

# Crowbook



# Crowbook

Élisabeth Henry



# Contents

<b>1</b>	<b>Crowbook</b>	<b>7</b>
1.1	Installing . . . . .	7
1.2	Usage . . . . .	8
1.3	Current features . . . . .	9
1.4	Acknowledgements . . . . .	10
1.5	ChangeLog . . . . .	10
1.6	Library . . . . .	10
1.7	License . . . . .	10
<b>2</b>	<b>The configuration file</b>	<b>11</b>
2.1	The list of files . . . . .	12
2.2	Crowbook options . . . . .	13
<b>3</b>	<b>Arguments</b>	<b>21</b>
3.1	--create . . . . .	22
3.2	--set . . . . .	23
3.3	--list-options . . . . .	24
3.4	--print-template . . . . .	24
3.5	--verbose . . . . .	25
3.6	--to . . . . .	25
3.7	--output . . . . .	25



# Chapter 1

## Crowbook

Render a markdown book in HTML, Epub or PDF.

Crowbook's purpose is to allow you to automatically generate multiple outputs formats from a book written in Markdown. Its main focus is novels, and the default settings should (hopefully) generate readable book with correct typography.

### 1.1 Installing

There are two ways to get **crowbook**: either use a precompiled binary or build it yourself.

#### Binaries

See [the releases page](#) to download a precompiled binary for your architecture (currently: Linux, Windows and MacOSX). Just extract the archive and run **crowbook** (you might also want to copy the binary somewhere in your PATH for later usage).

#### Building

You'll need to have the **Rust** compiler on your machine first; you can [download and install it here](#). Once it is down:

```
$ cargo install crowbook
```

will automatically download the latest **crowbook** release on [crates.io](#) and install it.

## 1.2 Usage

The simplest command is:

```
$ crowbook <BOOK>
```

Where `BOOK` is a configuration file. Crowbook will then parse this file and generate a book in HTML, Epub, LaTeX, and/or PDF, according to the settings in the configuration file. So if you clone this repository and run

```
$ crowbook book_example/config.book
```

you'll generate the example book in various formats. The HTML version should look [like that](#).

To create a new book, assuming you have a list of Markdown files, you can generate a template configuration file with the `--create` argument:

```
$ crowbook --create my.book chapter_*.md
```

This will generate a default `my.book` file, which you'll need to complete.

This configuration file contains some metadata, options, and lists the Markdown files. Here is a basic example:

```
author: Joan Doe
title: Some book
lang: en

output_html: some_book.html

+ chapter_1.md
+ chapter_2.md
+ chapter_3.md
+ ...
```

For more information see [the configuration file page](#), or the whole `book_example` source directory. (A (not necessarily up-to-date) [rendered version is available in HTML here](#)).

It is also possible to give additional parameters to `crowbook`; we have already seen `--create`, but if you want the full list, see [the arguments page](#).



## 1.3 Current features

### Output formats

Crowbook (to my knowledge) correctly supports HTML and EPUB (either version 2 or 3) as output formats: rendered files should pass respectively the [W3C validator](#) and the [IDPF EPUB validator](#) for a wide range of (correctly Markdown formatted) input files. See the example book rendered in [HTML](#) and [EPUB](#) on github.io.

LaTeX output is a bit more tricky: it should work reasonably well for novels (the primary target of Crowbook), but `pdflatex` might occasionally choke on some « weird » unicode character. Moreover, images are not yet implemented (but should come soon). See the example book rendered in [PDF](#) on github.io.

ODT output is experimental at best. It might work if your inputs files only include very basic formatting (basically, headers, emphasis and bold), it will probably look ugly in the rest of the cases, and it might miserably fail in some. See the example book rendered in [ODT](#) on github.io if you want to hurt your eyes.

### Input format

Crowbook uses [pulldown-cmark](#) and thus should support most of commonmark Markdown. Inline HTML, however, is not implemented, and probably won't be, as the goal is to have books that can also be generated in PDF (and maybe eventually ODT).

Maybe the most specific "feature" of Crowbook is that (by default, it can be deactivated) it tries to "clean" the input files. By default this doesn't do much (except removing superfluous spaces), but if the book's language is set to french it tries to respect french typography, replacing spaces with non-breaking ones when it is appropriate (e.g. in french you are supposed to put a non-breaking space before '?', '!', ',' or ':'). This feature is relatively limited at the moment, but I might try to add more options and support for more languages.

