

# **Imprimatur – a Texinfo Document Checker**

---

version 1.0, 19 September 2011

**Sergey Poznyakoff.**

---

Published by the Free Software Foundation, 51 Franklin Street, Fifth Floor, Boston, MA  
02110-1301 USA

Copyright © 2011 Sergey Poznyakoff

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, with any Front and Back-Cover. A copy of the license is included in the section entitled “GNU Free Documentation License”.

## Short Contents

1	Preface .....	1
2	Introduction .....	3
3	Renditions .....	5
4	Integrating and Using Imprimatur .....	7
5	Initialization .....	11
6	Texinfo Macros .....	13
7	Makefile Rules .....	15
8	check-docs.sh .....	19
9	Reporting Bugs .....	21
A	GNU Free Documentation License .....	23
	Concept Index .....	31



## Table of Contents

<b>1</b>	<b>Preface .....</b>	<b>1</b>
<b>2</b>	<b>Introduction.....</b>	<b>3</b>
<b>3</b>	<b>Renditions.....</b>	<b>5</b>
<b>4</b>	<b>Integrating and Using Imprimatur.....</b>	<b>7</b>
	4.1 Integration.....	7
	4.2 How to Use Imprimatur.....	8
<b>5</b>	<b>Initialization .....</b>	<b>11</b>
<b>6</b>	<b>Texinfo Macros .....</b>	<b>13</b>
<b>7</b>	<b>Makefile Rules .....</b>	<b>15</b>
<b>8</b>	<b>check-docs.sh .....</b>	<b>19</b>
<b>9</b>	<b>Reporting Bugs.....</b>	<b>21</b>
	<b>Appendix A GNU Free Documentation License</b>	
	.....	<b>23</b>
	A.1 ADDENDUM: How to use this License for your documents....	29
	<b>Concept Index.....</b>	<b>31</b>



# 1 Preface

**Imprimatur** is a set of Texinfo macros and Makefile rules to verify consistency and, to certain extent, completeness of Texinfo documentation.

This package was inspired in part by François Pinard's `'rendition.texi'` file we used in GNU tar's documentation. It was developed between 2005 and 2011 while writing documentation for several other free software projects its author maintains: GNU Mailutils, Dico and Pies, to name a few. It was reorganized into a standalone submodule project in 2011.





## 2 Introduction

Creating a documentation for software is a difficult task which is equally important as writing the software itself. In particular, it is important that the documentation be consistent with the software it describes and that it cover all aspects of its functionality and usage.

Usually the documentation, as well as the program itself, is not written in one pass. Rather it is created in a series of edits and proofreadings. During this process the author often needs to insert *annotations* which are not intended for the end user, but which, instead, serve as a reminder for the author himself about pending editing tasks. One may, for example, put a note saying that certain paragraph needs to be reworded or that some feature needs to be explained in greater detail and return to that note later in order to fix it. Quite often several authors take part in creating the documentation. In this case such annotations help coordinate their work.

When preparing the document for publishing, it is important to make sure that all such notes has been revised and none of them remain in the document.

Keeping documentation in sync with the software is another challenging task. After adding a new feature the author not always edits the documentation to reflect that. In fact, quite often it is impossible or undesirable, because the feature may imply some further changes to another parts of the program, and the author may prefer to update the docs when all the changes will have been finished. In this case, putting annotations in appropriate places of the document will remind him that certain parts of it have become inaccurate and need to be attended later.

*Imprimatur*<sup>1</sup> aims to facilitate all these tasks. It is intended for documents written in Texinfo (see [Section “Top” in \*Texinfo Manual\*](#)). The package provides a mechanism for editor’s annotations and a set of Makefile rules for verifying the consistency of the document.

---

<sup>1</sup> ‘*Imprimatur*’ is Latin for ‘let it be printed’. Historically, this term was used to indicate that there are no obstacles for printing a book.



## 3 Renditions

The basic concept used in `Imprimatur` is the document *rendition*. The rendition defines how much editor’s information and annotations should appear in the resulting document. There are three possible renditions:

**PROOF** [Rendition]

This rendition is used for proofreading the document. It inserts all annotations in a prominent form, which makes it easy to revise the document.

