

[Home page](#)

[What's new](#)

**Tutorial**

[License](#)

[Links](#)

[Plans](#)

[Mailing lists](#)

[Download](#)

# Tutorial

---

## GLOBAL source code tag system

Edition 4.0.1, for GLOBAL version 4.0.1

24 November 2000

by Shigio Yamaguchi

---

- [1. Overview of the tools](#)
- [2. Shell using GLOBAL](#)
  - [2.1 Preparation.](#)
  - [2.2 Basic usage.](#)
  - [2.3 Applied usage.](#)
- [3. Extended nvi using global](#)
  - [3.1 Features.](#)
  - [3.2 Preparation.](#)
  - [3.3 Basic usage.](#)
  - [3.4 Applied usage.](#)
- [4. Extended emacs using global](#)
  - [4.1 Features.](#)
  - [4.2 Preparation.](#)
  - [4.3 Basic usage.](#)
  - [4.4 Applied usage.](#)
- [5. Hypertext generator](#)
  - [5.1 Features.](#)
  - [5.2 Preparation.](#)
  - [5.3 Usage.](#)
- [6. Elvis using global](#)

- [6.1 Features.](#)
- [6.2 Preparation.](#)
- [6.3 Basic usage.](#)
- [6.4 Applied usage.](#)
- [7. Other topics](#)
  - [7.1 How to plug in a parser.](#)
  - [7.2 Compact format.](#)
  - [7.3 Incremental updating.](#)
- [GNU Free Documentation License](#)
  - [Preamble](#)
  - [APPLICABILITY AND DEFINITIONS](#)
  - [VERBATIM COPYING](#)
  - [COPYING IN QUANTITY](#)
  - [MODIFICATIONS](#)
  - [COMBINING DOCUMENTS](#)
  - [COLLECTIONS OF DOCUMENTS](#)
  - [AGGREGATION WITH INDEPENDENT WORKS](#)
  - [TRANSLATION](#)
  - [TERMINATION](#)
  - [FUTURE REVISIONS OF THIS LICENSE](#)
  - [ADDENDUM: How to use this License for your documents](#)
- [Option Index](#)

---

This document was generated on November, 24 2000 using [texi2html](#) 1.57.

---

[\[Top of page\]](#)

Copyright 2000,2001 Tama Communications Corporation. All rights reserved.  
Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1



# GLOBAL source code tag system

Edition 4.0, for GLOBAL version 4.0

5 September 2000

by Shigio Yamaguchi

---

Copyright (C) 2000 Tama Communications Corporation. This is the first edition of the GNU GLOBAL documentation,

and is consistent with 4.0.

Published by Tama Communications Corporation

Tama-shi, Tokyo, Japan.

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; with no Invariant Sections, with no Front-Cover Texts, and with no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

## 1. Overview of the tools

GNU GLOBAL is a source code tag system that works the same way across diverse environments. You can locate a specified function in the source files and move there easily. It is useful for hacking a large project containing many subdirectories, many `#ifdef` and many `main()` functions.

GNU GLOBAL has following features:

- support C, C++, Yacc and Java.
- work the same way across diverse environments. Currently, support followings:
  - Shell command line
  - Nvi editor (vi clone)
  - Elvis editor (vi clone)
  - Emacs editor
  - Web browser
- find the locations of a specified object quickly.
- locate not only object definitions but also object references.
- allows duplicate objects.
- locate also path which includes specified pattern.
- can treat a source tree containing subdirectories as a logical scope. You can get the relative path of objects from anywhere within that scope.
- search not only in a source tree but also in library paths.
- understand POSIX 1003.2 regular expressions.

- support external search engine (grep and idutils).
- generate hypertext of source code.
- tag files are independent of machine architecture.
- plugged-in parser is available to treat new language.
- compact format is available to save disk space.
- support incremental updating of tag files.
- support customizing with global.conf.
- generate completion list for completing input method.

## 2. Shell using GLOBAL

You can use tag facilities from shell command line. It is a big merit of GLOBAL compared with any other tag system.

