

---

# **Sourcery G++ Lite**

**ARM GNU/Linux**

**2007q1-21**

**Getting Started**



## **Sourcery G++ Lite: ARM GNU/Linux: 2007q1-21: Getting Started**

CodeSourcery, Inc.

Copyright © 2005, 2006, 2007 CodeSourcery, Inc.

All rights reserved.

---

# Table of Contents

Preface .....	ix
1. Intended Audience .....	x
2. Organization .....	x
3. Typographical Conventions .....	xi
1. Sourcery G++ Lite Licenses .....	1
1.1. Overview .....	2
1.2. Sourcery G++ <sup>TM</sup> Software License Agreement .....	2
2. Sourcery G++ Subscriptions .....	7
2.1. About Sourcery G++ Subscriptions .....	8
2.2. Accessing your Sourcery G++ Subscription Account .....	8
3. Sourcery G++ Lite for ARM GNU/Linux .....	9
3.1. Using Sourcery G++ Lite for ARM GNU/Linux .....	10
3.1.1. Target Kernel Requirements .....	10
3.1.2. Compiling for ARMv4t systems .....	10
3.1.3. NEON SIMD Code .....	10
3.2. Sourcery G++ Lite Release Notes .....	10
3.2.1. Changes in Sourcery G++ Lite 2007q1-21 .....	11
3.2.2. Changes in Sourcery G++ Lite 2007q1-10 .....	11
3.2.3. Changes in Sourcery G++ Lite 2007q1-3 .....	12
3.2.4. Changes in Sourcery G++ Lite 4.1-37 .....	14
3.2.5. Changes in Sourcery G++ Lite 4.1-34 .....	15
3.2.6. Changes in Sourcery G++ Lite 4.1-33 .....	15
3.2.7. Changes in Sourcery G++ Lite 4.1-31 .....	15
3.2.8. Changes in Sourcery G++ Lite 4.1-29 .....	15
3.2.9. Changes in Sourcery G++ Lite 4.1-28 .....	16
3.2.10. Changes in Sourcery G++ Lite 4.1-27 .....	16
3.2.11. Changes in Sourcery G++ Lite 4.1-23 .....	16
3.2.12. Changes in Sourcery G++ Lite 4.1-21 .....	16
3.2.13. Changes in Sourcery G++ Lite 4.1-19 .....	16
3.2.14. Changes in Sourcery G++ Lite 4.1-18 .....	17
3.2.15. Changes in Sourcery G++ Lite 4.1-16 .....	17
3.2.16. Changes in Sourcery G++ Lite 4.1-15 .....	17
3.2.17. Changes in Sourcery G++ Lite 4.1-13 .....	17
3.2.18. Changes in Sourcery G++ Lite 4.1-9 .....	18
3.2.19. Changes in Sourcery G++ Lite 4.1-8 .....	18
3.2.20. Changes in Sourcery G++ Lite 4.1-4 .....	18
3.2.21. Changes in Sourcery G++ Lite 4.1-1 .....	18
4. Installation and Configuration .....	19
4.1. Terminology .....	20
4.2. System Requirements .....	20
4.2.1. Host Operating System Requirements .....	20
4.2.2. Host Hardware Requirements .....	20
4.2.3. Target System Requirements .....	20
4.3. Downloading an Installer .....	20
4.4. Installing Sourcery G++ Lite .....	21
4.4.1. Installing Sourcery G++ Lite on Microsoft Windows .....	21
4.4.2. Installing Sourcery G++ Lite on GNU/Linux systems with an X Window System .....	21
4.4.3. Installing Sourcery G++ Lite on Solaris or GNU/Linux systems without an X Window System .....	21

4.4.4. Installing Sourcery G++ Lite on RPM-based GNU/Linux sys-	
tems .....	22
4.4.5. Installing the Java Runtime Environment .....	22
4.5. Uninstalling Sourcery G++ Lite .....	22
4.5.1. Uninstalling Sourcery G++ Lite on Microsoft Windows .....	23
4.5.2. Uninstalling Sourcery G++ Lite on Microsoft Windows Vista .....	23
4.5.3. Uninstalling Sourcery G++ Lite on GNU/Linux using the graphical	
uninstaller .....	23
4.5.4. Uninstalling Sourcery G++ Lite on RPM-based GNU/Linux sys-	
tems .....	23
4.5.5. Uninstalling Sourcery G++ Lite on GNU/Linux .....	23
4.6. Setting up the Environment .....	23
4.6.1. Setting up the Environment on Microsoft Windows .....	23
4.6.2. Setting up the Environment on GNU/Linux or Solaris .....	24
5. Using the Sourcery G++ Lite IDE .....	26
5.1. Overview .....	27
5.2. Building Applications .....	27
5.2.1. Setting Up .....	27
5.2.2. Writing Source Code .....	29
5.2.3. Using Cross-Reference Information .....	30
5.2.4. Dealing with Errors .....	31
5.2.5. Using Standard Make Mode .....	31
5.3. Debugging Applications .....	32
5.3.1. Starting the Debugger .....	32
5.3.2. Choosing a Debugging Mode .....	33
5.3.3. Controlling Execution .....	34
5.3.4. Low-Level Debugging .....	36
6. Using Sourcery G++ from the Command Line .....	38
6.1. Building an Application .....	39
6.2. Running an Application .....	39
A. GNU General Public License .....	41
B. GNU Lesser General Public License .....	47
B.1. Preamble .....	47
B.2. TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND	
MODIFICATION .....	49
B.2.1. Section 0 .....	49
B.2.2. Section 1 .....	49
B.2.3. Section 2 .....	49
B.2.4. Section 3 .....	50
B.2.5. Section 4 .....	50
B.2.6. Section 5 .....	51
B.2.7. Section 6 .....	51
B.2.8. Section 7 .....	52
B.2.9. Section 8 .....	52
B.2.10. Section 9 .....	53
B.2.11. Section 10 .....	53
B.2.12. Section 11 .....	53
B.2.13. Section 12 .....	53
B.2.14. Section 13 .....	54
B.2.15. Section 14 .....	54
B.2.16. NO WARRANTY Section 15 .....	54
B.2.17. Section 16 .....	54
B.3. How to Apply These Terms to Your New Libraries .....	55
C. GNU Free Documentation License .....	56

C.1. PREAMBLE .....	56
C.2. APPLICABILITY AND DEFINITIONS .....	56
C.3. VERBATIM COPYING .....	58
C.4. COPYING IN QUANTITY .....	58
C.5. MODIFICATIONS .....	58
C.6. COMBINING DOCUMENTS .....	60
C.7. COLLECTIONS OF DOCUMENTS .....	60
C.8. AGGREGATION WITH INDEPENDENT WORKS .....	61
C.9. TRANSLATION .....	61
C.10. TERMINATION .....	61
C.11. FUTURE REVISIONS OF THIS LICENSE .....	61
C.12. ADDENDUM: How to use this License for your documents .....	62

---

## List of Figures

5.1. Creating a Project .....	28
5.2. Adding a Source File .....	29
5.3. Using the Outline .....	30
5.4. Viewing Errors .....	31
5.5. Creating a Launch Configuration .....	32
5.6. Selecting a Program .....	33
5.7. Selecting a Debugger .....	34
5.8. Debug Perspective .....	35
5.9. Setting a Breakpoint .....	36
5.10. Low-Level Debugging .....	37

---

## List of Tables

1.1. Software Licenses .....	2
------------------------------	---

---

## List of Examples

5.1. Factorial Application .....	29
6.1. Hello, World (C) .....	39



---

# Preface

This preface introduces *Getting Started With Sourcery G++ Lite*. It explains the structure of this guide and lists other sources of information that relate to Sourcery G++ Lite.

# 1 Intended Audience

This guide is written for people who will install and/or use Sourcery G++ Lite. This guide provides a step-by-step guide to installing Sourcery G++ Lite and to building simple applications. Parts of this document assume that you have some familiarity with using the command-line interface. If you are an administrator installing Sourcery G++ Lite on a UNIX-like system for all of your users to use, you should also be familiar with the package-management software (such as the Red Hat Package Manager) for your system.

# 2 Organization

This document is organized into the following chapters and appendices:

Chapter 1, <i>Sourcery G++ Lite Licenses</i>	This chapter provides information about the software licenses that apply to Sourcery G++ Lite. Read this chapter to understand your legal rights and obligations as a user of Sourcery G++ Lite.
Chapter 2, <i>Sourcery G++ Subscriptions</i>	This chapter provides information about Sourcery G++ Lite subscriptions. CodeSourcery customers with Sourcery G++ Lite subscriptions receive comprehensive support for Sourcery G++ Lite. Read this chapter to find out how to obtain and use a Sourcery G++ Lite subscription.
Chapter 3, <i>Sourcery G++ Lite for ARM GNU/Linux</i>	This chapter provides information about this release of Sourcery G++ Lite including any special installation instructions, recent improvements, or other similar information. You should read this chapter before building applications with Sourcery G++ Lite.
Chapter 4, <i>Installation and Configuration</i>	This chapter describes how to download, install and configure Sourcery G++ Lite. This section describes the available installation options and explains how to set up your environment so that you can build applications.
Chapter 5, <i>Using the Sourcery G++ Lite IDE</i>	This chapter explains how to use the Sourcery G++ Lite IDE, which is based on Eclipse.
Chapter 6, <i>Using Sourcery G++ from the Command Line</i>	This chapter explains how to build applications with Sourcery G++ Lite using the command line. In the process of reading this chapter, you will build a simple application that you can use as a model for your own programs.
Appendix A, <i>GNU General Public License</i>	This appendix contains the full text of the GNU General Public License, the license that applies to certain components of Sourcery G++ Lite.

## 3 Typographical Conventions

The following typographical conventions are used in this guide:

`> command arg ...` A command, typed by the user, and its output. The “>” character is the command prompt.

**command** The name of a program, when used in a sentence, rather than in literal input or output.

`literal` Text provided to or received from a computer program.

*placeholder* Text that should be replaced with an appropriate value when typing a command.

---

# **Chapter 1**

## **Sourcery G++ Lite Licenses**

Sourcery G++ Lite contains software provided under a variety of licenses. Some components are "free" or "open source" software, while other components are proprietary. This chapter explains what licenses apply to your use of Sourcery G++ Lite. You should read this chapter to understand your legal rights and obligations as a user of Sourcery G++ Lite.

## 1.1 Overview

Sourcery G++ Lite may include any or all of the following components. Please consult the table below for the license terms applicable to each component.

Component	License
GNU Compiler Collection	GNU General Public License
GNU Binary Utilities	GNU General Public License
GNU Debugger	GNU General Public License
GNU Make	GNU General Public License
Newlib C Library	Newlib License
uClibc Library	GNU Lesser General Public License
Eclipse IDE	Eclipse Public License
GNU Manuals	GNU Free Documentation License
Sourcery G++ Lite Plug-in for Eclipse IDE	CodeSourcery License
Sourcery G++ Lite Debug Stub for ColdFire	CodeSourcery License
Sourcery G++ Lite USB Debug Stub for Stellaris	CodeSourcery License

The GNU Public License is available in Appendix A, *GNU General Public License*. The GNU Lesser General Public License is available in Appendix B, *GNU Lesser General Public License*. The Newlib License is available at <http://sources.redhat.com/newlib/COPYING.NEWLIB>. The Eclipse Public License is available at <http://www.eclipse.org/org/documents/epl-v10.php>. The GNU Free Documentation License is available in Appendix C, *GNU Free Documentation License*.

The CodeSourcery License is available in Section 1.2, "Sourcery G++™ Software License Agreement".

### Important

Although some of the licenses that apply to Sourcery G++ Lite are "free software" or "open source software" licenses, none of these licenses impose any obligation on you to reveal the source code of applications you build with Sourcery G++ Lite. You can develop proprietary applications and libraries with Sourcery G++ Lite.

