Evil

Extensible vi layer for Emacs

Frank Fischer and Vegard Øye

This manual is for Evil (version 0.1 of 2011-07-30), an extensible vi layer for Emacs.

The Evil team thanks everyone at gmane.emacs.vim-emulation for their feedback and contributions.

Table of Contents

1 Overview

Evil is an extensible vi layer for Emacs. It emulates the main features of Vim,[1] turning Emacs into a modal editor. Like Emacs in general, Evil is extensible in Emacs Lisp.

1.1 Installation

Evil lives in a Git repository. To download Evil, do:

```
git clone git://gitorious.org/evil/evil.git
```

Move Evil to `~/.emacs.d/evil`. Then add the following lines to `~/.emacs`:

```
(add-to-list 'load-path "~/.emacs.d/evil")
(require 'evil)
(evil-mode 1)
```

Evil requires `undo-tree.el` to provide linear undo and undo branches. It is available from EmacsWiki.[2] (A copy of `undo-tree.el` is also included in the Git repository.)

1.2 Modes and states

The next time Emacs is started, it will come up in *Normal state*, denoted by `<N>` on the mode line. This is where the main vi bindings are defined. Note that you can always disable Normal state with *C-z*, which switches to an \Emacs state" (denoted by `<E>`) in which vi keys are completely disabled. Press *C-z* again to switch back to Normal state.

Evil uses the term *state* for what is called a \mode" in vi, since \mode" already has its own meaning in Emacs. Evil defines a number of states, such as Normal state (`<N>`), Insert state (`<I>`), Visual state (`<V>`), Replace state (`<R>`), Operator-Pending state (`<O>`), Motion state (`<M>`) and Emacs state (`<E>`). Each state has its own keymaps and customization variables.

Meanwhile, a *mode* in Emacs is a set of key bindings for editing a certain sort of text, like `emacs-lisp-mode` for Emacs Lisp. Modes may include custom bindings for Evil states.

---

[1] Vim is the most popular version of *vi*, a modal text editor with many implementations. Vim also adds some functions of its own, like Visual selection and text objects. For more information, see: http://www.vim.org/

[2] http://www.emacswiki.org/emacs/UndoTree

2 Settings

Evil's behavior can be adjusted by setting various variables. The current values may be inspected by doing *M-x customize-group RET evil RET*.

To change the value of a variable, add a `setq' form to `~/.emacs`, preferably before Evil is loaded:[1]

```
(setq evil-shift-width 8)
;; Load Evil
(require 'evil) ...
```

Note that if a variable is bufer-local, you must use `setq-default' instead of `setq' to change its global value.

`evil-auto-indent` [Variable]
: Whether the current line is indented when entering Insert state. If `t` (the default), then the line is indented. If `nil`, then the line is not indented. Bufer-local.

`evil-shift-width` [Variable]
: The number of columns a line is shifted by the commands > and <.

`evil-repeat-move-cursor` [Variable]
: If `t` (the default), then repeating a command with . may change the position of the cursor. If `nil`, then the original position is preserved.

`evil-find-skip-newlines` [Variable]
: If `t`, then *f*, *F*, *t* and *T* may skip over newlines to fnd a character. If `nil` (the default), then they are restricted to the current line.

`evil-move-cursor-back` [Variable]
: If `t` (the default), then the cursor moves backwards when exiting Insert state. If `nil`, then the cursor does not move.

`evil-want-fine-undo` [Variable]
: If `t`, then a change-based action like *cw* may be undone in several steps. If `nil` (the default), then it is undone in one step.

`evil-regexp-search` [Variable]
: If `t` (the default), then */* and *?* use regular expressions for searching. If `nil`, they use plain text.

`evil-search-wrap` [Variable]
: If `t` (the default), then */* and *?* wrap the search around the bufer. If `nil`, then they stop at bufer boundaries.

`evil-flash-delay` [Variable]
: The number of seconds to fash search matches when pressing *n* and *N*.

