

# **Sourcery CodeBench Lite**

**MIPS Mentor Embedded GNU/Linux**

**Sourcery CodeBench Lite 2012.03-74**

**Getting Started**



---

# **Sourcery CodeBench Lite: MIPS Mentor Embedded GNU/Linux: Sourcery CodeBench Lite 2012.03-74: Getting Started**

CodeSourcery, Inc.

Copyright © 2005, 2006, 2007, 2008, 2009, 2010, 2011 CodeSourcery, Inc.

All rights reserved.

## **Abstract**

This guide explains how to install and build applications with Sourcery CodeBench Lite, CodeSourcery's customized and validated version of the GNU Toolchain. Sourcery CodeBench Lite includes everything you need for application development, including C and C++ compilers, assemblers, linkers, and libraries.

When you have finished reading this guide, you will know how to use Sourcery CodeBench from the command line.

---

---

# Table of Contents

Preface .....	iv
1. Intended Audience .....	v
2. Organization .....	v
3. Typographical Conventions .....	v
1. Quick Start .....	1
1.1. Installation and Set-Up .....	2
1.2. Configuring Sourcery CodeBench Lite for the Target System .....	2
1.3. Building Your Program .....	2
1.4. Running and Debugging Your Program .....	2
2. Installation and Configuration .....	4
2.1. Terminology .....	5
2.2. System Requirements .....	5
2.3. Downloading an Installer .....	6
2.4. Installing Sourcery CodeBench Lite .....	6
2.5. Installing Sourcery CodeBench Lite Updates .....	7
2.6. Setting up the Environment .....	7
2.7. Uninstalling Sourcery CodeBench Lite .....	8
3. Sourcery CodeBench Lite for MIPS Mentor Embedded GNU/Linux .....	9
3.1. Included Components and Features .....	10
3.2. Library Configurations .....	10
3.3. Target Architectures .....	11
3.4. Target Kernel Requirements .....	11
3.5. Using Sourcery CodeBench Lite on Mentor Embedded GNU/Linux Tar-gets .....	11
3.6. Using GDB Server for Debugging .....	14
4. Using Sourcery CodeBench from the Command Line .....	16
4.1. Building an Application .....	17
4.2. Running Applications on the Target System .....	17
4.3. Running Applications from GDB .....	18
5. Next Steps with Sourcery CodeBench .....	19
5.1. Sourcery CodeBench Knowledge Base .....	20
5.2. Example Programs .....	20
5.3. Manuals for GNU Toolchain Components .....	20
A. Sourcery CodeBench Lite Release Notes .....	22
A.1. Changes in Sourcery CodeBench Lite for MIPS Mentor Embedded GNU/Linux .....	23
B. Sourcery CodeBench Lite Licenses .....	24
B.1. Licenses for Sourcery CodeBench Lite Components .....	25
B.2. Sourcery CodeBench Software License Agreement .....	26
B.3. Attribution .....	29

---

# Preface

This preface introduces the Sourcery CodeBench Lite Getting Started guide. It explains the structure of this guide and describes the documentation conventions used.

# 1. Intended Audience

This guide is written for people who will install and/or use Sourcery CodeBench Lite. This guide provides a step-by-step guide to installing Sourcery CodeBench Lite and to building simple applications. Parts of this document assume that you have some familiarity with using the command-line interface.

# 2. Organization

This document is organized into the following chapters and appendices:

Chapter 1, “Quick Start”	This chapter includes a brief checklist to follow when installing and using Sourcery CodeBench Lite for the first time. You may use this chapter as an abbreviated guide to the rest of this manual.
Chapter 2, “Installation and Configuration”	This chapter describes how to download, install and configure Sourcery CodeBench Lite. This section describes the available installation options and explains how to set up your environment so that you can build applications.
Chapter 3, “Sourcery CodeBench Lite for MIPS Mentor Embedded GNU/Linux”	This chapter contains information about using Sourcery CodeBench Lite that is specific to MIPS Mentor Embedded GNU/Linux targets. You should read this chapter to learn how to best use Sourcery CodeBench Lite on your target system.
Chapter 4, “Using Sourcery CodeBench from the Command Line”	This chapter explains how to build applications with Sourcery CodeBench 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.
Chapter 5, “Next Steps with Sourcery CodeBench”	This chapter describes where you can find additional documentation and information about using Sourcery CodeBench Lite and its components. It also provides information about Sourcery CodeBench subscriptions. CodeSourcery customers with Sourcery CodeBench subscriptions receive comprehensive support for Sourcery CodeBench.
Appendix A, “Sourcery CodeBench Lite Release Notes”	This appendix contains information about changes in this release of Sourcery CodeBench Lite for MIPS Mentor Embedded GNU/Linux. You should read through these notes to learn about new features and bug fixes.
Appendix B, “Sourcery CodeBench Lite Licenses”	This appendix provides information about the software licenses that apply to Sourcery CodeBench Lite. Read this appendix to understand your legal rights and obligations as a user of Sourcery CodeBench Lite.