### Links handling

Crowbook tries to correctly translate local links in the input Markdown files: e.g. if you have a link to a markdown file that is part of your book, it will be transformed into a link inside the document.

See also [Bugs](#).

## 1.4 Acknowledgements

Besides the Rust compiler and standard library, Crowbook uses the following libraries:

- [pulldown-cmark](#) (for parsing markdown)
- [mustache](#) (for templating)
- [clap](#) (for parsing command line arguments)
- [chrono](#) (date and time library)
- [uuid](#) (to generate uuid)

It also uses configuration files from [rust-everywhere](#) to use [Travis][<https://travis-ci.org/>) and [Appveyor](#) to generate binaries for various platforms on each release.

While Crowbook directly doesn't use them, there was also inspiration from [Pandoc](#) and [mdBook](#).

## 1.5 ChangeLog

See [ChangeLog](#).

## 1.6 Library

While the main purpose of Crowbook is to be runned as a command line, the code is written as a library, so if you want to build on it you can use it as such. The code is currently badly documented (and badly in a general manner), but you can look at the generated documentation [here](#).

## 1.7 License

Crowbook is free software: you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License (LGPL), version 2.1 or (at your option) any ulterior version. See [LICENSE](#) file for more information.

## Chapter 2

# The configuration file

If you want to use Crowbook for your book, this configuration file is all you'll have to add (assuming you'll already have the book in Markdown files; if you don't, you'll also have to write a book first, but that's besides the scope of this document).

The format is not very complicated, and it looks a bit like YAML. This is an example of it:

```
# metadata
author: Joan Doe
title: Some book
lang: en

output.html: some_book.html

# list of chapters
- preface.md
+ chapter_1.md
+ chapter_2.md
+ chapter_3.md
+ chapter_4.md
- epilogue.md
```

Basically, it is divided in two parts:

- a list of options, under the form **key:** **value**;
- a list of Markdown files.

Files starting with the `#` characters are comments and are discarded by Crowbook when parsing the files. Note that `#` must be at the *beginning* of the line, so e.g.:

```
author: John Smith # aka John Doe
```

will set the `author` key to `John Smith # aka John Doe`.

## 2.1 The list of files

There are various options to include a markdown file.

- `+ file_name.md` includes a numbered chapter.
- `- file_name.md` includes an unnumbered chapter.
- `! file_name.md` includes a chapter whose title won't be displayed (except in the toc for epub); this is useful for e.g. including a copyright at the beginning of the book, or for short stories where there is only one chapter.
- `42. file_name.md` specifies the number for a chapter.

So a typical usage might look like this:

```
! copyright.md
- preface.md
# We want first chapter to be Chapter 0 because we are
programmers!
0. chapter_0.md
# Next chapters can be numbered automatically
+ chapter_1.md
+ chapter_3.md
...
```

There are two important things to note:

1. you must *not* use quotes around the file names
2. the path of these files are relative to the directory where your config file is, *not* to the directory where you are when running `crowbook`. E.g. you can run `crowbook books/my_trilogy/first_book/config.book` without being in the book's directory.

Also note that you don't have to specify a title. This is because the title of the chapter is inferred from the Markdown document. To go back to our previous example:

```
+ chapter_1.md
```

does not specify a chapter title, because it will read it directly in `chapter_1.md`, e.g.:

```
The day I was born
=====
```

```
...
```

You should have one and only one level-one header (i.e. chapter title) in each markdown file.

If you have more than one, Crowbook won't get too angry at you and will just print a warning and treat it as another chapter (numbered according to the scheme specified for including the file). It will however mess the table of contents if Crowbook tries to generate one (e.g. for Epub).

It's also a problem if you do *not* have a level-1 header in a markdown file. If it is a numbered chapter Crowbook will still be able to infer a chapter name, but if it is not numbered Crowbook will fail to generate an Epub file.

\*\*\*\*\*

So, to sum it up. *please*: one file = one chapter, a chapter starts with a title, and this way this will work nice.

## 2.2 Crowbook options

The first part of the configuration file is dedicated to pass options to Crowbook. Each one is of the form `option: value`. Note that you don't have to put string in quotes, e.g.:

```
title: My title
```

If you *do* use quotes, Crowbook will actually put those quotes in the string, so basically don't do that.