**DISTRIB** [Rendition]

This rendition is intended for the documentation distributed with the packaged versions and its online copies on the Web. It inserts most important annotations, so that the reader is aware about which parts of the document are not fully revised for adequacy. Dangling cross-references are also prominently marked as such (see [[Dangling cross-reference](#)], page 14).

**PUBLISH** [Rendition]

This rendition is intended for publishing printed copies of the documentation. It is supposed that by the time the document is ready for publishing, all editor’s and proofreader’s annotations and dangling references have been resolved and removed from the document. Consequently, the use of annotation macros in this rendition results in translation errors. This helps prevent publishing incomplete or inaccurate documentation.

Default rendition is determined when configuring the package. It can be redefined when building the output, by using the ‘`RENDITION`’ Makefile variable, e.g.:

```
$ make RENDITION=PUBLISH
```



## 4 Integrating and Using Imprimatur

This chapter describes how to integrate `Imprimatur` to your project and provides a brief overview on using it.

The project wishing to use `Imprimatur` must meet the following requirements: it must use `Autoconf` and `Automake` and it must use `Git` as a version control system.

### 4.1 Integration

`Imprimatur` is designed to be used as a `Git` submodule. If your project uses `git`, integrating it is quite straightforward.

1. First of all, you need to declare a submodule. To do so, change into the top source directory of your project and run:

```
git submodule add git://git.gnu.org.ua/imprimatur.git imprimatur
git submodule init
```

This step needs to be done only once. The first command will clone the project to the directory `'imprimatur'`. If you want another name or need to place it deeper in the directory hierarchy, change the last argument accordingly. For example, to place `Imprimatur` to the directory `'doc/aux'` use:

```
git submodule add git://git.gnu.org.ua/imprimatur.git doc/aux
```

The second command initializes the submodule.

2. Next step is to edit the top-level `'Makefile.am'`. Add `'-I imprimatur'` to the `'ACLOCAL_AMFLAGS'` variable, and `'imprimatur'` directory to the `'SUBDIRS'` variable. For example:

```
ACLOCAL_AMFLAGS = -I m4 -I imprimatur
SUBDIRS = imprimatur
```

If you chose another directory name during the first step, use it instead of `'imprimatur'` in the above example.

3. Next, you need to edit the `'configure.ac'` file and add a call to `'IMPRIMATUR_INIT'`. Normally no parameters are needed, just like that:

```
IMPRIMATUR_INIT
```

However, if you cloned `Imprimatur` into a directory with another name, the actual directory name must be supplied as the first argument. For example:

```
IMPRIMATUR_INIT(doc/aux)
```

See [Chapter 5 \[Init\], page 11](#), for a detailed description of the `'IMPRIMATUR_INIT'` macro.

4. Now edit the file `'Makefile.am'` located in the subdirectory where you have your `Texinfo` sources.
  - A. To the value of `'AM_MAKEINFOFLAGS'` variable add `'@IMPRIMATUR_MAKEINFOFLAGS@'`. This will inform `makeinfo` and related tools about the location of `Imprimatur` files (in particular, `'rendition.texi'`) and the selected rendition (see [Chapter 3 \[Renditions\], page 5](#)).

- B. Define the variable ‘`imprimatur_INPUT`’. It must contain the names of Texinfo sources to be verified by Imprimatur. Normally, the following definition is sufficient:

```
imprimatur_INPUT=$(info_TEXINFOS) $(base_TEXINFOS)
```

where *base* stands for the base name of your Texinfo document (e.g. ‘`foo`’, if it is named ‘`foo.texi`’).

See [Chapter 7 \[Rules\]](#), [page 15](#), for a discussion of this variable and its purposes.

- C. Include the file ‘`imprimatur.mk`’ from the Imprimatur directory using relative addressing. For example, if your documentation subdirectory is located at the same nesting level as the directory you cloned Imprimatur to, use:

```
include ../imprimatur/imprimatur.mk
```

*Do not use Automake substitutions nor Makefile variables in the argument to include.*

- D. If you plan to use the ‘`check-docs.sh`’ script, define a Makeinfo variable to access it, e.g.:

```
CHECK_DOCS=$(top_srcdir)/@IMPRIMATUR_MODULE_DIR@/check-docs.sh
```

See [Chapter 8 \[check-docs.sh\]](#), [page 19](#), for a discussion of this script.

