

Open Editing

Technique, Law, Trust, and Temperament

by Robert J. Chassell

Copyright © 2005 Robert J. Chassell

This is about open editing: the technical, legal, trust, and social issues associated with the creation and modification of freely redistributable documents, especially of legal textbooks and casebooks.

Moreover, this is intended as an example of how to write a freely redistributable document. To be a good primer, it needs much more work.

June 2005

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation; there is no Invariant Section, no Front-Cover Text, and no Back-Cover Text. A copy of the license is included in the section entitled “GNU Free Documentation License”.

Introduction

*This is an example of how to write a freely redistributable document.
To be a good primer, it needs more, much more.*

Open editing involves the freedom to create, modify, and distribute documents. It has has technical, legal, trust, and social issues.

We can handle the technical, legal, and trust issues. Unfortunately, social problems remain.

Technical

There are several aspects to the technical.

- The first aspect concerns audiences: producing different renderings for different audiences.
- The second aspect concerns method: there are several. At the time of first writing, this document focuses on Texinfo, which is a generation old and mature.
- The third aspect concerns the ways to include or exclude specific parts of an overall text for those who desire one or other content. (See “Texinfo”, page 4.)
- The fourth aspect concerns cooperation among those who wish to write together easily, distantly, and securely. (See “Arch”, page 9.)

In addition, to help novices and to make work and cooperation even easier, I would like to see an editor or word processor that shows at least two of the surface expressions simultaneously with the deep representation.

(See <http://www.rattlesnake.com/notions/multiple-output-preview.html>.)

Moreover, I would like to see easy bibliographic mechanisms, such as those in BibTeX. Perhaps SGML based formats, such as XML, should be adapted and replace Texinfo. Fortunately, none of these actions is necessary.

Legal

Legal problems have been solved by applying old laws to new circumstances. But an underlying question remains: for how long should the enabling law be enforced? (See “Legal Solutions: Different Licenses”, page 14.)

Trust

Technological advances have enabled us to create inexpensive new filters for credibility, to include many more people in the process, and to exclude the dangerous. Modern trust building mechanisms exist. They can be improved and adapted. (See “Trust”, page 16.)

Social

Unfortunately, society has not solved the social problems that have come with people’s desire to take advantage of the new technologies of the past

half century. Instead individuals continue with old habits and corporations continue with old business models.

Thus, many contemporary people write for just one output format. Similarly, many corporations sell inexpensive and non-rivalrous products as if they were expensive and rivalrous. (See “The Social Problems”, page 18.)

For success, we must overcome these hinderances.

Audiences and Multiple Renderings

In the past, people wrote with pen, pencil, or typewriter for a sighted audience. They wrote on paper and they had copies printed on paper. Paper was the one significant form that output could take. The only salient blind were the permanently blind and they are a small portion of the population.

Nowadays, paper is just one medium. Besides paper, people read on various kinds of electronic display. Or they listen. In the past, one ‘surface expression’ was salient, that of pen or print on paper. Now several different ‘surface expressions’ are salient.

Surface Expressions

The three major renderings are the high resolution, typeset format for printing on paper; HTML for Web pages; and Info for efficient online reading or listening. These are the surface expressions of a deep representation.

On a computer, plain text is common; most often, it is displayed in a fixed-width font. I am not going to talk about plain text here, except as an occasional output format.

For printed, typeset output on paper, the DVI, PDF, or PostScript output formats are common. These same outputs can also be viewed on an electronic display. (These files are created by the ‘`TEX`’, ‘`ATEX`’, ‘`debiandoc2dvi`’, or ‘`texi2dvi`’ commands using overtly marked text.)

The HTML of the World Wide Web is often viewed on an electronic display. (HTML files are created with the ‘`latex2html`’, ‘`debiandoc2html`’, and ‘`makeinfo`’ commands.)

Unlike paper output, in which the reader sees the design that the creator intended, World Wide Web output is formatted as the reader desires; the writer specifies only the content and the designer only hints.

Unfortunately, HTML does not distinguish between links within and without a document. This means that no one can search documents more than one electronic page long; no one can navigate efficiently.

When a Web document is limited to a single page, you can move around it comfortably and efficiently; but if not, you are stuck.

Moreover, people often fail to think of a search command as a movement command, even though it is.

Worse, many people are accustomed to search commands in which you must decide your string before you search. This means that the search fails when you type too little.

A better user interface, for both searching and navigation, offers incremental search. As you type characters, they add to the search string and are found. Even though incremental search has existed for more than a human generation, only recently has it entered the browser world.