### 2.1 Preparation.

First of all, you must execute `gtags(1)` at the root of source tree. For example, if you want to browse vi's source code:

```
% cd /usr/src/usr.bin/vi
% gtags
```

Gtags traverse subdirectories and makes four databases at the root of the source tree.

```
% ls G*
GPATH  GRTAGS  GSYMS  GTAGS
```

- ``GTAGS'` database of function definitions
- ``GRTAGS'` database of function references
- ``GSYMS'` database of other symbols
- ``GPATH'` database of path names

### 2.2 Basic usage.

Consider the following source tree:

```
ROOT/          <- the root of source tree (GTAGS,GRTAGS,...)
|
```

```

|- DIR1/
|
|  |- fileA.c      .....  +-----+
|  |               |main(){
|  |               |    func1();
|  |               |    func2();
|  |               |}
|  |               +-----+
|
|  |- fileB.c      .....  +-----+
|  |               |func1(){ ... }|
|  |               +-----+
|- DIR2/
|
|  |- fileC.c      .....  +-----+
|  |               |#ifdef X
|  |               |func2(){ i++; }
|  |               |#else
|  |               |func2(){ i--; }
|  |               |#endif
|  |               |func3(){
|  |               |    func1();
|  |               |}
|  |               +-----+

```

- You can get the relative path of your object from anywhere in the source tree. You need not specify where the tag file is. Global will locate the tag file by itself.

```

% cd ROOT
% global func1
DIR1/fileB.c           # func1() is defined in fileB.c
% cd DIR1
% global func1
fileB.c                # relative path from DIR1
% cd ../DIR2
% global func1
../DIR1/fileB.c       # relative path from DIR2

```

- The ``-r'` option locates function references.

```

% global -r func2
../DIR1/fileA.c       # func2() is referred from fileA.c

```

- You can use POSIX regular expressions.

```

% cd ROOT
% global 'func[1-3]'
DIR1/fileB.c           # func1, func2 and func3 are matched
DIR2/fileC.c

```

- The ``-x'` option shows the details. It is similar to the ``-x'` option in `ctags(1)`.

```

% global func2
DIR2/fileC.c
% global -x func2
func2                2 DIR2/fileC.c      func2(){ i++; }
func2                4 DIR2/fileC.c      func2(){ i--; }

```

- The ``-a'` option produces the absolute path name.

```

% global -a func1
/home/user/ROOT/DIR1/fileB.c

```

- The `-s` command locates any symbols other than functions.

```

% global -xs X
X                    1 DIR2/fileC.c #ifdef X

```

- The `-g` command locates any patterns including symbols. It is similar to `grep(1)`.

```

% global -xg '#ifdef'
#ifdef              1 DIR2/fileC.c #ifdef X

```

- The `-P` command enables you to locate path which includes specified string.

```

% global -P fileB
DIR1/fileB.c
% global -P '1/'
DIR1/fileA.c
DIR1/fileB.c
% global -P '\.c$'
DIR1/fileA.c
DIR1/fileB.c
DIR2/fileC.c

```

- The `-f` command enables you see the list of functions of specified file.

```
% global -f DIR2/fileC.c
func2          2 DIR2/fileC.c   func2(){ i++; }
func2          4 DIR2/fileC.c   func2(){ i--; }
func3          6 DIR2/fileC.c   func3(){
```

## 2.3 Applied usage.

You can make multiple tag files. For example, you can execute `gtags` at `ROOT/`, `version1.0/` and `version2.0/`.

```
ROOT/          <- the root of source tree      (GTAGS,...)
|
|- version1.0/ <- the root of version1.0      (GTAGS,...)
|
|  |- file.c   ..... +-----+
|                  |func1(){ i++; }|
|                  +-----+
|
|- version2.0/ <- the root of version2.0      (GTAGS,...)
|
|  |- file.c   ..... +-----+
|                  |func1(){ i--; }|
|                  +-----+
```

- When you are in the `version1.0` directory, `global` will only locate functions that are in `version1.0`.

```
% cd ROOT/version1.0
% global -x func1
func1          1 file.c          func1(){ i++; }
```

- When you are in the `version2.0`, `global` will only locate functions that are in `version2.0`.

```
% cd ROOT/version2.0
% global -x func1
func1          1 file.c          func1(){ i--; }
```

- If you are at `ROOT/`, or you set the `GTAGSROOT` environment variable to `ROOT`, then `global` will locate functions in both directories.

```

% cd ROOT
% global -x func1
func1          1 version1.0/file.c  func1(){ i++; }
func1          1 version2.0/file.c  func1(){ i--; }

```

There is another usage of GTAGSROOT.

