Running Mono

Hinne Hettema

h.hettema@auckland.ac.nz

Jaime Anguiano Olarra

jaime@genuara.ugr.es

This document describes how to install mono on your Linux system from the source and describes some simple experiments you can perform with this installation.

Table of Contents

Introduction

Mono is an open source implementation of the Common Language Infrastructure (CLI) specified in ECMA standard 335. It comes with a C# compiler and provides an extensible platform to develop and run applications that are interoperable with Microsoft .NET.

Mono is not finished--it is a project still under development. As a result, installation and configuration may not be as smooth as you will be used to from other Linux applications. Nevertheless, mono is in a state that will allow you to get it up and running and gain experience with it--which is, I would suggest, a very smart thing to do.

This document is limited in the following sense: it describes how to download a 'stable' version of mono as a 'tarball' and get it to run on your computer. I'm also assuming you run mono on Linux rather than on Windows. All the examples have been tested on a 'vanilla' Red Hat 7.3 installation.

There are two items this document will *not* cover: the first is how to self host the mcs compiler under linux and the second is the graphical user environment which is implemented as GTK#. These two interesting topics are the subject of two more howto's that I'm planning.

This document is also less useful for two types of people.

- If you are a very inexperienced linux user and want to know what mono is all about, there is an easier way to install mono. Go to mono.baselabs.org and download the rpm's ready for installation on your system. This site also runs a tutorial that is worth reading.
- If you are a very experienced user wanting to contribute to the mono code, you should probably regularly do a CVS download. At the moment, the mono code grows very fast, and the 'stable' releases still appear too slowly.

This document is aimed at the mono beginner, and aims to present a complete view of a minimal installation, which will allow you to experiment with mono to some degree. It also describes some of those experiments. We expect that after reading this document you'll go on to do either of two things:

1. Continue to contribute to the mono project in some shape or form. The website has some ideas and suggestions under the heading 'Contributing'.
2. Continue to write applications that run in mono.

We hope this document will be useful to you in your first steps with mono. Happy hacking!

A note on the development of this document

This document is expected to grow into a full 'running mono howto' over time. As of yet, it does not really discuss some of the excellent efforts of others to package mono into an rpm or deb and prepare it for easy installation. It is our aim to include this in future versions of this document.

Prerequisites

This document assumes you are somewhat familiar with the architecture of the Common Language Infrastructure and the C# programming language.

To compile mono on your system, you will also have to satisfy some dependencies on other software. A list of these are given on the mono download page. Read through this list to see if you've got all components.

On a practical note, I had no dependency issues with the installation of mono 0.12 on a 'standard issue' Red Hat 7.3 build.

Disclaimer

No liability for the contents of this document can be accepted. Use the concepts, examples and information at your own risk. There may be errors and inaccuracies, that could be damaging to your system. Proceed with caution, and although this is highly unlikely, the author(s) do not take any responsibility.

Credits / Contributors

This document was originally created by Jaime Anguiano Olarra, but has been extensively rewritten. Credits are due to

- Jaime Anguiano Olarra <jaime@geneura.ugr.es> for creating the original version of this document, and in fact the document that got me through installing mono for the first time.
- My partner Margaret for support on my more exotic ventures.

Feedback

Feedback is most certainly welcome for this document. Send your additions, comments and criticisms to the following email address: <mono-docs-list@ximian.com>. I am lurking on this list and will maintain this document as required.

Installing Mono

Obtaining Mono

There are several ways to get Mono running on your computer. This document will discuss only one of those: downloading the source tarball and utilising the make utilities to get mono up and running. There is only one mono package to worry about if you are just after a 'base' install of mono: the mono runtime. This package, found under the name "mono-x.xx" has got a compiled version of the compiler built in.

You'll be able to complete everything in this document if you just install the runtime, but taking a look at the compiler package is well worth the effort. The compiler is written in C# and is 'self hosting' which means it is able to compile itself.

As of this moment, it is my understanding that there are still some issues with the 'self hosting' bit of the compiler on linux, although this is expected to work in later versions of mono. If you are just interested in finding out how mono will work under linux, I would not worry about the self-hosting bit for now. The self-hosting of the compiler will be the topic of a future howto.

