GnuTLS-Guile

Guile binding for GNU TLS

for version 3.5.8, 30 October 2016

This manual is last updated 30 October 2016 for version 3.5.8 of GnuTLS.

Table of Contents

GnuTLS-Guile

This manual is last updated 30 October 2016 for version 3.5.8 of GnuTLS.

1 Preface

This manual describes the GNU Guile Scheme programming interface to GnuTLS, which is distributed as part of GnuTLS. The reader is assumed to have basic knowledge of the protocol and library. Details missing from this chapter may be found in Function reference, of the C API reference.

At this stage, not all the C functions are available from Scheme, but a large subset thereof is available.

2 Guile Preparations

The GnuTLS Guile bindings are available for Guile's 2.0 stable series, as well as the forthcoming 2.2 series and the legacy 1.8 series.

By default they are installed under the GnuTLS installation directory, typically `/usr/local/share/guile/site/`). Normally Guile will not find the module there without help. You may experience something like this:

```
$ guile
...
scheme@(guile-user)> (use-modules (gnutls))
ERROR: no code for module (gnutls)
```

There are two ways to solve this. The first is to make sure that when building GnuTLS, the Guile bindings will be installed in the same place where Guile looks. You may do this by using the `--with-guile-site-dir` parameter as follows:

```
$ ./configure --with-guile-site-dir=no
```

This will instruct GnuTLS to attempt to install the Guile bindings where Guile will look for them. It will use `guile-config info pkgdatadir` to learn the path to use.

If Guile was installed into `/usr`, you may also install GnuTLS using the same prefix:

```
$ ./configure --prefix=/usr
```

If you want to specify the path to install the Guile bindings you can also specify the path directly:

```
$ ./configure --with-guile-site-dir=/opt/guile/share/guile/site
```

The second solution requires some more work but may be easier to use if you do not have system administrator rights to your machine. You need to instruct Guile so that it finds the GnuTLS Guile bindings. Either use the `GUILE_LOAD_PATH` environment variable as follows:

```
$ GUILE_LOAD_PATH="/usr/local/share/guile/site:$GUILE_LOAD_PATH" guile
scheme@(guile-user)> (use-modules (gnutls))
scheme@(guile-user)>
```

Alternatively, you can modify Guile's `%load-path` variable (see Section "Build Config" in *The GNU Guile Reference Manual*).

At this point, you might get an error regarding `guile-gnutls-v-2` similar to:

```
gnutls.scm:361:1: In procedure dynamic-link in expression (load-extension "guile-gnutl
gnutls.scm:361:1: file: "guile-gnutls-v-2", message: "guile-gnutls-v-2.so: cannot open
```

In this case, you will need to modify the run-time linker path, for example as follows:

```
$ LD_LIBRARY_PATH=/usr/local/lib GUILE_LOAD_PATH=/usr/local/share/guile/site guile
scheme@(guile-user)> (use-modules (gnutls))
scheme@(guile-user)>
```

To check that you got the intended GnuTLS library version, you may print the version number of the loaded library as follows:

```
$ guile
scheme@(guile-user)> (use-modules (gnutls))
scheme@(guile-user)> (gnutls-version)
"3.5.8"
scheme@(guile-user)>
```

3 Guile API Conventions

This chapter details the conventions used by Guile API, as well as specificities of the mapping of the C API to Scheme.

3.1 Enumerates and Constants

Lots of enumerates and constants are used in the GnuTLS C API. For each C enumerate type, a disjoint Scheme type is used—thus, enumerate values and constants are not represented by Scheme symbols nor by integers. This makes it impossible to use an enumerate value of the wrong type on the Scheme side: such errors are automatically detected by type-checking.

The enumerate values are bound to variables exported by the `(gnutls)` module. These variables are named according to the following convention:

- All variable names are lower-case; the underscore `_` character used in the C API is replaced by hyphen `-`.
- All variable names are prepended by the name of the enumerate type and the slash `/` character.
- In some cases, the variable name is made more explicit than the one of the C API, e.g., by avoid abbreviations.