- If your source files are on a read-only device, such as CDROM, then you cannot make databases at the root of the source tree. In such cases, please do the following:

```

% mkdir /var/dbpath
% cd /cdrom/src          # the root of source tree
% gtags /var/dbpath     # make tag file in /var/dbpath
% setenv GTAGSROOT `pwd`
% setenv GTAGSDBPATH /var/dbpath
% global func

```

- If you want all references to a function that is not defined in the source tree to be treated as calls to library functions or system calls, then you can specify library directories with the GTAGSLIBPATH environment variable. You should execute gtags at each directory of the path. If `GTAGS` is not found in a directory, global ignores that directory.

```

% pwd
/develop/src/mh          # this is the source tree
% gtags
% ls G*TAGS
GRTAGS  GTAGS
% global mhl
uip/mhlsbr.c            # mhl() is found
% global strlen         # strlen() is not found
% (cd /usr/src/lib; gtags) # library source
% (cd /usr/src/sys; gtags) # kernel source
% setenv GTAGSLIBPATH /usr/src/lib:/usr/src/sys
% global strlen
../../../../usr/src/lib/libc/string/strlen.c # found in library
% global access
../../../../usr/src/sys/kern/vfs_syscalls.c # found in kernel

```

Of course, the user program does not call kernel functions directly, but at least it is useful.

- If you forget a function name, you can use the -c (complete) command.

```

% global -c kmem          # maybe k..k.. kmem..
kmem_alloc

```

```

kmem_alloc_pageable
kmem_alloc_wait
kmem_free
kmem_free_wakeup
kmem_init
kmem_malloc
kmem_suballoc          # This is what I need!
% global kmem_suballoc
../vm/vm_kern.c

```

- You can use the `-c` command with `tosh`'s complete command.

```

% set funcs=(`global -c`)
% complete global 'n/*/${funcs}/'
% global kmem_TAB
kmem_alloc          kmem_free_wakeup
kmem_alloc_pageable kmem_init
kmem_alloc_wait     kmem_malloc
kmem_free           kmem_suballoc
% global kmem_sTAB
% global kmem_suballoc
../vm/vm_kern.c

```

- You can edit all files that include a specified function by typing one command, for example:

```

% vi `global func1`      # edit fileB.c

```

- If you want to browse many files in order, do the following:

```

% global -xr fork | awk '{printf "view +%s %s\n", $2, $3}'
view +650 ../dev/aic7xxx/aic7xxx_asm.c
view +250 ibcs2/ibcs2_misc.c
view +401 linux/linux_misc.c
view +310 ../kern/init_main.c
view +318 ../kern/init_main.c
view +336 ../kern/init_main.c
view +351 ../kern/init_main.c
% !! | sh          # from now on, go to next tag with 'ZZ'.

```

# 3. Extended nvi using global

You can use GLOBAL as a tag system of Nvi editor instead of ctags.

## 3.1 Features.

- You can use most of GLOBAL's facilities from nvi editor.
- Extended nvi is completely backward-compatible with the original nvi. You can use GLOBAL's facilities only in 'gtags mode'.

## 3.2 Preparation.

First, do the preparation of global. (Please see See section [2.1 Preparation.](#)).

Second, to use global from vi, you need to get into *gtagmode*. There are several ways to do this:

1. Start vi with ``-G'` option

```
% vi -G file.c
```

2. Start vi and execute `set gtagmode`.

```
% vi file.c
~
~
~
:set gtagmode
```

3. Write the above set command to the ``.exrc'` or ``.nexrc'` and start vi

```
$HOME/.exrc
+-----
|set gtagmode
```

You must start vi under the source tree described in See section [2.1 Preparation.](#)

## 3.3 Basic usage.

- To go to `func1`, you can say

```
:tag func1
```

It seems the same as original vi, but extended vi use `GTAGS' instead of `tags'.

- To go to the referenced point of func1, add the option -r

```
:tag -r func1
```

Extended nvi use `GRTAGS'.

- If a number of functions are located, extended vi goes to the first tag. You can go to next tag by typing :tagnext and back by typing :tagprev.

```
Suggested .nexrc:  
set gtagsmode  
map ^N :tagnext^M  
map ^P :tagprev^M
```

- CTL-] command is available. In gtags mode, if current token is not a function then it is equivalent to :tag -s current token. Otherwise, if you are in the first column of a line, it is equivalent to :tag -r current token else it is equivalent to :tag current token.
- You can use the -s option. It locates any symbols other than functions.

```
:tag -s pat
```

Extended nvi use `GSYMS'.