Let’s summarize this step by an example:

```
AM_MAKEINFOFLAGS = @IMPRIMATUR_MAKEINFOFLAGS@
imprimatur_INPUT=$(info_TEXINFOS) $(foo_TEXINFOS)
include ../imprimatur/imprimatur.mk
CHECK_DOCS=$(top_srcdir)/@IMPRIMATUR_MODULE_DIR@/check-docs.sh
```

## 4.2 How to Use Imprimatur

To use Imprimatur, you need to include the file ‘`rendition.texi`’ into your main Texinfo source file:

```
@include rendition.texi
```

This file provides you with macros for annotating your documentation. These macros are described in detail in [Chapter 6 \[Macros\]](#), [page 13](#). Here we present a short overview.

The most common form of annotation is the ‘`@FIXME`’ macro. It introduces an editor’s note about something that needs a revision or clarification. For example:

```
The @var{fmt} argument is a database format identifier.
If it is valid, the function returns expiration interval for that
format. @FIXME{It is not clear how to obtain negative expiration
values.}
```

In ‘`PROOF`’ rendition, this Texinfo excerpt produces the following output:

The *fmt* argument is a database format identifier. If it is valid, the function returns expiration interval for that format.

**Editor’s note:**

It is not clear how to obtain negative expiration values.

In other renditions, the ‘@FIXME’ macro produces no output at all.

It is supposed that you will be progressively *resolving* your ‘@FIXME’s while reviewing your document. During this process you can obtain a listing of all pending ‘@FIXME’ notes by running `make imprimatur-fixmes`, e.g.:

```
$ make imprimatur-fixmes
Unresolved FIXMEs:
functions.texi:2231: It is not clear how to obtain negative expiration
```

It is especially handy if you use ‘`compile`’ mode of GNU Emacs, as you can then easily move editing point to the place when a particular note appears by placing the cursor on the corresponding line and hitting *Enter* (see Section “*Compilation Mode*” in *The Emacs Editor*).

Another common annotation is the ‘@UNREVISED’ macro. Placed after a sectioning command, it will draw reviewer’s attention to nodes that need a revision. A listing of such nodes can be obtained using the ‘`imprimatur-unrevised`’ rule:

```
$ make imprimatur-unrevised
Unrevised nodes:
mailfromd.texi:2567:@UNREVISED
functions.texi:1508:@UNREVISED
```

The ‘@UNREVISED’ macro produces output in ‘PROOF’ and ‘DISTRIB’ renditions. In ‘PUBLISH’ rendition it results in compilation error. This behavior is intended to help avoid unrevised nodes from appearing in final published texts.

When writing initial revision of a node you may need to mark a point where a cross-reference to another, not yet written node should be. The intent is to replace this mark with an actual cross-reference command when the node referred to has already been written. For that purpose Imprimatur provides three annotations: ‘@FIXME-ref’, ‘@FIXME-xref’, and ‘@FIXME-pxref’. These macros are similar to corresponding Texinfo commands, except that they take a single argument. In fact in ‘PUBLISH’ rendition these macros are equivalent to their Texinfo counterparts. In ‘PROOF’ and ‘DISTRIB’ rendition, however, they produce a cross-reference explicitly marked as referencing a non-existent node. For example:

```
Enable transcript of @acronym{SMTP} sessions to the log
channel. @FIXME-xref{Logging and Debugging}.
```

This Texinfo fragment produces:

```
Enable transcript of SMTP sessions to the log channel. See Logging and Debug-
ging (Editor’s note: dangling link).
```

To list unresolved cross-reference use the ‘`imprimatur-refs`’ Makefile rule:

```
$ make imprimatur-refs
Unresolved cross-references:
options.texi:4951: Logging and Debugging
```





## 5 Initialization

**IMPRIMATUR\_INIT** (*dir*, *options*) [Autoconf Macro]

Initializes *Imprimatur* framework. Both arguments are optional. The *dir* argument, if supplied, specifies the directory you cloned *Imprimatur* into. It defaults to ‘*imprimatur*’, so that if you follow the default setup described in [Section 4.2 \[Usage\]](#), [page 8](#) you don’t need to supply this argument.

