

vc-dwim

Version-control-agnostic commit, diff, and change logs.

Jim Meyering
Ralf Wildenhues

This manual is for `vc-dwim` (version 1.10, 26 January 2020), a version-control-agnostic ChangeLog diff and commit tool, and `vc-chlog`, a ChangeLog writing helper tool.

Copyright © 2006–2020 Free Software Foundation, Inc.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover texts and no Back-Cover Texts.

Table of Contents

| | | |
|-------------------|--|----------|
| 1 | Introduction | 1 |
| 2 | Version-control wrapper functionality | 2 |
| 2.1 | How can you use vc-dwim? | 2 |
| 2.2 | Initializing vc-dwim | 3 |
| 2.3 | Invoking vc-dwim | 3 |
| 3 | What can vc-chlog do? | 5 |
| 3.1 | Getting up and going with vc-chlog | 5 |
| 3.2 | Invoking vc-chlog | 6 |
| 3.3 | How vc-chlog works | 7 |
| 3.4 | Reporting bugs | 7 |
| Appendix A | GNU Free Documentation License .. | 8 |

1 Introduction

vc-dwim is a version-control-agnostic **ChangeLog** diff and commit tool. vc-chlog is a **ChangeLog** writing helper tool.

Both tools are useful if you like to maintain a **ChangeLog** file describing the changes you make to version-controlled files. vc-dwim works with the following version control systems: bzr, CVS, git, mercurial, SVN, and darcs. It should be easy to add support for more.

vc-dwim can save you from making some small mistakes when using version control programs from the command line.

For example, if you have unsaved changes in an editor buffer and use vc-dwim to print diffs or to commit changes involving that file, it will detect the problem, tell you about it, and fail. This works as long as you use Emacs or Vim.

Another common error you can avoid with this tool is the one where you create a new file, add its name to **Makefiles**, etc., mention the addition in a **ChangeLog**, but forget to e.g., **git add** (or **hg add**, etc.) the file to the version control system. vc-dwim detects this discrepancy and fails with a diagnostic explaining the probable situation. You might also have simply mistyped the file name in the **ChangeLog**.

Also, vc-dwim makes it a little easier/safer to commit a strict subset of the modified files in a working directory. But no one ever does **that**.

2 Version-control wrapper functionality

2.1 How can you use vc-dwim?

Print diffs of arbitrary files:

Use an alias like this to show all or specified diffs:

```
alias cv='vc-dwim --diff'
```

Use that when you want to see diffs of a specified file, regardless of whether you have written new `ChangeLog` entries for it. It works the same for `bzr`, `cvs`, `git`, `hg`, `svn`, and `darcs` repositories, as long as all you want are the difference between your local copy and the checked out version.

Print diffs of files with new `ChangeLog` entries:

Let's say you have made local changes to a file, and you've also added at least one corresponding entry in a `ChangeLog` file. Then, you can use `vc-dwim ChangeLog` to print the diffs for which there are `ChangeLog` entries, warning about the potential problems mentioned above (editor temporaries that can imply there are unsaved changes, and files listed in `ChangeLog`, but not `cvs` added). If your changes affect files covered by more than one `ChangeLog`, you might use `vc-dwim ChangeLog lib/ChangeLog`, or more concisely, `vc-dwim {,lib/}ChangeLog`.

Commit changes to files with new `ChangeLog` entries:

Use `vc-dwim --commit ChangeLog` or `vc-dwim --commit ChangeLog lib/ChangeLog src/ChangeLog` to commit the changes you would see without the `--commit` option.

