FFTW_Ada Version 2.0 $\alpha 1$

User and Reference Manual

FFTW_Ada is an Ada 95 binding to the FFTW library written at MIT by Matteo Frigo and Steven G. Johnson. FFTW is a library for computing the Fast Fourier transform, which is both fast, and copes with arbitrary sizes of array, in multiple dimensions. FFTW is written in C and the purpose of FFTW_Ada is to allow calls to FFTW from an Ada 95 program. It includes a thin binding (a fairly direct interface to the C code, using Ada 95 types, but C concepts) and a thick binding (an interface in Ada 95 style, which makes FFTW look like an Ada 95 package). There is also a test program. FFTW_Ada, like FFTW, is released under the GNU General Public License.

August 5, 2004

Contents

1 Introduction

This is the user guide and reference manual for FFTW_Ada, an Ada 95 binding to the FFTW library of Fast Fourier Transform (FFT) code written at MIT by Matteo Frigo and Steven G. Johnson. FFTW is written in C. The purpose of FFTW_Ada is to make it possible to use FFTW from an Ada 95 program. This is achieved by providing an Ada 95 interface to the C code (known as a *binding*). The binding is compiled by an Ada 95 compiler and linked with object code produced by compiling the FFTW C code. When linked into an Ada 95 application program, it appears as if the FFTW library were written in Ada 95. All of this is made possible by the clean design of FFTW and the advanced interfacing features of Ada 95 as defined in the Ada 95 Language Reference Manual[1], Appendix B [1]. This means that the binding is portable across machines and operating systems. This version of FFTW_Ada works with FFTW version 3. *It is not compatible with version 2 of FFTW, which had a significantly different application interface.* FFTW_Ada version 1.2 was the last version of the binding compatible with FFTW version 2 and it will remain available for the foreseeable future.

FFTW_Ada version 1 was written by Steve Sangwine while at the University of Reading, UK. FFTW_Ada version 2 was developed by Steve Sangwine at the University of Essex, UK. It represents a major revision of the package, with significant deletion of material, and addition of new material, necessitated by the major changes made to the API of FFTW between versions 2 and 3.

1.1 Some words of caution

FFTW_Ada does not install itself 'out of the box'. Compilation of the Ada 95 code is straightforward, but getting it fully working with the C code of FFTW will take some methodical work. The package is written to be portable, in the sense that the Ada 95 code uses only standard Ada 95[2].

The current version of FFTW_Ada (2.0 α1) should not be considered fully tested. A test program is included, and this has been used to run a fairly comprehensive, and random, test of some of the high level facilities of FFTW_Ada (the thick binding). There are many facilities in the thin binding (low-level Ada 95) that have not been tested, because they are not exercised by the test program. FFTW_Ada has been tested with tasking, using a simple test in the test program provided (three tasks, testing 1D, 2D and 3D FFTs respectively), but the task safety of FFTW_Ada should not be considered totally reliable without more extensive testing.

1.2 FFTW itself

This manual gives only minimal information about FFTW itself. You must refer to the manual for FFTW for most of the information about it, including details of what it computes, and how, and how to install/compile it.

[1] Referred to from here on as the LRM.

[2] A harmless exception is `pragma Assert` which may not be implemented by all compilers, but probably will be in the future, as this pragma is part of the current draft amendment to the language. If pragma Assert is not supported, the compiler is required to give a warning (LRM 2.8(13)), but otherwise ignore the pragma (LRM 2.8(11)).

1.3 Floating-point precision

FFTW provides three different floating-point precisions: single, double, and long double. These are implemented using a macro which is expanded three times, once for each precision, giving different names to the functions and so on for each precision. This allows a compiled program to have all three precisions in use at once (versions of FFTW prior to version 3 could not support this). FFTW_Ada has to work in a similar manner, and it provides three different precisions, using the floating-point types declared in the standard package `Interfaces.C`.

For technical reasons which are discussed later (in section 2.1), FFTW_Ada has separate non-generic child packages for each precision. This is a major change from FFTW_Ada version 1, where generics were used to provide different precisions.