## 1.2 Sourcery G++™ Software License Agreement

- Parties.** The parties to this Agreement are you, the licensee ("You" or "Licensee") and CodeSourcery. If You are not acting on behalf of Yourself as an individual, then "You" means Your company or organization.
- The Software.** The Software licensed under this Agreement consists of computer programs and documentation referred to as Sourcery G++™ Lite Edition (the "Software").
- Definitions.**

- 3.1. **CodeSourcery Proprietary Components.** The components of the Software that are owned and/or licensed by CodeSourcery and are not subject to a "free software" or "open source" license, such as the GNU Public License. The CodeSourcery Proprietary Components of the Software include, without limitation, the Sourcery G++ Installer, any Sourcery G++ Eclipse plug-ins, and any Sourcery G++ Debug Sprite. For a complete list, refer to the "Getting Started Guide" included with this distribution.
- 3.2. **Open Source Software Components.** The components of the Software that are subject to a "free software" or "open source" license, such as the GNU Public License.
- 3.3. **Proprietary Rights.** All rights in and to copyrights, rights to register copyrights, trade secrets, inventions, patents, patent rights, trademarks, trademark rights, confidential and proprietary information protected under contract or otherwise under law, and other similar rights or interests in intellectual or industrial property.
4. **License Grant to Proprietary Components of the Software.** You are granted a non-exclusive, royalty-free license to install and use the CodeSourcery Proprietary Components of the Software, transmit the CodeSourcery Proprietary Components over an internal computer network, and/or copy the CodeSourcery Proprietary Components for Your internal use only.
5. **Restrictions.** You may not: (i) copy or permit others to use the CodeSourcery Proprietary Components of the Software, except as expressly provided above; (ii) distribute the CodeSourcery Proprietary Components of the Software to any third party; or (iii) reverse engineer, decompile, or disassemble the CodeSourcery Proprietary Components of the Software, except to the extent this restriction is expressly prohibited by applicable law.
6. **"Free Software" or "Open Source" License to Certain Components of the Software.** This Agreement does not limit Your rights under, or grant You rights that supersede, the license terms of any Open Source Software Component delivered to You by CodeSourcery. For a list of which license applies to each component, refer to the "Getting Started Guide" included with this distribution.
7. **CodeSourcery Trademarks.** Notwithstanding any provision in a "free software" or "open source" license agreement applicable to a component of the Software that permits You to distribute such component to a third party in source or binary form, You may not use any CodeSourcery trademark, whether registered or unregistered, including without limitation, CodeSourcery™, Sourcery G++™, the CodeSourcery crystal ball logo, or the Sourcery G++ splash screen, or any confusingly similar mark, in connection with such distribution, and You may not recompile the Open Source Software Components with the `--with-pkgversion` or `--with-bugurl` configuration options that embed CodeSourcery trademarks in the resulting binary.
8. **Term and Termination.** This Agreement shall remain in effect unless terminated pursuant to this provision. CodeSourcery may terminate this Agreement upon seven (7) days written notice of a material breach of this Agreement if such breach is not cured; provided that the unauthorized use, copying, or distribution of the CodeSourcery Proprietary Components of the Software will be deemed a material breach that cannot be cured.
9. **Transfers.** You may not transfer any rights under this Agreement without the prior written consent of CodeSourcery, which consent shall not be unreasonably withheld. A condition to any transfer or assignment shall be that the recipient agrees to the terms of this Agreement. Any attempted transfer or assignment in violation of this provision shall be null and void.
10. **Ownership.** CodeSourcery owns and/or has licensed the CodeSourcery Proprietary Components of the Software and all intellectual property rights embodied therein, including copyrights

and valuable trade secrets embodied in its design and coding methodology. The CodeSourcery Proprietary Components of the Software are protected by United States copyright laws and international treaty provisions. CodeSourcery also owns all rights, title and interest in and with respect to its trade names, domain names, trade dress, logos, trademarks, service marks, and other similar rights or interests in intellectual property. This Agreement provides You only a limited use license, and no ownership of any intellectual property.

11. **Warranty Disclaimer; Limitation of Liability.** CODESOURCERY AND ITS LICENSORS PROVIDE THE SOFTWARE "AS-IS" AND PROVIDED WITH ALL FAULTS. CODESOURCERY DOES NOT MAKE ANY WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. CODESOURCERY SPECIFICALLY DISCLAIMS THE IMPLIED WARRANTIES OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, SYSTEM INTEGRATION, AND DATA ACCURACY. THERE IS NO WARRANTY OR GUARANTEE THAT THE OPERATION OF THE SOFTWARE WILL BE UNINTERRUPTED, ERROR-FREE, OR VIRUS-FREE, OR THAT THE SOFTWARE WILL MEET ANY PARTICULAR CRITERIA OF PERFORMANCE, QUALITY, ACCURACY, PURPOSE, OR NEED. YOU ASSUME THE ENTIRE RISK OF SELECTION, INSTALLATION, AND USE OF THE SOFTWARE. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS AGREEMENT. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.
12. **Local Law.** If implied warranties may not be disclaimed under applicable law, then ANY IMPLIED WARRANTIES ARE LIMITED IN DURATION TO THE PERIOD REQUIRED BY APPLICABLE LAW.
13. **Limitation of Liability.** INDEPENDENT OF THE FORGOING PROVISIONS, IN NO EVENT AND UNDER NO LEGAL THEORY, INCLUDING WITHOUT LIMITATION, TORT, CONTRACT, OR STRICT PRODUCTS LIABILITY, SHALL CODESOURCERY BE LIABLE TO YOU OR ANY OTHER PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY KIND, INCLUDING WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER MALFUNCTION, OR ANY OTHER KIND OF COMMERCIAL DAMAGE, EVEN IF CODESOURCERY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY TO THE EXTENT PROHIBITED BY APPLICABLE LAW. IN NO EVENT SHALL CODESOURCERY'S LIABILITY FOR ACTUAL DAMAGES FOR ANY CAUSE WHATSOEVER, AND REGARDLESS OF THE FORM OF ACTION, EXCEED THE AMOUNT PAID BY YOU IN FEES UNDER THIS AGREEMENT DURING THE PREVIOUS ONE YEAR PERIOD.
14. **Export Controls.** You agree to comply with all export laws and restrictions and regulations of the United States or foreign agencies or authorities, and not to export or re-export the Software or any direct product thereof in violation of any such restrictions, laws or regulations, or without all necessary approvals. As applicable, each party shall obtain and bear all expenses relating to any necessary licenses and/or exemptions with respect to its own export of the Software from the U.S. Neither the Software nor the underlying information or technology may be electronically transmitted or otherwise exported or re-exported (i) into Cuba, Iran, Iraq, Libya, North Korea, Sudan, Syria or any other country subject to U.S. trade sanctions covering the Software, to individuals or entities controlled by such countries, or to nationals or residents of such countries other than nationals who are lawfully admitted permanent residents of countries not subject to such sanctions; or (ii) to anyone on the U.S. Treasury Department's list of Specially Designated Nationals and Blocked Persons or the U.S. Commerce Department's Table of Denial Orders. By downloading or using the Software, Licensee agrees to the foregoing and represents and warrants that it complies with these conditions.

15. **U.S. Government End-Users.** The Software is a "commercial item," as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of "commercial computer software" and "commercial computer software documentation," as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire the Software with only those rights set forth herein.
16. **Licensee Outside The U.S.** If You are located outside the U.S., then the following provisions shall apply: (i) Les parties aux presentes confirment leur volonte que cette convention de meme que tous les documents y compris tout avis qui siy rattache, soient rediges en langue anglaise (translation: "The parties confirm that this Agreement and all related documentation is and will be in the English language."); and (ii) You are responsible for complying with any local laws in your jurisdiction which might impact your right to import, export or use the Software, and You represent that You have complied with any regulations or registration procedures required by applicable law to make this license enforceable.
17. **Severability.** If any provision of this Agreement is declared invalid or unenforceable, such provision shall be deemed modified to the extent necessary and possible to render it valid and enforceable. In any event, the unenforceability or invalidity of any provision shall not affect any other provision of this Agreement, and this Agreement shall continue in full force and effect, and be construed and enforced, as if such provision had not been included, or had been modified as above provided, as the case may be.
18. **Arbitration.** Except for actions to protect intellectual property rights and to enforce an arbitrator's decision hereunder, all disputes, controversies, or claims arising out of or relating to this Agreement or a breach thereof shall be submitted to and finally resolved by arbitration under the rules of the American Arbitration Association ("AAA") then in effect. There shall be one arbitrator, and such arbitrator shall be chosen by mutual agreement of the parties in accordance with AAA rules. The arbitration shall take place in Granite Bay, California, and may be conducted by telephone or online. The arbitrator shall apply the laws of the State of California, USA to all issues in dispute. The controversy or claim shall be arbitrated on an individual basis, and shall not be consolidated in any arbitration with any claim or controversy of any other party. The findings of the arbitrator shall be final and binding on the parties, and may be entered in any court of competent jurisdiction for enforcement. Enforcements of any award or judgment shall be governed by the United Nations Convention on the Recognition and Enforcement of Foreign Arbitral Awards. Should either party file an action contrary to this provision, the other party may recover attorney's fees and costs up to \$1000.00.
19. **Jurisdiction And Venue.** The courts of Placer County in the State of California, USA and the nearest U.S. District Court shall be the exclusive jurisdiction and venue for all legal proceedings that are not arbitrated under this Agreement.
20. **Independent Contractors.** The relationship of the parties is that of independent contractor, and nothing herein shall be construed to create a partnership, joint venture, franchise, employment, or agency relationship between the parties. Licensee shall have no authority to enter into agreements of any kind on behalf of CodeSourcery and shall not have the power or authority to bind or obligate CodeSourcery in any manner to any third party.
21. **Force Majeure.** Neither CodeSourcery nor Licensee shall be liable for damages for any delay or failure of delivery arising out of causes beyond their reasonable control and without their fault or negligence, including, but not limited to, Acts of God, acts of civil or military authority, fires, riots, wars, embargoes, or communications failure.
22. **Miscellaneous.** This Agreement constitutes the entire understanding of the parties with respect to the subject matter of this Agreement and merges all prior communications, representations, and agreements. This Agreement may be modified only by a written agreement signed by the



parties. If any provision of this Agreement is held to be unenforceable for any reason, such provision shall be reformed only to the extent necessary to make it enforceable. This Agreement shall be construed under the laws of the State of California, USA, excluding rules regarding conflicts of law. The application of the United Nations Convention of Contracts for the International Sale of Goods is expressly excluded. This license is written in English, and English is its controlling language.

---

## **Chapter 2**

# **Sourcery G++ Subscriptions**

CodeSourcery provides support contracts for Sourcery G++. This chapter describes these contracts and explains how CodeSourcery customers can access their support accounts.

## 2.1 About Sourcery G++ Subscriptions

CodeSourcery offers Sourcery G++ subscriptions. Professional Edition subscriptions provide unlimited support, with no per-incident fees. CodeSourcery's support covers questions about installing and using Sourcery G++, the C and C++ programming languages, and all other topics relating to Sourcery G++. CodeSourcery provides updated versions of Sourcery G++ to resolve critical problems. Personal Edition subscriptions do not include support, but do include free upgrades as long as the subscription remains active.

CodeSourcery's support is provided by the same engineers who build Sourcery G++. A Sourcery G++ subscription is like having a team of compiler engineers and programming language experts available as consultants!

If you would like more information about Sourcery G++ subscriptions, including a price quote or information about evaluating Sourcery G++, please send email to <sales@codesourcery.com>.

## 2.2 Accessing your Sourcery G++ Subscription Account

If you have a Sourcery G++ subscription, you may access your account by visiting the Sourcery G++ Portal<sup>1</sup>. If you have a support account, but are unable to log in, send email to <support@codesourcery.com>.

---

<sup>1</sup> <https://support.codesourcery.com/GNUToolchain/>

---

## **Chapter 3**

# **Sourcery G++ Lite for ARM GNU/Linux**

This chapter contains information about using Sourcery G++ Lite on your target system. This chapter also contains information about changes in this release of Sourcery G++ Lite. You should read this chapter to learn how to best use Sourcery G++ Lite on your target system.

## 3.1 Using Sourcery G++ Lite for ARM GNU/Linux

### 3.1.1 Target Kernel Requirements

The GNU C library supplied with Sourcery G++ Lite uses the new EABI-based kernel syscall interface. This means applications compiled with Sourcery G++ Lite require at least a 2.6.16 kernel with EABI syscalls enabled.

### 3.1.2 Compiling for ARMv4t systems

By default Sourcery G++ Lite generates Linux binaries that require an ARMv5 or later CPU. To build applications or libraries capable of running on ARMv4t CPUs, use the `-march=armv4t` command-line option.

Runtime libraries suitable for ARMv4t systems are supplied in the `armv4t` subdirectory.

Code compiled for ARMv4t is ABI compatible with ARMv5 code. Code and binaries compiled for different architectures may be mixed freely.

#### Caution

There are several other ways to tell the compiler to generate ARMv4t code. However `-march=armv4t` must be used when linking to ensure the correct libraries and startup code are selected.

### 3.1.3 NEON SIMD Code

Sourcery G++ Lite contains preliminary support for automatic generation of NEON SIMD vector code. Autovectorization is a compiler optimization where loops involving normal integer or floating-point code are transformed into loops that use NEON SIMD instruction to process several data elements at once.