Assuming you have completed a change and have documented everything in one or more `ChangeLog` file, run `vc-dwim --commit ChangeLog` to commit that `ChangeLog` file and the files "implied" by the new `ChangeLog` lines. The commit log message is derived from the added `ChangeLog` lines. With a single `ChangeLog` file, the log message is nearly identical to the list of added lines. One leading `TAB` is removed and any `date user-name <email>` lines are elided. When there are two or more `ChangeLog` files, the log message includes a line for each indicating the affected directory. For example:

```
[ChangeLog]
* some-file-in-top-level-dir: ...
[lib/ChangeLog]
* lib.c: ...
[m4/ChangeLog]
* foo.m4: ...
```

After committing a change, do not erase or edit the `ChangeLog` file. When writing further changes, just prepend to the top of `ChangeLog`, as usual. `vc-dwim` requires that the only changes to `ChangeLog` are additions at the beginning of the file. If you feel the need to edit past log entries, this needs to be done in the VC log, not by editing `ChangeLog`.

2.2 Initializing vc-dwim

vc-dwim operates based on a ChangeLog file; if you maintain a ChangeLog as part of your source tree, you don't need to do anything special to set up to use vc-dwim; just run it.

However, it is common practice nowadays for source trees not to have ChangeLog files, but instead to have them derived for distributions or on demand (`make ChangeLog`, with one common GNU infrastructure). So, how to use vc-dwim in this case? It still needs a ChangeLog.

The answer is to run `vc-dwim --initialize`. This sets up your own, private ChangeLog file in a git repository rooted at `VCDIR/c/`, with a symlink to it from the top level directory of the source tree in which you'd like to use vc-dwim.

Your own source tree need not use git; `VCDIR` in the above refers to the version-control system's control directory, such as `.svn`, `_darcs`, etc., as well as `.git`. The `VCDIR/c/.git` repository is always independent of your own source tree.

Then you can make ChangeLog entries as usual, and proceed with use of vc-dwim.

You can also set up your own repository of whatever type and make your own symlinked top-level ChangeLog into it; there's nothing magical about `--initialize` or the `VCDIR/c/` location.

2.3 Invoking vc-dwim

Synopsis:

```
vc-dwim [option...] [changelog_file...]
vc-dwim [option...] --commit changelog_file...
vc-dwim [option...] --diff [file...]
vc-dwim [option...] --print-vc-list
vc-dwim [option...] --initialize
```

By default, each command line argument should be a locally modified, version-controlled ChangeLog file. If there is no command line argument, `vc-dwim` tries to use the ChangeLog file in the current directory. In this default mode, `vc-dwim` works by first computing diffs of those files and parsing the diff output to determine which named files are being changed. Then, it diffs the affected files and prints the resulting output. One advantage of using this tool is that before printing any diffs, it warns you if it sees that a ChangeLog or an affected file has unsaved changes. It detects that by searching for an editor temporary file corresponding to each affected file. Another common error you can avoid with this tool is the one where you create a new file, add its name to `Makefiles`, etc., mention the addition in ChangeLog but forget to e.g., `git add` (or `hg add`, etc.) the file to the version control system. `vc-dwim` detects this discrepancy and fails with a diagnostic explaining the probable situation. You might also have simply mistyped the file name in the ChangeLog.

Once you are happy with your ChangeLog-derived diffs, you can commit those changes and the ChangeLog simply by rerunning the command with the `--commit` option.

But what if you'd like to use `vc-dwim` on a project that doesn't have or want a ChangeLog file? In that case, you can maintain your own ChangeLog file. This is what the `--initialize` option sets up for you, or you can perform equivalent steps by hand (see Section 2.2 [Initializing vc-dwim], page 3).

`vc-dwim` accepts the following options:

- `--help` Display help and exit.
- `--version`
 Output version information and exit.
- `--author="User Name <user@example.org>"`
 Specify the user name and email address of the author of this change set.
- `--commit` Perform the commit, too.
- `-n`
- `--dry-run`
 Print the commands that would be run instead of running them.
- `--diff` Determine which version control system manages the first *file*, then use that to print diffs of the named *files*. If no *file* is specified, print all diffs for the current hierarchy.
- `--print-vc-list`
 Print the list of recognized version control names, then exit.
- `--initialize`
 Prepare a source tree with no `ChangeLog` file for use with `vc-dwim` (see Section 2.2 [Initializing `vc-dwim`], page 3).
- `--verbose`
 Generate verbose output.
- `--debug` Generate debug output; implies `--verbose`.