1.4 Components

FFTW_Ada is structured as a hierarchy of packages and consists of the following main components:

- A root package `FFTW_Ada`. This defines types and constants used in all three precisions of the binding. It also defines two procedures for locking and unlocking when calling procedures or functions which are not threadsafe.
- A binding to FFTW consisting of a package specification and body. This binding exists in three versions, generated by preprocessing of two master source code files. The package names are:

`FFTW_Ada.Single_Precision`	single precision
`FFTW_Ada.Double_Precision`	double precision
`FFTW_Ada.Long_Double_Precision`	long double precision

  Each binding contains both ‘thick’ and ‘thin’ elements in one package. ‘Thin’ refers to Ada functions and procedures which are directly mapped onto the underlying C code (that is they have no bodies, and are imported from the C code). ‘Thick’ subprograms provide a more Ada-like interface to the C code, using higher-level parameters.
- Library unit instantiations of the package `Ada.Numerics.Generic_Complex_Types` for the three precisions. These instantiations are called:

`FFTW_Ada.Single_Complex_Types`	single precision
`FFTW_Ada.Double_Complex_Types`	double precision
`FFTW_Ada.Long_Double_Complex_Types`	long double precision

- A test program which exercises some of the features of the binding. Like the binding itself, it exists in three versions, for each of the supported precisions. The three versions are generated by preprocessing from a single master file and their names are:

`Single_Test`	single precision
`Double_Test`	double precision
`Long_Double_Test`	long double precision

The test program verifies randomly chosen FFT results against a very simple (and slow) DFT implementation computed using the maximum precision available. It accepts command line parameters to specify the maximum array sizes, the proportion of results to be verified, whether to enable tasking, and the filenames to use for import/export of wisdom and logging of results. It tests the 1D, 2D and 3D complex FFTs through the thick binding, exercising the forward/backward directions, with each of the four different methods of planning, and with input data destroyed or preserved, for randomly chosen sizes of array. Any results which differ from the DFT implementation by more than a certain tolerance are logged.

The test program does not exercise all the facilities of the binding.

1.5 Tasking safety

FFTW is not thread safe. Although it can compute FFTs re-entrantly, it can compute plans only one at a time. The FFTW_Ada thick binding takes care of this so that multiple planning calls from a tasking application will not cause problems. (Subsequent calls are queued until the first completes using a protected flag – transparent to the user, of course.)

Of course, tasking may mess up the timing of the FFT experiments used in planning, so it would be advisable to generate measured plans using non-tasking code. (One way to do this is to run the test program *with tasking disabled* to generate wisdom, ensuring that the test program is not time-sharing with other applications.)

2 Implementation details and design rationale

2.1 Floating-point precision

FFTW_Ada version 2 does not use generics to provide different floating-point precisions (as was done with version 1), but instead has separate non-generic packages for each precision (generated from common source code using preprocessing). The reason for this is that the names of the FFTW C routines differ between precisions. For example, the function to calculate an FFT is called `fftwf_execute`, `fftw_execute`, or `fftwl_execute`, depending on which precision is desired (single, double or long double respectively). Since these link names have to be used in `Import` pragmas, and the link names in these pragmas have to be static (LRM B.1(27)), they cannot be implemented using generic parameters. This is why there are three versions of the child packages, generated from one source using preprocessing.

2.2 FFTW facilities supported

The binding provides direct translations of most of the FFTW facilities into Ada 95, using C concepts like pointers. Some facilities are omitted, because they are not likely to be useful in an Ada program. These

Ada versions of the FFTW facilities constitute a 'thin' binding. The names used in the thin binding match those in the FFTW code fairly obviously. It is straightforward to understand the facilities in the thin binding by reference to the FFTW manual where the C code is documented, but of course, using these facilities directly in an Ada program requires the Ada code to do quite a lot of work to match the C concepts. The thin binding routines do not currently include the 'guru' interface.

Higher-level (*i.e.* more Ada-like) facilities are provided by the thick binding, which consists of functions and procedures in Ada-style that perform conversions and mappings between higher-level parameters and those used in the thin binding and the C code. Arrays, for example, are passed in Ada style as array parameters, whereas the C code needs pointers to the first element.

