

The GNU D Compiler

For GCC version 16.0.0 (pre-release)

(GCC)

David Friedman, Iain Buclaw

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

Copyright © 2006-2025 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. A copy of the license is included in the section entitled “GNU Free Documentation License”.

Option Index	63
Keyword Index	65

- `'c++23'` Sets `__traits(getTargetInfo, "cppStd")` to 202302.
- `-finclude-imports`
Include imported modules in the compilation, as if they were given on the command line. When this option is enabled, all imported modules are compiled except those that are part of libphobos.
- `-fno-invariants`
Turns off code generation for class `invariant` contracts.
- `-fmain` Generates a default `main()` function when compiling. This is useful when unittesting a library, as it enables running the unittests in a library without having to manually define an entry-point function. This option does nothing when `main` is already defined in user code.
- `-fno-moduleinfo`
Turns off generation of the `ModuleInfo` and related functions that would become unreferenced without it, which may allow linking to programs not written in D. Functions that are not be generated include module constructors and destructors (`static this` and `static ~this`), `unittest` code, and DSO registry functions for dynamically linked code.
- `-fonly=filename`
Tells the compiler to parse and run semantic analysis on all modules on the command line, but only generate code for the module specified by *filename*.
- `-fno-postconditions`
Turns off code generation for postcondition `out` contracts.
- `-fno-preconditions`
Turns off code generation for precondition `in` contracts.
- `-fpreview=id`
Turns on an upcoming D language change identified by *id*. The following values are supported:
- `'all'` Turns on all upcoming D language features.
 - `'bitfields'`
Implements bit-fields in D.
 - `'dip1000'` Implements <https://github.com/dlang/DIPs/blob/master/DIPs/other/DIP1000.md> (Scoped pointers).
 - `'dip1008'` Implements <https://github.com/dlang/DIPs/blob/master/DIPs/other/DIP1008.md> (Allow exceptions in `@nogc` code).
 - `'dip1021'` Implements <https://github.com/dlang/DIPs/blob/master/DIPs/accepted/DIP1021.md> (Mutable function arguments).
 - `'dip25'` Implements <https://github.com/dlang/DIPs/blob/master/DIPs/archive/DIP25.md> (Sealed references).
 - `'dtorfields'`
Turns on generation for destructing fields of partially constructed objects.

<code>'fieldwise'</code>	Turns on generation of struct equality to use field-wise comparisons.
<code>'fixaliasthis'</code>	Implements new lookup rules that check the current scope for <code>alias this</code> before searching in upper scopes.
<code>'fiximmutableconv'</code>	Disallows unsound immutable conversions that were formerly incorrectly permitted.
<code>'in'</code>	Implements <code>in</code> parameters to mean <code>scope const [ref]</code> and accepts rvalues.
<code>'inclusiveincontracts'</code>	Implements <code>in</code> contracts of overridden methods to be a superset of parent contract.
<code>'nosharedaccess'</code>	Turns off and disallows all access to shared memory objects.
<code>'rvaluerefparam'</code>	Implements rvalue arguments to <code>ref</code> parameters.
<code>'systemvariables'</code>	Disables access to variables marked <code>@system</code> from <code>@safe</code> code.

-frelease

Turns on compiling in release mode, which means not emitting runtime checks for contracts and asserts. Array bounds checking is not done for `@system` and `@trusted` functions, and assertion failures are undefined behavior.

This is equivalent to compiling with the following options:

```
gdc -fno-assert -fbounds-check=safe -fno-invariants \
    -fno-postconditions -fno-preconditions -fno-switch-errors
```

-frevert=

Turns off a D language feature identified by *id*. The following values are supported:

<code>'all'</code>	Turns off all revertable D language features.
<code>'dip1000'</code>	Reverts https://github.com/dlang/DIPs/blob/master/DIPs/other/DIP1000.md (Scoped pointers).
<code>'dip25'</code>	Reverts https://github.com/dlang/DIPs/blob/master/DIPs/archive/DIP25.md (Sealed references).
<code>'dtorfields'</code>	Turns off generation for destructing fields of partially constructed objects.
<code>'intpromote'</code>	Turns off C-style integral promotion for unary <code>+</code> , <code>-</code> and <code>~</code> expressions.