To enable generation of NEON vector code specify `-ftree-vectorize -mfpu=neon -mfloat-abi=softfp`. `-mfpu=neon` also enables generations of VFPv3 scalar floating-point code.

Sourcery G++ Lite also contains preliminary support for manual generation of NEON SIMD code using C intrinsic functions. These intrinsics, the same as those supported by the ARM RVCT compiler, are defined in the `arm_neon.h` header and are documented in the 'ARM NEON Intrinsics' section of the GCC manual. The options `-mfpu=neon -mfloat-abi=softfp` must be specified to use these intrinsics; `-ftree-vectorize` is not required.

NEON support is still under active development. It has not been subject to extensive testing, and may not yet take full advantage of all the features provided by the NEON architecture.

## 3.2 Sourcery G++ Lite Release Notes

This section documents Sourcery G++ Lite changes for each released revision.

### 3.2.1 Changes in Sourcery G++ Lite 2007q1-21

**NEON coprocessor system registers.** The assembler now accepts the MVFR0 and MVFR1 coprocessor registers in `fmxr` and `fmxr` instructions.

**Disabling diagnostics for use of system header and library directories.** The warnings for use of options such as `-I/usr/include` when cross compiling can be disabled with a new option `-Wno-poison-system-directories`. This option is intended for use in chroot environments when such directories contain the correct headers and libraries for the target system rather than the host.

**Thumb-2 doubleword writeback addressing modes.** An assembler bug that caused writeback addressing modes for `ldrd` and `strd` to be incorrectly encoded has been fixed.

**Thumb-2 shift instruction aliases.** The assembler now accepts `mov` with shifted operands as an alias for Thumb-2 shift instructions. For example `mov r0, r1, lsl r2` is encoded as `lsl r0, r1, r2`.

**Thumb-2 branches to shared libraries.** The linker can now generate PLT stubs for `R_ARM_THM_JUMP24` and `R_ARM_THM_JUMP19` relocations. This occurs when Thumb-2 branch instructions refer to symbols imported from shared libraries.

**EABI object attribute merging.** The linker now properly merges EABI object attributes into its output file.

**Thumb-2 exception return instructions.** An assembler bug that caused `subs pc, lr, #const` and `movs pc, lr` to be incorrectly encoded has been fixed.

**Tag\_ABI\_PCS\_wchar\_t object attributes.** Objects generated with `-fshort-wchar` are now given the correct `Tag_ABI_PCS_wchar_t` EABI object attribute annotations.

**Uppercase special register names.** The assembler now accepts both uppercase and lowercase special register names when assembling `msr` and `mrs` instructions for the Microcontroller profile of the ARM Architecture.

### 3.2.2 Changes in Sourcery G++ Lite 2007q1-10

**Disassembly of overlapping sections.** A bug in the disassembler that caused code to be displayed as data (and vice-versa) in files with overlapping sections has been fixed. This mainly affects the `objdump` utility.

**Installer hangs while refreshing environment.** The Sourcery G++ installer for Microsoft Windows now updates the `PATH` environment variable without waiting for open applications to acknowledge the update. This change prevents open applications from blocking the installer's progress.

**Improved assembler diagnostics for 8-bit offsets.** The assembler now correctly diagnoses out-of-range offsets to instructions such as `LDRD` as 8-bit rather than half-word offsets.

**Less disk space required for installation.** Sourcery G++ Lite packages are smaller because multiple copies of files have been replaced with hard and/or symbolic links when possible. Both the size of the installer images and the amount of disk space required for an installed package have been reduced.

**Thumb register corruption fix.** A bug in the compiler that could cause register corruption in Thumb mode has been fixed. The compiler was formerly emitting code to restore registers on function return that was not interrupt safe.

**\_\_aeabi\_lcmp.** An error in the libgcc implementation of \_\_aeabi\_lcmp that caused incorrect results to be returned has been fixed. This is a support routine defined by the ARM EABI. GCC does not normally use this routine directly, however it may be used by third-party code.

**The \@ assembler pseudo-variable.** A bug in the assembler that caused uses of the \@ pseudo-variable to be mis-parsed as comments has been fixed.

**Crash when generating vector code.** A bug that sometimes caused the compiler to crash when invoked with the `-ftree-vectorize` option has been fixed.

**Propagation of Thumb symbol attributes.** Symbols referring to Thumb functions on ARM targets now have their Thumb attribute correctly propagated to any aliases defined with `.set` or `.symver`.

**Linking of non-ELF images.** A linker bug that could cause a crash when linking non-ELF objects for ARM targets has been fixed.

**Invalid load instructions.** A bug in the compiler which caused it to generate invalid assembly (e.g. `ldrd r0, [#0, r2]`) has been fixed.

**VFPv3/NEON debug information.** A bug in the compiler which caused it to generate incorrect debug information for code using VFPv3/NEON registers has been fixed. The debugger is now able unable to locate and display values held in these registers.

**ARMv6-M system instructions.** An assembler bug that caused some ARMv6-M system instructions to be incorrectly rejected has been fixed. The affected instructions are `msr`, `mrs`, `yield`, `wfi`, `wfe` and `sev`.

**Assembly of Thumb-2 load/store multiple instructions.** The Thumb-2 `ldm` and `stm` assembly mnemonics are now assembled to `ldr` and `str` instructions when a single register is transferred, as specified in the Thumb-2 Architecture Supplement.

**Conditional Thumb-2 branch instructions.** A linker bug that could cause objects involving conditional Thumb-2 branch instructions to be incorrectly rejected has been fixed.

**Alignment bug fix.** A bug has been fixed that formerly caused incorrect code to be generated in some situations for copying structure arguments being passed by value. The incorrect code caused alignment errors on stack accesses on some targets.

### 3.2.3 Changes in Sourcery G++ Lite 2007q1-3

**Thumb-2 runtime libraries.** Sourcery G++ Lite now includes runtime libraries built as Thumb-2 code for use on ARMv7 systems. These can be found in the `libc/thumb2/` directory.

**Marvell Feroceon support.** Sourcery G++ Lite now generates code optimized for Marvell Feroceon CPUs when the `mcpu=marvell-f` option is specified. This option also selects runtime libraries optimized for this processor.

**Assembly of SRS instructions.** An assembler bug that resulted in incorrect encoding of the Thumb-2 SRS instruction has been fixed. In addition the assembler supports explicit specification of the base register, as accepted by other ARM toolchains.

**VFP disassembly crash.** A bug that caused crashes when disassembling some forms of the VFP `fmrx` and `fmxr` instructions has been fixed.

**Improved debugging for optimized code.** GDB's ability to print and change variables' values in optimized code is improved. GDB now tracks variable scopes more accurately, making better use of the detailed debugging information produced by Sourcery G++ Lite compilers.

**Improved handling of Windows paths in GDB.** GDB now properly recognizes the names of source files that were passed to the compiler using an absolute path on Windows. You may refer to the file either by its base name (without any leading directory components), by the exact path passed to the compiler, or by its absolute path.

**ARM Cortex-R4 performance improvements.** Sourcery G++ Lite now generates faster code when compiling for the ARM Cortex-R4 processor by scheduling instructions for the processor's pipelines. To generate code for this processor, use the `-mcpu=cortex-r4` command-line option.

**GDB update.** The included version of GDB has been updated to 6.6.50.20070228. This update includes numerous bug fixes and improved support for C++ pointers to members.

**Assembling Thumb store-multiple instructions.** The assembler now issues an error message instead of crashing on load/store multiple instructions that incorrectly use Thumb-2 addressing modes (e.g., `ldmdb`) in legacy Thumb syntax mode. If you want to use these address modes, you should use unified syntax mode instead.

**ARM Cortex-A8 performance improvements.** Sourcery G++ Lite now generates faster code when compiling for the ARM Cortex-A8 processor by scheduling instructions for the processor's dual-issue pipelines. To generate code for this processor, use the `-mcpu=cortex-a8` command-line option.

**GCC version 4.2.** Sourcery G++ Lite for ARM GNU/Linux is now based on GCC version 4.2. For more information about changes from GCC version 4.1 that was included in previous releases, see <http://gcc.gnu.org/gcc-4.2/changes.html>.

**Fix `--gc-sections` and C++ exceptions.** A bug in the `--gc-sections` linker option has been fixed. Previously this would incorrectly remove unwinding tables, breaking C++ applications that use exceptions.

**Symbols defined in linker scripts.** A bug is fixed that caused the linker to crash in some circumstances when a linker script defined a symbol in an output section. Typically usage is where the script contained a `__DATA_LOAD = LOADADDR(.data)` statement in the `.data` section.

**ARM NEON store intrinsics bug fix.** A compiler bug that incorrectly caused calls to ARM NEON store intrinsics (such as `vst1_u8`) to be optimized away has been fixed.

**Improvements to ARM NEON support.** The ARM NEON support in GCC has been enhanced to comply with new rules for containerized vector types specified in the ARM procedure call standard. Additionally, the compiler now rejects implicit conversions between NEON polynomial vector types and NEON integer vector types of the same layout.

**Complex numbers bug fix.** A bug that could lead to incorrect code generation for code using complex numbers has been fixed.

**Use of system header and library directories diagnosed.** The compiler and linker now diagnose the incorrect use of native system header and library directories for cross-compilation. This typically arises from options such as `-I/usr/X11R6/include` hard-coded in build scripts written without a view to cross-compilation.



**Initialization priorities.** The constructor and destructor function attributes now accept an optional priority argument. Constructors with small priorities are run before those with larger priorities; the opposite is true for destructors. For example:

```
void f __attribute__((constructor(500)));  
void f() {  
    /* Perform initialization. */  
}
```

defines a function `f` with priority 500. This function will be run before constructors with larger priorities. Constructors and destructors with no explicit priority argument have priority 65535, the maximum permitted value.

**Thumb-2 IT block code generation error fixed.** A bug in Thumb-2 code generation has been fixed. This bug would result in missing IT instructions, causing the assembler to reject the code.

**iWMMXt compiler errors.** A compiler bug that caused invalid assembly when generating iWMMXt code has been fixed.

**Thumb-2 stack decrement misassembly.** An assembler bug that resulted in incorrect encoding of the 32-bit Thumb-2 form of the `sub sp, sp, #const` instruction has been fixed. Previously this was misassembled as `subs`.

**Naked functions.** Functions marked with `__attribute__((naked))` no longer contain prologue and epilogue code. Please refer to the GCC manual for the proper use of this attribute.

**Destructor execution order.** Prioritized destructors (whether specified with the `destructor` function attribute in C or the `init_priority` variable attribute in C++) are now executed in the correct order. Previous releases ran the destructors in an indeterminate order.

**Fix `addr2line` defect.** The binary utility `addr2line` now operates correctly on 64-bit targets with DWARF2 debug information.

**Thumb-2 assembler infinite loop.** An assembler bug that would cause it to enter an infinite loop when processing some Thumb-2 assembly has been fixed.

**Assembler warnings about overlapping multiplication operands.** The assembler no longer warns about overlapping `Rd` and `Rm` operands when assembling `mul` and `mla` instructions for the ARM architecture version six or above.

**Improve handling of corrupt debug information.** The binary utility `readelf` now copes more gracefully with corrupted DWARF 2 information.

**Smaller C++ programs.** Rarely-used functions in the C++ runtime library have been isolated into separate object files so that they will not be included unless needed. As a result, most statically linked C++ programs are smaller.

### 3.2.4 Changes in Sourcery G++ Lite 4.1-37

**Preserve volatile accesses.** Reads from volatile memory are no longer incorrectly optimized away at higher optimization levels.

### 3.2.5 Changes in Sourcery G++ Lite 4.1-34

**Implicit conversions between generic vector types.** Implicit conversions between generic vector types are now only permitted when the two vectors in question have the same number of elements and compatible element types. (Note that the restriction involves *compatible* element types, not implicitly-convertible element types: thus, a vector type with element type `int` may not be implicitly converted to a vector type with element type `unsigned int`.) This restriction, which is in line with specifications for SIMD architectures such as AltiVec, may be relaxed using the flag `-flax-vector-conversions`. This flag is intended only as a compatibility measure and should not be used for new code.

**type\_info comparison fix.** Comparison of `type_info` objects now uses pointer comparison where possible.

**C++ forced unwinding fixes.** Some bugs relating to forced unwinding through C++ code have been fixed.

**Support for additional Stellaris boards.** Linker scripts are provided for the 6xx and 8xx series Stellaris boards.

**Linux support for USB Debug Sprite.** A new driver is included to allow the Sourcery G++ Lite USB Debug Sprite to run on Linux hosts. See Chapter 3, *Sourcery G++ Lite for ARM GNU/Linux* for additional information.