- The -g, -f and -P option are also available. It works like command line. Extended nvi use no index file.

```
:tag -g pat
```

- Other tag commands are also available:

```
CTL-T
```

Return to the most recent tag context.

```
:tagpop
```

Go to the specified tag in the tags stack.

```
:tagtop
```

Go to the top tag in the tags stack.

```
:display tags
```

Display the tags stack.

## 3.4 Applied usage.

- In large projects that include many main() function like MH, you can start vi like this:

```
% vi -G -t main
```

You can browse all commands sequentially.

- When you want to check functions the name of which start with "set" or "get", use:

```
% vi -G -t '^[sg]et'
```

Of course, the following command is also available:

```
:tag ^[sg]et
```

- If your source files are on a read only device like a CD-ROM, please do the following:

```
% mkdir /var/dbpath           # directory for the tag file
% cd /cdrom/src               # the root of the source tree
% gtags /var/dbpath           # make tag files in /var/dbpath
% setenv GTAGSROOT `pwd`
% setenv GTAGSDBPATH /var/dbpath
% vi -G -t main
```

- If you want all references to function that are not defined in the source tree to be treated as references to library functions or as system calls, do the following:

```
% cd /usr/src/lib
% gtags                       # probably as a root
% cd /usr/src/sys
% gtags
% setenv GTAGSLIBPATH /usr/src/lib:/usr/src/sys
```

- If you examine vi's source,

```
% cd /usr/src/usr.bin/vi
% gtags
% vi -G -t main
```

You can start from vi and browse the whole unix world as if you were using hypertext.

## 4. Extended emacs using global

You can use GLOBAL as a tag system of Emacs editor instead of etags.

### 4.1 Features.

Addition to the extended vi,

- More intelligent recognition of the current token and its type.
- Tag completion is available for input tag name.
- Mouse events are supported.

### 4.2 Preparation.

First, do the preparation of global. (Please see See section [2.1 Preparation.](#)).

Second, to use global from emacs, you need to load the `gtags.el' and execute gtags-mode function in it.

1. Write the autoload function to the `~/.emacs', start emacs and execute the gtags-mode function.

```
$HOME/.emacs
+-----
| (autoload 'gtags-mode "gtags" "" t)
|
% emacs
|
| J_:-----Mule: *scratch*          (Lisp Interaction)--L16--All-----
| M-x gtags-mode[RET]
+-----
```

If you want to get into gtags-mode on c-mode then you can append followings into the `~/.emacs'.

```
(setq c-mode-hook
      '(lambda ()
          (gtags-mode 1)
        ))
```

2. Specify the root directory of the source tree using `gtags-visit-rootdir`. If you have tag files in

/usr/src/sys then please do like this:

```
Visit root directory: /usr/src/sys
```

## 4.3 Basic usage.

- To go to `func1`, invoke `gtags-find-tag` and you can see a prompt in mini-buffer. Then input the tag name.

```
Find tag: func1                                <- 'Find tag: ' is a prompt
```

- To go to a point that references `func1`, invoke `gtags-find-rtag`.

```
Find tag (reference): func1
```

- Tag name completion is available. You need to execute `gtags-make-complete-list` command before it.

```
Find tag: fuTAB
```

```
Find tag: func1                                <- 'nc1' is appended by emacs
```

- If a number of functions are located, emacs goes into *GTAGS SELECT MODE* like this:

```
+-----+
|main          347 i386/isa/ultra14f.c main()
|main          128 kern/init_main.c  main(framep)
|main          104 netiso/clnp_debug.c main()
|main          164 netiso/xebec/main.c main(argc, argv)
|
|J_:--%*-Mule: *scratch*          (Gtags Select)--L1--All----
|[GTAGS SELECT MODE] 4 lines
+-----+
```

You can select a tag line by using any emacs command and pressing RET, and you can go to the tag's

point. When you want to go to the next or the previous tag, you can return to 'GTAGS SELECT MODE' with `gtags-pop-stack` and `reselect`.