It is possible to use multiline strings with either `>` or `|`, and then indenting the lines that are part of the string:

```
title: >
  A
  long
  title
author: Joan Doe
```

will set `title` to "A long title", whereas

```
title: >
  A
  long
  title
author: Joan Doe
```

will set `title` to "A\nlong\ntitle\n" (replicating line returns).

This feature is useful for options like `description` who may take a long string.

Here is the complete list of options, with a short description. The usage of some of them is detailed later on.

## Metadata

- **author**
  - **type**: string
  - **default value**: Anonymous
  - The author of the book
- **title**
  - **type**: string
  - **default value**: Untitled
  - The title of the book
- **lang**
  - **type**: string
  - **default value**: en
  - The language of the book

- **subject**
  - **type:** string
  - **default value:** not set
  - Subject of the book (used for EPUB metadata)
- **description**
  - **type:** string
  - **default value:** not set
  - Description of the book (used for EPUB metadata)
- **cover**
  - **type:** string
  - **default value:** not set
  - File name of the cover of the book

## Output options

- **output.epub**
  - **type:** string
  - **default value:** not set
  - Output file name for EPUB rendering
- **output.html**
  - **type:** string
  - **default value:** not set
  - Output file name for HTML rendering
- **output.tex**
  - **type:** string
  - **default value:** not set
  - Output file name for LaTeX rendering
- **output.pdf**
  - **type:** string
  - **default value:** not set

- Output file name for PDF rendering
- `output.odt`
  - **type**: string
  - **default value**: not set
  - Output file name for ODT rendering

## Misc options

- `numbering`
  - **type**: integer
  - **default value**: 1
  - The maximum heading levels to number (0: no numbering, 1: only chapters, ..., 6: all)
- `display_toc`
  - **type**: boolean
  - **default value**: false
  - If true, display a table of content in the document
- `toc_name`
  - **type**: string
  - **default value**: Table of contents
  - Name of the table of contents if toc is displayed in line
- `autoclean`
  - **type**: boolean
  - **default value**: true
  - Toggles cleaning of input markdown (not used for LaTeX)
- `verbose`
  - **type**: boolean
  - **default value**: false
  - Toggle verbose mode
- `side_notes`



- **type:** boolean
- **default value:** false
- Display footnotes as side notes in HTML/Epub
- **nb\_char**
  - **type:** char
  - **default value:** ' '
  - The non-breaking character to use for autoclean when lang is set to fr
- **temp\_dir**
  - **type:** string
  - **default value:** .
  - Path where to create a temporary directory
- **numbering\_template**
  - **type:** string
  - **default value:** {{number}}. {{title}}
  - Format of numbered titles

## HTML options

- **html.template**
  - **type:** string
  - **default value:** not set
  - Path of an HTML template
- **html.css**
  - **type:** string
  - **default value:** not set
  - Path of a stylesheet to use with HTML rendering

## EPUB options

- `epub.version`
  - **type:** integer
  - **default value:** 2
  - The EPUB version to generate
- `epub.css`
  - **type:** string
  - **default value:** not set
  - Path of a stylesheet to use with EPUB rendering
- `epub.template`
  - **type:** string
  - **default value:** not set
  - Path of an epub template for chapter

## LaTeX options

- `tex.links_as_footnotes`
  - **type:** boolean
  - **default value:** true
  - If set to true, will add footnotes to URL of links in LaTeX/PDF output
- `tex.command`
  - **type:** string
  - **default value:** `pdflatex`
  - LaTeX flavour to use for generating PDF
- `tex.template`
  - **type:** string
  - **default value:** not set
  - Path of a LaTeX template file

## Output options

These options specify which files to generate. You must at least set one of this option, or Crowbook won't do anything.

Recall that all file paths are relative to the directory where the config file is, not to the one where you run `crowbook`. So if you set

```
output.epub = foo.epub
```

and runs

```
$ crowbook some/dir/config.book
```

`foo.epub` will be generated in `some/dir`, not in your current directory.

Crowbook will try to generate each of the `output.xxx` files that are specified. That means that you'll have to set at least one of those if you want a call to

```
$ crowbook my.book
```

to generate anything. (It's still possible to generate a specific format, and only this one, by using the `--to` argument on the command line).