Consider for instance this C-side enumerate:

```
typedef enum
{
  GNUTLS_CRD_CERTIFICATE = 1,
  GNUTLS_CRD_ANON,
  GNUTLS_CRD_SRP,
  GNUTLS_CRD_PSK
} gnutls_credentials_type_t;
```

The corresponding Scheme values are bound to the following variables exported by the `(gnutls)` module:

```
credentials/certificate
credentials/anonymous
credentials/srp
credentials/psk
```

Hopefully, most variable names can be deduced from this convention.

Scheme-side "enumerate" values can be compared using `eq?` (see Section "Equality" in *The GNU Guile Reference Manual*). Consider the following example:

```
(let ((session (make-session connection-end/client)))

  ;;
  ;; ...
  ;;

  ;; Check the ciphering algorithm currently used by SESSION.
```

```
(if (eq? cipher/arcfour (session-cipher session))
    (format #t "We're using the ARCFOUR algorithm"))
```

In addition, all enumerate values can be converted to a human-readable string, in a type-specific way. For instance, `(cipher->string cipher/arcfour)` yields `"ARCFOUR 128"`, while `(key-usage->string key-usage/digital-signature)` yields `"digital-signature"`. Note that these strings may not be sufficient for use in a user interface since they are fairly concise and not internationalized.

3.2 Procedure Names

Unlike C functions in GnuTLS, the corresponding Scheme procedures are named in a way that is close to natural English. Abbreviations are also avoided. For instance, the Scheme procedure corresponding to `gnutls_certificate_set_dh_params` is named `set-certificate-credentials-dh-parameters!`. The `gnutls_` prefix is always omitted from variable names since a similar effect can be achieved using Guile's nifty binding renaming facilities, should it be needed (see Section "Using Guile Modules" in *The GNU Guile Reference Manual*).

Often Scheme procedure names differ from C function names in a way that makes it clearer what objects they operate on. For example, the Scheme procedure named `set-session-transport-port!` corresponds to `gnutls_transport_set_ptr`, making it clear that this procedure applies to session.

3.3 Representation of Binary Data

Many procedures operate on binary data. For instance, `pkcs3-import-dh-parameters` expects binary data as input.

Binary data is represented on the Scheme side using bytevectors (see Section "Bytevectors" in *The GNU Guile Reference Manual*). Homogeneous vectors such as SRFI-4 `u8vectors` can also be used[1].

As an example, generating and then exporting Diffie-Hellman parameters in the PEM format can be done as follows:

```
(let* ((dh  (make-dh-parameters 1024))
       (pem (pkcs3-export-dh-parameters dh
                                        x509-certificate-format/pem)))
  (call-with-output-file "some-file.pem"
    (lambda (port)
      (uniform-vector-write pem port))))
```

For an example of OpenPGP key import from a file, see Section 4.3 [Importing OpenPGP Keys Guile Example], page 10.

3.4 Input and Output

The underlying transport of a TLS session can be any Scheme input/output port (see Section "Ports and File Descriptors" in *The GNU Guile Reference Manual*). This has to be specified using `set-session-transport-port!`.

[1] Historically, SRFI-4 `u8vectors` are the closest thing to bytevectors that Guile 1.8 and earlier supported.

However, for better performance, a raw file descriptor can be specified, using `set-session-transport-fd!`. For instance, if the transport layer is a socket port over an OS-provided socket, you can use the `port->fdes` or `fileno` procedure to obtain the underlying file descriptor and pass it to `set-session-transport-fd!` (see Section "Ports and File Descriptors" in *The GNU Guile Reference Manual*). This would work as follows:

```
(let ((socket (socket PF_INET SOCK_STREAM 0))
      (session (make-session connection-end/client)))

  ;;
  ;; Establish a TCP connection...
  ;;

  ;; Use the file descriptor that underlies SOCKET.
  (set-session-transport-fd! session (fileno socket)))
```