There are 12 thick binding planners, implemented in Ada as generic functions that take input parameters like the arrays to be transformed and the planning options and return a plan. The plan is then passed to a single `Execute` procedure to compute the transform. The planners are generic because this allows the user to have total control over the types used for indexing the arrays. The generic parameters of these planner functions are: index types (1, 2, or 3 according to the dimension of the transform to be planned); and a constrained array type of real or complex values as appropriate. The user is not forced to use a particular index type (any discrete type is acceptable, but the real gain is that the user is not forced to use `0 .. N - 1`, or `1 .. N`, or any other specific indexing scheme). The choice of a constrained type was dictated by generality. If an unconstrained type was used, the user would have to use unconstrained arrays. Constrained arrays can be declared as subtypes of an unconstrained type, so the choice of constrained arrays does not preclude the use of unconstrained arrays in the user's application (but an instantiation of the generic package for the appropriate number of dimensions will be needed for each constrained subtype). To explain how this operates, for those who have not made extensive use of generics, or those new to Ada 95, consider an image processing application in which images are read from a file and an FFT has to be computed. The image size will not be known until after the file has been opened, and some header information read from the file. At this point, a constrained subtype of an unconstrained 2-dimensional complex array type may be declared to match the image size. Then, the generic 2D planner may be instantiated, a plan created, and the FFT computed.

2.3 Wisdom

Wisdom is implemented in a straightforward way by passing strings in and out of FFTW. The facilites in FFTW which provide for reading and writing wisdom from and to files are not supported. Instead, the binding provides two Ada 95 procedures to read and write wisdom from and to files with a given filename (using Ada input/output).

3 Installation

The recommended way to install FFTW_Ada is to start by installing FFTW by following all the relevant instructions in the FFTW manual, plus any platform specific advice given on the FFTW website at http://www.fftw.org/.

It is best to have a working C system before attempting to get the Ada 95 binding working. Since FFTW comes with a fairly comprehensive test program, it is a good idea to use this test program to make sure all is well.

3.1 Compiling and testing the bindings

This part of the installation is fairly straightforward. All you need to do is compile the various Ada 95 packages. A test program is provided to allow you to verify that the compiled code will link and work with FFTW. As a side effect, the test program will create a wisdom file which could be used in an application. (You could use a long run of the test program to generate a lot of wisdom for many array sizes, and perhaps not need to write your application code to generate wisdom.)

The main complication when linking C code with Ada 95 code is getting the linker to find the C object code. The end result of the FFTW installation will usually be a library file which can be linked with your Ada 95 application code. If your Ada 95 implementation supports **pragma** `Linker_Options` you can put this pragma into your application code and then just follow your usual steps for compiling Ada 95 code. The pragma has not been used in the distributed code of FFTW_Ada because the parameter of the pragma is not portable. If **pragma** `Linker_Options` is not possible, you will need to specify the linker options and object files when you run your linker.

4 License, Copyright and Lack of Warranty

The following notice is included in all the source files of FFTW_Ada:

```
------------------------------------------------------------------------
------------------------------------------------------------------------
-- FFTW_Ada -- An Ada95 Binding to the FFTW library (www.fftw.org) --
--                                                                    --
-- Copyright (©) 2000 Dr Stephen J. Sangwine (S.Sangwine@IEEE.org) --
-- Version 2 (©) 2004                                                 --
--                                                                    --
-- This software was created by Stephen J. Sangwine. He hereby     --
-- asserts his Moral Right to be identified as author of this      --
-- software.                                                          --
------------------------------------------------------------------------
------------------------------------------------------------------------
-- FFTW_Ada is free software; you can redistribute it and/or       --
-- modify it under the terms of the GNU General Public License as  --
-- published by the Free Software Foundation; either version 2 of  --
-- the License, or (at your option) any later version.             --
--                                                                    --
-- FFTW_Ada is distributed in the hope that it will be useful, but --
-- WITHOUT ANY WARRANTY; without even the implied warranty of      --
-- MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the    --
-- GNU General Public License for more details.                    --
--                                                                    --
-- You should have received a copy of the GNU General Public       --
-- License along with this software (in the file gpl.txt); if not, --
-- contact the Free Software Foundation, or access www.fsf.org.    --
------------------------------------------------------------------------
------------------------------------------------------------------------
```

Since FFTW_Ada is only a binding (interface) to FFTW, the license for FFTW_Ada does not grant any rights to use FFTW. Since both FFTW_Ada and FFTW are released under the Gnu General Public License, this will not be a problem for uses covered by that license. If you wish to use FFTW_Ada under terms different to those of the GNU General Public License, please contact the author. A commercial license for FFTW_Ada version 2 may be negotiable, but it may require negotiation with the two universities where the author has worked (and you will also need a license from MIT to use FFTW).