The *options* argument is a whitespace-separated list of options. Following options are recognized:

**PROOF** [Option]

Set the default rendition to ‘PROOF’.

**DISTRIB** [Option]

Set the default rendition to ‘DISTRIB’.

**PUBLISH** [Option]

Set the default rendition to ‘PUBLISH’.

If none of these three options is given, **IMPRIMATUR\_INIT** will select default rendition based on the version number of the hosting package, i.e. the ‘**VERSION**’ variable, as set by **AC\_INIT** macro. The version number should consist of up to three integers, separated by dots. The first integer is the *major number*, the second one is the *minor number*, and the third one is the *patchlevel*. Given that, the following algorithm is used:

1. If *patchlevel* is present, then:
  - a. If it is less than 50, ‘DISTRIB’ rendition is selected.
  - b. Otherwise, the package is considered an alpha release and ‘PROOF’ rendition is selected.
2. If *patchlevel* is not present, ‘DISTRIB’ is used.

**frenchspacing** [Option]

Assume French sentence spacing. See [Section “frenchspacing” in \*Texinfo Manual\*](#), for a description of the two possible sentence spacings.

By default, American sentence spacing is assumed. This option affects the ‘*imprimatur-check-sentence-spacing*’ Makefile rule, which verifies whether sentence spacing is used consistently throughout the document. See [\[\*imprimatur-check-sentence-spacing\*\]](#), [page 16](#).

**makedoc** [Option]

Enables creation of a separate makefile for building *Imprimatur* documentation. This file is called ‘*Makedoc*’ and contains rules for building this manual in various output formats. With this option on, the ‘*Makefile*’ in the ‘*imprimatur*’ subdirectory will also contain several additional shortcut rules, namely:

‘*make info*’

‘*make imprimatur.info*’

Creates the manual in info format (a shortcut for `make -f Makedoc imprimatur.info`).

`'make pdf'` Creates a PDF output (a shortcut for `make -f Makedoc pdf`).

`'make dvi'` Creates a DVI output.

`'make ps'` Creates a PostScript output.

This is intended mostly for `Imprimatur` maintainers, but you may use this option to build your own, perhaps customized, copies of this manual.

`dist-info` [Option]  
Distribute the `'imprimatur.info'` file. This option is valid only in conjunction with `'makedoc'`.

The `IMPRIMATUR_INIT` macro defines several substitution variables, which you can use in your `'Makefile.am'` files.

`IMPRIMATUR_MODULE_DIR` [Substitution Variable]  
The name of the directory `Imprimatur` was cloned into. This name is relative to the top source directory. It helps address files from the `Imprimatur` subdirectory. For example:

```
CHECK_DOCS=$(top_srcdir)/@IMPRIMATUR_MODULE_DIR@/check-docs.sh
```

`IMPRIMATUR_MAKEINFOFLAGS` [Substitution Variable]  
Additional options for `makeinfo` and compatible tools, which inform them about location of `Imprimatur` Texinfo files. You need to add this variable to your `'AM_MAKEINFOFLAGS'` variable:

```
AM_MAKEINFOFLAGS = @IMPRIMATUR_MAKEINFOFLAGS@
```

`RENDITION` [Substitution Variable]  
Default rendition. You may need this variable to inform various tools about the selected rendition level, e.g.:

```
TEXI2DVI=texi2dvi -t '@set $(RENDITION)'
```

Note, that `'IMPRIMATUR_MAKEINFOFLAGS'` includes the option `'-D $(RENDITION)'`, so you don't need to define it explicitly if the tool in question uses this variable.

## 6 Texinfo Macros

The file ‘`rendition.texi`’ defines helper macros which serve to annotate Texinfo sources. Expansion of these macros depends on the rendition of the document.