### 3.2.6 Changes in Sourcery G++ Lite 4.1-33

**Linker scripts.** A bug is fixed where an erroneous linker script would cause a linker crash. An error message is now produced.

**Newlib memory use improvements.** The memory overhead of linking with newlib is reduced. Applications that use only a minimal set of library features may now require significantly less memory.

### 3.2.7 Changes in Sourcery G++ Lite 4.1-31

**Compiler alias analysis.** The type-based alias analysis performed by the compiler when compiling with `-O2` or with `-fstrict-aliasing` is now more conservative. The more aggressive analysis used in previous versions sometimes resulted in incorrect code generation.

**Fully relocatable preprocessor.** When cross-compiling, the default preprocessor search path includes only the directories present in the installed toolchain. This speeds up the preprocessor and prevents the unintentional use of unrelated files and directories on the machine where it is installed.

### 3.2.8 Changes in Sourcery G++ Lite 4.1-29

**Support for new-style symbol hashing.** Support has been added in binutils and the prelinker for new-style (also known as `DT_GNU_HASH`) symbol hashing. This can dramatically speed up symbol resolution time and is particularly applicable in environments where full prelinking is not possible (for example where shared libraries are dynamically opened at runtime). The new-style hashing may be enabled by passing `--hash-style=gnu` to the linker.

**Prelinker update.** The prelinker has been updated to the current upstream sources and some bugs affecting operation have been fixed.

### 3.2.9 Changes in Sourcery G++ Lite 4.1-28

**Improved support for ROM debugging.** GDB now determines ROM regions automatically from the memory map included in target configuration files. This information is used to determine when hardware breakpoints should automatically be used (for instance the **step**, **next** and **finish** commands). Separate ROM configurations have been removed from the Eclipse debugger menu. The Eclipse GUI has been extended to provide improved support for debugging programs in ROM, when a memory map is not automatically available.

### 3.2.10 Changes in Sourcery G++ Lite 4.1-27

**Rename Windows executables.** The Windows host tools **make.exe** and **rm.exe** are now named **cs-make.exe** and **cs-rm.exe**. This change avoids conflicts with tools provided by other distributors.

**iWMMXt bug fixes.** Some bugs involving incorrect code generation and internal compiler errors when generating iWMMXt code have been fixed.

**Cortex-M3 startup code.** The ARMv7M startup code (`armv7m-crt0.o`) incorrectly contained ARM code. This has been replaced with Thumb-2 code.

**ARM EABI coverage testing support.** Coverage testing using GCOV is now supported for the ARM EABI target. Please refer to the *GNU C Compiler Manual (HTML)* for more information on coverage testing.

### 3.2.11 Changes in Sourcery G++ Lite 4.1-23

**Windows debugging fix.** In recent releases of Sourcery G++ Lite, the GDB **target remote |** command would hang on Windows. This affected both command line and Eclipse debugging when using the Sourcery G++ Lite Debug Sprite.

**Stellaris USB Debug Sprite improvements.** The former USB Debug Stub, **armswd**, is now known as the USB Debug Sprite, and has been renamed to **arm-stellaris-eabi-sprite**. In addition, its initialization sequence has been updated to recognize the r1p1 release of the Cortex-M3 processor.

**Incompatible changes to Stellaris linker scripts.** Sourcery G++ Lite now supports linking executables to run from RAM as well as ROM. As part of this change, there are now separate RAM and ROM versions of the linker scripts for each supported board, and the former ROM-based versions have been renamed. For example, if you were formerly linking with `-T lm3s10x.ld`, you should now use `-T lm3s10x-rom.ld` to get the same behavior.

### 3.2.12 Changes in Sourcery G++ Lite 4.1-21

**Eclipse debuggers.** Eclipse configurations for debugging arm-none-eabi applications using the GDB simulator and remote debug stubs have been added.

**iWMMXt2 support.** The assembler and disassembler now support iWMMXt2 instructions.

**NEON intrinsics support.** GCC now supports NEON intrinsics defined in the `arm_neon.h` header. These are the same intrinsics supported by the ARM RVCT compiler and are documented in the 'ARM NEON Intrinsics' section of the GCC manual.

### 3.2.13 Changes in Sourcery G++ Lite 4.1-19

**ARMv4t linux multilib.** Linux configurations now support ARMv4t CPUs.

**Linker scripts.** Several problems with the linker scripts for bare-metal targets have been fixed.

### 3.2.14 Changes in Sourcery G++ Lite 4.1-18

**Binutils update.** The binutils in this release is based on the final binutils 2.17 release.

**GDB update.** The included version of GDB has been upgraded to 6.5.50.20060822. This includes numerous bug fixes from the previous version.

**GDB support for flash memory.** The GDB **load** command can now write to flash memory, if the remote debugging stub contains appropriate support.

**Compiler support for NEON.** Initial GCC support for autovectorization and generation of NEON SIMD instructions has been added.

**Bare metal Cortex-M3 configurations.** Bare metal configurations now support generating images for use on ARMv7M devices (eg. Cortex-M3).

**iWMMXt support in GLIBC.** GLIBC's `setjmp` and `longjmp` now support saving and restoring iWMMXt registers on hardware with those registers. This requires a kernel reporting `iwmmxt` in the `Features` entry in `/proc/cpuinfo`.

**iWMMXt exception handling support.** Exception handling now restores the values of iWMMXt registers correctly.

**Corrected IPC functions.** A bug in GLIBC's `msgctl`, `semctl`, and `shmctl` functions has been corrected.

### 3.2.15 Changes in Sourcery G++ Lite 4.1-16

**GCC update.** This release is based on GCC 4.1.1.

**Fully relocatable compiler.** The compiler now searches for its components only in the directory where it has been installed, and no longer also searches pathnames matching the directory where it was configured. This speeds up the compiler and prevents problems with unintentionally finding unrelated files or directories on the machine where it has been installed.

**Stack permission marking for ARM GNU/Linux.** Non-executable stacks can provide increased security against some forms of buffer overflow attacks. The tools involved must coordinate the annotation of required stack permissions, either executable, or non-executable. For ARM GNU/Linux targets the compiler now outputs annotations indicating the required stack permissions.

### 3.2.16 Changes in Sourcery G++ Lite 4.1-15

**Stabs debugging information support.** Using the Stabs debugging format (available with `-gstabs` or `-gstabs+`) now works in conjunction with `-mthumb`. CodeSourcery recommends the default DWARF debugging format (available with `-g`) as DWARF is a more comprehensive debugging format.

### 3.2.17 Changes in Sourcery G++ Lite 4.1-13

**Stellaris linker scripts in IDE.** Linker scripts may now be selected via a drop-down menu in Eclipse.

**Stellaris linker scripts for 3xx series CPUs.** The linker scripts for 3xx Series CPUs now place the ISR vector at address zero, as required by all Cortex-M3 cores.

**Stellaris USB Debug Sprite improvements.** Bug fixes and new features include:

- A bug that caused the stub not to correctly update the program counter and other register values was fixed. As a result of this fix, it is now possible to run programs residing in SRAM using the `continue` command from GDB.
- The stub no longer prints status messages via GDB console output when invoked with the `-q` command-line option.
- The stub's initialization sequence was updated to recognize revision C Cortex-M3 hardware.

### 3.2.18 Changes in Sourcery G++ Lite 4.1-9

**Stellaris USB Debug Sprite improvements.** Program images exceeding 4K can now be uploaded to flash memory.

**Additional Stellaris boards supported.** The Stellaris 301, 310, 315, and 316 CPUs are now supported. Linker scripts have been added for these boards.

### 3.2.19 Changes in Sourcery G++ Lite 4.1-8

**Stellaris USB Debug Sprite improvements.** Several bug fixes and enhancements were made to the USB Debug Stub. In particular:

- Bugs in the implementation of `open`, `read`, and `lseek` were fixed.
- Support was added for `isatty`, `rename`, `unlink`, and `system`.
- Memory reads that span 4K block boundaries now work correctly.

### 3.2.20 Changes in Sourcery G++ Lite 4.1-4

**Runtime libraries.** Support for ARMv7 including Cortex-M3 and pure Thumb-2.

**Assembler.** Support for NEON and VFPv3, including unified NEON/VFP syntax.

### 3.2.21 Changes in Sourcery G++ Lite 4.1-1

**Initial release.** This release is based on GCC 4.1.0.

---

# Chapter 4

## Installation and Configuration

This chapter explains how to install Sourcery G++ Lite. You will learn how to:

1. Verify that you can install Sourcery G++ Lite on your system.
2. Download the appropriate Sourcery G++ Lite installer.
3. Install Sourcery G++ Lite.
4. Configure your environment so that you can use Sourcery G++ Lite.

## 4.1 Terminology

Throughout this document, the term *host system* refers to the system on which you run Sourcery G++ Lite while the term *target system* refers to the system on which the code produced by Sourcery G++ Lite runs. The target system for this version of Sourcery G++ Lite is "arm-none-linux-gnueabi".

If you are developing a workstation or server application to run on the same system that you are using to run Sourcery G++ Lite, then the host and target systems are the same. On the other hand, if you are developing an application for an embedded system, then the host and target systems are probably different.

## 4.2 System Requirements

### 4.2.1 Host Operating System Requirements

Sourcery G++ Lite supports the following host operating systems:

- Microsoft Windows NT 4, Windows 2000, Windows XP, and Windows Vista systems using IA32, AMD64, and EM64T processors.
- GNU/Linux systems using the IA32, AMD64, or EM64T processors, including Debian 3.0 (and later), Red Hat Enterprise Linux 3 (and later), SuSE Enterprise Linux 8 (and later).
- Solaris 2.8 (and later) systems using SPARC processors.

Not all combinations of host and target systems are available. Therefore, Sourcery G++ Lite for your target system may not be available on all of the above host systems.

Sourcery G++ Lite is built as a 32-bit application. Therefore, even when running on a 64-bit GNU/Linux host system, Sourcery G++ Lite requires 32-bit host libraries. If these libraries are not already installed on your system, you must install them before installing and using Sourcery G++ Lite. Consult your operating system documentation for more information about obtaining these libraries.

### 4.2.2 Host Hardware Requirements

In order to install and use Sourcery G++ Lite, you must have at least 128MB of available memory.

The amount of disk space required for a complete Sourcery G++ Lite installation directory depends on the host operating system and the number of target libraries included. Typically, you should plan on at least 400MB. In addition, the graphical installer requires a similar amount of scratch space during the installation process.

### 4.2.3 Target System Requirements

See Chapter 3, *Sourcery G++ Lite for ARM GNU/Linux* for requirements that apply to the target system.

## 4.3 Downloading an Installer

If you have received Sourcery G++ Lite on a CD, or other physical media, then you do not need to download an installer. You may skip ahead to Section 4.4, "Installing Sourcery G++ Lite".

If you have a Sourcery G++ Lite subscription (or evaluation), then you can log into the Sourcery G++ Portal<sup>1</sup> to download your Sourcery G++ Lite toolchain(s). CodeSourcery also makes some toolchains available to the general public from the Sourcery G++ web site<sup>2</sup>. These publicly available toolchains do not include all the functionality of CodeSourcery's product releases.

Once you have navigated to the appropriate web site, download the installer that corresponds to your host operating system. For Microsoft Windows systems, the Sourcery G++ Lite installer is provided as an executable, with the `.exe` extension. For GNU/Linux systems with an X Window System, Sourcery G++ Lite is provided as a graphical installer with the `.bin` extension. For Solaris, and GNU/Linux systems without an X Window System, Sourcery G++ Lite is provided as a compressed archive `.tar.bz2`. If installing on a RPM-based GNU/Linux system you may download the `.rpm` file.

On Microsoft Windows systems, save the installer to the desktop. On GNU/Linux and Solaris systems, save the download package in your home directory.

## 4.4 Installing Sourcery G++ Lite

The method used to install Sourcery G++ Lite depends on your host system.

### 4.4.1 Installing Sourcery G++ Lite on Microsoft Windows

If you have received Sourcery G++ Lite on CD, insert the CD in your computer. On most computers, the installer then starts automatically. If your computer has been configured not to automatically run CDs, open My Computer, and double click on the CD. If you downloaded Sourcery G++ Lite, double-click on the installer.

After the installer starts, follow the on-screen dialogs to install Sourcery G++ Lite. This package comes with a bundled Java Runtime Environment; you do not have to download any additional software.