A third output format, Info, does distinguish between references that are to locations within a document and those without. This means that you can move through a multi-page document readily.

(Info files are created with the ‘`debiandoc2info`’ and ‘`makeinfo`’ commands.)

The Texinfo and DebianDoc packages offer commands for generating all renderings. Other packages do not.

What Computers Can Do

Every appropriately marked up document produced with a computer has multiple renderings in addition to the deep representation in which it is stored electronically. (Inappropriately marked up documents are limited; they may only format to one rendering.)

The three renderings of an appropriately marked up document are the high resolution, typeset format for printing on paper, usually DVI, PDF, or PostScript; HTML for Web pages; and Info for efficient online reading and listening.

Info provides a good listening format for Emacspeak. In effect, the text-to-speech mechanism provides a fourth rendering. (Emacspeak provides an audio desktop; it is different than a screen reader. It is also different from Emacs from which it derives. Of the four types of user integrating environment common today, command line, graphic, virtual lisp machine, and audio, Emacs is the third and Emacspeak the fourth.)

Consider a book: it consists of words (and perhaps images). Often, nowadays, it comes only in one surface expression, that of a typeset, printed book. But the words can be typed into a computer.

It takes editing to create a decent deep representation. But once that is done, you can produce a typeset and printed book, a frugal format for reading or listening online, and a typeset Web page, all with commands that take less than a second to give.

When you do not offer both the auditory and the frugal, you exclude the ‘situationally blind’, such as car drivers, who should keep their eyes on the road. And you exclude people who suffer from low bandwidth connections or who are using old or small devices, such as cell phones or PDAs.

As time passes, low bandwidth connections should become faster and small devices more powerful. It will become less and less important to be frugal. However, over the next few years, the number of ‘situationally blind’, will

blossom. People driving will want to listen. For those with cell phones or PDAs the technology for listening already exists. For those with laptops, listening is simply a matter of downloading, installing, and learning existing free software.

Texinfo

*Texinfo*¹ is a documentation system that uses a single source file to produce both online information and printed output. This means that instead of writing multiple documents, one for the printed work, a second for the Web site, and a third for Info, you need write only one document. Therefore, when the work is revised, you need revise only that one document.

Emacspeak enables Info to provide a fourth rendering to which you listen.

Unfortunately, many people treat computers as if they were 19th century typesetting machines that display printed output immediately. Worse, some programs offer a single, frozen typeset output in a WYSIWYG or ‘What You See Is What You Get’ interface.

In WYSIWYG interface, authors specify the document layout interactively while typing. Writers often specify the layout of a printed document by italicizing technical terms and the like. They impose structure on a document by typesetting it for printing.

In contrast the programs for Texinfo, as well as for DebianDoc, HTML, XML, SGML, \TeX , and \LaTeX impose structure by themselves. The writer tells the various formatters that a chapter title is a chapter title, that examples of code are examples of code, and so on. Computer programs do the presentation.

Thus, a computer program specifies in which voice to speak for titles, in which for examples of code, and in which for body text. For a printed document, it specifies in which fonts to typeset code, titles, and body text. You can, of course, tell the computer which font or which voice goes with which circumstance. But you need not tell it every time.

Currently, programs that help you write a Texinfo document do not show the various output expressions immediately. I would like to see an editor or word processor that shows at least two of the surface expressions simultaneously with the Texinfo deep representation.

See <http://www.rattlesnake.com/notions/multiple-output-preview.html>

Fortunately, such a program is nice but not necessary.

¹ The first syllable of “Texinfo” is pronounced like “speck”, not “hex”. This odd pronunciation is derived from, but is not the same as, the pronunciation of \TeX . In the word \TeX , the ‘X’ is actually the Greek letter “chi” rather than the English letter “ex”. Pronounce \TeX as if the ‘X’ were the last sound in the name ‘Bach’; but pronounce Texinfo as if the ‘x’ were a ‘k’. Spell “Texinfo” with a capital “T” and the other letters in lower case.

A Texinfo file consists of content plus markup.

You can learn many Texinfo commands. But most simply, you adapt ‘boiler plate’ for the beginning and use a few commands in chapters and sections. That is all.

Structure

Regardless of output format, most people write documents with the same structure. People tend to write a book, for example, in a specific order with chapters, sections, paragraphs, and sentences. Texinfo offers more freedom although few make use of it. An online expression need not have the same order or the same contents as a printed expression. You can see this in the Emacs manual.