**WRITEME** [Macro]

Use this macro to mark the nodes which need to be written. Normally, you would place it immediately after a sectioning command. When in ‘PUBLISH’ rendition, this macro forces compilation error, because one would hardly want empty nodes to appear in the published document. Otherwise, it produces the following text:

**Editor’s note:**

This node is to be written.

**UNREVISED** [Macro]

Use this macro to mark the nodes which need a revision. Normally, you would place it immediately after a sectioning command.

This macro acts similarly to ‘@WRITEME’. When in ‘PUBLISH’ rendition, it forces a compilation error. In ‘PROOF’ and ‘DISTRIB’ renditions, it produces the following expansion:

**Editor’s note:**

The information in this node may be obsolete or otherwise inaccurate.  
This message will disappear, once this node revised.

**FIXME** (*text*) [Macro]

The ‘@FIXME’ macro adds an editor’s note to the file. The argument *text* gives the content of the note.

The purpose of the note is to draw reviewer’s attention to something near this point of the document. It is of some importance only to the author and reviewers of the document, so it produces output only in ‘PROOF’ revision. Consider, for example, this text:

```
The @option{-I @var{DIR}} command line option adds a directory
to the program’s search path. @FIXME{Document the initial value
of the search path.}
```

In ‘PROOF’ rendition it would produce the following:

```
The ‘-I DIR’ command line option adds a directory to the program’s
search path.
```

**Editor’s note:**

Document the initial value of the search path.

The three macros below serve as placeholders for the corresponding cross-reference commands:

**FIXME-ref** (*text*) [Macro]  
 Use this macro for ‘@ref’-style cross-references. In ‘PUBLISH’ rendition it is entirely equivalent to @ref{*text*}.

**FIXME-xref** (*text*) [Macro]  
 Use this macro for ‘@xref’-style cross-references. In ‘PUBLISH’ rendition it is equivalent to @xref{*text*}.

**FIXME-pxref** (*text*) [Macro]  
 Use this macro for ‘@pxref’-style cross-references. In ‘PUBLISH’ rendition it is entirely equivalent to @pxref{*text*}.

Use one of these macros when you need to add a cross-reference to a node or anchor which does not exist yet. They will create a *dangling reference* which allows for translating Texinfo source without errors, and at the same time serves as a reminder to you that the node referred to still needs to be written (see [Chapter 7 \[Rules\], page 15](#)). Once you create the node, you will fix the reference by simply removing the ‘FIXME-’ prefix after the ‘@’ sign.

To illustrate the output produced by these macros, consider the following Texinfo fragment:

```
@FIXME-xref{regexp}, for more information about regular expressions.
```

In ‘PROOF’ mode it produces:

```
See regexp (Editor’s note: dangling link), for more information about regular expressions.
```

In ‘DISTRIB’ rendition, the output is:

```
See regexp, for more information about regular expressions.
```

## 7 Makefile Rules

The file `imprimatur.mk` contains Makefile rules for verifying the Texinfo documents. It should be included into `Makefile.am` located in the directory where Texinfo sources reside. For example, if your documentation subdirectory is located at the same nesting level as the directory you cloned `imprimatur` into, then add the following statement:

```
include ../imprimatur/imprimatur.mk
```

Note that Automake is supposed to expand this statement, i.e. to replace it with the actual contents of `imprimatur.mk` in the resulting `Makefile.in`. This is possible only if the argument to `include` is a literal string. Do not use Automake substitutions nor Makefile variables in the argument to `include`.

The rules defined in `imprimatur.mk` operate on Texinfo documents listed in the variable `imprimatur_INPUT`

`imprimatur_INPUT` [Makefile Variable]

Whitespace-delimited list of Texinfo source files to process. It is often convenient to specify source files via the `name_TEXINFOS` Automake variables. For example, if your main source has the name `myproj.texi` it would normally be added to the Makefile variable `info_TEXINFOS` and the variable `myproj_TEXINFOS` would contain subordinate Texinfo sources it depends on. Then, you would declare `imprimatur_INPUT` as follows:

```
imprimatur_INPUT=$(info_TEXINFOS) $(myproj_TEXINFOS)
```

Once this variable is defined, you can use the following rules:

`imprimatur-format` [Makefile Rule]

Checks whether the files contain tabs (ASCII 9). If so the following diagnostics is displayed:

```
Sources contain tabs; run make imprimatur-untabify
```

See [\[imprimatur-untabify\]](#), page 17.