### 4.4.2 Installing Sourcery G++ Lite on GNU/Linux systems with an X Window System

Start the graphical installer by invoking the executable shell script:

```
> /bin/sh ./path/to/package.bin
```

After the installer starts, follow the on-screen dialogs to install Sourcery G++ Lite. This package comes with a bundled Java Runtime Environment; you do not have to download any additional software.

### 4.4.3 Installing Sourcery G++ Lite on Solaris or GNU/Linux systems without an X Window System

You do not need to be a system administrator to install Sourcery G++ Lite on a GNU/Linux or Solaris system. You may install Sourcery G++ Lite using any user account and in any directory to which you have write access. This guide assumes that you have decided to install Sourcery G++ Lite in the `$HOME/CodeSourcery` subdirectory of your home directory and that the filename of the package

---

<sup>1</sup> <https://support.codesourcery.com/GNUToolchain/>

<sup>2</sup> [http://www.codesourcery.com/gnu\\_toolchains/](http://www.codesourcery.com/gnu_toolchains/)



you have downloaded is `/path/to/package.tar.bz2`. After installation the toolchain will be in `$HOME/CodeSourcery/sourceryg++-4.1` or similar.

First, uncompress the package file:

```
> bunzip2 /path/to/package.tar.bz2
```

Next, create the directory in which you wish to install the package:

```
> mkdir -p $HOME/CodeSourcery
```

Change to the installation directory:

```
> cd $HOME/CodeSourcery
```

Unpack the package:

```
> tar xf /path/to/package.tar
```

If you are installing a native toolchain, it is then necessary to run a post-install script found in the share directory:

```
> /bin/sh sourceryg++-4.1/share/postinst-*
```

The `.tar.bz2` package is not bundled with a Java Runtime Environment.

#### 4.4.4 Installing Sourcery G++ Lite on RPM-based GNU/Linux systems

On a RPM-based system you should use RPM to install the provided package. Execute the following command as root (administrator):

```
> rpm -ivh /path/to/package.rpm
```

The `.rpm` package is not bundled with a Java Runtime Environment.

#### 4.4.5 Installing the Java Runtime Environment

Some versions of Sourcery G++ Lite include the Eclipse Integrated Development Environment. Because Eclipse is an optional component, the installer allows you to choose whether or not to install it. Eclipse is a Java application and requires the Java Runtime Environment (JRE). The Java Runtime Environment is available at no charge from Sun Microsystems Java website<sup>3</sup>. You may download either the Java Runtime Environment (JRE) or the Java Development Kit (JDK). (The JDK includes the JRE.)

### 4.5 Uninstalling Sourcery G++ Lite

The method used to uninstall Sourcery G++ Lite depends on your host system. If you have modified any files in the installation it is recommended that you back up these changes. The uninstall procedure may remove the files you have altered.

---

<sup>3</sup> <http://java.sun.com/j2se/>

### 4.5.1 Uninstalling Sourcery G++ Lite on Microsoft Windows

Select Start, then Control Panel. Select Add or Remove Programs. Scroll down and click on Sourcery G++ for ARM GNU/Linux. Select Change/Remove and follow the on-screen dialogs to uninstall Sourcery G++ Lite.

### 4.5.2 Uninstalling Sourcery G++ Lite on Microsoft Windows Vista

Select Start, then Settings and finally Control Panel. Select the Uninstall a program task. Scroll down and double click on Sourcery G++ for ARM GNU/Linux. Follow the on-screen dialogs to uninstall Sourcery G++ Lite.

### 4.5.3 Uninstalling Sourcery G++ Lite on GNU/Linux using the graphical uninstaller

If you installed on GNU/Linux using the graphical installer, then you must use the graphical uninstaller to remove Sourcery G++ Lite. The `arm-none-linux-gnueabi` directory located in the install directory will be removed entirely by the uninstaller. Please back up any changes you have made to this directory, such as modified linker scripts.

Start the graphical uninstaller by invoking the executable shell script:

```
> /bin/sh ./path/to/install/\
    Sourcery_G++/\
    Uninstall_Sourcery_GXX_for_ARM_GNU/Linux
```

After the uninstaller starts, follow the on-screen dialogs to uninstall Sourcery G++ Lite.

### 4.5.4 Uninstalling Sourcery G++ Lite on RPM-based GNU/Linux systems

On a RPM-based system you should use RPM to uninstall the installed package. Execute the following command as root (administrator):

```
> rpm -e sourceryg++-arm-none-linux-gnueabi
```

### 4.5.5 Uninstalling Sourcery G++ Lite on GNU/Linux

If you did not use the graphical installer or RPM, uninstall Sourcery G++ Lite by manually deleting the installation directory created in the install procedure.

## 4.6 Setting up the Environment

As with the installation process itself, the steps required to set up your environment depend on your host operating system. The name of the Sourcery G++ Lite commands all begin with **arm-none-linux-gnueabi** so that you can install Sourcery G++ Lite for multiple target systems in the same directory.

### 4.6.1 Setting up the Environment on Microsoft Windows

On a non-Vista Microsoft Windows system, the installer automatically adds Sourcery G++ Lite to your PATH. You can test that your PATH is set up correctly by using the following command:

```
> arm-none-linux-gnueabi-g++ -v
```

and verifying that the last line of the output contains: Sourcery G++ 2007q1-21.

On a Microsoft Windows Vista system, the installer does not automatically add Sourcery G++ Lite to your PATH. The Sourcery G++ IDE does not need this step to function correctly. This step is only required if you wish to use the tools from the command line on a Microsoft Windows Vista system. To set up your PATH on Microsoft Windows Vista, use the following command in a `cmd.exe` shell:

```
setx "%PATH%;C:\Program Files\Sourcery G++\bin"
```

where `C:\Program Files\Sourcery G++` should be changed to the path of your Sourcery G++ Lite installation. You can verify that the command worked by starting a second `cmd.exe` shell and running:

```
arm-none-linux-gnueabi-g++ -v
```

Verify that the last line of the output contains: Sourcery G++ 2007q1-21.

#### 4.6.1.1 Working with Cygwin

Sourcery G++ Lite does not require Cygwin or any other UNIX emulation environment. You can use Sourcery G++ Lite directly from the Eclipse IDE or from the Windows command shell. You can also use Sourcery G++ Lite from within the Cygwin environment, if you prefer.

The Cygwin emulation environment translates Windows path names into UNIX path names. For example, the Cygwin path `/home/user/hello.c` corresponds to the Windows path `c:\cygwin\home\user\hello.c`. Because Sourcery G++ Lite is not a Cygwin application, it does not, by default, recognize Cygwin paths.

If you are using Sourcery G++ Lite from Cygwin, you should set the `CYGPATH` environment variable. If this environment variable is set, Sourcery G++ Lite automatically translates Cygwin path names into Windows path names. To set this environment variable, type the following command in a Cygwin shell:

```
> export CYGPATH=cygpath
```

To resolve Cygwin path names, Sourcery G++ Lite relies on the **cygpath** utility provided with Cygwin. You must provide Sourcery G++ Lite with the full path to `cygpath` if **cygpath** is not in your PATH. For example:

```
> export CYGPATH=c:/cygwin/bin/cygpath
```

directs Sourcery G++ Lite to use `c:/cygwin/bin/cygpath` as the path conversion utility. The value of `CYGPATH` must be an ordinary Windows path, not a Cygwin path.

#### 4.6.2 Setting up the Environment on GNU/Linux or Solaris

If you installed Sourcery G++ Lite using the `.bin` graphical installer then you may skip this step. The graphical installer does this setup for you.

Before using Sourcery G++ Lite you should add Sourcery G++ Lite to your PATH. The command you must use varies with the particular command shell that you are using. If you are using the C Shell (**csh** or **tcsh**), use the command:

```
> setenv PATH $HOME/CodeSourcery/sourceryg++-4.1/bin:$PATH
```

If you are using Bourne Shell (**sh**), the Korn Shell (**ksh**), or another shell, use:

```
> export PATH=$HOME/CodeSourcery/sourceryg++-4.1/bin:$PATH
```

If you are not sure which shell you are using, try both commands. In both cases, if you have installed Sourcery G++ Lite in an alternate location, you must replace the directory above with `bin` subdirectory of the directory in which you installed Sourcery G++ Lite.

You may also wish to set the `MANPATH` environment variable so that you can access the Sourcery G++ Lite manual pages, which provide additional information about using Sourcery G++ Lite. To set the `MANPATH` environment variable, follow the same steps shown above, replacing `PATH` with `MANPATH`, and `bin` with `share/doc/sourceryg++-2007q1-21-arm-none-linux-gnueabi/man`.

You can test that your `PATH` is set up correctly by using the following command:

```
> arm-none-linux-gnueabi-g++
```

and verifying that you receive the message:

```
arm-none-linux-gnueabi-g++: no input files
```

---

# Chapter 5

## Using the Sourcery G++ Lite IDE

This chapter explains how to use the Sourcery G++ Lite IDE to build a C or C++ application. This chapter assumes you have installed Sourcery G++ Lite as described in Chapter 4, *Installation and Configuration*. If you prefer to use the command line to build your applications, you may refer to Chapter 6, *Using Sourcery G++ from the Command Line* instead.

The IDE is not included in Sourcery G++ Lite. If you are using Sourcery G++ Lite, you may skip this chapter.

## 5.1 Overview

This chapter explains how to create, modify, and debug a program using the Sourcery G++ Lite IDE. After working through the example program in this chapter, you can use the same techniques to create your own programs.

This chapter is divided into two sections. The first explains how to create and build a program; the second section explains how to debug and run a program once it has been built.

### Learning More About Eclipse

The Sourcery G++ Lite IDE is based on Eclipse. While this chapter explains how to accomplish basic tasks using the Sourcery G++ Lite IDE, it is not a comprehensive reference manual. If you want to learn more about Eclipse visit the Eclipse web site<sup>1</sup>.

## 5.2 Building Applications

In the Sourcery G++ Lite IDE, every program is a *project*. The project contains all of the source files required to build the program. So, the first step is to create a project.

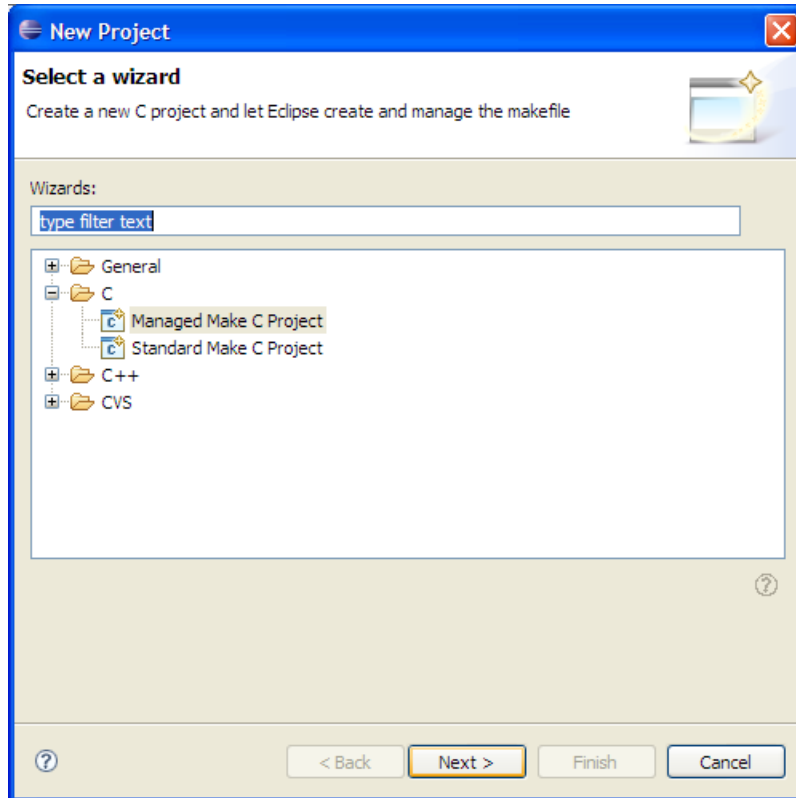
There are two kinds of projects: “Managed Make” and “Standard Make” projects. In general, if you intend to do all of your development from within the IDE, you should use a Managed Make project. In this mode, the IDE automatically handles building your project for you. However, if you are working with code that has previously been built with **make**, you may wish to use a Standard Make project instead. The following several sections explain how to create and work with a Managed Make project. If you wish to use a Standard Make project instead, skip ahead to section Section 5.2.5, “Using Standard Make Mode”.