However, many helper programs presume a standard format for all output expressions, whether read in a book, viewed online, or heard.

Texinfo mode in GNU Emacs provides quick ways to update a Texinfo file and to insert the most frequently used commands.

Also regardless of output format, you may choose which content to format. For example, if you are teaching in Massachusetts and are preparing documents for your students, you can choose to format the text of a Massachusetts law. On the other hand, if you are teaching in Texas, you can choose to format the text of a not-quite similar Texan law.

Texinfo provides two ways to select. The ‘arch’ version control system provides a third (see “Arch”, page 9).

To write a document, adapt ‘boiler plate’ for the beginning. Specify filename, title, author or authors, and copyright or copyrights.

For the content itself, you need to learn a few of the structuring commands so you can divide the work into chapters and sections. In addition, you may want to learn how to emphasize words, how to make cross references, and how to make index entries.

The Texinfo formatting commands automatically generate a table of contents. If you have made index entries and created the position for an index by inserting another stretch of boiler plate, the formatting commands will automatically generate an index, too.

If you wish, you can create footnotes, citations, and tables, and include images for various sorts for your different output formats. You can do a lot. But that takes more learning.

A Few Texinfo Commands

Texinfo commands start with ‘@’, an ‘at-sign’. To tell the computer what a command covers and what it does not, inline commands employ braces around the command arguments, { . . . }. Commands that go on a line by themselves do not need or use braces.

Two other common mark up languages, HTML and XML, use angle brackets, `< . . . >`, instead of at-signs. (HTML is used for Web pages.) HTML and XML derive from their predecessor, SGML (used in DebianDoc), which was invented in the same decade, the 1970s, as the predecessor to Texinfo.

In addition to the commands embedded in the ‘boiler plate’, the basic Texinfo commands are these:

@node

@chapter To enable node names to be shorter than chapter and section titles, the node and heading commands are separate.

Texinfo provides more heading commands than ‘chapter’: it provides section and subsection commands (and subsection commands too). Also, it provides the equivalents for unnumbered chapters (called **@unnumbered**, **@unnumberedsec**) and for appendices. The numbered and unnumbered chapters and appendices are listed in the table of contents. In addition, there is an unnumbered equivalent that does not appear in the table of contents.

The only two commands you need to learn early on are **@node** and **@chapter**.

For more detail, see [section “Structuring Command Types” in *The Texinfo Manual*](#).

@emph

@strong To emphasize text, use the **@emph** and **@strong** commands. The **@strong** command emphasizes text strongly. For paper typesetting, the **@emph** command usually prints in italics and the **@strong** command prints in bold. Emacspeak uses different voices.

For more detail, see [section “Emphasizing Text” in *The Texinfo Manual*](#).

@samp

@code To show a sample of text, use **@samp**; to show a sample of code, use **@code**. Texinfo has a more than a dozen highlighting commands.

You will use only a few frequently. You need not remember them all; although it is useful to know which exist.

For more detail, see [section “Marking Words and Phrases” in *The Texinfo Manual*](#).

@xref

To indicate a cross reference, use **@xref**. Besides **@xref**, there are four more variations for particular situations, such as the last part of a sentence.

Use **@uref** to produce a reference to a uniform resource locator (a URL).

See [section “Cross References” in *The Texinfo Manual*](#).

`@bye` End your Texinfo document with `@bye`.

Of course, you can, if you wish, learn more. For example, the previous two column table used the `@table` and `@end table` commands. Neither of those commands took braces; each stood on a line by itself.

You can format examples with `@example` and `@end example`, although I sometimes prefer the equivalent commands, `@smallexample` and `@end smallexample`, which typeset text in a smaller font.

There are many more ‘block enclosing commands’ (see [section “Block Enclosing Commands” in *The Texinfo Manual*](#)). Like the highlighting commands, these need not be learned, although it is useful to know which exist.

Perhaps the most critical additional command is the one for quotations.

`@quotation ... @end quotation`

The quotation command marks quotation. Like the other ‘block enclosing commands’, this command does not use braces. Each part, fits on a line by itself.

Personally, I like emphasising what I quote, so I usually write:

```
@quotation
@emph{Text of quotation}
@end quotation
```

Since examples and quotations often fit within paragraphs, it is a good idea to put `@noindent` on the line just preceding the next part so that a formatter does not inadvertently indent the text as a new paragraph. (You may turn off such indentation altogether, as I have done for this document, but not everyone likes that. Since it is easy to turn on such indentation, I have kept the `@noindent` commands in the Texinfo deep representation.)