Once a TLS session is established, data can be communicated through it (i.e., *via* the TLS record layer) using the port returned by `session-record-port`:

```
(let ((session (make-session connection-end/client)))

  ;;
  ;; Initialize the various parameters of SESSION, set up
  ;; a network connection, etc.
  ;;

  (let ((i/o (session-record-port session)))
    (display "Hello peer!" i/o)
    (let ((greetings (read i/o)))

      ;; ...

      (bye session close-request/rdwr))))
```

Note that each write to the session record port leads to the transmission of an encrypted TLS "Application Data" packet. In the above example, we create an Application Data packet for the 11 bytes for the string that we write. This is not efficient both in terms of CPU usage and bandwidth (each packet adds at least 5 bytes of overhead and can lead to one `write` system call), so we recommend that applications do their own buffering.

A lower-level I/O API is provided by `record-send` and `record-receive!` which take a bytevector (or a SRFI-4 vector) to represent the data sent or received. While it might improve performance, it is much less convenient than the session record port and should rarely be needed.

3.5 Exception Handling

GnuTLS errors are implemented as Scheme exceptions (see Section "Exceptions" in *The GNU Guile Reference Manual*). Each time a GnuTLS function returns an error, an exception with key `gnutls-error` is raised. The additional arguments that are thrown include an error code and the name of the GnuTLS procedure that raised the exception. The error code is pretty much like an enumerate value: it is one of the `error/` variables exported by the `(gnutls)` module (see Section 3.1 [Enumerates and Constants], page 4). Exceptions can be turned into error messages using the `error->string` procedure.

The following examples illustrates how GnuTLS exceptions can be handled:

```
(let ((session (make-session connection-end/server)))

  ;;
  ;; ...
  ;;

  (catch 'gnutls-error
    (lambda ()
      (handshake session))
    (lambda (key err function . currently-unused)
      (format (current-error-port)
              "a GnuTLS error was raised by `~a': ~a~%"
              function (error->string err)))))
```

Again, error values can be compared using `eq?`:

```
    ;; `gnutls-error' handler.
    (lambda (key err function . currently-unused)
      (if (eq? err error/fatal-alert-received)
          (format (current-error-port)
                  "a fatal alert was caught!~%")
          (format (current-error-port)
                  "something bad happened: ~a~%"
                  (error->string err))))
```

Note that the `catch` handler is currently passed only 3 arguments but future versions might provide it with additional arguments. Thus, it must be prepared to handle more than 3 arguments, as in this example.

4 Guile Examples

This chapter provides examples that illustrate common use cases.

4.1 Anonymous Authentication Guile Example

Anonymous authentication is very easy to use. No certificates are needed by the communicating parties. Yet, it allows them to benefit from end-to-end encryption and integrity checks.

The client-side code would look like this (assuming *some-socket* is bound to an open socket port):

```
;; Client-side.

(let ((client (make-session connection-end/client)))
  ;; Use the default settings.
  (set-session-default-priority! client)

  ;; Don't use certificate-based authentication.
  (set-session-certificate-type-priority! client '())

  ;; Request the "anonymous Diffie-Hellman" key exchange method.
  (set-session-kx-priority! client (list kx/anon-dh))

  ;; Specify the underlying socket.
  (set-session-transport-fd! client (fileno some-socket))

  ;; Create anonymous credentials.
  (set-session-credentials! client
                            (make-anonymous-client-credentials))

  ;; Perform the TLS handshake with the server.
  (handshake client)

  ;; Send data over the TLS record layer.
  (write "hello, world!" (session-record-port client))

  ;; Terminate the TLS session.
  (bye client close-request/rdwr))
```

The corresponding server would look like this (again, assuming *some-socket* is bound to a socket port):

```
;; Server-side.

(let ((server (make-session connection-end/server)))
  (set-session-default-priority! server)
  (set-session-certificate-type-priority! server '())
  (set-session-kx-priority! server (list kx/anon-dh))
```

```
  ;; Specify the underlying transport socket.
  (set-session-transport-fd! server (fileno some-socket))

  ;; Create anonymous credentials.
  (let ((cred (make-anonymous-server-credentials))
        (dh-params (make-dh-parameters 1024)))
    ;; Note: DH parameter generation can take some time.
    (set-anonymous-server-dh-parameters! cred dh-params)
    (set-session-credentials! server cred))

  ;; Perform the TLS handshake with the client.
  (handshake server)

  ;; Receive data over the TLS record layer.
  (let ((message (read (session-record-port server))))
    (format #t "received the following message: ~a~%"
            message)

    (bye server close-request/rdwr)))
```

This is it!