- `gtags-find-tag-from-here` command is available. If current token is a definition, it is equivalent to Find tag (reference): current tokenRET, otherwise it is equivalent to Find tag: current tokenRET. (GLOBAL decides this intelligently, but may sometimes misunderstand.)
- To go to any symbols other than function, try `gtags-find-symbol`.

```
Find symbol: lbolt          <- 'Find symbol:' is a prompt
```

- To go to any strings other than symbols, try `gtags-find-with-grep`.

```
Find pattern: Copyright
```

## 4.4 Applied usage.

- You can use POSIX regular expressions.

```
Find tag: ^put_           <- locate tags start with 'put_'
```

- If your source files are on a read-only device like a CDROM, please do the following:

```
% mkdir /var/dbpath          # directory for the tag file
% cd /cdrom/src              # the root of the source tree
% gtags /var/dbpath          # make tag files in /var/dbpath
% setenv GTAGSROOT `pwd`
% setenv GTAGSDBPATH /var/dbpath
% emacs -f gtags-mode
```

- If you want all references to functions that are not defined in the source tree to be treated as references to library functions or as system calls, do the following:

```
% cd /usr/src/lib
% gtags                      <- probably as a root
% cd /usr/src/sys
% gtags
% setenv GTAGSLIBPATH /usr/src/lib:/usr/src/sys
% emacs -f gtags-mode
```

- Mouse command is available. If you use X version emacs, try the following Move the mouse cursor to a

function name and click the middle button. You will then go to the function's definition, or to its references, depending on the context. In 'GTAGS SELECT MODE', move the mouse cursor to a line and click the center button. To return to the previous position, click the right button.

## 5. Hypertext generator

You can use GLOBAL's facilities from WWW browser.

### 5.1 Features.

- Htags makes hypertext from C, C++, Yacc and Java source files.
- Once the hypertext is generated, you need nothing other than a WWW browser.
- You can move the hypertext to anywhere. It is independent of the source code.
- You can use all of your browser's functions, such as search, history, bookmark, save, frames, windows.

### 5.2 Preparation.

At first, you must ensure that you have a lot of disk space. Hypertext needs a great amount of disk space. For example, the source code of FreeBSD kernel needs:

```
source code(/usr/src/sys)      14.0MB
GTAGS                          1.5MB
GRTAGS                         8.0MB
GSYMS                          12.0MB
HTML/                          55MB(!!!)
-----
total                          77MB
```

Please do the following:

```
(at your source directory)
% gtags          # make the tag database(GTAGS,GRTAGS,GSYMS)
% htags         # make the hypertext(HTML/)
```

Then you will find an `HTML' subdirectory in the current directory.



```
:tag func1
```

It seems the same as original elvis, but elvis execute `global -t func1` internally and read it instead of tags file.

- To go to the referenced point of `func1`, add ``-r'` option.

```
:tag -r func1
```

Elvis executes command like `global -t -r func1` internally.

- To go to any symbols other than function, try this.

```
:tag -s lbolt
```

- To go to any strings other than symbols, try this.

```
:tag -g Copyright
```

- When using `-r`, `-s` or `-g`, you had better to use `browse` command.

```
:browse -r fork
```

It brings a following selection list. You can select tag and go to the point.

```
Browse -r fork (2 matches)
+-----+-----+-----+
| TAG NAME          | SOURCE FILE      | SOURCE LINE      |
+-----+-----+-----+
| fork              | ux/linux_misc.c | (line 565)       |
| fork              | ern/init_main.c | (line 191)       |
+-----+-----+-----+
```

- To get list of functions in a file, use `-f` command.

```
:browse -f main.c          <- locate functions in main.c
```

- Other tag commands are also available:

```
CTL- ]
```

go to the definition of current token.

CTL-T

return to the most recent tag context.

:tag

without argument, go to the next tag.

:pop

return to the most recent tag context.

:stack

display the tags stack.

:stag

creates a new window and moves its cursor to the tag's definition point.

:sbrowse

same with 'browse' but show in a new window.

## 6.4 Applied usage.

- You can use POSIX regular expressions.

```
:tag ^put_                <- locate functions start with 'put_'  
:browse -g 'fseek(.*L_SET)' <- locate fseek() using L_SET argument
```

- You can browse functions list of many files.

```
:browse -f *.c            <- locate functions in *.c
```

- You can browse project files whose path includes specified pattern.

```
:browse -P /vm/           <- under vm/ directory  
:browse -P \.h$           <- all include files  
:browse -P init           <- path including 'init'
```

- You can use mouse to select tag.

```
+-----+-----+-----+  
| TAG NAME          | SOURCE FILE      | SOURCE LINE  
+-----+-----+-----+  
| fork              | ux/linux_misc.c | (line 565)  
| fork              | ern/init_main.c | (line 191)
```



- [1] tag name
- [2] line number the tag appeared
- [3] path name. It must be equal to argument path name.
- [4] line image

Otherwise, you can make a suitable wrapper for the pug-in parser.