See [section “@noindent: Omitting Indentation” in *The Texinfo Manual*](#).

Other useful commands are `@footnote` and `@cite`. (See [section “Footnote Commands” in *The Texinfo Manual*](#), and [section “@cite” in *The Texinfo Manual*](#).)

You can cause the various formatters to ignore a block of text with the `@ignore ... @end ignore` command, or to ignore the rest of a line with the `@c` or `@comment` command. You can put in an elipsis, three dots, `...`, with the `@dots{}` command. Since this command works within a line, it includes empty braces so the formatters can easily find the end of the command.

Conditional Text

There are three ways to cause the various formatters to include or exclude specific parts of the overall text.

- `@ifset` and `@ifclear`,
- `@include`, and
- ‘arch’ branches

The first two methods are built into Texinfo.

The first method uses the `@set` and `@clear` commands to specify which parts to include or exclude. I used those commands to mark another document whose Texinfo deep representation is ‘GNU-GPL-and-Commentaries.texi’. The specific parts to be included or excluded are marked with `@ifset ... @end ifset` and `@ifclear ... @end ifclear`.

The second method enables you to include files as include chapters with the `@include` command. That is how this document is written. When you comment out a chapter with `@c`, that chapter is excluded. (Incidentally, when you use the `@include` command, all the included files must be at the level of chapter. Moreover, you must update the document with `texinfo-multiple-files-update` rather than `texinfo-all-menus-update` and save all the files.)

The third method uses the ‘arch’ version control system (see “Arch”, page 9).

Formatting

You can use DVI or PDF or Postscript files for printing,

To format the Texinfo deep representation for this document to DVI, use

```
texi2dvi open-editing.texi
```

For PDF, use

```
texi2pdf open-editing.texi
```

For Postscript, you must first create a DVI file and then convert it:

```
dvi2ps open-editing.dvi > open-editing.ps
```

For Info, for Web sites, and for plain text, use ‘makeinfo’

Thus for Info, use

```
makeinfo open-editing.texi
```

However, I myself like the program to explain what it is doing, `--verbose`, prefer one very long file rather than numerous short ones (presuming the Texinfo file is long), `--no-split`, prefer to wrap at a fill column that is two less than the default, `--fill-column=70`, do not want to indent paragraphs, `--paragraph-indent=0`, and prefer to force output even if the file contains errors, `--force`.

So I use this command:

```
makeinfo --verbose --no-split --fill-column=70 \  
--paragraph-indent=0 --force open-editing.texi
```

Naturally, I do not type all those letters — I hate typing. I simply copy the command from some other file and change the file name.

I format a Web page with this command:

```
makeinfo --verbose --no-split --html open-editing.texi
```

And I format a plain text file with this one;

```
makeinfo --verbose --no-split --fill-column=70 \
  --paragraph-indent=0 --force \
  --no-headers --output=open-editing.txt open-editing.texi
```

Output Expressions

To see or listen to a file in Info, I use the `info` command in GNU Emacs or in Emacspeak.

That command has a keybinding, which is ‘`control-h i`’.

(It goes without saying that I make sure my control key is to the left of my `@` key; some keyboards label that key as ‘`Caps_Lock`’, as if they were labelling keys for an 1880s typewriter. No one in their right mind puts a Control anywhere else. In ancient times, stone masons built temples as if they were wood; these keylabelings are a contemporary example of the same habit.)

When I have not specified the location of the Info file in the appropriate ‘`dir`’ file, such as ‘`/usr/local/info/dir`’, I call the `info` command with a prefix, ‘`control-u control-h i`’.

(I never use the standalone Info command.)

To see a printed expression of the document on my screen I invoke one or the other of:

```
xdvi open-editing.dvi
```

```
gv open-editing.pdf
```

```
gv open-editing.ps
```

```
xpdf open-editing.pdf
```

Arch

Arch is a modern revision control system that grants developers the ability to work independently and merge code with one another when and if they choose.

Tom Lord wrote the current version, called `tla` (for ‘Tom Lord’s Arch’).

If you are not familiar with revision control systems, please look at

<http://www.gnuarch.org/web/gnu-arch/writings/what-is-revctl.html>.

For an introduction that is good for someone new, see

http://wiki.gnuarch.org/Quick_20Introduction.

It may be followed by Tom Lord’s tutorial:

<http://regexps.srparish.net/tutorial-tla/arch.html>.