See the mono download site[1] for the source of the tarballs.

This page also lists the current versions of the software mono depends on. Make sure that your system has all the required versions, otherwise mono won't compile.

At a minimum for mono 0.12, you'll need to:

- download and install pkg-config[2].
- download and install glib 1.3[3].

Installing the tarballs is done via GNU autoconf and automake. The general upshot of running autoconf and automake is that you can type **./configure** and then **make** to do the build. Typing **make install** completes the installation of mono on your system.

To build the mono runtime package, unzip the tarball to some useful location. I tend to use **/usr/src** as the location, but you could really do it anywhere. For the install process, it is also a good idea to be 'root'. The installation process will shuttle some executables into **/usr/local/bin** (more about that later).

Unzipping the tarballs will have created two directories in **/usr/src/**, one called something like mono-x.xx and the other mcs-x.xx. The one called 'mono' is the runtime environment, and this contains all that is initially needed to run mono and investigate its inner workings.

To start building the mono runtime, first type **./configure**. You will see an output like this:

```
[root@taurus mono-0.12]# ./configure
loading cache ./config.cache
checking host system type... i686-pc-linux-gnu
checking target system type... i686-pc-linux-gnu
checking build system type... i686-pc-linux-gnu
checking for a BSD compatible install... (cached) /usr/bin/install -c
checking whether build environment is sane... yes
checking whether make sets ${MAKE}... (cached) yes
checking for working aclocal... found


snip ....


checking BASE_DEPENDENCIES_CFLAGS...  -I/usr/include/glib-2.0 -I/usr/lib/glib-
2.0/include
checking BASE_DEPENDENCIES_LIBS...  -lglib-2.0
checking for GC_malloc in -lgc... (cached) no
configure: warning: Compiling mono without GC.
checking if off_t is 64 bits wide... no
checking if _FILE_OFFSET_BITS=64 gives 64 bit off_t... ok

snip ...

creating Makefile
creating mono/Makefile
creating mono/utils/Makefile
creating mono/metadata/Makefile
creating mono/dis/Makefile
creating mono/cil/Makefile
creating mono/arch/Makefile
creating mono/os/Makefile
creating mono/os/win32/Makefile
creating mono/os/unix/Makefile
creating mono/arch/x86/Makefile
creating mono/arch/ppc/Makefile
creating mono/arch/sparc/Makefile
creating mono/arch/arm/Makefile
creating mono/interpreter/Makefile
creating mono/tests/Makefile
creating mono/benchmark/Makefile
creating mono/monoburg/Makefile
creating mono/monograph/Makefile
```

```
creating mono/jit/Makefile
creating mono/io-layer/Makefile
creating mono/handles/Makefile
creating runtime/Makefile
creating scripts/Makefile
creating man/Makefile
creating doc/Makefile
creating docs/Makefile
creating config.h
config.h is unchanged


        GC:     auto


[root@taurus mono-0.12]#
```

This means that the configure script has now created all the makefiles necessary to complete the build on your system. Note the "GC: auto" at the end of the list of commands. GC stands for "garbage collection" and it will be addressed in a later section of this document.

You are now ready to start the mono "build" process. To kick this off, you can type **make** at the command prompt. You'll see something like this:

```
[root@taurus mono-0.12]# make
make  all-recursive
make[1]: Entering directory '/usr/src/mono-0.12'
Making all in mono
make[2]: Entering directory '/usr/src/mono-0.12/mono'
Making all in utils
make[3]: Entering directory '/usr/src/mono-0.12/mono/utils'
gcc -DHAVE_CONFIG_H -I. -I. -I../.. -I/usr/include/glib-2.0
-I/usr/lib/glib-2.0/include   -I/usr/include/glib-2.0
-I/usr/lib/glib-2.0/include   -I../.. -I../../mono
-D_FILE_OFFSET_BITS=64 -D_GNU_SOURCE   -g -Wall
-Wunused -Wmissing-prototypes -Wmissing-declarations
-Wstrict-prototypes  -Wmissing-prototypes -Wnested-externs
-Wpointer-arith -Wno-cast-qual -Wcast-align -Wwrite-strings
-c mono-hash.c
```

and this will continue for a while. On my 1 GHz Duron with 256 MB of RAM the total build process takes about three minutes. All the terrible flags after gcc have essentially been set by the configure script and there is no need to worry about these.