## 7.2 Compact format.

You can save disk space with the compact format. The compact format needs source files.

- To specify the use of the compact format on the command line, add the ``-c'` option:

```
% gtags -c
```

- To specify the use of the compact format in the configuration file:

```
+-----
| ...
| default:\
|         format=compact:...

```

- If you will publish hypertext generated by htags then use the ``-c'` option of htags too:

```
% htags -c
```

With the ``-c'` option, htags makes gzipped hypertext. You need to set up an HTTP server so that gzipped files can be read (see ``HTML/.htaccess.skel'`).

Example:

	Standard	Compact	Compressed rate
GTAGS	1744896 bytes	720896 bytes	-59%
GRTAGS	10133504 bytes	1409024 bytes	-86%
GSYMS	11911168 bytes	9306112 bytes	-22%
	Standard	Compact	Compressed rate
HTML/	56618 kbytes	15219 kbytes	-73%

## 7.3 Incremental updating.

Modifying some source files, you need not remake whole tag files. Instead, you can use incremental updating facility (`-i` option).

```
% gtags
% cd kern
% vi tty.c                # modify tty.c
...
:wq
% global -vi              # -v means verbose
[Sun Dec  6 16:27:47 JST 1998] Gtags started
Tag found in '/usr/src/sys'.
Incremental update.
Updating tags of 'kern/tty.c' ...GTAGS..GRTAGS..GSYMS.. Done.
Global databases have been modified.
[Sun Dec  6 16:28:30 JST 1998] Done.
% global -vi              # try again
[Sun Dec  6 16:28:48 JST 1998] Gtags started
Tag found in '/usr/src/sys'.
Incremental update.
Global databases are up to date.      # do nothing
[Sun Dec  6 16:28:52 JST 1998] Done.
```

# GNU Free Documentation License

Version 1.1, March 2000

Copyright (C) 2000 Free Software Foundation, Inc.

59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

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

## Preamble

The purpose of this License is to make a manual, textbook, or other written 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.

## APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you".

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. (For example, 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.

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 "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, whose contents can be viewed and edited directly and 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 has been designed to thwart or discourage subsequent modification by readers is not Transparent. 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 designed for human modification. Opaque formats include PostScript, PDF, 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 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.

# 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.

## COPYING IN QUANTITY

If you publish printed copies 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 publicly-accessible computer-network location containing a complete Transparent copy of the Document, free of added material, which the general network-using public has access to download anonymously at no charge using public-standard network protocols. 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.

## 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:

1. 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.
2. 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 less than five).

3. State on the Title page the name of the publisher of the Modified Version, as the publisher.
4. Preserve all the copyright notices of the Document.
5. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
6. 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.
7. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
8. Include an unaltered copy of this License.
9. Preserve the section entitled "History", and 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.
10. 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.
11. In any section entitled "Acknowledgements" or "Dedications", preserve the section's title, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
12. 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.
13. Delete any section entitled "Endorsements". Such a section may not be included in the Modified Version.
14. Do not retitle any existing section as "Endorsements" or to conflict in title with any Invariant Section.

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.

# 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.

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."

# 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.

# 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, does not as a whole count as a Modified Version of the Document, provided no compilation copyright is claimed for the compilation. Such a compilation is called an "aggregate", and this License does not apply to the other self-contained works thus compiled with the Document, on account of their being thus compiled, if they 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 quarter of the entire aggregate, the Document's Cover Texts may be placed on covers that surround only the Document within the aggregate. Otherwise they must appear on covers around the whole aggregate.

# 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 provided that you also include the original English version of this License. In case of a disagreement between the translation and the original English version of this License, the original English version will prevail.

# 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.

# 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.

# 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 1.1
or any later version published by the Free Software Foundation;
with the Invariant Sections being LIST THEIR TITLES, with the
Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.
A copy of the license is included in the section entitled "GNU
Free Documentation License".
```

If you have no Invariant Sections, write "with no Invariant Sections" instead of saying which ones are

invariant. If you have no Front-Cover Texts, write "no Front-Cover Texts" instead of "Front-Cover Texts being LIST"; likewise for Back-Cover Texts.

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.

# Option Index

Jump to:

---

This document was generated on September, 5 2000 using [texi2html](#) 1.57.