5 Comments and bug reports

Please send comments, suggestions, technical contributions or bug reports to: S.Sangwine@IEEE.org

References

[1] Ada 95 reference manual, January 1995. ANSI/ISO/IEC-8652:1995.

A Distribution

FFTW_Ada is available from http://privatewww.essex.ac.uk/~sjs/fftw_ada/fftwa.html. It is distributed in zip format[3].

If your machine or operating system does not have an unzip utility already installed, then note that free software is available from the Info-Zip site for nearly all operating systems.
See: http://www.info-zip.org/pub/infozip/.

FFTW is available from: http://www.fftw.org/

The FFTW_Ada distribution consists of the 22 files listed in the table below, plus this manual, making 23 in all. The source code files use the ISO-8859-1 Latin 1 character set, but the actual Ada code (outside comments) is restricted to the ISO-646 subset (originally known as ASCII). The source and text files use the DOS end of line convention (CR + LF)[4].

1	`gpl.txt`	GNU General Public License.
2 3	`fftw_ada.ads` `fftw_ada.adb`	Root package specification and body.
4 5	`fftw_ada-single_precision.ads` `fftw_ada-single_precision.adb`	Single precision binding.
6 7	`fftw_ada-double_precision.ads` `fftw_ada-double_precision.adb`	Double precision binding.
8 9	`fftw_ada-long_double_precision.ads` `fftw_ada-long_double_precision.adb`	Long double precision binding.

10
11
12
	`single_test.adb`
	`double_test.adb`
	`long_double_test.adb`
		Test programs in the three different precisions.

13
14
15
	`fftw_ada-single_complex_types.ads`
	`fftw_ada-double_complex_types.ads`
	`fftw_ada-long_double_complex_types.ads`
		Instantiations of complex types package for each precision.

16 17	`fftw_ada-$_precision.ads` `fftw_ada-$_precision.adb`	Master files for the binding.
18	`$test.adb`	Master file for test program.

19
20
21
22
	`single.def`
	`double.def`
	`long_double.def`
	`preprocess.bat`
		Preprocessor definition files and DOS batch file for running the preprocessor (gnatprep).

Files from 16 to 22 in the above table are not needed unless you plan to make changes to the code. They are the master files before preprocessing to create the compilable source code files listed above them in the table.

FFTW_Ada distributed files are accompanied by detached PGP signature files (extension `.sig`). To verify these (and it is not essential to do so) you will need a version of PGP which supports DH encryption (older

[3] Earlier versions were also distributed in tar format, but since nearly all operating systems now have support or free software for the zip format, I don't intend to make FFTW_Ada available in tar format in future.

[4] If you are using a *nix system you may need to convert the files to the LF convention using `dos2unix` or similar, but if your editor and compiler support the DOS convention you will not need to do this.