*If you are are accustomed to an older revision control system such as CVS, please remember that **arch** is modern. It will be unfamiliar because of its additional capabilities, such as easy branching and merging.*

You can check out a project from someone else or start your own. Either way, **arch** makes it easy to cooperate with others.

The sources, whether originally from you or someone else need not be in a directory (or ‘folder’ as some people say) below the directory with the ‘**archive-name**’, although they can be.

It is a good idea to put your first archive, as named with ‘**archive-name**’, under a more general directory, so you can have several archives.

Note that an archive itself can have just one project in it, named with ‘**archive-name**’, or several.

At the beginning, the naming and such may seem complex; that is because **arch** is designed for projects that grow.

As an initial step, I will talk about getting a project from elsewhere.

Getting From Elsewhere

First, you need to tell **tla** who you are:

```
tla my-id "Robert J. Chassell <bob@rattlesnake.com>"
```

By convention, your **arch** ID is an email address along with your name.

To check out a project from elsewhere, such as **tla**, you need to register that archive so **tla** can learn its location:

```
tla register-archive http://www.seyza.com/archives/lord@emf.net--gnu-arch-2004
```

This results in a message saying

```
Registering archive: lord@emf.net--gnu-arch-2004
```

You can check that it is registered and its location with:

```
tla archives
```

To download the project’s source code, you need;

1. The archive’s name:

For example,

```
lord@emf.net--gnu-arch-2004
```

```
bob@rattlesnake.com--2005-law-keynote
```

```
stefan@xsteve.at--public-2005
```

2. Its location as a URI

For example,

```
http://www.seyza.com/archives/lord@emf.net--gnu-arch-2004
```

```
file://home/u/Legal-center-speech
```

```
http://arch.xsteve.at/2005
```

3. And its category, branch, and version

For example,

```
tla--devo--1.3

legal-keynote--main--0.1

xtla--main--1.1
```

(xtla is an Emacs mode for tla. ‘2005-law-keynote’ refers to the text of my speech at the 2005 Conference for Law School Computing.)

If you do not know the category, branch, and version of a project, you can discover them:

```
tla categories -A lord@emf.net--gnu-arch-2004
```

This results in a message saying

```
docs-tla
tla
```

Then

```
tla branches -A lord@emf.net--gnu-arch-2004 tla
tla versions -A lord@emf.net--gnu-arch-2004 tla--devo
```

which results in

```
tla--devo--1.3
```

Then you can switch to the directory that holds the repository

```
cd /usr/local/src/tla-1.3
```

and check more thoroughly

```
tla revisions --summary -A lord@emf.net--gnu-arch-2004
```

In my case that tells me:

```
base-0
  Top Level Directory for tla mainline development config
patch-1
  tla-1.3pre1 testing candidate release
patch-2
  tla-1.3pre2 testing candidate release
patch-3
  tla-1.3pre3 testing candidate release
version-0
  tla-1.3 released!
```

Incidentally

```
tla revisions = tla logs + tla missing
```

You can run either of the other commands with the ‘--summary’ option

```
tla logs -A lord@emf.net--gnu-arch-2004 --summary
```

```
tla missing -A lord@emf.net--gnu-arch-2004 --summary
```

In my case,

```
tla missing -A lord@emf.net--gnu-arch-2004 --summary
```

tells me nothing is missing — the command returns nothing at all. This means my repository is up to date. But let us suppose it were empty.

To get source the first time, give a `tla get` command:

```
tla get -A lord@emf.net--gnu-arch-2004 tla--devo--1.3
```

This, by the way, is the same as what you do to update your repository with others' work. I show updating in [“Updating from Others' Work”, page 13](#).

The `tla get` command has options and two arguments; the second argument may be left out. The `-A stefan@xsteve.at--public-2005` option tells `tla get` where to retrieve the source.

The first argument is `tla--devo--1.3`. Its double dashes separate the parts that tell `arch` the category, branch, and version to retrieve. This is version 1.3 of the development branch.

The second argument, which may be left out, is the name of the destination directory. This would be the name of the directory into which the repository is going. I am not showing it in this case. Were I to move one directory level up, it would be `'tla-1.3'`.

In case the remote sources go away, you can cache the sources locally,

```
tla cacherev tla--devo--1.3--version-0
```

where the optional argument tells you which revision to cache. The revision name has *four* parts separated by double dashes: category, branch, version, and revision

You do not need to specify the revision: as the documentation says, “If no revision is specified, but the command is run from within a project tree, cache the latest revision . . . ”

Making a Mirror

A mirror is useful if you work off line, but still want to make your changes permantly public. A mirror is an online repository to which you can write using the `archive-mirror` command and which others can retrieve.