### 5.2.1 Setting Up

Create a new project by selecting **File** → **New** → **Project**. Expand the **C** label and select **Managed Make C Project**. (If you want to build a C++ application, expand the **C++** label instead.) Click the **Next** button.

---

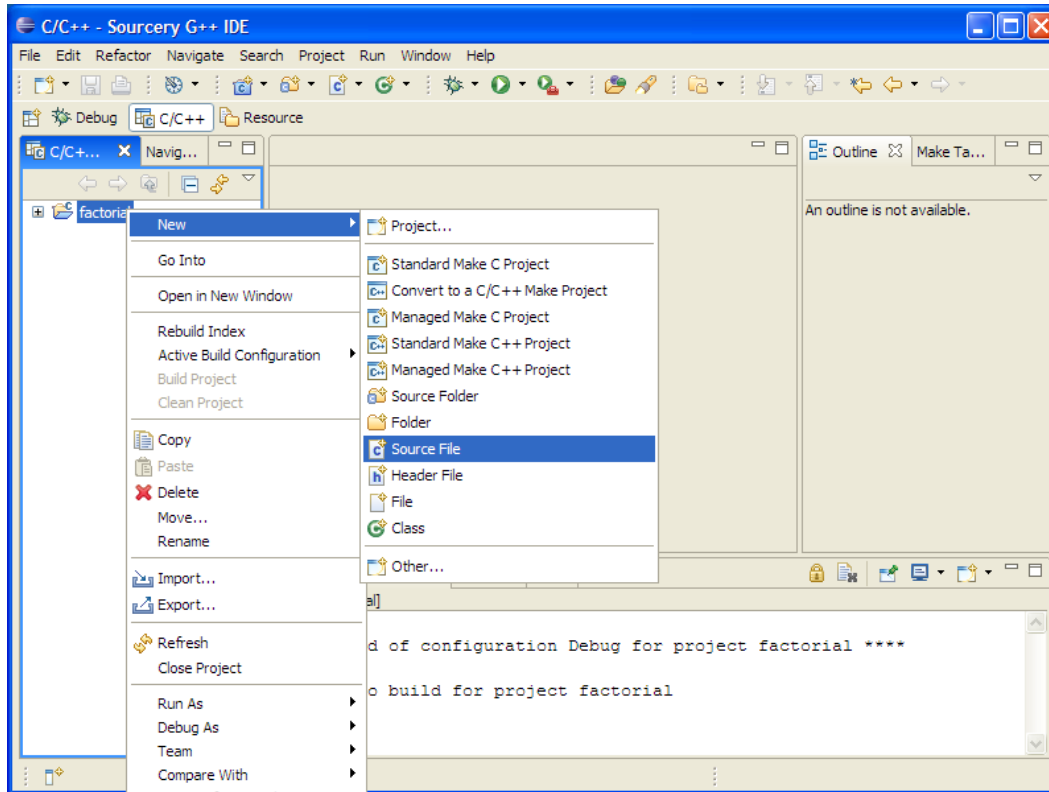
<sup>1</sup> <http://www.eclipse.org>



Expand the C folder and select the Managed Make C Project to create a new program.

Give the project the name `factorial` and click the Next button. From the Project Type menu select Executable (Sourcery G++ Lite for ARM GNU/Linux) and click Finish. If you are asked whether or not to open a new perspective, click the Yes button.

At this point, the project exists, but there is no associated source code. So, the next step is to create the main program. Right-click on the `factorial` project, and select New → Source File. Give the new file the name `main.c` and click the Finish button.



Right-click on the project name to add a new source file.

## 5.2.2 Writing Source Code

Whenever you create or save a file, the Sourcery G++ Lite IDE attempts to rebuild the program. Because the program is empty at this point, the compilation does not succeed, and you may notice some messages in the Console tab indicating errors. Those errors will go away when the program is completed.

The Sourcery G++ Lite IDE now displays an editing window for you to use to create the program. Type (or cut-and-paste) the following program into the editor:

```
#include <stdio.h>

int factorial(int n) {
    if (n == 0)
        return 1;
    return n * factorial (n - 1);
}

int main () {
    int i;
    int n;
    for (i = 0; i < 10; ++i) {
        n = factorial (i);
        printf ("factorial(%d) = %d\n", i, n);
    }
    return 0;
}
```



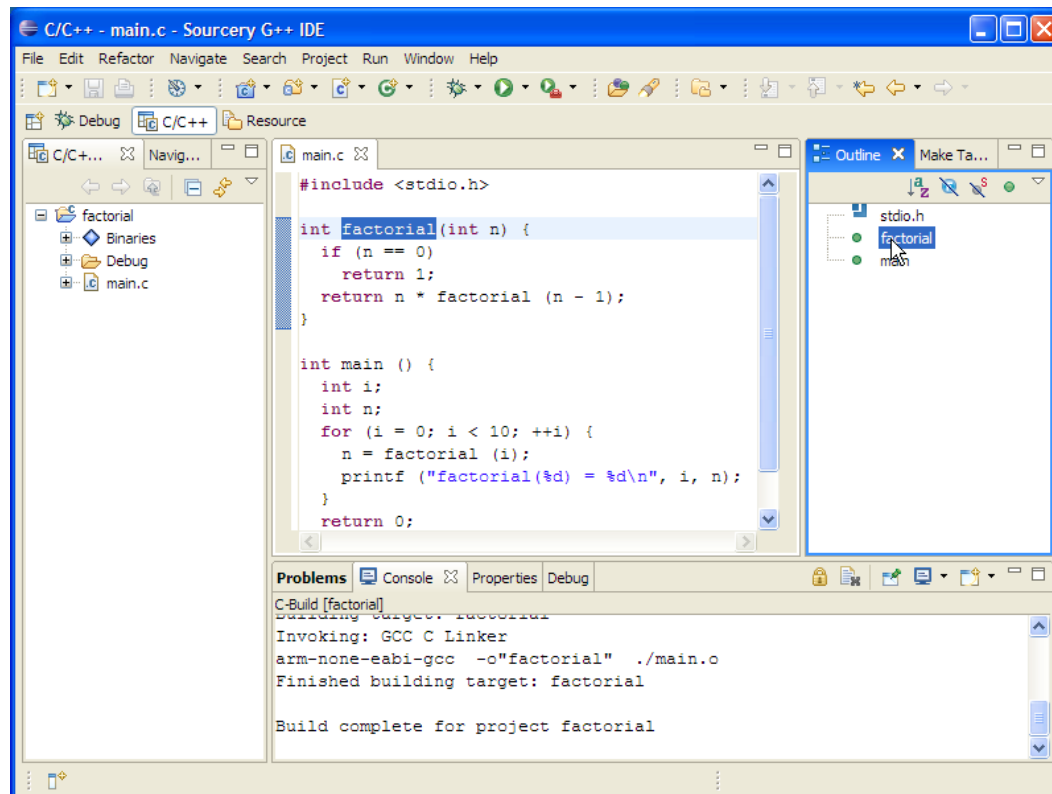
When you are done, save the file with **File** → **Save** (**Ctrl-S**).

When you save the file, the Sourcery G++ Lite IDE rebuilds the project. The output of the commands run by the IDE is displayed in the **Console** tab. You should see the following output at the bottom of the console:

```
Build complete for project factorial
```

### 5.2.3 Using Cross-Reference Information

Whenever it rebuilds your project, the Sourcery G++ Lite IDE also computes cross-reference information. You can see some of this information in the **Outline** pane. In particular, each of the two functions in the program (**factorial** and **main**) are shown in the **Outline** pane. When you click on name of a function or variable in the **Outline** pane, the IDE repositions the cursor to show you that entity.





Click a function name in the **Outline** to jump to it in the editor.

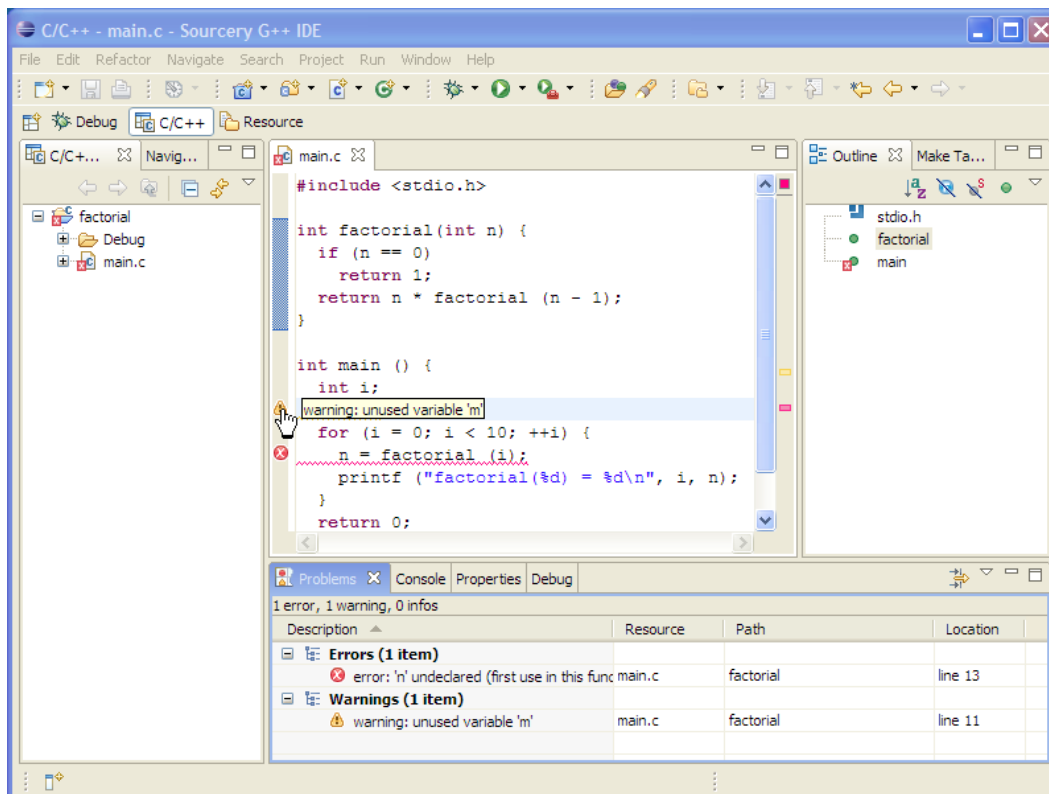
You can also use the cross-reference information to jump from the place where a function is called to the definition of the function. For example, find the line in **main** which calls **factorial** and place the cursor over the name **factorial**. Then, right-click and select **Open Declaration** (**F3**) to jump to the point at which **factorial** is declared. The cross-reference functionality works even if the function call is in a different file from the declaration of the function.

## 5.2.4 Dealing with Errors

If you pasted the sample application into the IDE, the program probably compiled correctly the first time. But, of course, that rarely happens when writing a large program from scratch. To see how the Sourcery G++ Lite IDE deals with errors, you can intentionally introduce an error.

Change the declaration of `n` in `main` to declare `m`, instead of `n`, and save the file. This change makes the program invalid because there are references to `n` in the function, but no declaration. In addition, the new variable `m` is not serving any useful purpose (since there are no references to it). Sourcery G++ Lite informs you of both issues by flagging the problematic lines of source code.

The IDE places a circular red symbol  next to lines that cause errors and a triangular yellow symbol  on lines that cause warnings. There are several ways to get more detailed information about the problems. One way is to click on the Problems pane at the bottom of the IDE. This pane shows the error and/or warning messages issued by the compiler. Also, when you place the cursor over the error indicators, the IDE displays the error message.



Place the cursor over a warning or error indicator to see the cause of the problem.

Before proceeding, you must correct the error by changing `m` back to `n`.

## 5.2.5 Using Standard Make Mode

This section explains how to use the advanced Standard Make mode, instead of the simpler Managed Make mode described above. If you are just getting started with Sourcery G++ Lite, you should skip this section and proceed directly to Section 5.3, “Debugging Applications”.

**Caution**

Using Standard Make Mode requires that you manually maintain information about how your program is built. If you use this mode, you need to be familiar with the **make** utility.


If you want to import an existing project for use with the Sourcery G++ Lite IDE, and that project uses **make**, or some similar command-line tool to manage the build process, you should use a Standard Make project, instead of a Managed Make project. In Standard Make mode, the IDE invokes **make** (or an alternative program that you specify) to build your program. If you add new files to your project, you have to manually update the `Makefile` for your project.