Note that some formats depend on some commands being installed on your system. Most notably, Crowbook depends on LaTeX (`pdflatex` by default, though you can specify the command to use with `tex.command`) to generate a PDF file, so PDF rendering won't work if it is not installed on your system. Crowbook also uses the `zip` command to generate the EPUB and ODT, files. want to use.)

## Generic options for rendering

### numbering

An integer that represents the maximum level of numbering for your book. E.g., 1 will only number chapters, while 2 will number chapters, sections, but not anything below that. 6 is the maximum level and turns numbering on for all headers.

**default::** 1

## **numbering\_template**

A string that will be used for chapter titles. You can use `{{number}}` and `{{title}}` in this string, e.g.:

`numbering_template: Chapter {{number}} {{title}}`

Note that:

- this string isn't used for unnumbered chapters;
- this string isn't used by LaTeX, either.

## **autoclean**

This option cleans a bit the input markdown. With the default implementation, it only removes consecutive spaces, which has not real impact (they are ignored anyway both by HTML viewers and by LaTeX).

However, if `lang` is set to `fr`, it also tries to add non-breaking spaces in front (or after) characters like `'?', '!', ','` to respect french typography.

## **nb\_char**

This option allows you to specify the non breaking character used by the french cleaning method (see above). Probably not really something you need to modify.

## **Additional options**

### **temp\_dir**

When it is generating epub or pdf files, Crowbook creates a temporary directory (which is then removed), named from a random uuid (so we can be pretty certain it's not gonna exist). This option specify where to create this directory. E.g., if you set:

`temp_dir: /tmp`

crowbook might create a temporary directory `/tmp/7fcbe41e-1676-46ba-b1a7-40c2fa37a3a7`.

By default, this temporary directory is created where the config file is.

**default:** `.`

# Chapter 3

## Arguments

Crowbook can takes a list of arguments:

Render a markdown book in Epub, PDF or HTML.

USAGE:

```
crowbook [OPTIONS] [--] [ARGS]
```

OPTIONS:

<code>--create &lt;FILES&gt;...</code>	Creates a new book
with existing markdown files.	
<code>-h, --help</code>	Prints help
information	
<code>-l, --list-options</code>	Lists all possible
option	
<code>--list-options-md</code>	List all options,
formatted in Markdown	
<code>-o, --output &lt;FILE&gt;</code>	Specifies output
file.	
<code>--print-template &lt;TEMPLATE&gt;</code>	Displays the
default value of a template.	
<code>-s, --set &lt;KEY_VALUES&gt;</code>	Sets a list of book
options	
<code>-t, --to &lt;FORMAT&gt;</code>	Generate specific
format [values: epub, pdf, html, tex, odt]	
<code>-V, --version</code>	Prints version
information	
<code>-v, --verbose</code>	Activate verbose

mode

ARGS:

<BOOK>      File containing the book configuration.

Command line options allow to override options defined in <BOOK> configuration file. E.g., even if this file specifies 'verbose: false', calling 'crowbook --verbose <BOOK>' will activate verbose mode.

Note that Crowbook generates output files relatively to the directory where <BOOK> is:

```
$ crowbook foo/bar.book --to pdf --output baz.pdf
```

will thus generate baz.pdf in directory foo and not in current directory.

The most important option obviously <BOOK>, i.e. the file configuration book. It is mandatory for most options: if you don't pass it, **crowbook** will simply display this help message. In a normal use case this is the only argument you'll need to pass, and **crowbook** will generate the book in all formats specified in the configuration file.

It is, however, possible to pass more arguments to **crowbook**.

## 3.1 --create

**Usage:** **crowbook** [**B00K**] --create file\_1.md file\_2.md ...

Creates a new book from a list of Markdown files. It will generate a book configuration file with all file names specified as chapter. It either prints the result to stdout (if **B00K** is not specified) or generate the file **B00K** (or abort if it already exists).

## Examples

```
crowbook foo.book --create  README.md ChangeLog.md
LICENSE.md
```

will generate a file **foo.book** containing:

```
author: Your name
title:  Your title
lang:  en
```

```
# Uncomment and fill to generate files
# output.html: some_file.html
# output.epub: some_file.epub
# output.pdf: some_file.pdf

# Uncomment and fill to set cover image (for Epub)
# cover: some_cover.png

# List of chapters
+ README.md
+ ChangeLog.md
+ LICENSE.md

while

crowbook --create README.md ChangeLog.md LICENSE.md
```

will prints the same result, but to stdout (without creating a file).