To finally install mono, you can type **make install** to get a working version of mono. This command will copy the scripts that run the executables to a place where the operating system will be ale to find them. This location is **/usr/local/bin** on my system.

Garbage Collection

To make the mint interpreter work with garbage collection, you first need to install a garbage collection package on your system. Mono works with Boehm Garbage collection, which is found at http://www.hpl.hp.com/personal/Hans_Boehm/gc/ [4].

To get garbage collection to work, I downloaded the source tarball into **/usr/src/** and typed **./configure** followed by **make** and **make install**.

There is one glitch to work around. The header file for the garbage collection **gc.h** is found in the **/usr/src/gc6.0/include** directory (at least on my system). The mono build process will break on this location: it is looking for the header file in some other place called **/usr/include/gc/** and won't be able to find the correct header file.

The solution is to build a symbolic link between these two locations as follows

```
[root@taurus include]# ln -s /usr/src/gc6.0/include/ /usr/include/gc
```

after which you'll be able to list the directory contents as if they were in **/usr/include**

```
[root@taurus include]# pwd
/usr/include
[root@taurus include]# ls /usr/include/gc
cord.h               gc_backptr.h  gc_inline.h             javaxfc.h
ec.h                 gc_cpp.h      gc_local_alloc.h        leak_detector.h
gc                   gc_gcj.h      gc_mark.h               new_gc_alloc.h
gc_alloc.h           gc.h          gc_pthread_redirects.h  private
gc_amiga_redirects.h gc_inl.h      gc_typed.h              weakpointer.h
[root@taurus include]#
```

To get mono to work with garbage collection, you'll have to rebuild the environment. This is done easily by typing **make distclean** to clean up any existing executables.

Continue the build by typing **./configure**, and look at the last line. It should read:

```
snip...

creating config.h


        GC:     boehm


[root@taurus mono-0.12]#
```

The GC: boehm tells you that garbage collection has been turned on. Now to compile the runtime, you type **make** and **make install** as previously.

You now have mono working with Garbage Collection.

Although my experimentation is incomplete as of this time, I have found some issues with running Garbage Collection continuously, and my recommendation would be to turn it off for now.

To turn garbage collection off again, you have to go to the **/usr/src** directory where you extracted the tarball. Enter the **gc6.0** directory and type **make uninstall** to uninstall Garbage Collection. Then in your **mono** directory, type **make distclean** followed by **./configure**, **make** and **make install**.

Running mono

Basic steps

To work with mono, you first have to create a C# program. Open up your favourite editor, and type in the following code:

```
using System;

class Hello
{
public static void Main(String[] args)
 {
 Console.WriteLine("mono:: is alive and well...");

 for (int i = 0; i < args.Length; i++)
  Console.WriteLine("Argument {0} = {1}", i, args[i]);
 }
```

```
}
```

Save the file as hello.cs. To compile this into a working program, type mcs hello.cs. If you get the following:

```
[hinne@taurus hello]$ mcs hello.cs
RESULT: 0
[hinne@taurus hello]$
```

you know the compile worked fine. If you see some strange error messages including the word 'parser' somewhere, you made a mistake in your program. Fix this up first.

You are now ready to execute your first mono program. To execute the code, type

```
[hinne@taurus hello]$ mono hello.exe arg1 arg2 arg 3
```

(where we have given some arguments just for fun) and you'll see the following:

```
mono:: is alive and well...
Argument 0 = arg1
Argument 1 = arg2
Argument 2 = arg
Argument 3 = 3
RESULT: 0
```

As you can see, mono printed the line "mono:: is alive and well" and printed the list of arguments. This completes the creation and execution of your first mono program.