4.2 OpenPGP Authentication Guile Example

GnuTLS allows users to authenticate using OpenPGP certificates. Using OpenPGP-based authentication is not more complicated than using anonymous authentication. It requires a bit of extra work, though, to import the OpenPGP public and private key of the client/server. Key import is omitted here and is left as an exercise to the reader (see Section 4.3 [Importing OpenPGP Keys Guile Example], page 10).

Assuming *some-socket* is bound to an open socket port and *pub* and *sec* are bound to the client's OpenPGP public and secret key, respectively, client-side code would look like this:

```
;; Client-side.

(define %certs (list certificate-type/openpgp))

(let ((client (make-session connection-end/client))
      (cred   (make-certificate-credentials)))
  (set-session-default-priority! client)

  ;; Choose OpenPGP certificates.
  (set-session-certificate-type-priority! client %certs)

  ;; Prepare appropriate client credentials.
  (set-certificate-credentials-openpgp-keys! cred pub sec)
  (set-session-credentials! client cred)

  ;; Specify the underlying transport socket.
```

```
    (set-session-transport-fd! client (fileno some-socket))

    (handshake client)
    (write "hello, world!" (session-record-port client))
    (bye client close-request/rdwr))
```

Similarly, server-side code would be along these lines:

```
;; Server-side.

(define %certs (list certificate-type/openpgp))

(let ((server (make-session connection-end/server))
      (dh     (make-dh-parameters 1024)))
  (set-session-default-priority! server)

  ;; Choose OpenPGP certificates.
  (set-session-certificate-type-priority! server %certs)

  (let ((cred (make-certificate-credentials)))
    ;; Prepare credentials with Diffie-Hellman parameters.
    (set-certificate-credentials-dh-parameters! cred dh)
    (set-certificate-credentials-openpgp-keys! cred pub sec)
    (set-session-credentials! server cred))

  (set-session-transport-fd! server (fileno some-socket))

  (handshake server)
  (let ((msg (read (session-record-port server))))
    (format #t "received: ~a~%" msg)

    (bye server close-request/rdwr)))
```

4.3 Importing OpenPGP Keys Guile Example

The following example provides a simple way of importing "ASCII-armored" OpenPGP keys from files, using the `import-openpgp-certificate` and `import-openpgp-private-key` procedures.

```
(use-modules (srfi srfi-4)
             (gnutls))

(define (import-key-from-file import-proc file)
  ;; Import OpenPGP key from FILE using IMPORT-PROC.

  ;; Prepare a u8vector large enough to hold the raw
  ;; key contents.
  (let* ((size (stat:size (stat path)))
         (raw  (make-u8vector size)))
```

```
    ;; Fill in the u8vector with the contents of FILE.
    (uniform-vector-read! raw (open-input-file file))

    ;; Pass the u8vector to the import procedure.
    (import-proc raw openpgp-certificate-format/base64)))


(define (import-public-key-from-file file)
  (import-key-from-file import-openpgp-certificate file))

(define (import-private-key-from-file file)
  (import-key-from-file import-openpgp-private-key file))
```

The procedures `import-public-key-from-file` and `import-private-key-from-file` can be passed a file name. They return an OpenPGP public key and private key object, respectively (see Chapter 5 [Guile Reference], page 12).

5 Guile Reference

This chapter lists the GnuTLS Scheme procedures exported by the `(gnutls)` module (see Section "The Guile module system" in *The GNU Guile Reference Manual*).

(Guile not available, documentation not generated.)

Appendix A Copying Information

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 noncommercially. 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 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 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. 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.

Procedure Index

E

M

P

R

S

Concept Index