`imprimatur-refs` [Makefile Rule]

Checks if the sources contain unresolved cross-references, i.e. any of the macros `FIXME-ref`, `FIXME-xref` or `FIXME-pxref` (see [Chapter 6 \[Macros\]](#), page 13). If so, the prominent diagnostics is displayed and the rule exits with code 2. For example:

```
Unresolved cross-references:
mailfromd.texi:3535: existing categories
mailfromd.texi:4951: optimizer
mailfromd.texi:9899: debug configuration
functions.texi:1273: filter chains
```

`imprimatur-fixmes` [Makefile Rule]

Checks whether the sources contain `@FIXME` macros (see [Chapter 6 \[Macros\]](#), page 13). If so, list the location of each `@FIXME` along with its content. If the content has several lines, only first of them is displayed. For example:

```

Unresolved FIXMEs:
functions.texi:1518: Check references to 'Polling functions'
functions.texi:1562: Give more details and references.
functions.texi:2231: How to obtain negative

```

#### imprimatur-writemes [Makefile Rule]

Checks whether the sources contain empty nodes, i.e. nodes marked with the '@WRITEME' macro (see [Chapter 6 \[Macros\], page 13](#)). If so, it displays locations of each '@WRITEME', e.g.:

```

Empty nodes:
calloutd.texi:6:@WRITEME
mfdbttool.texi:148:@WRITEME

```

#### imprimatur-empty-nodes [Makefile Rule]

An improved version of 'imprimatur-writemes'. This rule reports nodes which do not have any text and those marked with the '@WRITEME' macro.

Unlike the rest of rules, 'imprimatur-empty-nodes' analyzes files in '\$(info\_TEXINFOS)'. It reads each file and parses it, honoring any '@include' statements.

This rule outputs to standard error the names and locations of empty nodes. For nodes marked with the '@WRITEME' macro, location refers to the place where the macro was encountered and the actual node name is prefixed with '(@WRITEME)', as shown in the example below:

```

Empty nodes:
./calloutd.texi:6: calloutd
./mfdbttool.texi:148: Configuring mfdbttool
./upgrade.texi:31: (@WRITEME) Upgrading from 7.0 to 8.0

```

If you wish to excerpt any of the input files from being verified by this rule, you can do so by adding the following *pragmatic comment* at the beginning of such files:

```
@c imprimatur-ignore
```

#### imprimatur-unrevised [Makefile Rule]

Checks whether the sources contain nodes which need revision, i.e. the ones marked with the '@UNREVISED' macro (see [Chapter 6 \[Macros\], page 13](#)). If so, display locations of each '@UNREVISED', e.g.:

```

Unrevised nodes:
mailfromd.texi:2567:@UNREVISED
mailfromd.texi:2694:@UNREVISED

```

#### imprimatur-check-sentence-spacing [Makefile Rule]

Checks if sentence spacing is correct throughout the sources. By default, American sentence spacing is assumed. French sentence spacing is used if IMPRIMATUR\_INIT was called with the 'frenchspacing' option (see [Chapter 5 \[Init\], page 11](#)).

If the sentence spacing is wrong, this rule displays the following diagnostics:

```

Sources contain single-space sentence separators.
Run make imprimatur-fix-sentence-spacing to fix.

```

If French spacing is declared, the diagnostics will read:

Sources contain double-space sentence separators.  
Run `make imprimatur-fix-sentence-spacing` to fix.

`imprimatur-fix-sentence-spacing` [Makefile Rule]

Fixes sentence spacing. The action of this rule depends on the setting of ‘`frenchspacing`’ option (see [Chapter 5 \[Init\]](#), page 11).