Interpreter

But mono will allow you to do more. First of all, mono is the compiled mono execution environment which uses the Just in Time (JIT) compiler. Mono also comes with an interpreted environment, which can be accessed using the command 'mint' as follows

```
[hinne@taurus hello]$ mint hello.exe arg1 arg 2
mono:: is alive and well...
Argument 0 = arg1
Argument 1 = arg
Argument 2 = 2
[hinne@taurus hello]$
```

As you can see, it makes no difference to mono output which environment you use, but what happens under the hood is very different. If you use 'mono' as the command line tool, you call the 'production' execution environment which will read your portable executable (PE) file, and call the just in time (JIT) compiler to compile the PE code down to machine level code (in my case, an x86 architecture) after which it is executed.

If you use mint, the JIT is not used, and the PE code is interpreted into x86 instructions for execution. In fact, for our simple 'hello' mint is slightly faster. The point is that the JIT compiler will take some time to compile the code of our program and store it in some location in memory, but the subsequent execution of the code is faster with mono.

You can see what happens below (the thing to look for is the 'user' time: 0.1 seconds with mono and 0.06 seconds with mint):

```
[hinne@taurus hello]$ time mono hello.exe arg1 arg 2
mono:: is alive and well...
Argument 0 = arg1
Argument 1 = arg
Argument 2 = 2
```

```
RESULT: 0

real    0m0.575s
user    0m0.100s
sys     0m0.010s
[hinne@taurus hello]$ time mint hello.exe arg1 arg 2
mono:: is alive and well...
Argument 0 = arg1
Argument 1 = arg
Argument 2 = 2

real    0m0.545s
user    0m0.060s
sys     0m0.000s
[hinne@taurus hello]$
```

After this simple run of mono, it is time to play with some options. I won't cover these in detail since there are quite a few, and also because I assume you downloaded mono to hack it around in the first place. So I'll leave some pointers.

Debugging

Mono supports a debugging option when you specify the "-d" flag while running the runtime. Utilising this flag will get you a significant amount of output, and it may be an idea to specify an output file as well. The interesting aspect of this file is that it allows you to see to some extent (quite precisely, actually) what the JIT compiler is up to.

Statistics

It is also possible to collect some runtime statistics on your program. These will give you some idea of the resource utilisation of your program.

```
[hinne@taurus hello]$ mono --stats hello.exe
mono:: is alive and well...
RESULT: 0
Mono Jit statistics
Compiled methods:       58
Methods cache lookup:   15
Method trampolines:     698
Basic blocks:           188
Max basic blocks:       15
Allocated vars:         238
Analyze stack repeat:   61
Compiled CIL code size: 2450
Native code size:       10167
Max code size ratio:    7.13 (FileStream::FlushBuffer)
Biggest method:         1016 (StreamWriter::Write)
Code reallocs:          27
Allocated code size:    22073
Inlineable methods:     17
Inlined methods:        22

Created object count:   18
Initialized classes:    127
Used classes:           37
Static data size:       288
VTable data size:       8292
```