[1] Strictly speaking, the order only matters if the variable afects the way Evil is loaded. This is the case with some of the `evil-want-' variables.

`evil-want-C-i-jump` [Variable]
: If `t` (the default), then *C-i* jumps forwards in the jump list. If `nil`, then *C-i* inserts a tab.

`evil-want-C-u-scroll` [Variable]
: If `t`, then *C-u* scrolls the bufer. If `nil` (the default), then *C-u* begins a numeric prefx argument.

2.1 The cursor

A state may change the cursor's appearance. The cursor settings are stored in the variables below, which may contain a cursor type as per the `` `cursor-type' `` variable, a color string as passed to the `` `set-cursor-color' `` function, a zero-argument function for changing the cursor, or a list of the above. For example, the following changes the cursor in Replace state to a red box:

```
(setq evil-replace-state-cursor '("red" box))
```

If the state does not specify a cursor, `` `evil-default-cursor' `` is used.

`evil-default-cursor` [Variable]
: The default cursor.

`evil-normal-state-cursor` [Variable]
: The cursor for Normal state.

`evil-insert-state-cursor` [Variable]
: The cursor for Insert state.

`evil-visual-state-cursor` [Variable]
: The cursor for Visual state.

`evil-replace-state-cursor` [Variable]
: The cursor for Replace state.

`evil-operator-state-cursor` [Variable]
: The cursor for Operator-Pending state.

`evil-motion-state-cursor` [Variable]
: The cursor for Motion state.

`evil-emacs-state-cursor` [Variable]
: The cursor for Emacs state.

2.2 The initial state

By default, a new bufer comes up in Normal state. This can be changed with the function `` `evil-set-initial-state' ``.

`evil-set-initial-state` *mode state* [Function]
: Set the initial state for a bufer in which *mode* is active to *state*. *mode* should be a major mode such as `text-mode`, although minor modes work as well.

3 Keymaps

Evil's key bindings are stored in a number of keymaps. Each state has a *global keymap*, where the default key bindings for the state are stored. For example, the global keymap for Normal state is `evil-normal-state-map', and the key bindings in this map are seen in all bufers that are currently in Normal state.

Keymaps are modifed with the Emacs function `define-key':

```
(define-key evil-normal-state-map "w" 'foo)
```

This binds the key `w` to the command `foo' in Normal state. The fle `evil-maps.el` contains all the key bindings.

`evil-normal-state-map` [Variable]
The global keymap for Normal state.

`evil-insert-state-map` [Variable]
The global keymap for Insert state.

`evil-visual-state-map` [Variable]
The global keymap for Visual state.

`evil-replace-state-map` [Variable]
The global keymap for Replace state.

`evil-operator-state-map` [Variable]
The global keymap for Operator-Pending state.

`evil-motion-state-map` [Variable]
The global keymap for Motion state.

Each state also has a *buffer-local keymap*, which is specifc to the current bufer and has precedence over the global keymap. These maps may be changed from a mode hook.

`evil-normal-state-local-map` [Variable]
Bufer-local keymap for Normal state.

`evil-insert-state-local-map` [Variable]
Bufer-local keymap for Insert state.

`evil-visual-state-local-map` [Variable]
Bufer-local keymap for Visual state.

`evil-replace-state-local-map` [Variable]
Bufer-local keymap for Replace state.

`evil-operator-state-local-map` [Variable]
Bufer-local keymap for Operator-Pending state.

`evil-motion-state-local-map` [Variable]
Bufer-local keymap for Motion state.

3.1 ‘evil-define-key’

Finally, Evil provides the function `evil-define-key' for adding state bindings to a regular keymap.

`evil-define-key` *state keymap key def* [Function]

In *keymap*, create a binding from *key* to *def* in *state*. *state* is one of `normal', `insert', `visual', `replace', `operator' and `motion'. The other parameters are like those of `define-key'.