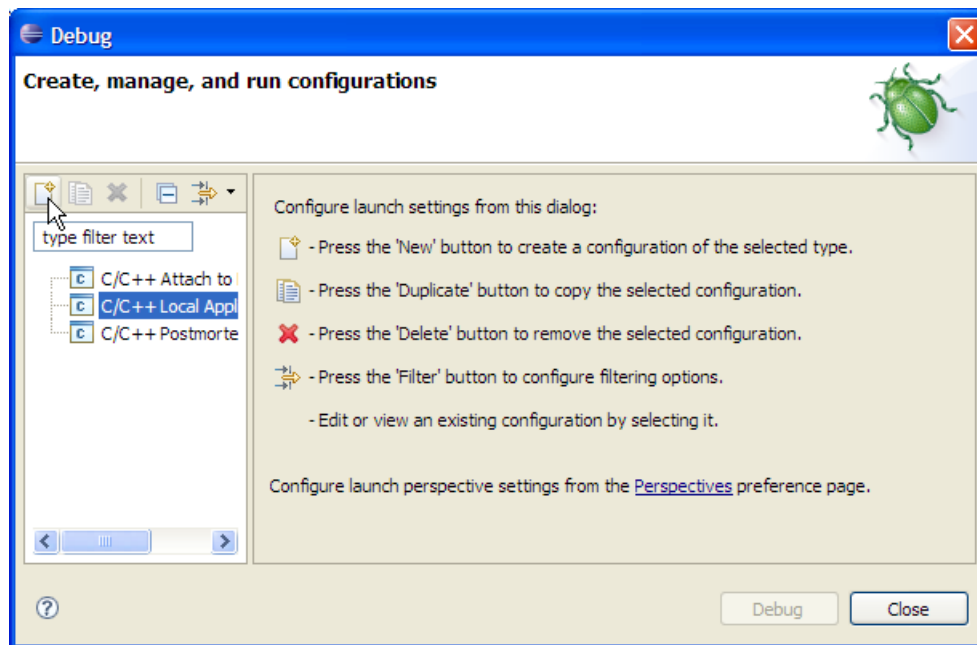
To set up the Standard Make mode to work with Sourcery G++ Lite, you have to make a few changes to the default project settings. When you create the project, the IDE displays a window that permits you to define the project settings.

Select the `Discovery Options` tab and set the `Compiler invocation command` to **arm-none-linux-gnueabi-gcc** instead of the default `gcc`. That change tells the IDE to use the Sourcery G++ Lite compilers when scanning your program code to determine cross-reference information. You may also have to adjust your `Makefile` to use Sourcery G++ Lite. For example, you might need to set the `CC` variable in your `Makefile` to **arm-none-linux-gnueabi-gcc**.

## 5.3 Debugging Applications

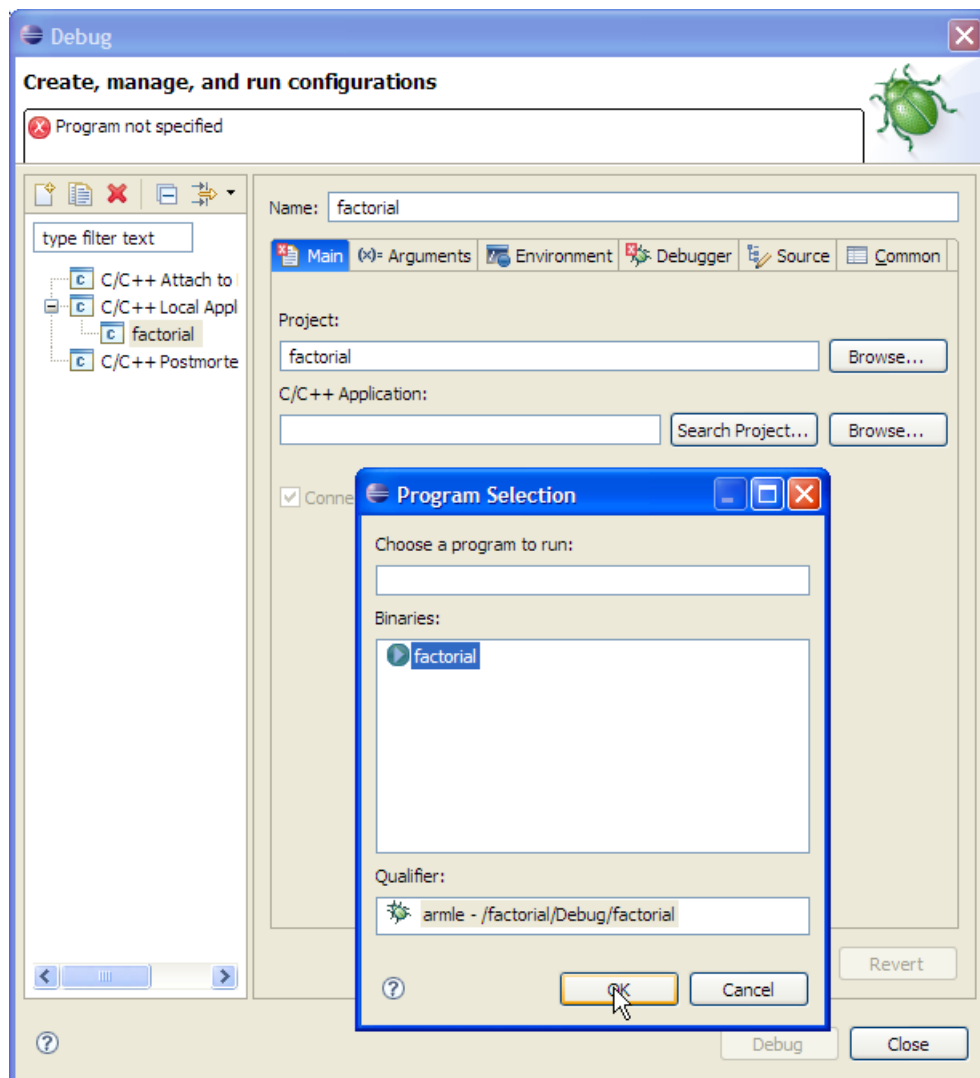
### 5.3.1 Starting the Debugger

After you build your application, choose `Run`. Select the `C/C++ Local Application` label in the `Configurations` pane. Then, click the `New` icon  positioned towards the upper left of the window.



Click the `New` icon to create to create a new debug configuration.

When you create the launch configuration, a new window appears. On the Main tab, use the Browse... button to select your project, if it is not already selected. Then, use the Search Project... button to select your application.



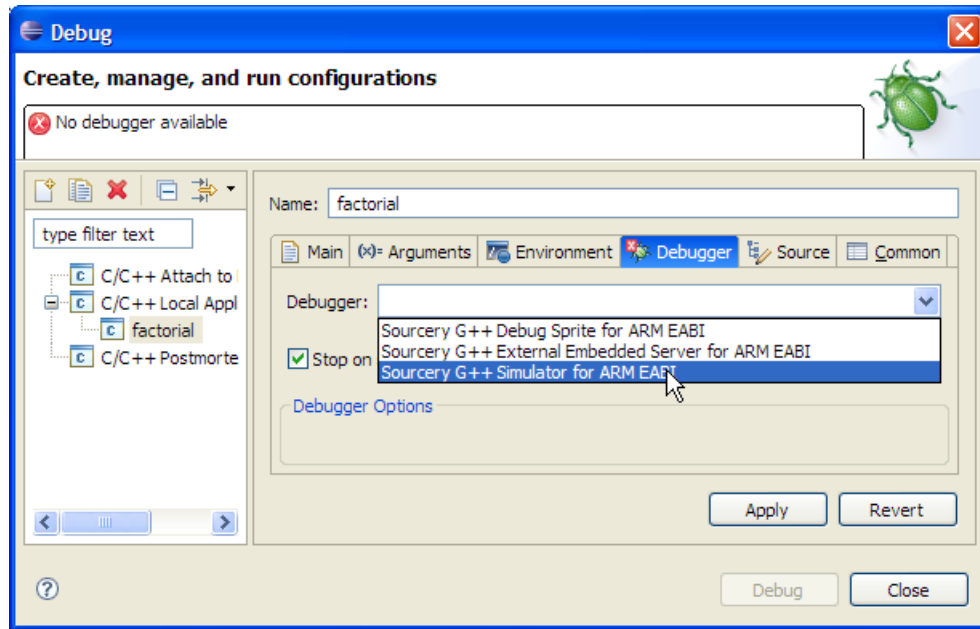
Use the Search Project... button to locate your program.

## 5.3.2 Choosing a Debugging Mode

Before you can use the Sourcery G++ Lite IDE to debug your application, you must decide which debugging mode to use. Sourcery G++ Lite supports several debugging modes, as described below.

### 5.3.2.1 Selecting a Debugger

Once you have decided which debugger to use, switch to the Debugger tab and select the appropriate Sourcery G++ Lite option.



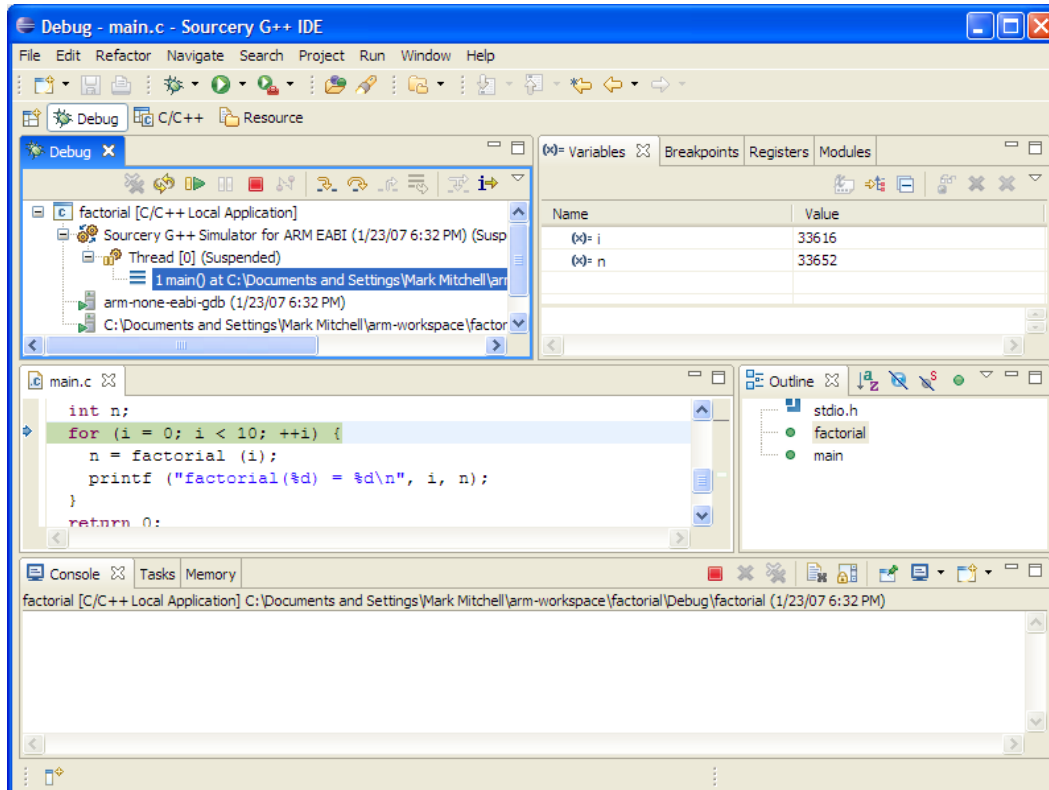
Pick the debugger that you want to use.

Once you have made any necessary adjustments, click the Debug button to start the debugger.

You do not need to repeat the debugger selection process the next time you launch the debugger. Instead, you can select **Run → Debug Last Launched** to start the debugger using the settings you have selected.

### 5.3.3 Controlling Execution

When you start the debugger, the IDE switches from the C/C++ perspective to the debug perspective. Instead of showing panes that help you to develop your application, the IDE now shows panes that help you to debug your application.