When `crowbook` is runned with `--create`, it can also uses the keys/values set by `--set` (see below):

```
$ crowbook foo.book --create file1.md file2.md --set
author "Pierre
Dupont" title "Mon œuvre" lang fr
```

will generate a `foo.book` file containing

```
author: Pierre Dupont
title: Mon œuvre
lang: fr
```

```
# List of chapters
+ file1.md
+ file2.md
```

## 3.2 --set

**usage:** ‘`crowbook <BOOK> --set [KEY] [VALUE]...`  
(or `crowbook <BOOK> -s [KEY] [VALUE]...`)

This options takes a list `KEY VALUE` pairs and allows to set or override a book configuration option. All valid options in the configuration files are valid as keys. For more information, see [the configuration file page](#).

## Examples

```
$ crowbook foo.book --set html.css style.css
```

will override the CSS for HTML generation (the `html.css` key) to `style.css`.

```
$ crowbook foo.book --set author Foo --title Bar
```

will override the book title to `Bar` and its author to `Foo`.

### 3.3 --list-options

**usage:** `crowbook --list-options`  
(or `crowbook -l`)

Displays all the valid options to use either in a book configuration file or with `--set`, with a short description. There is also `--list-options-md`, which outputs markdown.

### 3.4 --print-template

**usage:** `crowbook --print-template template`

Prints to stdout the built-in template. Useful if you want to customize the appearance of your document. E.g., if you want to modify the CSS used for HTML rendering:

```
$ crowbook --print-template html.css > my_style.css
# edit my_style.css in your favourite editor
$ crowbook my.book --set html.css my_style.css
# or add "html.css: my_style.css" in my.book
```

Note that it is possible to use this option in conjunction with `--set`, though it is currently only useful for EPUB template:

```
$ crowbook --print-template epub.template --set
epub.version 2
# Returns the template for Epub 2 (default one)
$ crowbook --print-template epub.template --set
epub.version 3
# Returns the template for Epub 3
```



### 3.5 --verbose

**usage:** `crowbook <BOOK> --verbose`

If this flag is set, Crowbook will print some more messages.

### 3.6 --to

**usage:** `crowbook <BOOK>--to [FORMAT]`  
(or `crowbook <BOOK> -t [FORMAT]`)

Generate only the specified format. `FORMAT` must be either `epub`, `pdf`, `html`, `odt` or `tex`.

If an output file for the format is not specified in the book configuration file, `crowbook` will fail to render PDF, ODT and Epub (whereas it will print HTML and Tex files on stdout). It is however possible to specify a file with the `--output` option.

## Examples

```
crowbook --to html foo.book
```

will generate some HTML, and prints it either to the file specified by `output.html` in `foo.book`, or to stdout.

```
crowbook --to pdf --output foo.pdf foo.book
```

will (try to) generate a `foo.pdf` file,.

### 3.7 --output

**usage:** `crowbook <BOOK> --to <FORMAT> --output <FILE>`  
(or `crowbook -t <FORMAT> -o <FILE> <BOOK>`)

Specifies an output file. Only valid when `--to` is used.

Note that Crowbook generates output files relatively to the directory where `BOOK` is:

```
$ crowbook foo/bar.book --to pdf --output baz.pdf
```

will thus generate `baz.pdf` in directory `foo` and not in current directory.



# ChangeLog

## 0.3.0 (unreleased)

- Crowbook now tries to convert local links. That is, if you link to a Markdown file that is used in the book (e.g. `README.md`), it *should* link to an appropriate inner reference inside the book.
- Latex renderer now supports (local) images.
- Epub renderer now embed (local) images in the EPUB file.
- Internal (or usage as a library):
  - Crowbook no longer changes current directory, which worked in the binary but could cause problem if library was used in multithreaded environment (e.g. in `cargo test`).
  - Modules `toc` and `escape` are now private.
  - More methods are now private.
  - Improved documentation.
  - Added more unit tests.
- Bugfixes:
  - Epub renderer now correctly renders unnumbered chapter without a number in its `toc.ncx` file

## 0.2.2 (2016-02-25)

- Bugfixes:
  - French cleaner now correctly replaces space after `—` (in e.g. dialogs) with `"em space"`.