versions support only RSA), and my public key, which is given below (copy and paste it to a file and import into PGP). It is also available from key servers and from my web page at http://privatewww.essex.ac.uk/~sjs.

```
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: PGPfreeware 6.0.2i

mQGiBDU3GkARBADyn6tgErL8GsO/+UkVIMmMR65FgaFhsUue+/90hIRBqRJGF0Xd
RbiW9dm8MNb/2vl2XAcuGxrvokWzdt1fullFJAUuwa/tKOWX8fXuThUjdfS0GpJS
saqlJf74z1R8FT+YTJe2JCOl4vk733ctUGsSYc9hvCgFRsu9e1d9Qvhe8wCg/7Zb
Pazd/WpLuqOFy1WiOzbuN4UEAOOQqo/3Z1/bZeEoCoQNIkSnHzybQLbaZlNlvhcY
IuslKshJT6Moi4ERPpSUJHRh/B+hHcDb8a0dFYdQ89v8EvSYmFW9hL7geH2Wh27z
4giDlTpIpsj+cbdbkp9JV7OYDjuFvblvr+T+r04+D9dUOjeFnVVkwyveYTHovwHh
W4ysA/4vMojMUw2M2hI2dzL6i+eL59przRP40s62G6tQUWvLw2+wlSGw4RncG0/p
xKNCm9Sry/EZimygFU8I6ymBWcZ6FLnjON2xyaVp+VV15nSFUhdA1xcSXY2y41zU
UtEg5706gy/K1On7D1nbrDN/yzjJZ+fLK61R5h3K5Icz6n7Sc7QrU3RldmUgU2Fu
Z3dpbmUgPFMuSi5TYW5nd2luZUBSZWFkaW5nLmFjLnVrPokASwQQEQIACwUCNYpQ
zAQLAwIBAAoJEEGGEV9WsZDtMdsAn1EcgHkoNWUqC6TUZPUQ3S/Az3vGAKDgOdXj
LbvPdh3j/UD8NoDvsqzKMbQpU3RlcGhlbiBKLiBTYW5nd2luZSA8Uy5TYW5nd2lu
ZUBJRUVFLm9yZz6JAEsEEBECAAsFAjU3GkAECwMCAQAKCRBBhhFfVrGQ7QygAKDC
SBp0U+00JiQBV6WHNWA4YrR6CQCgnliLaMqUuEHbvKs8KA3qs9LPkcu0JlN0ZXZl
IFNhbmd3aW5lIDxzanNhbmd3aW5lQGllZS5vcmcudWs+iQBLBBARAgALBQI1ilCg
BAsDAgEACgkQQYYRX1axkO3zJgCg8XMgd/+0+Us2zfsMDfYXdeiY9oQAoOvr7Xk5
k2OAw0qDU5TI7/5iVSAAuQINBDU3GkAQCAD2Qle3CH8IF3KiutapQvMF6PlTETlP
tvFuuUs4INoBp1ajFOmPQFXz0AfGy0OplK33TGSGSfgMg71l6RfUodNQ+PVZX9x2
Uk89PY3bzpnhV5JZzf24rnRPxfx2vIPFRzBhznzJZv8V+bv9kV7HAarTW56NoKVy
OtQa8L9GAFgr5fSI/VhOSdvNILSd5JEHNmszbDgNRR0PfIizHHxbLY7288kjwEPw
pVsYjY67VYy4XTjTNP18F1dDox0YbN4zISy1Kv884bEpQBgRjXyEpwpy1obEAxnI
Byl6ypUM2Zafq9AKUJsCRtMIPWakXUGfnHy9iUsiGSa6q6Jew1XpMgs7AAICB/9P
24ofRoqQVvyRv1julDbGThnmv7BJhxItoh5U1/MVksv9I6WktzgFWqMzSASoEzfs
tc2DSnnmKR9yIiX1jESFHHYkZE9ba6sPM1+de57p301isU6FaTcbcwHOv1lHXG0T
0xz4H4sBw0ZQ+3DzpMoXN248/BZWEaP96WyV1JGNEs9ijC4krDZKY8XwvGDWwc6E
t1XofqiOsR6+QkurMgdg9RrR20OlW4FEraiOn7/RMPFnlGAz/kcds7VwUuAfS1GK
zkyHDLH7NrE1Rtg4rLCHRbt++zIWN7ng7pMY0T3UwBEIQFR5+Xo2/2+MMOKpkYvX
jxu1OTEA+80diyGIGfOmiQBGBBgRAgAGBQI1NxpAAAoJEEGGEV9WsZDtLYkAnjDa
hdvZazzXkJ2ZUzNZpViY5AcqAJ988JCPvUTngaxYHvMx2V2LLdszYg==
=hhoI
-----END PGP PUBLIC KEY BLOCK-----
```

B Change history

Version 2.0 is the first release for use with FFTW version 3 and represents a substantial rewrite and restructuring of the binding to reflect the major changes made in FFTW. For details of the previous history, see the manual for FFTW_Ada version 1.2, which was the last version prior to version 2.

Version 2.0 α 1 is a minor change to the specification. The 2.0 α release included two types `int_Ptr` and `Kind_Ptr` which have now been deleted. Planner functions which used them now use access parameters instead (which is how the corresponding values were passed in version 1).

C GNU Free Documentation License

Version 1.2, November 2002

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

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.

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 for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

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.