hsize\bgroup\begin{minipage}{\hsize}}%
{\end{minipage}\hss\egroup\egroup\end{center}}
```

`\centerHook`

Default: `\centerline{#1}`

Attention! Although this is a block hook by layout, it is not so in terms of its  $\text{\LaTeX}$  realisation. In fact it is a  $\text{\LaTeX}$  *macro* as those inline hooks, see section 4.2 [Overriding LaTeX hooks] on page 14.

Note that if you activate syntax highlighting (see section 3.1 [Command line options] on page 10), some `@lisp` and `@smalllisp` blocks will be typeset using  $\text{\LaTeX}$ 's ‘listings’ package. This cannot be included in a hook. However, the ‘`lispHook`’ is still wrapped around it.

### 4.3 XSLT customisation layer

I don't want to spent too many words on that because people trying it must know what they are doing anyway. Basically it means that you call Saxon with a stylesheet provided by you, instead of ‘`texi2latex.xml`’. However, you include ‘`texi2latex.xml`’ in your stylesheet. So, your stylesheet will look like this:

```
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

  <xsl:import href="texi2latex.xml"/>
```

*Your templates ...*

```
</xsl:stylesheet>
```

Put this file in the same directory where ‘`texi2latex.xml`’ resides, or, if you don't like that, put it anywhere and insert the full path to ‘`texi2latex.xml`’ in the href attribute.

“*Your templates ...*” may be a copy of one or more original templates from the `texi2latex` source, plus a couple of modifications by you. They will override their original counterparts. With this you have full control of the  $\text{\LaTeX}$  that is generated.

(Damn, I wanted to say nothing, but now I told the whole story. But it doesn't help much, does it?)

# Appendix A

## Why XSLT?

In section “Output Formats” in *Texinfo*, Richard Stallman writes:

If you are a programmer and would like to contribute to the GNU project by implementing additional output formats for Texinfo, that would be excellent. But please do not write a separate translator `texi2foo` for your favorite format `foo`! That is the hard way to do the job, and makes extra work in subsequent maintenance, since the Texinfo language is continually being enhanced and updated. Instead, the best approach is modify `makeinfo` to generate the new format.

Stallman had good reason for this advice: An extension of `makeinfo` makes very fast conversions possible, maintaining is easier at first sight, and availability for the Texinfo community is more direct.

But despite that, I ignored the advice for my Texinfo to  $\LaTeX$  converter. The reason is that `makeinfo` has a rather good XML export. XML itself is perfectly suited to be transformed into an arbitrary output format. It’s drop-dead simple to use XSLT in order to do the conversion. My full-featured implementation is only 142 kB large, although XSLT is a verbose XML dialect full of redundancies. By the way, 12% of the code are GPL notices, and 22% are translation snippets for non-English languages.

But even maintenance is at least not more difficult with XSLT. New macros, for example, must be added to `makeinfo` as they have to be added to `texi2latex`. The parser is in both cases already available. Even better, extensions that require a look at the whole document are a lot easier in XSLT.

On the downside, the use of XSLT makes `texi2latex` slow. And of course it needs extra installation. This means that it is impossible to call `texi2latex` in the default `make` run, for example. It’s just a way to get beautiful printouts and “static” PDFs, that’s all.



# GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright © 1989, 1991 Free Software Foundation, Inc.  
59 Temple Place – Suite 330, Boston, MA 02111-1307, USA

Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

## 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.

## **TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION**

- o. 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 Subsection 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 judgement 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.

## **NO WARRANTY**

11. 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.
12. 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.



# Index

All L<sup>A</sup>T<sub>E</sub>X identifiers are printed in typewriter style; additionally, macros start with ‘\’ (in contrast to environments).

- `\abbrevdescHook`, 15
- `\abbrevwordHook`, 14
- `\acronymdescHook`, 15
- `\acronymwordHook`, 15
- `\bHook`, 17
- `\centerHook`, 20
- `\citeHook`, 16
- `\codeHook`, 15
- `\commandHook`, 16
- `\dfnHook`, 16
- `\dmnHook`, 17
- `\emailHook`, 16
- `\emphHook`, 16
- `\envHook`, 16
- `\fileHook`, 16
- `\footnoteHook`, 17
- `\iHook`, 17
- `\kbdHook`, 15
- `\keyHook`, 15
- `\mathHook`, 17
- `\optionHook`, 16
- `\rHook`, 17
- `\sampHook`, 14
- `\sansserifHook`, 17
- `\scHook`, 17
- `\slantedHook`, 17
- `\strongHook`, 16
- `\tHook`, 17
- `\tabletermHook`, 14
- `\urlHook`, 14, 15
- `\varHook`, 16
- `\verbHook`, 16
- block hooks, 17
- cartoucheHook, 19
- command line options, 10
- config file, 13
- config file (global), 13
- configuration, 13
- copyingHook, 18
- Courier, 13
- CTAN, 7
- customisation, 13
- customisation layer (XSLT), 20
- description, term hook, 14
- display (L<sup>A</sup>T<sub>E</sub>X environment), 17
- displayHook, 18
- displays (Texinfo), hooks, 17
- document type declaration, 6
- document-class (option), 10
- DTD, 6

- EPS (images), 9
- example (Texinfo), hook, 17
- exampleHook, 18
- flushleftHook, 19
- flushrightHook, 19
- formatHook, 18
- global-options (option), 10
- GPL, 23
- graphics formats, 9
- groupHook, 19
- Helvetica, 13
- hooks, L<sup>A</sup>T<sub>E</sub>X, 14, 17
- ignore-tabs (option), 10
- image formats, 9
- index generation, 9
- inline hooks, 14
- installation, 7
- installation, by hand, 8
- invocation, 9
- JPEG, 9
- L<sup>A</sup>T<sub>E</sub>X, 7
- L<sup>A</sup>T<sub>E</sub>X block hooks, 17
- L<sup>A</sup>T<sub>E</sub>X inline hooks, 14
- L<sup>A</sup>T<sub>E</sub>X packages, 7
- licence, 23
- lisp (Texinfo environment), 20
- lispHook, 18
- make call, 7
- Makefile, 7
- makeindex, 9
- makeinfo, 7, 21
- options (command line), 10
- packages, L<sup>A</sup>T<sub>E</sub>X, 7
- parameters (command line), 10
- PDF (images), 9
- pdfL<sup>A</sup>T<sub>E</sub>X, 9
- PNG, 9
- preamble.mod.xsl, 17
- prerequisites, 7
- quotationHook, 18
- Saxon, 7
- smalldisplayHook, 19
- smallexampleHook, 19
- smallformatHook, 19
- smalllisp (Texinfo environment), 20
- smalllispHook, 19
- special displays (Texinfo), hook, 17
- Stallman, Richard, 21
- syntax highlighting, 20
- syntax-highlighting (option), 10
- table (two-column), term hook, 14
- tbents.txt, 8
- TBLIBDIR, 8
- tbrplent, 6–8
- templates (XSLT), 20
- texi2latex (options), 10
- texi2latex (shell script), 9
- texi2latex.cfg, 13
- texi2latex.xsl, 20
- texinfo.dtd, 6
- Times, 13
- two-column table, term hook, 14
- URLs, 14
- usage, 9
- usage, step by step, 11
- verbatim (Texinfo), hook, 17
- verbatimHook, 18



Xalan, 8

XSLT, 7

XSLT customisation layer, 20

XSLT parameters, 10

XSLT processor, 7, 8

XSLT, rationale, 21