- MQ *target***
Same as **-MT**, but it quotes any characters which are special to **make**.
- MD**
This option is equivalent to **-M -MF *file***. The driver determines *file* by removing any directory components and suffix from the input file, and then adding a **.deps** suffix.
- MMD**
Like **-MD** but does not mention imported modules from the D standard library package directories.
- X**
Output information describing the contents of all source files being compiled in JSON format to a file. The driver determines *file* by removing any directory components and suffix from the input file, and then adding a **.json** suffix.
- Xf *file***
Same as **-X**, but writes all JSON contents to the specified *file*.
- fdoc**
Generates Ddoc documentation and writes it to a file. The compiler determines *file* by removing any directory components and suffix from the input file, and then adding a **.html** suffix.
- fdoc-dir=*dir***
Same as **-fdoc**, but writes documentation to directory *dir*. This option can be used with **-fdoc-file=*file*** to independently set the output file and directory path.
- fdoc-file=*file***
Same as **-fdoc**, but writes documentation to *file*. This option can be used with **-fdoc-dir=*dir*** to independently set the output file and directory path.
- fdoc-inc=*file***
Specify *file* as a Ddoc macro file to be read. Multiple **-fdoc-inc** options can be used, and files are read and processed in the same order.
- fdump-c++-spec=*file***
For D source files, generate corresponding C++ declarations in *file*.
- fdump-c++-spec-verbose**
In conjunction with **-fdump-c++-spec=** above, add comments for ignored declarations in the generated C++ header.
- fsave-mixins=*file***
Generates code expanded from D **mixin** statements and writes the processed sources to *file*. This is useful to debug errors in compilation and provides source for debuggers to show when requested.

1.5 Warnings

Warnings are diagnostic messages that report constructions that are not inherently erroneous but that are risky or suggest there is likely to be a bug in the program. Unless **-Werror** is specified, they do not prevent compilation of the program.

- Wall**
Turns on all warnings messages. Warnings are not a defined part of the D language, and all constructs for which this may generate a warning message are valid code.

1.7 Developer Options

This section describes command-line options that are primarily of interest to developers or language tooling.

-fdump-d-original

Output the internal front-end AST after the **semantic3** stage. This option is only useful for debugging the GNU D compiler itself.

-v

Dump information about the compiler language processing stages as the source program is being compiled. This includes listing all modules that are processed through the **parse**, **semantic**, **semantic2**, and **semantic3** stages; all **import** modules and their file paths; and all **function** bodies that are being compiled.

ImportC does not have a preprocessor. It is designed to compile C files after they have been first run through the C preprocessor. If the C file has a `.i` extension, the file is presumed to be already preprocessed. Preprocessing can be run manually:

```
gcc -E file.c > file.i
```

ImportC collects all the `#define` macros from the preprocessor run when it is run automatically. The macros that look like manifest constants, such as:

```
#define COLOR 0x123456
```

are interpreted as D manifest constant declarations of the form:

```
enum COLOR = 0x123456;
```

The variety of macros that can be interpreted as D declarations may be expanded, but will never encompass all the metaprogramming uses of C macros.

GNU D does not directly compile C files into modules that can be linked in with D code to form an executable. When given a source file with the suffix `.c`, the compiler driver program `gdc` instead runs the subprogram `cc1`.

```
gdc file1.d file2.c // d2i file1.d -o file1.s
                  // cc1 file2.c -o file2.s
                  // as file1.s -o file1.o
                  // as file2.s -o file2.o
                  // ld file1.o file2.o
```

2.4 Inline Assembly

The `asm` keyword allows you to embed assembler instructions within D code. GNU D provides two forms of inline `asm` statements. A *basic* `asm` statement is one with no operands, while an *extended* `asm` statement includes one or more operands.

```
asm FunctionAttributes {
    AssemblerInstruction ;
}

asm FunctionAttributes {
    AssemblerTemplate
    : OutputOperands
    [ : InputOperands
    [ : Clobbers
    [ : GotoLabels ] ] ] ;
}
```

The extended form is preferred for mixing D and assembly language within a function, but to include assembly language in a function declared with the `naked` attribute you must use basic `asm`.

```
uint incr (uint value)
{
    uint result;
    asm { "incl %0"
        : "=a" (result)
        : "a" (value);
    }
    return result;
}
```


`float std.math.rounding.floor (float x)` [Function]
`double std.math.rounding.floor (double x)` [Function]
`real std.math.rounding.floor (real x)` [Function]
 Returns the value of *x* rounded downward to the next integer (toward negative infinity).
 This function is evaluated during CTFE as the GCC built-in function `__builtin_floor`.

`real std.math.rounding.round (real x)` [Function]
 Return the value of *x* rounded to the nearest integer. If the fractional part of *x* is exactly 0.5, the return value is rounded away from zero.
 This function is evaluated during CTFE as the GCC built-in function `__builtin_round`.

`real std.math.rounding.trunc (real x)` [Function]
 Returns the integer portion of *x*, dropping the fractional portion.
 This function is evaluated during CTFE as the GCC built-in function `__builtin_trunc`.