You cannot make ordinary commits to a mirror archive, since it is designed to reflect your home system. Instead, the `tla archive-mirror` command writes to it.

First, you need to make the mirror. Here is what I did with secure FTP, `sftp`, onto an ISP whose machines stay on the Internet full time, unlike my home machine. The backslash followed by a carriage return at the end of the first line is an old mechanism for continuing the line without breaking it.

```
tla make-archive --listing --mirror bob@rattlesnake--2005-legal \  
sftp://bob@shell.isp.net:/home/bob/www/archives/2005-legal
```

Second, you can update the mirror:

```
tla archive-mirror bob@rattlesnake--2005-legal
```

Editing and Committing

You may edit your sources.

Then there are two steps to putting modified sources into `tla`:

- Make a log entry.
- Commit the sources and the log.

To do this, edit and invoke the appropriate commands:

1. Edit your sources ...
2. `tla make-log`
3. Edit the log entry
4. `tla commit`

Updating from Others' Work

You can use the `tla get` command to update your own repository, just as you can use it to get source the first time.

First, let us see what we have for the Emacs mode, `xtla`:

```
tla categories -A stefan@xsteve.at--public-2005
xtla
tla branches -A stefan@xsteve.at--public-2005 xtla
tla--dev
xtla--main
tla versions -A stefan@xsteve.at--public-2005 xtla--main
xtla--main--1.1
```

Then be sure to change to the directory that holds the `'xtla-main'` sources. (If you forget, as I did initially, you will confuse yourself.)

```
cd /usr/local/src/xtla-main
tla missing -A stefan@xsteve.at--public-2005
patch-22
patch-23
patch-24
patch-25
patch-26
patch-27
patch-28
patch-29
patch-30
patch-31
patch-32
```

I am missing quite a bit. I looked at a summary which I am not showing here and then updated with the `tla get` command:

```
tla get -A stefan@xsteve.at--public-2005 xtla--main--1.1
```

'xtla--main--1.1' comes from the results of the `tla versions` command and '-A' is the name of the archive.

Rather than copy `-A stefan@xsteve.at--public-2005` into your command each time, you can set your default with a command such as

```
tla my-default-archive stefan@xsteve.at--public-2005
```

Legal Solutions: Different Licenses

With the dropping cost of incremental and even initial production, restriction on writings serve primarily to hinder. The legal challenge is to enable people to stand tall rather than become victims.

Legally, how do you harness the power of police and courts to protect you and others, especially the small and the weak?

Although a computer program, a technical manual, and a poem are all bits on a computer, they have different characteristics to humans. For example, a computer program is like a cook's recipe, any part of which may change. A poem, on the other hand, is fixed.

Except as parody, you would not want to change Wordsworth's famous lines,

... all at once I saw a crowd A host of dancing Daffodills;

to

... all at once I saw a crowd A host of dancing Tulips;

Poems In Two Volumes, Vol 2,
by William Wordsworth

So a poem, like an editorial or a scientific paper, but unlike a computer program, is properly invariant. (You can make fair use changes regardless of license.)

For such a work, you can use a verbatim license. It should appear in the '@copying' section of a Texinfo document like this:

```
Verbatim copying and distribution of this entire document is permitted
in any medium, provided this notice is preserved.
```

Parts of a textbook, legal casebook, or technical manual are invariant, too, such as the introduction. The body, however, must be able to change.

The GNU Free Documentation License was designed for such a situation. You may specify invariant sections. Other sections may be modified; if modifications are made, they must be noted.

Moreover, since publishers seek monopoly or partial monopoly rather than change their business model to that of providing services like lawyers, the GFDL restricts you and me a little. You may require up to five words that you specify on the front cover and up to 25 words on the back cover, and if

another publishes more than one hundred printed copies, that text must be included.

The GNU Free Documentation License was designed to balance competitive, free markets with hard copy publishers' desire for monopoly or partial monopoly. It offers a clear definition of what is 'commercial'.

Copyright Law

The various licenses employ copyright law. That is why they are so strong. As Eben Moglen, who is a professor of law and legal history at Columbia University Law School as well as the General Counsel of the Free Software Foundation, points out,²

Licenses are not contracts: the work's user is obliged to remain within the bounds of the license not because she voluntarily promised, but because she doesn't have any right to act at all except as the license permits.

The Purpose of Copyright

Licenses are enforced through police and courts. In the United States of America, according to its Constitution, the purpose of copyright is

To promote the progress of science and useful arts, . . .