### 0.2.1 (2016-02-25)

- Bugfixes:
  - HTML/Epub rendering no longer incorrectly increment chapter count for unnumbered chapters.
  - Latex: makes what is possible to avoid overflowing the page.
- Minor changes:
  - Latex: improvement of the default way URLs are displayed.

### 0.2.0 (2016-02-25)

- Command line arguments:
  - New argument `--print-template` now allows to print a built-in template to stdout.
  - New argument `--list-options` prints out all valid options in a config file (or in `set`), their type and default value.
  - New argument `--set` allows to define or override whatever option set in a book configuration.
  - `--create` can now be used without specifying a `BOOK`, printing its result on `stdout`.
- Configuration file:
  - Added support for multiline strings in `.book` files, with either `'|'` (preserving line returns) or `'>'` (transforming line returns in spaces)
  - New option `display_toc` allows to display the table of contents (whose name, at least for HTML, is specified by `toc_name`) in HTML and PDF documents.
  - Option `numbering` now takes an int instead of a boolean, allowing to specify the maximum level to number (e.g. 1: chapters only, 2: chapters and sections, ..., 6: everything).
- Rendering:
  - Added support for numbering all headers, not just level-1 (e.g., having a subsection numbered 2.3.1).

- Tables and Footnotes are now implemented for HTML/Epub and LaTeX output.
- Internal:
  - Refactored `Book` to use an `HashMap` of `BookOptions` instead of having like 42 fields.

## 0.1.0 (2016-02-21)

- initial release



# Bugs

- epub: what happens if cover point to "../foo" or "/foo/bar"? nothing good, I fear...
- epub: no way to embed custom files to epub
- epub: toc.ncx currently only displays chapters (first level headers)
- latex: support for images is not good
- odt: no support for... a lot of stuff, actually
- config: can't parse specified numbers for negative chapter (is it really a bug?)

# ToDo

- better debugging/warning/verbose messages
- generate cleaner codes (e.g. indentation)
- clean CSS and templates files
- html: add option to generate multi-pages html





# GNU LESSER GENERAL PUBLIC LICENSE

Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc. 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts as the successor of the GNU Library Public License, version 2, hence the version number 2.1.]

## Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, 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 and use pieces of it in new free programs; and that you are informed that you can do these things.

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

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages

are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

#### GNU LESSER GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

1. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion

of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, 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 library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete 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 distribute a copy of this License along with the Library.

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.

1. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, 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) The modified work must itself be a software library.

b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, 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 Library, 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 Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other

work under the scope of this License.

1. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

1. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you 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.

If distribution of 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 satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

1. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be

a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

1. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified

executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be 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.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

1. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work

based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact

that part of it is a work based on the Library, and explaining

where to find the accompanying uncombined form of the same work.

1. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library 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.
2. 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 Library or its derivative works. These actions

are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

3. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library 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 with this License.
1. 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 Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library 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 Library.

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.

1. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library 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.
2. The Free Software Foundation may publish revised and/or new versions of the Lesser 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 Library 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 Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

1. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, 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

2. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "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 LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

3. 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 LIBRARY 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 LIBRARY (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 LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### END OF TERMS AND CONDITIONS

#### How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the library's name and a brief idea of  
what it does.>  
Copyright (C) <year> <name of author>
```

This library is free software; you can redistribute it  
and/or  
modify it under the terms of the GNU Lesser General Public

License as published by the Free Software Foundation;  
either  
version 2.1 of the License, or (at your option) any later  
version.

This library is distributed in the hope that it will be  
useful,  
but WITHOUT ANY WARRANTY; without even the implied  
warranty of  
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See  
the GNU  
Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General  
Public  
License along with this library; if not, write to the Free  
Software  
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston,  
MA 02110-1301 USA

Also add information on how to contact you by electronic and paper  
mail.

You should also get your employer (if you work as a programmer)  
or your school, if any, to sign a "copyright disclaimer" for the library,  
if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library  
'Frob' (a library for tweaking knobs) written by James Random Hacker.  
<signature of Ty Coon>, 1 April 1990 Ty Coon, President of Vice  
That's all there is to it!