`evil-define-key' can be used to augment existing modes with state bindings, as well as create packages for custom bindings. For example, the following will create a minor mode `foo-mode` with Normal state bindings for the keys `w` and `e`:

```
(define-minor-mode foo-mode
  "Foo mode."
  :keymap (make-sparse-keymap))

(evil-define-key 'normal foo-mode-map "w" 'bar)
(evil-define-key 'normal foo-mode-map "e" 'baz)
```

This minor mode can then be enabled in any buffers where the custom bindings are desired:

```
(add-hook 'text-mode-hook 'foo-mode) ; enable alongside text-mode
```

If the minor mode is put into its own file `foo.el` with a `(provide 'foo)` statement, it becomes an Emacs package.

4 Hooks

A *hook* is a list of functions to execute. Hooks are modifed with the Emacs function `add-hook'. Evil provides entry and exit hooks for all of its states.

`evil-normal-state-entry-hook` [Variable]
Run when entering Normal state.

`evil-normal-state-exit-hook` [Variable]
Run when exiting Normal state.

`evil-insert-state-entry-hook` [Variable]
Run when entering Insert state.

`evil-insert-state-exit-hook` [Variable]
Run when exiting Insert state.

`evil-visual-state-entry-hook` [Variable]
Run when entering Visual state.

`evil-visual-state-exit-hook` [Variable]
Run when exiting Visual state.

`evil-replace-state-entry-hook` [Variable]
Run when entering Replace state.

`evil-replace-state-exit-hook` [Variable]
Run when exiting Replace state.

`evil-operator-state-entry-hook` [Variable]
Run when entering Operator-Pending state.

`evil-operator-state-exit-hook` [Variable]
Run when exiting Operator-Pending state.

`evil-motion-state-entry-hook` [Variable]
Run when entering Motion state.

`evil-motion-state-exit-hook` [Variable]
Run when exiting Motion state.

When these hooks are run, the variables `evil-next-state' and `evil-previous-state' hold information about the states being switched to and from.

`evil-next-state` [Variable]
The state being switched to.

`evil-previous-state` [Variable]
The state being switched from.

5 Macros

Evil is implemented in terms of reusable macros. Package writers can use these to define new commands.

5.1 Motions

A *motion* is a command which moves the cursor, such as `w` and `e`. Motions are defined with the macro `evil-define-motion'. Motions not defined in this way should be declared with `evil-declare-motion'.

`evil-declare-motion` *command* [Function]
Declare *command* to be a motion. This ensures that it works properly in Visual state.

`evil-define-motion` *motion* (*count args...*) *doc keyword-args... body...* [Macro]
Define a movement command *motion*. A motion can have any number of arguments, but the first argument, if any, has a predefined meaning as the *count*. It is a positive or negative number, or `nil`. The argument list is followed by the documentation string *doc*, which is followed by optional keyword arguments:

`:type` *type*
: The *type* determines how the motion works after an operator. If *type* is `inclusive', then the ending position is included in the motion range. If *type* is `line', then the range is expanded to linewise positions. If *type* is `block', then the range is blockwise. The default is `exclusive', which means that the range is used as-is.

`:jump` *jump*
: If *jump* is `t`, then the previous position is stored in the jump list so it can be restored with `C-o`. The default is `nil`.

The keyword arguments are followed by the *body*, which is where the motion's behavior is defined. For instance:

```
(evil-define-motion foo-forward (count)
  "Move to the right by COUNT characters."
  :type inclusive
  (forward-char (or count 1)))
```

For more examples, you can view the source code for any command with `C-h k`. For instance, `evil-goto-line' may be viewed by typing `C-h k G` and following the file link.

5.2 Operators

An *operator* is a command which acts on the text moved over by a motion, such as `c`, `d` and `y`. Operators are defined with the macro `evil-define-operator'.