(U. S. Constitution, Article I, Section 8)

How long should police and courts enforce the license? Currently, in the U. S., the duration extends from President Herbert Hoover's time. Is this duration enough, not enough, or too much?

What if the time were limited to seven years? Would that be enough time?

Legal Textbooks and Casebooks

I am told that nowadays in the United States, legal texts are priced higher than they would be in a competitive, free market. An economist would refer to the top three publishers, all foreign owned, as an oligopoly. (Incidentally, in school forty years ago, I was taught how to price fix in such an oligopoly without breaking U. S. law.)

Thus, in law schools, as John Mayer told me,³

. . . faculty teach, use each other's textbooks and improve on them in the classroom — but the chain is broken there. There is little opportunity and no incentive for a faculty member to contribute ideas, changes and refinements back to the author or publisher of

² *Enforcing the GNU GPL*,
Eben Moglen, 10 September 2001,
<http://www.gnu.org/philosophy/enforcing-gpl.html>

³ Personal communication

the original textbook. There are a million ways to teach, but very few ways to teach something well.

Often, nowadays, a legal casebook comes in only one surface expression, that of a typeset, printed book. But the words can be typed into a computer. Indeed, many legal cases are already online; so is the United States Code. With the advance of technology, textbooks and casebooks need not be limited to one rendering. Their production need not fit the desires of an oligopoly.

Trust

When you teach from or study a textbook or casebook, you must trust it. It does you no good if you are fooled into studying a falsehood. Organizations that distribute textbooks and casebooks must be trustworthy.

A fellow asked me,

How do I know exactly unless I was there?

My response was straightforward:

You don't.

That is the problem. You must depend on what others tell you. A writer citing laws and legal cases cannot go to all courts in session simultaneously; that writer cannot go to court sessions before he or she was born.

The only way to discover which laws were passed and what decisions were made is to trust others. Without trust, no one can succeed.

But then I pointed out that an advantage of the Internet is that organizations can use it to create inexpensive new forms of ‘credibility filter’.

You do not have to depend on the old mechanisms of banks, insurance companies, and encyclopedias, all of which were and still are trust based institutions (See <http://www.rattlesnake.com/notions/trust-based-institutions.html>)

One modern mechanism is to provide tags tell you how others judge reputations. (And provide MD5 sums or an equivalent to ensure that you receive the information as intended.)

Take a look at how Slashdot does this. (Slashdot is an online news and commentary site for ‘nerds’. I am not saying that it provides the only or best mechanism, merely that its method is good enough for its forum. As far as I am concerned, Slashdot’s level 5 items are far better than its level 1 and less than one-tenth as commonplace. To me, Slashdot shows how to provide reviewing services that are probabilistically OK. See <http://slashdot.org/index.pl?mode=flat>.)

The Slashdot trust-building mechanism makes use of randomly selected, temporary judges who also fit additional criteria. Basically, a judge, called a ‘moderator’ in Slashdot, must be logged onto the system. (See <http://slashdot.org/faq/com-mod.shtml>.)

Of course, if your effort is successful, many hostile people will log on, or a few people will log on with many aliases. Presume that millions of dollars can be spent in an attack against you . (Funding can come from PR and advertising, from an organization of believers, or from the covert operations budget of another country.)

The Slashdot mechanism requires that a judge be a long time regular and willing to serve. These characteristics are open to abuse by a well-funded opponent, but will force the opponent to spend more.

Also, a judge must be judged well by other judges. The practice is called *metamoderation*.

Each judge receives a number of points. Points expire three days after the last judgement. Moreover, each judgement costs a point. When a judge uses all his or her points or after they expire, he or she stops being a judge.

The latter conditions means that a fraud must first gain the respect of other judges and then wait until randomly chosen to be a judge. Obviously, a well funded opposition can undermine such a constitution, but that is expensive.

A similar trust-building mechanism can be applied elsewhere. A backstop, for example, enables trust-building to be applied to governance. Without a backstop, people will trust what they believe which may be quite foolish, as was the belief that tuberculosis germs could not develop resistance. (See <http://www.rattlesnake.com/notions/Choice-and-Constraint.html#Science>.)

While some may disbelieve a particular investigator, temporary judges will know several. It is likely that temporary judges will not know investigators personally, but they will have figured out whether to adapt their judgments to the results of those investigations.

And the rest of us can, probabilistically speaking, come to depend on these judgements.