# 3. Typographical Conventions

The following typographical conventions are used in this guide:

<code>&gt; command arg ...</code>	A command, typed by the user, and its output. The “>” character is the command prompt.
<code>command</code>	The name of a program, when used in a sentence, rather than in literal input or output.
<code>literal</code>	Text provided to or received from a computer program.
<code>placeholder</code>	Text that should be replaced with an appropriate value when typing a command.
<code>\</code>	At the end of a line in command or program examples, indicates that a long line of literal input or output continues onto the next line in the document.

---

# Chapter 1

## Quick Start

This chapter includes a brief checklist to follow when installing and using Sourcery CodeBench Lite for the first time. You may use this chapter as an abbreviated guide to the rest of this manual.

Sourcery CodeBench Lite for MIPS Mentor Embedded GNU/Linux is intended for developers working on embedded Mentor Embedded GNU/Linux applications. It may also be used for Linux kernel development and debugging, or to build a Mentor Embedded GNU/Linux distribution.

Follow the steps given in this chapter to install Sourcery CodeBench Lite and build and run your first application program. The checklist given here is not a tutorial and does not include detailed instructions for each step; however, it will help guide you to find the instructions and reference information you need to accomplish each step. Note that this checklist is also oriented towards application debugging rather than kernel debugging.

You can find additional details about the components, libraries, and other features included in this version of Sourcery CodeBench Lite in Chapter 3, “Sourcery CodeBench Lite for MIPS Mentor Embedded GNU/Linux”.

## 1.1. Installation and Set-Up

**Install Sourcery CodeBench Lite on your host computer.** You may download an installer package from the Sourcery CodeBench web site<sup>1</sup>, or you may have received an installer on CD. The installer is an executable program that pops up a window on your computer and leads you through a series of dialogs to configure your installation. When the installation is complete, it offers to launch the Getting Started guide. For more information about installing Sourcery CodeBench Lite, including host system requirements and tips to set up your environment after installation, refer to Chapter 2, “Installation and Configuration”.

## 1.2. Configuring Sourcery CodeBench Lite for the Target System

**Identify your target libraries.** Sourcery CodeBench Lite supports libraries optimized for different targets. Libraries are selected automatically by the linker, depending on the processor and other options you have specified. Refer to Section 3.2, “Library Configurations” for details.

**Install runtime libraries on your target machine.** In order to run programs built with the Sourcery CodeBench runtime libraries on target hardware, you must install these libraries, the Sourcery CodeBench dynamic linker, and other runtime support files -- collectively referred to as the *sysroot* -- on your Mentor Embedded GNU/Linux target system. Typically, this involves either using third-party tools to build a complete root filesystem including the Sourcery CodeBench sysroot, or using special commands when linking or running your program so it can find the sysroot installed in another location on the target. Refer to Section 3.5, “Using Sourcery CodeBench Lite on Mentor Embedded GNU/Linux Targets” for full discussion of these options.

## 1.3. Building Your Program

**Build your program with Sourcery CodeBench command-line tools.** Create a simple test program, and follow the directions in Chapter 4, “Using Sourcery CodeBench from the Command Line” to compile and link it using Sourcery CodeBench Lite.

## 1.4. Running and Debugging Your Program

The steps to run or debug your program depend on your target system and how it is configured. Choose the appropriate method for your target.

---

<sup>1</sup> <http://go.mentor.com/codebench/>



**Run your program on the MIPS Mentor Embedded GNU/Linux target.** To run a program using the included Sourcery CodeBench libraries, you must install the sysroot on the target, as previously discussed. Copy the executable for your program to the target system. The method you use for launching your program depends on how you have installed the libraries and built your program. In some cases, you may need to invoke the Sourcery CodeBench dynamic linker explicitly. Refer to Section 3.5, “Using Sourcery CodeBench Lite on Mentor Embedded GNU/Linux Targets” for details.

**Debug your program on the target using GDB server.** You can use GDB server on a remote target to debug your program. When debugging a program that uses the included Sourcery CodeBench libraries, you must use the `gdbserver` executable included in the sysroot, and similar issues with respect to the dynamic linker as discussed previously apply. See Section 3.6, “Using GDB Server for Debugging” for detailed instructions. Once you have started GDB server on the target, you can connect to it from the debugger on your host system. Refer to Section 4.3, “Running Applications from GDB” for instructions on remote debugging from command-line GDB.