`evil-define-operator` *operator* (*beg end type args...*) *doc keyword-args... body...* [Macro]
Define an operator command *operator*. An operator must have at least two or three arguments, which have predefined meanings. *beg* is the beginning position, *end* is the

ending position, and *type*, if given, is the type of the motion range. The argument list is followed by the documentation string *doc*, which is followed by optional keyword arguments:

`:type` *type*
: Make the input range be a certain *type*. For example, an operator which only works with whole lines may set *type* to `` `line' ``.

`:motion` *motion*
: Use the motion *motion* instead of reading one from the keyboard. This does not affect the behavior in Visual state, where the selection boundaries are used instead.

`:repeat` *repeat*
: If *repeat* is `t` (the default), then `.` will repeat the operator. If *repeat* is `nil`, then the operator will not be repeated.

`:move-point` *move-point*
: If *move-point* is `t` (the default), then the cursor is positioned at the beginning of the range. If *move-point* is `nil`, then the original position is preserved.

`:keep-visual` *keep-visual*
: If *keep-visual* is `t`, then the selection is not disabled when the operator is run in Visual state; it is up to the operator to do this. The default is `nil`, which means that Visual state is exited automatically.

The keyword arguments are followed by the *body*, which is where the operator's actions on *beg* and *end* are defined. For example, `` `evil-rot13' ``, which is bound to *g?* and performs ROT13 encryption on the text, may be defined as:

```
(evil-define-operator evil-rot13 (beg end)
  "ROT13 encrypt text."
  (rot13-region beg end))
```

Pressing *g?w* will encrypt a word by calling `` `rot13-region' `` on the text moved over by the *w* motion.

5.3 Text objects

A *text object* is a special kind of motion which sets a beginning position as well as an ending position, such as *iw* and *a(*. In Visual state, text objects alter both ends of the selection. Text objects are defined with the macro `` `evil-define-text-object' ``.

`evil-define-text-object` *object* (*count args...*) *doc keyword-args... body...* [Macro]
: Define a text object *object*. The first argument has a predefined meaning as the *count*: it is a positive or negative number. The argument list is followed by the documentation string *doc*, which is followed by optional keyword arguments:

`:type` *type*
: Use the type *type* after an operator. In Visual state, this is the type of the selection.

`:extend-selection` *extend-selection*
: If *extend-selection* is `t` (the default), then the text object always enlarges the current selection. If `nil`, then the object replaces the selection.

The keyword arguments are followed by the *body*, which should evaluate to a list (*beg end*) of two positions in the buffer. For example, a text object which selects three characters following the current position could be defined as:

```
(evil-define-text-object foo (count)
  "Select three characters."
  (list (point) (+ (point) 3)))
```

Evil provides several functions which return a list of positions, for use in the definition of a text object. These functions follow the rule that a positive *count* selects text after the current position, while a negative *count* selects text before it.

`evil-inner-object-range` *count forward backward* [Function]
: Return a text range (*beg end*) of *count* \inner" text objects (e.g., *iw*, *is*). *forward* is a function which moves to the end of an object, and *backward* is a function which moves to the beginning.

`evil-an-object-range` *count forward backward* [Function]
: Return a text range (*beg end*) of *count* text objects with whitespace (e.g., *aw*, *as*). *forward* is a function which moves to the end of an object, and *backward* is a function which moves to the beginning.

`evil-paren-range` *count open close* **&optional** *exclusive* [Function]
: Return a text range (*beg end*) of *count* delimited blocks (e.g., *i(*, *a(*). *open* and *close* are characters. If *exclusive* is non-nil, then the delimiters are excluded from the range. This function uses Emacs' syntax table and is only applicable for single-character delimiters; use `evil-regexp-range' to match multiple characters.

`evil-regexp-range` *count open close* **&optional** *exclusive* [Function]
: Return a text range (*beg end*) of *count* delimited blocks (e.g., *it*, *at*). *open* and *close* are regular expressions. If *exclusive* is non-nil, then the delimiters are excluded from the range.