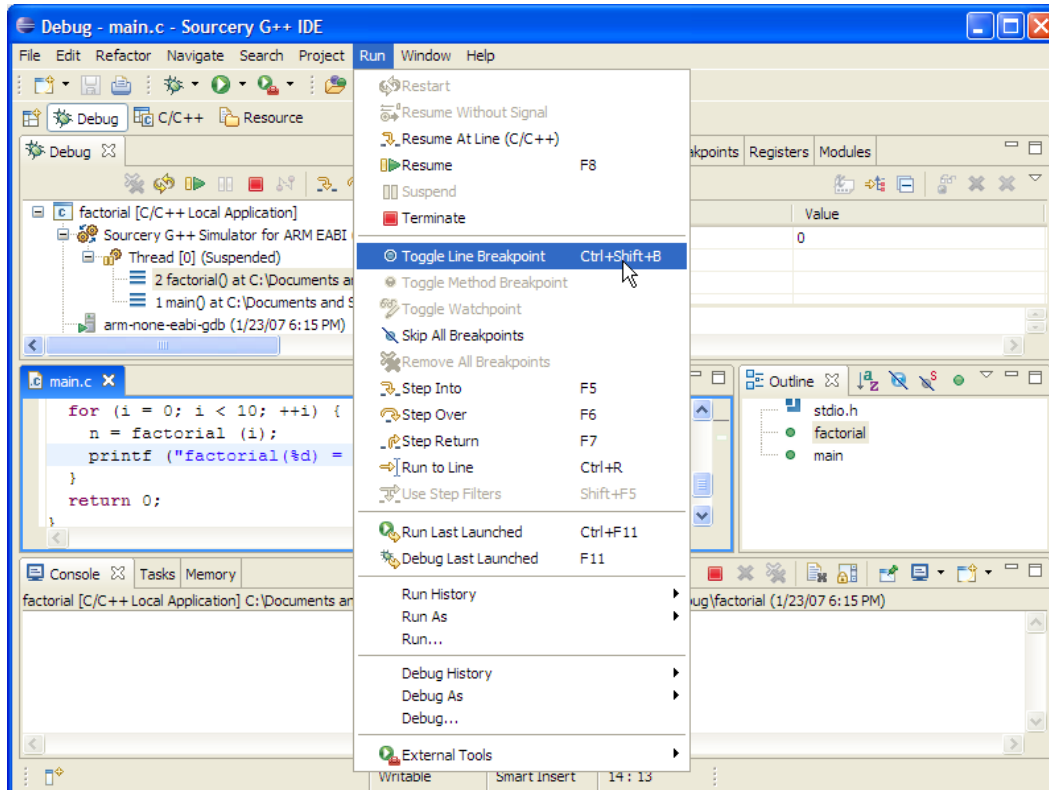


The debug perspective displays the stack, local variables, and the current location.

The debugger automatically stops on the first line of `main`. The currently active source line is highlighted. The pane at the upper left shows the application threads and the stack associated with each thread. The pane at the upper right shows the values of local variables. (At this point, `i` and `n` have not yet been initialized, so their values are indeterminate.)

Use `Run → Step Over (F6)` to advance by a single line. Because the program has changed the value of `i`, the IDE highlights the value in the variable pane.

By looking at the code, you can see that the program calls `factorial` and then calls `printf` to print out the resulting value. You can set a breakpoint right before the call to `printf` by clicking anywhere on that line, and then using `Run (Ctrl-Shift-B)`.



Set a breakpoint by highlighting the line where you want to stop and then using the Run menu.

After setting the breakpoint, use Run → Step Into (F5) to step into the body of factorial.

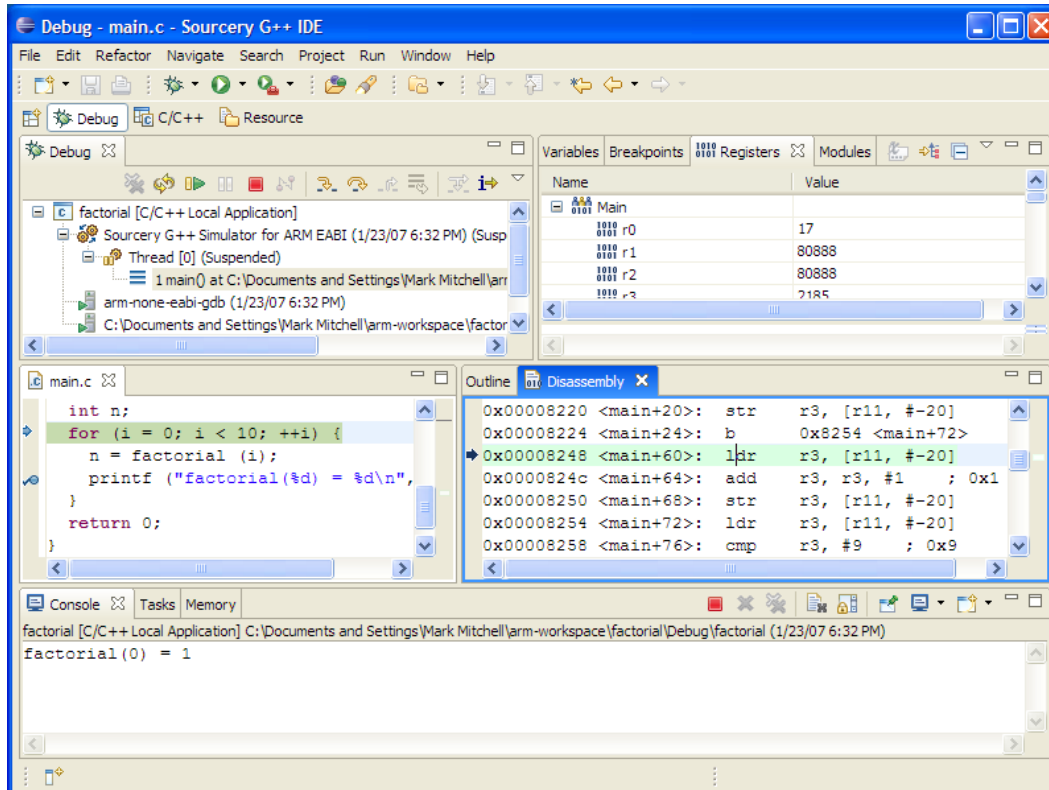
The IDE no longer displays the value of `i` because there is no local variable `i` within `factorial`. If you wish to see the value of `i` (from `main`), select the stack frame for `main` in the pane at the upper left. The IDE displays the variables for whichever frame is presently selected.

Now, proceed to the breakpoint by using Run → Resume (F8). The variable `n` now has the value 1 because the factorial of zero is one. Step over the call to `printf` to print the value in the console.

### 5.3.4 Low-Level Debugging

You may sometimes need to debug at the machine level, rather than at the source code level. For example, if you are working with an assembly code device driver, you may wish to see the values stored in machine registers and step through the code instruction by instruction.

To view machine registers, click on the Registers tab, and expand the Main register group. To see the instructions being executed, use Window → Show View → Disassembly.



The Sourcery G++ Lite IDE can display machine registers and assembly code.

When the disassembly window is active, the Step Over and Step Into commands operate at the assembly level, rather than at the source code level. So, a Step Over command advances by a single machine instruction. When the values of registers change, the registers are highlighted in the IDE. You can set breakpoints on particular machine instructions in the same way that you can set breakpoints on source code.



---

## Chapter 6

# Using Sourcery G++ from the Command Line

This chapter demonstrates the use of Sourcery G++ Lite from the command line. This chapter assumes you have installed Sourcery G++ Lite as described in Chapter 4, *Installation and Configuration*. If you prefer to use an integrated development environment to build your applications, you may refer to Chapter 5, *Using the Sourcery G++ Lite IDE* instead.

## 6.1 Building an Application

This chapter explains how to build an application with Sourcery G++ Lite using the command line. As elsewhere in this manual, this section assumes that your target system is `arm-none-linux-gnueabi`. If you are using a different target system, you must replace commands that begin with **arm-none-linux-gnueabi** with the name of your target system.

Using an editor (such as **notepad** on Microsoft Windows or **vi** on UNIX-like systems), create a file named `hello.c` containing the following simple program:

```
#include <stdio.h>

int
main (void)
{
    printf("Hello World!\n");
    return 0;
}
```

Compile and link this program using the command:

```
> arm-none-linux-gnueabi-gcc -o hello hello.c
```

There should be no output from the compiler. (If you are building a C++ application, instead of a C application, replace **arm-none-linux-gnueabi-gcc** with **arm-none-linux-gnueabi-g++**.)

## 6.2 Running an Application

If the target system is the same as the host system (e.g., if you are running Sourcery G++ Lite on IA32 GNU/Linux to build an application for IA32 GNU/Linux), then you can just run the resulting application. On a Microsoft Windows system, you may use the command:

```
> hello
```

On a GNU/Linux or Solaris system, use the slightly more complex:

```
> ./hello
```

command. In either case, you should see:

```
Hello world!
```

If the target system is not the same as the host system, then you cannot run the application directly. Instead, you must run the application on the target system. You should consult the manuals for your target system to determine the exact procedures required to run the application.

On some systems, Sourcery G++ Lite includes a simulator that can be used to run the program. To use the simulator run:

```
> arm-none-linux-gnueabi-run hello
```

The simulator is available if you see the expected output:

```
Hello, world!
```

There is no simulator for your target system if you see a message like:

```
'arm-none-linux-gnueabi-run' is not recognized as an internal or external command
```

or:

```
arm-none-linux-gnueabi-run: command not found
```

---

# Appendix A. GNU General Public License

Version 2, June 1991

Copyright © 1989, 1991 Free Software Foundation, Inc.

Free Software Foundation, Inc.

51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Version 2, June 1991

## Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software - to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps:

1. copyright the software, and
2. offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in

effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

## TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:
  - a. You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
  - b. You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
  - c. If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License.

**Exception:** If the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2 in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:
  - a. Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
  - b. Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
  - c. Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program 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.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.
6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.
7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.
9. The Free Software Foundation may publish revised and/or new versions of the 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 a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.
11. **NO WARRANTY.** BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, 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.
12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE 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.

END OF TERMS AND CONDITIONS

## How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.> Copyright (C) <year>  
<name of author>



This program 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.

This program 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 program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'. This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989 Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

---

# Appendix B. GNU Lesser General Public License

This is the first released version of the Lesser GPL. It also counts as the successor of the GNU Library Public License, version 2, hence the version number 2.1.  
Copyright © 1991, 1999 Free Software Foundation, Inc.

Free Software Foundation, Inc.  
51 Franklin Street, Fifth Floor,  
Boston,  
MA  
02110-1301  
USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.  
Version 2.1, February 1999

## B.1 Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method:

1. we copyright the library, and

2. we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the *Lesser* General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a “work based on the library” and a “work that uses the library”. The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

## **B.2 TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION**

### **B.2.1 Section 0**

This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called “this License”). Each licensee is addressed as “you”.

A “library” means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The “Library”, below, refers to any such software library or work which has been distributed under these terms. A “work based on the Library” means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term “modification”.)

“Source code” for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

### **B.2.2 Section 1**

You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

### **B.2.3 Section 2**

You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a. The modified work must itself be a software library.
- b. You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

- c. You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.
- d. If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

## **B.2.4 Section 3**

You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

## **B.2.5 Section 4**

You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

## **B.2.6 Section 5**

A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a “work that uses the Library”. Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a “work that uses the Library” with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a “work that uses the library”. The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a “work that uses the Library” uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

## **B.2.7 Section 6**

As an exception to the Sections above, you may also combine or link a “work that uses the Library” with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

- a. Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable “work that uses the Library”, as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

- b. Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.
- c. Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.
- d. If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.
- e. Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the “work that uses the Library” must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

## **B.2.8 Section 7**

You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

- a. Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.
- b. Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

## **B.2.9 Section 8**

You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library 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.

### **B.2.10 Section 9**

You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

### **B.2.11 Section 10**

Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

### **B.2.12 Section 11**

If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

### **B.2.13 Section 12**

If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.



### **B.2.14 Section 13**

The Free Software Foundation may publish revised and/or new versions of the Lesser 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 Library specifies a version number of this License which applies to it and “any later version”, you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

### **B.2.15 Section 14**

If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

### **B.2.16 NO WARRANTY Section 15**

BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY “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 LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

### **B.2.17 Section 16**

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY 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 LIBRARY (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 LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

## B.3 How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the “copyright” line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.> Copyright (C) <year>  
<name of author>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library 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 Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a “copyright disclaimer” for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library `Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990 Ty Coon, President of Vice

That's all there is to it!

---

# Appendix C. GNU Free Documentation License

Version 1.2, November 2002  
Copyright © 2000,2001,2002 Free Software Foundation, Inc.

Free Software Foundation, Inc.  
51 Franklin St, Fifth Floor,  
Boston,  
MA  
02110-1301  
USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Version 1.2, November 2002

## C.1 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. Secondly, 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.

## C.2 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.

## C.3 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.

## C.4 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.

## C.5 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 distri-

bution 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.

## C.6 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".

## C.7 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.

## C.8 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.

## C.9 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.

## C.10 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.

## C.11 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.

## C.12 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:

### **Sample Invariant Sections list**

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 1.2 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:

### **Sample Invariant Sections list**

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.