This tool can be useful to you only if you use a version control system. It's most useful if you maintain a `ChangeLog` file and create a log entry per file per "commit" operation.

Relies on fairly strict adherence to recommended `ChangeLog` syntax. Detects editor temporaries created by Emacs and Vim. Patches to detect temporaries created by other editors are welcome.

3 What can vc-chlog do?

vc-chlog is about writing GNU Coding Standards-compliant `ChangeLog` entries easily, Section “`ChangeLog`” in *The GNU Coding Standards*.

3.1 Getting up and going with vc-chlog

Say you have made some changes to your code, ready to be committed. The only remaining part is to write one or more `ChangeLog` entries: for each `ChangeLog` governing a part of the package, collect the list of changed files, in each file list the changed functions, and mention all of those, in order to afterwards describe the changes:

```
1984-01-01  A.U. Thor  <email@address>

* file1.c (foo, bar, ...): ...
* file2.c (baz): ...
```

vc-chlog attempts to help with this step. It scans the diff (obtained by `vc-dwim --diff` or passed on standard input with `--stdin`) for the files that were touched and the set of lines that have been changed. It then uses the `ctags` program to try to find out in which functions those changes have occurred, and formats the file and functions names in a prototype `ChangeLog` entry form on standard output.

Finding identifiers for what has changed:

There is a crucial assumption behind this idea to work well in practice: `ctags` should be able to generate tags for the identifiers that changed. For example, it should list functions in C source files (but not function-local or other nested entities); it should list macros in M4 files (e.g., those that serve as input to Autoconf), or it should list `@nodes` in Texinfo files. The output of `vc-chlog` improves with that of `ctags`.

Exuberant Ctags is a powerful and extensible implementation of this command, and therefore preferred. For example, with the `.ctags` file in this package:

```
--langmap=Sh:+.in
```

it detects the shell script `vc-chlog` as such. With a `~/ctags` containing

```
--langdef=Texinfo
--langmap=Texinfo:.texi.txi.texinfo
--regex-texinfo=/~@node[ ]+([\^,]+)/\1/d,definition/
```

it detects Texinfo node names (`vc-chlog` uses some heuristics to deal with spaces in the identifier names when Exuberant Ctags is used).

Autoconf macros may be tagged by options such as

```
--langdef=m4
--langmap=m4:.m4.at.ac.as.m4sh
--regex-m4=/^(m4_def(ine|un(|_once))|A[CU]_DEFUN(|_ONCE)|AU_ALIAS)\
  \([\^*([a-zA-Z0-9_()]+)]+\)/\5/d,definition/
```

vc-chlog tries to find out about added as well as removed identifiers by examining both the new and the old version of a file. Here, it works hard not to change any file in your working directory, by using `ctags -x` and keeping all intermediate files in a temporary directory.

For some languages, `vc-chlog` attempts to guess where functions end, and thus not attribute changes past that end to the previous identifier.

Typically, `vc-chlog` is exact in the list of files that changed; false negatives in the list of identifiers stem from a `ctags` that failed to enumerate all identifiers properly, or changes before a function, and false positives typically stem from constructs like nested functions.

Use `vc-chlog` with multiple ChangeLog files in a project:

If a project uses multiple `ChangeLog` files, `vc-chlog` assumes that changes are to be recorded in the log file that is nearest up the directory tree. One possibility is to invoke `vc-chlog` always from the project root and put the output of

```
find . -name ChangeLog | sed 's,^\./,--changelog ,'
```

into the `.vc-chlogrc` file at the root.

3.2 Invoking `vc-chlog`

Synopsis:

```
vc-chlog [option...] [file...]
```

`vc-dwim` accepts the following options:

`--address address`

Use *address* as email address.

`--changelog cfile`

For all changes to files below the dirname of *cfile*, generate an entry for *cfile* with stripped names. This option may be passed multiple times.