5.4 Types

A *type* is a transformation on a pair of buffer positions. Evil defines the types `exclusive', `inclusive', `line' and `block', which are used for motion ranges and Visual selection. Types are defined with the macro `evil-define-type'.

`evil-define-type` *type doc keyword-args. . .* [Macro]
: Define a type *type*, described by the documentation string *doc*. Then follows keyword arguments:

`:expand` *expand*
: A function which takes two buffer positions and returns a list (*beg end*) of expanded positions.

`:contract` *contract*
: A function which takes two expanded bufer positions and returns a list `(`*beg end*`)` of unexpanded positions. Optional.

`:normalize` *normalize*
: A function which takes two unexpanded bufer positions and returns a list `(`*beg end*`)` of adjusted positions. Optional.

`:injective` *injective*
: If `t` (the default), then expansion is one-to-one { i.e., *expand* followed by *contract* always returns the original positions. If `nil`, then several positions may expand to the same (for example, the `` `line' `` type is one-to-many as it expands to the containing lines).

Further keywords and functions may be specifed. These are understood to be transformations on bufer positions, like *expand* and *contract*.

5.5 States

States are defned with the macro `` `evil-define-state' ``. The macro defnes the necessary hooks, keymaps and variables for a state, as well as a toggle function `` `evil-`*state*`-state' `` for entering the state, and a predicate function `` `evil-`*state*`-state-p' `` which returns `t` when the state is active, and `nil` otherwise.

`evil-define-state` *state doc keyword-args... body...* [Macro]
: Defne an Evil state *state*, described by the documentation string *doc*. Then follows optional keyword arguments:

  `:tag` *tag* Mode line indicitor, e.g., `"<T>"`.

  `:message` *message*
  : String shown in the echo area.

  `:cursor` *cursor*
  : Cursor specifcation.

  `:enable` *enable*
  : List of other modes and states to enable. A state may enable another state's keymaps in addition to its own.

  This is followed the *body*, which is executed whenever the state is enabled or disabled. The state's predicate function may be used to distinguish between the two.

6 Other internals

6.1 Command properties

Evil defnes *command properties* to store information about commands, such as whether they should be repeated. A command property is a `:keyword` with an associated value, e.g., `:repeat nil`.

`evil-add-command-properties` *command* **&rest** *properties* [Function]
Add *properties* to *command*. The properties should be specifed as a list of keywords and values:

```
(evil-add-command-properties 'my-command :repeat t)
```

`evil-set-command-properties` *command* **&rest** *properties* [Function]
Like `evil-add-command-properties', but resets all previous properties.

`evil-get-command-property` *command property* [Function]
Return the value of a command property.

`evil-define-command` *command* (*args...*) *doc keyword-args... body...* [Macro]
Defne a command with command properties *keyword-args*.

For setting repeat properties, Evil provides the following functions:

`evil-declare-repeat` *command* [Function]
Declare *command* to be repeatable.

`evil-declare-not-repeat` *command* [Function]
Declare *command* to be nonrepeatable.

`evil-declare-change-repeat` *command* [Function]
Declare *command* to be repeatable by bufer changes rather than keystrokes.

Appendix A GNU Free Documentation License

Version 1.3, 3 November 2008

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. Secondarily, 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 modifcations in the Modifed Version, together with at least fve of the principal authors of the Document (all of its principal authors, if it has fewer than fve), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modifed Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifcations adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modifed 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 Modifed 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 Modifed 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 Modifed Version.

N. Do not retitle any existing section to be Entitled \Endorsements" or to confict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modifed 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 Modifed 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 Modifed Version by various parties | for example, statements of peer review or that the text has been approved by an organization as the authoritative defnition of a standard.

You may add a passage of up to fve 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 Modifed 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 Modifed Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defned in section 4 above for modifed versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodifed, 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 diferent 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 modifcation, 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 fnally 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 notifes you of the violation by some reasonable means, this is the frst 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 dife r 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 specifes 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 specifed 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 specifes 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 frst 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.