Inspecting IL Assembly code

Mono also provides a small tool that will let you disassemble the executable (.exe) file so you can have a peek under the hood. This tool is monodis, and is run as follows:

```
[hinne@taurus hello]$ monodis hello.exe
.assembly extern mscorlib
{
  .ver 0:0:0:0
}
.assembly 'hello'
{
  .hash algorithm 0x00008004
  .ver  0:0:0:0
}
  .class private auto ansi beforefieldinit Hello
        extends [mscorlib]System.Object
  {

    // method line 1
    .method public hidebysig  specialname  rtspecialname
           instance default void .ctor()  cil managed
    {
        // Method begins at RVA 0x20ec
        // Code size 7 (0x7)
        .maxstack 8
        IL_0000: ldarg.0
        IL_0001: call instance void System.Object::.ctor()
        IL_0006: ret
    } // end of method instance default void .ctor()

    // method line 2
    .method public static
           default void Main(string[] args)  cil managed
    {
        // Method begins at RVA 0x20f4
        .entrypoint
        // Code size 56 (0x38)
        .maxstack 5
        .locals init (
                int32   V_0,
                int32   V_1)
        IL_0000: ldstr "mono:: is alive and well..."
        IL_0005: call void System.Console::WriteLine(string)
        IL_000a: ldc.i4.0
        IL_000b: stloc.0
        IL_000c: ldloc.0
        IL_000d: ldarg.s 0
        IL_000f: ldlen
        IL_0010: clt
        IL_0012: brfalse IL_0037

        IL_0017: ldstr "Argument {0} = {1}"
        IL_001c: ldloc.0
        IL_001d: box [mscorlib]System.Int32
        IL_0022: ldarg.s 0
        IL_0024: ldloc.0
        IL_0025: ldelem.ref
        IL_0026: call void System.Console::WriteLine(string, object, object)
        IL_002b: nop
        IL_002c: ldloc.0
        IL_002d: ldc.i4.1
        IL_002e: add
        IL_002f: stloc.1
        IL_0030: ldloc.1
        IL_0031: stloc.0
```

```
        IL_0032: br IL_000c

        IL_0037: ret
    } // end of method default void Main(string[] args)

  } // end of type Hello

[hinne@taurus hello]$
```

This is the listing of the code of your program in a language called IL assembly, or Common Intermediate Language (CIL). The CIL provides the portability of the mono platform, and ensures that code compiled with Microsoft's .NET framework will work on mono and vice versa.

Man pages

Mono has man pages already installed, and generally, typing **man** before the command you wish to execute should help in getting a list of the options. In many cases, it's still up to you to figure out what they do.

Problems

With mono being as new as it is, it is likely that you will have some problems with installation. The following are some ideas to help you out in a bind:

1. See the Ximian Bugzilla page to find out if there is a bug report about your specific issue.
2. Read this document. If it does not solve your problem, we want to know about it. Please send a message to the email address listed for feedback at the beginning of the document.
3. Visit the mono mailing lists' archives and do a little research in there for threads talking about the problem you have.
4. If you still cannot correct the problem, send a message to the mono list. When you do this, please be as precise as possible--i.e. mention the system you are running, the version of mono that you have the problem with, and give any error codes and other output that might appear.

Known Problems

mcs fails to compile in Linux. To the best of my knowledge, as of yet mcs cannot compile in Linux. Try to install the already made packages from http://mono.baselabs.org/index.php/software.

Copyright and License

GNU Free Documentation License

Version 1.1, March 2000

PREAMBLE

The purpose of this License is to make a manual, textbook, or other written 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.

APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you".

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. (For example, 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.

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 "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, whose contents can be viewed and edited directly and 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 has been designed to thwart or discourage subsequent modification by readers is not Transparent. 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 designed for human modification. Opaque formats include PostScript, PDF, 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 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.

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.

COPYING IN QUANTITY

If you publish printed copies 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 publicly-accessible computer-network location containing a complete Transparent copy of the Document, free of added material, which the general network-using public has access to download anonymously at no charge using public-standard network protocols. 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.

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 less than five).

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", and 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. In any section entitled "Acknowledgements" or "Dedications", preserve the section's title, 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 as "Endorsements" or to conflict in title with any Invariant Section.

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.

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.

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."

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.

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, does not as a whole count as a Modified Version of the Document, provided no compilation copyright is claimed for the compilation. Such a compilation is called an "aggregate", and this License does not apply to the other self-contained works thus compiled with the Document, on account of their being thus compiled, if they 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 quarter of the entire aggregate, the Document's Cover Texts may be placed on covers that surround only the Document within the aggregate. Otherwise they must appear on covers around the whole aggregate.

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 provided that you also include the original English version of this License. In case of a disagreement between the translation and the original English version of this License, the original English version will prevail.

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.

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.

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 1.1 or any later version published by the Free Software Foundation; with the Invariant

Notes

1. http://www.go-mono.com/download
2. http//www.freedesktop.org/software/pkgconfig
3. ftp://ftp.gtk.org/pub/gtk/v1.3/glib-1.3.12.tar.gz
4. http://www.hpl.hp.com/personal/Hans_Boehm/gc/
5. http://www.gnu.org/copyleft/