---

# Chapter 2

## Installation and Configuration

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

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

## 2.1. Terminology

Throughout this document, the term *host system* refers to the system on which you run Sourcery CodeBench while the term *target system* refers to the system on which the code produced by Sourcery CodeBench runs. The target system for this version of Sourcery CodeBench is `mips-mentor-linux-gnu`.

If you are developing a workstation or server application to run on the same system that you are using to run Sourcery CodeBench, 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.

## 2.2. System Requirements

### 2.2.1. Host Operating System Requirements

This version of Sourcery CodeBench supports the following host operating systems and architectures:

- GNU/Linux systems using IA32, AMD64, or Intel 64 processors, including Debian 3.1 (and later), Red Hat Enterprise Linux 3 (and later), SuSE Enterprise Linux 8 (and later), and Ubuntu 8.04 (and later).

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

#### Installing on Ubuntu and Debian GNU/Linux Hosts

The Sourcery CodeBench graphical installer is incompatible with the `dash` shell, which is the default `/bin/sh` for recent releases of the Ubuntu and Debian GNU/Linux distributions. To install Sourcery CodeBench Lite on these systems, you must make `/bin/sh` a symbolic link to one of the supported shells: `bash`, `csh`, `tcsh`, `zsh`, or `ksh`.

For example, on Ubuntu systems, the recommended way to do this is:

```
> sudo dpkg-reconfigure -plow dash
Install as /bin/sh? No
```

This is a limitation of the installer and uninstaller only, not of the installed Sourcery CodeBench Lite toolchain.

### 2.2.2. Host Hardware Requirements

In order to install and use Sourcery CodeBench Lite, you must have at least 512MB of available memory.

The amount of disk space required for a complete Sourcery CodeBench Lite installation directory depends on the host operating system and the number of target libraries included. When you start the graphical installer, it checks whether there is sufficient disk space before beginning to install. Note that the graphical installer also requires additional temporary disk space during the installation process. On Linux hosts, the installer puts temporary files in the directory specified by the `IATEMPDIR` environment variable, or `/tmp` if that is not set.

### 2.2.3. Target System Requirements

See Chapter 3, “Sourcery CodeBench Lite for MIPS Mentor Embedded GNU/Linux” for requirements that apply to the target system.

## 2.3. Downloading an Installer

If you have received Sourcery CodeBench Lite on a CD, or other physical media, then you do not need to download an installer. You may skip ahead to Section 2.4, “Installing Sourcery CodeBench Lite”.

You can download Sourcery CodeBench Lite from the Sourcery CodeBench web site<sup>1</sup>. This free version of Sourcery CodeBench, which is made available to the general public, does not include all the functionality of CodeSourcery's product releases. If you prefer, you may instead purchase or register for an evaluation of CodeSourcery's product toolchains at the Sourcery CodeBench Portal<sup>2</sup>.

Once you have navigated to the appropriate web site, download the installer that corresponds to your host operating system. For GNU/Linux systems Sourcery CodeBench Lite is provided as an executable installer package with the `.bin` extension. You may also install from a compressed archive with the `.tar.bz2` extension.

On GNU/Linux systems, save the download package in your home directory.

## 2.4. Installing Sourcery CodeBench Lite

The method used to install Sourcery CodeBench Lite depends on your host system and the kind of installation package you have downloaded.

### 2.4.1. Using the Sourcery CodeBench Lite Installer on GNU/Linux Hosts

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 CodeBench Lite.

If you prefer, or if your host system does not run the X Window System, you can run the installer in console mode rather than using the graphical interface. To do this, invoke the installer with the `-i console` command-line option. For example:

```
> /bin/sh ./path/to/package.bin -i console
```

### 2.4.2. Installing Sourcery CodeBench Lite from a Compressed Archive

You do not need to be a system administrator to install Sourcery CodeBench Lite from a compressed archive. You may install Sourcery CodeBench 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 CodeBench Lite in the `$HOME/CodeSourcery` subdirectory of your home directory and that the filename of the package you have downloaded is `/path/to/package.tar.bz2`. After installation the toolchain will be in `$HOME/CodeSourcery/sourceryg++-2012.03`.

First, uncompress the package file:

---

<sup>1</sup> <http://go.mentor.com/codebench/>

<sup>2</sup> <https://sourcery.mentor.com/GNUToolchain/>

```
> 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
```

## 2.5. Installing Sourcery CodeBench Lite Updates

If you have already installed an earlier version of Sourcery CodeBench Lite for MIPS Mentor Embedded GNU/Linux on your system, it is not necessary to uninstall it before using the installer to unpack a new version in the same location. The installer detects that it is performing an update in that case.