`imprimatur-basic-checks` [Makefile Rule]

Runs the following rules in that order:

- `imprimatur-format`
- `imprimatur-check-sentence-spacing`
- `imprimatur-refs`
- `imprimatur-fixmes`
- `imprimatur-empty-nodes`
- `imprimatur-unrevised`

`imprimatur-master-menu` [Makefile Rule]

Creates a master menu for each Texinfo file in ‘`#{info_TEXINFOS}`’, taking into account included files.

This rule uses `Emcas` to create a master menu that follows the top node. The master menu includes every entry from all the other menus. It replaces any existing ordinary menu that follows the top node.

The effect of this rule differs from that of `texinfo-master-menu` (see [Section “Updating Commands” in \*Texinfo Manual\*](#)) in that it takes into account included Texinfo files.

`imprimatur-untabify` [Makefile Rule]

Converts all tabs in Texinfo files to multiple spaces, preserving columns. This rule uses `emacs` to do the job.

`imprimatur-final` [Makefile Rule]

Runs ‘`imprimatur-untabify`’ and ‘`imprimatur-master-menu`’.





## 8 check-docs.sh

The shell script ‘`check-docs.sh`’ provides a mechanism for verifying if all items declared in a program are documented in its Texinfo docs. By *items* we mean here any external entities visible by the end user and designed for his use, such as command line options, configuration statements, external functions and the like. Any such item is supposed to have the following properties:

- class*        A descriptive name for the group of items this one belongs to. It is intended to be human readable and can consist of arbitrary number of characters. E.g. ‘`Command line option`’ or ‘`External function`’, or the like.  
The script prints class name at the beginning of its output.
- identifier*    A string of characters uniquely identifying this item among other items of the same class. E.g., a function name if the item is a function, or option name (without the leading dashes) if the item is a command line option.
- declaration*    A statement in the source code where this item is declared. E.g. a function declaration or option definition. The declaration must contain item identifier so that it can be extracted from it using a regular expression.
- description*    A statement in the Texinfo document which contains a description of this item. It must meet the same requirement as the declaration, i.e. it should be possible to define a regular expression for extracting the item identifier from the description.

The script is invoked as follows:

```
check-docs.sh code-sed doc-sed sources -- makeinfo-args
```

Notice mandatory double-dash.

Its arguments are:

- class*        Item class. Make sure you quote it, if it contains characters special for the shell.
- code-sed*      A regular expression to extract item identifiers from declarations. It must use a parenthesized group to mark the identifier (see [Section “REPLACEMENT” in Sed Manual](#)), and the ‘`p`’ flag or command to output it (see [Section “Print out the pattern space” in Sed Manual](#)).  
For example, the following expression extracts function name from its declaration in C code:
- ```
s/^\([\[_a-zA-Z] [\_a-zA-Z0-9]*\) *(.*\/\1/p
```
- It assumes that identifiers start in column 0 and is simplified a bit.
- doc-sed*      A regular expression to extract item identifiers from descriptions. It is in all respects similar to *code-sed*.  
For example, the expression below extracts identifiers from ‘`@defmac`’ statements:

```
’/@defmac/s/@defmac *\[ \_a-zA-Z] [\_a-zA-Z0-9]*\) .*\/\1/p’
```

*sources* A list of source files to apply *code-sed* to.

*makeinfo-args*

A command which produces preprocessed Texinfo source at its output. Normally, it is `makeinfo -E -` with some additional arguments. The *doc-sed* expression is applied to its output.

The `check-docs.sh` script collects identifiers produced by the two sed expressions and computes two sets of differences: identifiers missing from the docs, but present in the sources and the ones missing from the sources but present in the docs. If both lists are empty, the script exits silently with status `'0'`. Otherwise, it reports the differences and exits with status `'1'`.