`--stdin` Read unified diff from stdin instead of with `vc-dwim`.

`--tabsize cols`

Assume tab stops at each *cols* instead of 8.

`--user-name name`

Use *name* as user name.

`--width cols`

Wrap `ChangeLog` entry at *cols* instead of 72.

`--no-functions`

Do not try to parse the names of functions or other identifiers that changed.

`--dirty-workdir`

Assume that the working directory has other changes unrelated to the patch (usually given with `--stdin`). This will let the file be recreated from the version control index or cache, if any.

`--debug` Inhibit temp file cleanup and show much shell execution.

`--help` Display this help and exit.

`--version`

Output version information and exit.

The environment variables `TMPDIR` and `USER` are honored.

The files `~/vc-chlogrc` and `.vc-chlogrc` may contain default options, possibly with double-quoted arguments, and comment lines starting with `#`.

3.3 How vc-chlog works

At the heart of `vc-chlog`, there is a long pipe that processes a prototype `ChangeLog` entry from a patch and a source tree. This source tree is assumed to be patched, i.e., already has the patch applied.

From the patch text, `func_extract_changed_lines` generates a list of changes. For each changed file, it outputs a line containing the `filename`, a strip prefix, a couple of integers `d1` and `d2`, then a list of ranges, all separated with spaces. The strip prefix is the argument that needs to be passed to the `-p` option of the `patch` command. The integers `d1` and `d2` denote the line numbers of the patch text that deal with `filename`, while the ranges deal with line numbers within the file `filename`.

A range consists of four minus-separated integers `a-b-c-d`. They denote that, in the old version of `filename`, lines `a` through `b` inclusively have been changed and end up as lines `c` through `d` inclusively in the new file. Purely added or purely removed lines are denoted with a `0-0` pair.

This information is passed to `func_extract_identifiers`, the heart of the script. Unless identifier extraction has been disabled with `--no-functions`, for each changed file, it calls a helper function `func_taglist`, which uses `ctags -x` to obtain a list of identifiers in both the new, and the old version of the file. These lists are ordered by line number. Changes between two consecutive identifiers will later be assumed to change the earlier of the two.

While this works reasonably well for functions, it fails, for example, for global variable declarations or changes to comments outside of functions. So `func_taglist` applies some heuristics to guess the end of a function, e.g. a closing brace in the first column, and inserts an empty tag there.

With all tags in place, `func_extract_identifiers` selects those that match the change intervals, merges the lists from the old and the new file, and outputs them, avoiding duplicates and roughly maintaining the order of the identifiers in the new file.

Care is taken throughout that the tags may contain spaces, so that Texinfo `@nodes` may be listed as tags.

In a couple of final steps, the sets of files with identifiers are keyed to one or more `ChangeLog` files throughout the source tree using `func_sortby_chlog`, and a GCS-style `ChangeLog` entry stub is formatted with `func_pretty_changelog_entry`.

The fact that `vc-chlog` is written as a portable shell script, using `awk`, `sed`, and other POSIX tools freely, adds to the compactness of the script, added to the fun in writing it, necessitates slightly awkward data constructs, and probably impedes easy modification and debugging. This might be construed as a bug.

3.4 Reporting bugs

Email bug reports and all other discussion to `bug-vc-dwim@gnu.org`, a mailing list whose web page is <https://lists.gnu.org/mailman/listinfo/bug-vc-dwim>.

Appendix A GNU Free Documentation License

Version 1.3, 3 November 2008

Copyright © 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc.

<https://fsf.org/>

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.

The “publisher” means any person or entity that distributes copies of the Document to the public.

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 under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

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 <https://www.gnu.org/licenses/>.

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. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy’s public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

“Massive Multiauthor Collaboration Site” (or “MMC Site”) means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A “Massive Multiauthor Collaboration” (or “MMC”) contained in the site means any set of copyrightable works thus published on the MMC site.

“CC-BY-SA” means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

“Incorporate” means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is “eligible for relicensing” if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

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 1.3  
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.