If you are installing an update from a compressed archive, it is recommended that you remove any previous installation in the same location, or install in a different directory.

Note that the names of the Sourcery CodeBench commands for the MIPS Mentor Embedded GNU/Linux target all begin with `mips-mentor-linux-gnu`. This means that you can install Sourcery CodeBench for multiple target systems in the same directory without conflicts.

## 2.6. Setting up the Environment

As with the installation process itself, the steps required to set up your environment depend on your host operating system.

### 2.6.1. Setting up the Environment on GNU/Linux Hosts

If you installed Sourcery CodeBench Lite using the graphical installer then you may skip this step. The installer does this setup for you.

Before using Sourcery CodeBench Lite you should add it 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 installdir/bin:$PATH
```

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

```
> PATH=installdir/bin:$PATH  
> export PATH
```

If you are not sure which shell you are using, try both commands. In both cases, replace `installdir` with the full pathname of your Sourcery CodeBench Lite installation directory.

You may also wish to set the `MANPATH` environment variable so that you can access the Sourcery CodeBench manual pages, which provide additional information about using Sourcery CodeBench. To set the `MANPATH` environment variable, follow the same steps shown above, replacing `PATH`

with `MANPATH`, and `bin` with `share/doc/sourceryg++-mips-mentor-linux-gnu/man`.

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