The script is normally used from Makefile rules in the `Makefile.am` file located in the documentation directory. You may have any number of such rules, each one for a particular class of items. It is recommended to declare a Makeinfo variable pointing to the script, such as:

```
CHECK_DOCS = $(top_srcdir)/@IMPRIMATUR_MODULE_DIR@/check-docs.sh
```

The *makeinfo-args* parameter is usually as follows:

```
$(MAKEINFO) $(AM_MAKEINFOFLAGS) $(MAKEINFOFLAGS) -E - $(info_TEXINFOS)
```

To illustrate this, here is a Makefile rule used by Imprimatur itself to verify that all rules from `imprimatur.mk` are documented:

```
check-rules:
    @$(CHECK_DOCS) "Makefile rules" \
    '/^imprimatur-.*:/s/:.*//p' \
    's/@deffn {Makefile Rule} */p' \
    imprimatur.mk -- \
    $(MAKEINFO) $(AM_MAKEINFOFLAGS) $(MAKEINFOFLAGS) \
    -I $(srcdir) -E - $(info_TEXINFOS)
```

To illustrate its output, suppose that the rule `imprimatur-fixmes` is not documented. Then, you will get:

```
$ make -k check-rules
Not documented Makefile rules:
imprimatur-fixmes
```

Similarly, if the docs document rule `imprimatur-fixme` (presumably a typo: the `'s'` is missing), you'll get:

```
$ make -k check-rules
Non-existing Makefile rules:
imprimatur-fixme
```

In both cases, the exit code will indicate an error.

## 9 Reporting Bugs

Send bug reports and suggestions to [gray+imprimatur@gnu.org.ua](mailto:gray+imprimatur@gnu.org.ua).

If you think you've found a bug, please be sure to include maximum information available to reliably reproduce, or at least to analyze it. Please, include the version of the package you are using and a description of the bug: what you did, what result did you get and how it differs from what you expected. If possible, include a short test case with the steps needed to reproduce the bug.

Any errors, typos or omissions found in this manual also qualify as bugs. Please report them, if you happen to find any.



# 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 non-commercially. 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.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.



# Concept Index

This is a general index of all issues discussed in this manual:

## A

AC\_INIT ..... 11  
 ACLOCAL\_AMFLAGS ..... 7  
 AM\_MAKEINFOFLAGS ..... 7, 12  
 annotation ..... 3

## C

check-docs.sh ..... 8  
 check-docs.sh ..... 19  
 configure.ac ..... 7

## D

default rendition ..... 11  
 dist-info ..... 12  
 DISTRIB ..... 5, 11  
 doc/Makefile.am ..... 7

## F

FDL, GNU Free Documentation License ..... 23  
 FIXME ..... 13  
 FIXME-pxref ..... 14  
 FIXME-ref ..... 14  
 FIXME-xref ..... 14  
 François Pinard ..... 1  
 frenchspacing ..... 11

## G

git ..... 7

## I

imprimatur ..... 3  
 imprimatur-basic-checks ..... 17  
 imprimatur-check-sentence-spacing ..... 16  
 imprimatur-empty-nodes ..... 16  
 imprimatur-final ..... 17  
 imprimatur-fix-sentence-spacing ..... 17  
 imprimatur-fixmes ..... 15  
 imprimatur-format ..... 15  
 imprimatur-ignore ..... 16  
 imprimatur-master-menu ..... 17  
 imprimatur-refs ..... 15  
 imprimatur-unrevised ..... 16  
 imprimatur-untabify ..... 17

imprimatur-writemes ..... 16  
 imprimatur.mk ..... 8, 15  
 IMPRIMATUR\_INIT ..... 11  
 imprimatur\_INPUT ..... 8, 15  
 IMPRIMATUR\_MAKEINFOFLAGS ..... 7, 12  
 IMPRIMATUR\_MODULE\_DIR ..... 12  
 info\_TEXINFOS ..... 15, 17

## M

makedoc ..... 11  
 Makedoc ..... 11  
 Makefile ..... 15  
 Makefile.am, top level ..... 7

## P

PROOF ..... 5, 11  
 PUBLISH ..... 5, 11

## R

RENDITION ..... 5, 12  
 rendition, default ..... 11  
 rendition, defined ..... 5  
 rendition.texi ..... 8, 13  
 rules, Makefile ..... 15

## S

SUBDIRS ..... 7  
 submodule ..... 7

## T

Texinfo ..... 3

## U

UNREVISED ..... 13

## V

VERSION ..... 11

## W

WRITEME ..... 13