`R std.math.traits.copysign (R, X)(R to, X from)` [Template]
 Returns a value composed of *to* with *from*'s sign bit.
 This function is evaluated during CTFE as the GCC built-in function `__builtin_copysign`.

`bool std.math.traits.isFinite (X)(X x)` [Template]
 Returns true if *x* is finite.
 This function is evaluated during CTFE as the GCC built-in function `__builtin_isfinite`.

`bool std.math.traits.isInfinity (X)(X x)` [Template]
 Returns true if *x* is infinite.
 This function is evaluated during CTFE as the GCC built-in function `__builtin_isinf`.

`bool std.math.traits.isNaN (X)(X x)` [Template]
 Returns true if *x* is NaN.
 This function is evaluated during CTFE as the GCC built-in function `__builtin_isnan`.

`float std.math.trigoometry.tan (float x)` [Function]
`double std.math.trigoometry.tan (double x)` [Function]
`real std.math.trigonometry.tan (real x)` [Function]
 Returns tangent of *x*, where *x* is in radians.
 This intrinsic is the same as the GCC built-in function `__builtin_tan`.

MIPS32
MIPS64
MIPS_EABI
MIPS_HardFloat
MIPS_N32
MIPS_N64
MIPS_O32
MIPS_O64
MIPS_SoftFloat
 Versions relating to the MIPS family of processors.

NetBSD Version relating to NetBSD systems.

OpenBSD Version relating to OpenBSD systems.

OSX Version relating to OSX systems.

Posix Version relating to POSIX systems (includes Linux, FreeBSD, OSX, Solaris, etc).

PPC
PPC64
PPC_HardFloat
PPC_SoftFloat
 Versions relating to the PowerPC family of processors.

RISCV32
RISCV64 Versions relating to the RISC-V family of processors.

S390
SystemZ Versions relating to the S/390 and System Z family of processors.

S390X Deprecated; use `SystemZ` instead.

Solaris Versions relating to Solaris systems.

SPARC
SPARC64
SPARC_HardFloat
SPARC_SoftFloat
SPARC_V8Plus
 Versions relating to the SPARC family of processors.

Thumb Deprecated; use `ARM_Thumb` instead.

D_X32
X86
X86_64 Versions relating to the x86-32 and x86-64 family of processors.

Windows
Win32
Win64 Versions relating to Microsoft Windows systems.

2.8 Special Enums

Special `enum` names are used to represent types that do not have an equivalent basic D type. For example, C++ types used by the C++ name mangler.

Special enums are declared opaque, with a base type explicitly set. Unlike regular opaque enums, special enums can be used as any other value type. They have a default `.init` value, as well as other enum properties available (`.min`, `.max`). Special enums can be declared in any module, and will be recognized by the compiler.

```
import gcc.builtins;
enum __c_long : __builtin_clong;
__c_long var = 0x800A;
```

The following identifiers are recognized by GNU D.

```
__c_complex_double
    C _Complex double type.

__c_complex_float
    C _Complex float type.

__c_complex_real
    C _Complex long double type.

__c_long    C++ long type.

__c_longlong
    C++ long long type.

__c_long_double
    C long double type.

__c_ulong
    C++ unsigned long type.

__c_ulonglong
    C++ unsigned long long type.

__c_wchar_t
    C++ wchar_t type.
```

The `core.stdc.config` module declares the following shorthand alias types for convenience: `c_complex_double`, `c_complex_float`, `c_complex_real`, `cpp_long`, `cpp_longlong`, `c_long_double`, `cpp_ulong`, `cpp_ulonglong`.

It may cause undefined behavior at runtime if a special enum is declared with a base type that has a different size to the target C/C++ type it is representing. The GNU D compiler will catch such declarations and emit a warning when the `-Wmismatched-special-enum` option is seen on the command-line.

2.9 Traits

Traits are extensions to the D programming language to enable programs, at compile time, to get at information internal to the compiler. This is also known as compile time reflection.

GNU D implements a `__traits(getTargetInfo)` trait that receives a string key as its argument. The result is an expression describing the requested target information.

```
version (OSX)
```


On x86 targets, all intrinsics are available as functions in the `gcc.builtins` module, and have predictable equivalents.

```
version (DigitalMars)
{
    __simd(XMM.PSLLW, op1, op2);
    __simd_ib(XMM.PSLLW, op1, imm8);
}
version (GNU)
{
    __builtin_ia32_psllw(op1, op2);
    __builtin_ia32_psllwi(op1, imm8);
}
```

TypeInfo-based `va_arg`

The Digital Mars D compiler implements a version of `core.vararg.va_arg` that accepts a run-time `TypeInfo` argument for use when the static type is not known. This function is not implemented by GNU D. It is more portable to use variadic template functions instead.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public 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.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License “or any later version” applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

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

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

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.

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/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.