Trust-building is not so necessary when an organizational system compensates: that is why it is useful to emply the Christian presumption that people are *Fallen*, even if you are not Christian. In politics, that presumption leads you to figure that ‘power corrupts’ and to install various ‘checks and balances’. Otherwise, you will call for ‘virtuous rulers’, yet find that you are ruled by frauds who pretend to be virtuous but are not.

But many governing systems lack checks and balances. Even with them, and with better policies, it helps to bear in mind, as Michael Fromkin said, that

... collaborative reputational systems can help identify who the group believes is advancing the debate and who is impeding it. ...

(See <http://islandia.law.yale.edu/isp/GlobalFlow/paper/Fromkin.pdf> and <http://www.rattlesnake.com/notions/petals-of-cooperation.html>.)

So far and fortunately, advances in technology have enabled us to deal with the trust issues that arise as more and more people deal with strangers. We need to adapt our current institutions and create new ones to continue to benefit.

The Social Problems

The social problems of open and freely redistributable documentation have not been solved. They have not been solved on a personal level and they have not been solved on a corporate level.

On a personal level, many contemporary people write for just one output format. This is from habit. In the past, people wrote onto paper, either with pen or typewriter. Paper was the one significant form that output could take. In the past, the only salient blind were the permanently blind, and they are a small portion of the population.

Put in a way that might appeal to those who like acronyms,

- Paper provides a W T S W I W T R S interface (spoken as ‘**what-swi-wi-ters**’), ‘What The Sender Wrote Is What The Recipient Sees’.
- Computers provide a Y R C H T V O L format (spoken as ‘**yrch-ti-vol**’), ‘Your Recipient Chooses How To View Or Listen’.

Yet today, many people think only of WYSIWYG. They think of a ‘What You See Is What You Get’ program that provides a single, frozen typeset output format. They do not think of a frugal but highly efficient online format or of a typeset online format.

A generation ago, man pages and Texinfo were developed as formats for multiple outputs — typeset outputs for paper and the ‘linked’ outputs for online work.

Sad to say, even computer geeks tend to write for just one format. I have seen this often.

On a corporate level, many companies resist changing their business models. As the cost of reduplicating information drops, they seek more governmental policing. Like the old time dumping of toxic wastes, governmental policing costs them nothing.

For civilization, to police against advances in technology is to fail.

The better business solution is to adapt, as IBM is doing for example. Businesses can sell services rather than package them as products that depend on policing to keep their prices high.

Such companies could become more like you, who sell your services.

As I said, for legal textbooks and casebooks, the technical problems have been solved. We can and should improve the technical solutions, but the critical work has been done.

The legal problems have also been solved. Other legal questions exist and should be addressed; but for documentation, the work has been done.

Appendix A GNU Free Documentation License

Version 1.2, November 2002

Copyright © 2000,2001,2002 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.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document *free* in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of “copyleft”, which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The “Document”, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as “you”. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A “Modified Version” of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A “Secondary Section” is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document’s overall subject (or to related matters) and contains nothing that could fall directly within

that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The “Invariant Sections” are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The “Cover Texts” are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A “Transparent” copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not “Transparent” is called “Opaque”.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The “Title Page” means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, “Title Page” means the text near the most prominent appearance of the work’s title, preceding the beginning of the body of the text.

A section “Entitled XYZ” means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as “Acknowledgements”, “Dedications”, “Endorsements”, or “History”.) To “Preserve the Title” of such a section when you modify the Document means that it remains a section “Entitled XYZ” according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document’s license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque

copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If

there is no section Entitled “History” in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the “History” section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled “Acknowledgements” or “Dedications”, Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled “Endorsements”. Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled “Endorsements” or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version’s license notice. These titles must be distinct from any other section titles.

You may add a section Entitled “Endorsements”, provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled “History” in the various original documents, forming one section Entitled “History”; likewise combine any sections Entitled “Acknowledgements”, and any sections Entitled “Dedications”. You must delete all sections Entitled “Endorsements.”

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an “aggregate” if the copyright resulting from the compilation is not used to limit the legal rights of the compilation’s users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire

aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document 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.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation 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. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

A.0.1 ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

```
Copyright (C)  year  your name.
Permission is granted to copy, distribute and/or modify this document
under the terms of the GNU Free Documentation License, Version 1.2
or any later version published by the Free Software Foundation;
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.
A copy of the license is included in the section entitled ‘‘GNU
Free Documentation License’’.
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the “with...Texts.” line with this:

```
with the Invariant Sections being list their titles, with
the Front-Cover Texts being list, and with the Back-Cover Texts
being list.
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.