```
> mips-mentor-linux-gnu-g++ -v
```

Verify that the last line of the output contains: `Sourcery CodeBench Lite 2012.03-74`.

## 2.7. Uninstalling Sourcery CodeBench Lite

The method used to uninstall Sourcery CodeBench Lite depends on the method you originally used to install it. 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. In particular, the `mips-mentor-linux-gnu` directory located in the `install` directory will be removed entirely by the uninstaller.

### 2.7.1. Using the Sourcery CodeBench Lite Uninstaller on GNU/Linux

You should use the provided uninstaller to remove a Sourcery CodeBench Lite installation originally created by the executable installer script. Start the graphical uninstaller by invoking the executable `Uninstall` shell script located in your installation directory. After the uninstaller starts, follow the on-screen dialogs to uninstall Sourcery CodeBench Lite.

You can run the uninstaller in console mode, rather than using the graphical interface, by invoking the `Uninstall` script with the `-i console` command-line option.

### 2.7.2. Uninstalling a Compressed Archive Installation

If you installed Sourcery CodeBench Lite from a `.tar.bz2` file, you can uninstall it by manually deleting the installation directory created in the `install` procedure.

---

# **Chapter 3**

## **Sourcery CodeBench Lite for MIPS Mentor Embedded GNU/Linux**

This chapter contains information about features of Sourcery CodeBench Lite that are specific to MIPS Mentor Embedded GNU/Linux targets. You should read this chapter to learn how to best use Sourcery CodeBench Lite on your target system.

## 3.1. Included Components and Features

This section briefly lists the important components and features included in Sourcery CodeBench Lite for MIPS Mentor Embedded GNU/Linux, and tells you where you may find further information about these features.

Component	Version	Notes
<b>GNU programming tools</b>		
GNU Compiler Collection	4.6.3	Separate manual included.
GNU Binary Utilities	2.21.53	Includes assembler, linker, and other utilities. Separate manuals included.
<b>Debugging support and simulators</b>		
GNU Debugger	7.2.50	Separate manual included.
GDB Server	N/A	Included with GDB. See Section 3.6, “Using GDB Server for Debugging”.
<b>Target libraries</b>		
GNU C Library	2.15	Separate manual included.
Linux Kernel Headers	3.2.10	
<b>Other utilities</b>		
GNU Make	N/A	Build support on Windows hosts.
GNU Core Utilities	N/A	Build support on Windows hosts.

## 3.2. Library Configurations

Sourcery CodeBench Lite for MIPS Mentor Embedded GNU/Linux includes the following library configuration.

<b>MIPS32 revision 2 - Big-Endian, O32</b>	
Command-line option(s):	default
Sysroot subdirectory:	./
Dynamic linker:	lib/ld.so.1

<b>MIPS32 revision 2 - Little-Endian, O32</b>	
Command-line option(s):	-EL
Sysroot subdirectory:	el/
Dynamic linker:	lib/ld.so.1

<b>MIPS64 revision 2 - Big-Endian, N64</b>	
Command-line option(s):	-mabi=64
Sysroot subdirectory:	./
Dynamic linker:	lib64/ld.so.1



<b>MIPS64 revision 2 - Little-Endian, N64</b>	
Command-line option(s):	-EL -mabi=64
Sysroot subdirectory:	el/
Dynamic linker:	lib64/ld.so.1

<b>XLP - Big-Endian, N64</b>	
Command-line option(s):	-march=xlp -mabi=64
Sysroot subdirectory:	xlp/
Dynamic linker:	lib64/ld.so.1

Sourcery CodeBench includes copies of run-time libraries that have been built with optimizations for different target architecture variants or other sets of build options. Each such set of libraries is referred to as a *multilib*. When you link a target application, Sourcery CodeBench selects the multilib matching the build options you have selected.

Each multilib corresponds to a *sysroot* directory which contains the files that should be installed on the target system. The *sysroot* contains the dynamic linker used to run your applications on the target as well as the libraries. Refer to Section 3.5, “Using Sourcery CodeBench Lite on Mentor Embedded GNU/Linux Targets” for instructions on how to install and use these support files on your target Mentor Embedded GNU/Linux system. You can find the *sysroot* directories provided with Sourcery CodeBench in the `mips-mentor-linux-gnu/libc` directory of your installation. In the tables below, the dynamic linker pathname is given relative to the corresponding *sysroot*.

## 3.3. Target Architectures

By default, Sourcery CodeBench Lite for MIPS Mentor Embedded GNU/Linux generates code for MIPS32r2 processors. If you wish to generate code for another MIPS processor, you must use an appropriate `-march` option when you build your application. Refer to the GCC manual for additional information about supported targets.

## 3.4. Target Kernel Requirements

The GNU C library supplied with this version of Sourcery CodeBench Lite requires that Linux kernel version 2.6.12 or later be installed on the target in order to run applications.

To support hardware watchpoints with `gdbserver`, Linux 2.6.28 or later is required.

## 3.5. Using Sourcery CodeBench Lite on Mentor Embedded GNU/Linux Targets

In order to run and debug programs produced by Sourcery CodeBench on a Mentor Embedded GNU/Linux target, you must install runtime support files on the target. You may also need to set appropriate build options so that your executables can find the correct dynamic linker and libraries at runtime.

The runtime support files, referred to as the *sysroot*, are found in the `mips-mentor-linux-gnu/libc` directory of your Sourcery CodeBench Lite installation. The *sysroot* consists of the contents of the `etc`, `lib`, `sbin`, and `usr` directories. There may be other directories in `mips-mentor-linux-gnu/libc` that contain additional *sysroots* customized for particular

combinations of command-line compiler flags, or *multilibs*. Refer to Section 3.2, “Library Configurations” for a list of the included multilibs in this version of Sourcery CodeBench Lite, and the corresponding sysroot directory pathnames.

You have these choices for installing the sysroot on the target:

- You can install the files in the filesystem root on the target (that is, installing the files directly in `/etc/`, `/lib/`, and so on). All applications on the target then automatically use the Sourcery CodeBench libraries. This method is primarily useful when you are building a Mentor Embedded GNU/Linux root filesystem from scratch. If your target board already has a Mentor Embedded GNU/Linux filesystem installed, overwriting the existing C library files is not recommended, as this may break other applications on your system, or cause it to fail to boot.
- You can install the sysroot in an alternate location and build your application with the `-rpath` and `--dynamic-linker` linker options to specify the sysroot location.
- You can install the sysroot in an alternate location and explicitly invoke your application through the dynamic linker to specify the sysroot location. If you are just getting started with Sourcery CodeBench Lite, this may be the easiest way to get your application running, but this method does not support use of the debugger.

Setting the environment variable `LD_LIBRARY_PATH` on the target is not sufficient, since executables produced by Sourcery CodeBench depend on the Sourcery CodeBench dynamic linker included in the sysroot as well as the Sourcery CodeBench runtime libraries.

### 3.5.1. Installing the Sysroot

If you are modifying an existing system, rather than creating a new system from scratch, you should place the sysroot files in a new directory, rather than in the root directory of your target system.

If you choose to overwrite your existing C library, you may not be able to boot your system. You should back up your existing system before overwriting the C library and ensure that you can restore the backup even with your system offline.

The next step is to identify the correct sysroot subdirectory in the Sourcery CodeBench Lite install directory on your host system. The sysroot you copy to the target must be the one that corresponds to the linker options you are using to build your applications. The tables in Section 3.2, “Library Configurations” tell you which sysroot subdirectories correspond to which sets of command-line options. From the command line, you can identify the appropriate sysroot for your program by invoking the compiler with `-print-sysroot` added to your other build options. This causes GCC to print the host sysroot pathname and exit.

The mechanism you use for copying the sysroot to your target board depends on its hardware and software configuration. You may be able to use FTP or SSH with a server already running on your target. If your target board does not have networking configured, you may be able to copy files using an SD card or USB memory stick, or via a file transfer utility over a serial line. The instructions that come with your board may include specific suggestions.

When running Sourcery CodeBench on a GNU/Linux host, as an alternative to copying files to the target system, you may be able to NFS-mount the Sourcery CodeBench Lite installation directory from your host system on the target system. It is especially convenient for debugging if you can make the sysroot pathname on the target system be identical to that on the GNU/Linux host system; refer to Section 3.6.3, “Setting the Sysroot in the Debugger” for further discussion of this issue.

Otherwise, you must copy files from the appropriate `sysroot` subdirectory in the `mips-mentor-linux-gnu/libc` directory of your Sourcery CodeBench Lite install to the target system. In many cases, you do not need to copy all of the files in the `sysroot`. For example, the `usr/include` subdirectory contains files that are only needed if you will actually be running the compiler on your target system. You do not need these files for non-native compilers. You also do not need any `.o` or `.a` files; these are used by the compiler when linking programs, but are not needed to run programs. You should definitely copy all `.so` files and the executable files in `usr/bin` and `sbin`.

### 3.5.2. Using Linker Options to Specify the Sysroot Location

If you have installed the `sysroot` on the target in a location other than the file system root, you can use the `-rpath` and `--dynamic-linker` linker options to specify the `sysroot` location.

If you are using Sourcery CodeBench from the command line, follow these steps:

1. First find the correct `sysroot` directory, dynamic linker, and library subdirectory for your selected multilib. Refer to Section 3.2, “Library Configurations”. In the following steps, `sysroot` is the absolute path to the `sysroot` directory on the target corresponding to your selected multilib. For the default multilib, the dynamic linker path relative to the `sysroot` is `lib/ld.so.1`, and the library subdirectory is `lib`. This is used in the example below.
2. When invoking `mips-mentor-linux-gnu-gcc` to link your executable, include the command-line options:

```
-Wl,-rpath=sysroot/lib:sysroot/usr/lib \  
-Wl,--dynamic-linker=sysroot/lib/ld.so.1
```

where `sysroot` is the absolute path to the `sysroot` directory on the target corresponding to your selected multilib.

3. Copy the executable to the target and execute it normally.

Note that if you specify an incorrect path for `--dynamic-linker`, the common failure mode seen when running your application on the target is similar to

```
> ./factorial  
./factorial: No such file or directory
```

or

```
> ./factorial  
./factorial: bad ELF interpreter: No such file or directory
```

This can be quite confusing since it appears from the error message as if it is the `./factorial` executable that is missing rather than the dynamic linker it references.

### 3.5.3. Specifying the Sysroot Location at Runtime

You can invoke the Sourcery CodeBench dynamic linker on the target to run your application without having to compile it with specific linker options.

To do this, follow these steps:

1. Build your application on the host, without any additional linker options, and copy the executable to your target system.

2. Find the correct sysroot directory, dynamic linker, and library subdirectory for your selected multilib. Refer to Section 3.2, “Library Configurations”. In the following steps, *sysroot* is the absolute path to the sysroot directory on the target corresponding to your selected multilib. For the default multilib, the dynamic linker is `lib/ld.so.1`, and the library subdirectory is `lib`. This is used in the example below.
3. On the target system, invoke the dynamic linker with your executable as:

```
> sysroot/lib/ld.so.1 \  
  --library-path sysroot/lib:sysroot/usr/lib \  
  /path/to/your-executable
```

where *sysroot* is the absolute path to the sysroot directory on the target corresponding to your selected multilib.

Invoking the linker in this manner requires that you provide either an absolute pathname to your executable, or a relative pathname prefixed with `./`. Specifying only the name of a file in the current directory does not work.

## 3.6. Using GDB Server for Debugging

The GDB server utility provided with Sourcery CodeBench Lite can be used to debug a Mentor Embedded GNU/Linux application. While Sourcery CodeBench runs on your host system, `gdbserver` and the target application run on your target system. Even though Sourcery CodeBench and your application run on different systems, the debugging experience when using `gdbserver` is very similar to debugging a native application.

### 3.6.1. Running GDB Server

The GDB server executables are included in the sysroot in ABI-specific subdirectories of *sysroot/usr*. Use the executable from the sysroot and library subdirectory that match your program. See Section 3.2, “Library Configurations” for details.

You must copy the sysroot to your target system as described in Section 3.5.1, “Installing the Sysroot”. You must also copy the executable you want to debug to your target system.

If you have installed the sysroot in the root directory of the filesystem on the target, you can invoke `gdbserver` as:

```
> gdbserver :10000 program arg1 arg2 ...
```

where *program* is the path to the program you want to debug and *arg1 arg2 ...* are the arguments you want to pass to it. The `:10000` argument indicates that `gdbserver` should listen for connections from GDB on port 10000. You can use a different port, if you prefer.

If you have installed the sysroot in an alternate directory, invoking `gdbserver` becomes more complicated. You must build your application using the link-time options to specify the location of the sysroot, as described in Section 3.5.2, “Using Linker Options to Specify the Sysroot Location”. You must also invoke `gdbserver` itself using the dynamic linker provided in the Sourcery CodeBench sysroot, as described in Section 3.5.3, “Specifying the Sysroot Location at Runtime”. In other words, the command to invoke `gdbserver` in this case would be similar to:

```
> sysroot/lib/ld.so.1 \  
  --library-path sysroot/lib:sysroot/usr/lib \  
  sysroot/usr/lib/bin/gdbserver :10000 program arg1 arg2 ...
```

### 3.6.2. Connecting to GDB Server from the Debugger

You can connect to GDB server by using the following command from within GDB:

```
(gdb) target remote target:10000
```

where *target* is the host name or IP address of your target system.

When your program exits, `gdbserver` exits too. If you want to debug the program again, you must restart `gdbserver` on the target. Then, in GDB, reissue the `target` command shown above.

### 3.6.3. Setting the Sysroot in the Debugger

In order to debug shared libraries, GDB needs to map the pathnames of shared libraries on the target to the pathnames of equivalent files on the host system. Debugging of multi-threaded applications also depends on correctly locating copies of the libraries provided in the sysroot on the host system.

In some situations, the target pathnames are valid on the host system. Otherwise, you must tell GDB how to map target pathnames onto the equivalent host pathnames.

In the general case, there are two GDB commands required to set up the mapping:

```
(gdb) set sysroot-on-target target-pathname  
(gdb) set sysroot host-pathname
```

This causes GDB to replace all instances of the *target-pathname* prefix in shared library pathnames reported by the target with *host-pathname* to get the location of the equivalent library on the host.

If you have installed the sysroot in the root filesystem on the target, you can omit the `set sysroot-on-target` command, and use only `set sysroot` to specify the location on the host system.

Refer to Section 3.5.1, “Installing the Sysroot” for more information about installing the sysroot on the target. Note that if you have installed a stripped copy of the provided libraries on the target, you should give GDB the location of an unstripped copy on the host.

---

# **Chapter 4**

## **Using Sourcery CodeBench from the Command Line**

This chapter demonstrates the use of Sourcery CodeBench Lite from the command line.

## 4.1. Building an Application

This chapter explains how to build an application with Sourcery CodeBench Lite using the command line. As elsewhere in this manual, this section assumes that your target system is `mips-mentor-linux-gnu`, as indicated by the `mips-mentor-linux-gnu` command prefix.

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

```
#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;
}
```

Compile and link this program using the command:

```
> mips-mentor-linux-gnu-gcc -o factorial main.c
```

There should be no output from the compiler. (If you are building a C++ application, instead of a C application, replace `mips-mentor-linux-gnu-gcc` with `mips-mentor-linux-gnu-g++`.)

## 4.2. Running Applications on the Target System

You may need to install the Sourcery CodeBench runtime libraries and dynamic linker on the target system before you can run your application. Refer to Chapter 3, “Sourcery CodeBench Lite for MIPS Mentor Embedded GNU/Linux” for specific instructions.

To run your program on a Mentor Embedded GNU/Linux target system, use the command:

```
> factorial
```

You should see:

```
factorial(0) = 1
factorial(1) = 1
factorial(2) = 2
factorial(3) = 6
factorial(4) = 24
factorial(5) = 120
factorial(6) = 720
factorial(7) = 5040
```

```
factorial(8) = 40320  
factorial(9) = 362880
```

## 4.3. Running Applications from GDB

You can run GDB, the GNU Debugger, on your host system to debug programs running remotely on a target board or system.

When starting GDB, give it the pathname to the program you want to debug as a command-line argument. For example, if you have built the factorial program as described in Section 4.1, “Building an Application”, enter:

```
> mips-mentor-linux-gnu-gdb factorial
```

While this section explains the alternatives for using GDB to run and debug application programs, explaining the use of the GDB command-line interface is beyond the scope of this document. Please refer to the GDB manual for further instructions.

### 4.3.1. Connecting to an External GDB Server

Sourcery CodeBench Lite includes a program called `gdbserver` that can be used to debug a program running on a remote MIPS Mentor Embedded GNU/Linux target. Follow the instructions in Chapter 3, “Sourcery CodeBench Lite for MIPS Mentor Embedded GNU/Linux” to install and run `gdbserver` on your target system.

From within GDB, you can connect to a running `gdbserver` or other debugging stub that uses the GDB remote protocol using:

```
(gdb) target remote host:port
```

where *host* is the host name or IP address of the machine the stub is running on, and *port* is the port number it is listening on for TCP connections.



---

# **Chapter 5**

## **Next Steps with Sourcery**

### **CodeBench**

This chapter describes where you can find additional documentation and information about using Sourcery CodeBench Lite and its components.

## 5.1. Sourcery CodeBench Knowledge Base

The Sourcery CodeBench Knowledge Base is available to registered users at the Sourcery CodeBench Portal<sup>1</sup>. Here you can find solutions to common problems including installing Sourcery CodeBench, making it work with specific targets, and interoperability with third-party libraries. There are also additional example programs and tips for making the most effective use of the toolchain and for solving problems commonly encountered during debugging. The Knowledge Base is updated frequently with additional entries based on inquiries and feedback from customers.

## 5.2. Example Programs

Sourcery CodeBench Lite includes some bundled example programs. You can find the source code for these examples in the `share/sourceryg++-mips-mentor-linux-gnu-examples` directory of your Sourcery CodeBench installation.

### 5.2.1. Other Examples

The subdirectories contain a number of small, target-independent test programs. You may find these programs useful as self-contained test cases when experimenting with configuring the correct compiler and debugger settings for your target, or when learning how to use the debugger or other features of the Sourcery CodeBench toolchain.

## 5.3. Manuals for GNU Toolchain Components

Sourcery CodeBench Lite includes the full user manuals for each of the GNU toolchain components, such as the compiler, linker, assembler, and debugger. Most of the manuals include tutorial material for new users as well as serving as a complete reference for command-line options, supported extensions, and the like.

When you install Sourcery CodeBench Lite, links to both the PDF and HTML versions of the manuals are created in the shortcuts folder you select. If you elected not to create shortcuts when installing Sourcery CodeBench Lite, the documentation can be found in the `share/doc/sourceryg++-mips-mentor-linux-gnu/` subdirectory of your installation directory.

In addition to the detailed reference manuals, Sourcery CodeBench Lite includes a Unix-style manual page for each toolchain component. You can view these by invoking the `man` command with the pathname of the file you want to view. For example, you can first go to the directory containing the man pages:

```
> cd $INSTALL/share/doc/sourceryg++-mips-mentor-linux-gnu/man/man1
```

Then you can invoke `man` as:

```
> man ./mips-mentor-linux-gnu-gcc.1
```

Alternatively, if you use `man` regularly, you'll probably find it more convenient to add the directory containing the Sourcery CodeBench man pages to your `MANPATH` environment variable. This should go in your `.profile` or equivalent shell startup file; see Section 2.6, “Setting up the Environment” for instructions. Then you can invoke `man` with just the command name rather than a pathname.

---

<sup>1</sup> <https://sourcery.mentor.com/GNUToolchain/>

Finally, note that every command-line utility program included with Sourcery CodeBench Lite can be invoked with a `--help` option. This prints a brief description of the arguments and options to the program and exits without doing further processing.

---

# **Appendix A**

## **Sourcery CodeBench Lite Release Notes**

This appendix contains information about changes in this release of Sourcery CodeBench Lite for MIPS Mentor Embedded GNU/Linux. You should read through these notes to learn about new features and bug fixes.

## **A.1. Changes in Sourcery CodeBench Lite for MIPS Mentor Embedded GNU/Linux**

This section documents Sourcery CodeBench Lite changes for each released revision.

### **A.1.1. Changes in Sourcery CodeBench Lite 2012.03-74**

**Initial release.** This is the initial release for MIPS Mentor Embedded GNU/Linux.

---

# **Appendix B**

## **Sourcery CodeBench Lite**

### **Licenses**

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

## B.1. Licenses for Sourcery CodeBench Lite Components

The table below lists the major components of Sourcery CodeBench Lite for MIPS Mentor Embedded GNU/Linux and the license terms which apply to each of these components.

Some free or open-source components provide documentation or other files under terms different from those shown below. For definitive information about the license that applies to each component, consult the source package corresponding to this release of Sourcery CodeBench Lite. Sourcery CodeBench Lite may contain free or open-source components not included in the list below; for a definitive list, consult the source package corresponding to this release of Sourcery CodeBench Lite.

Component	License
GNU Compiler Collection	GNU General Public License 3.0 <a href="http://www.gnu.org/licenses/gpl.html">http://www.gnu.org/licenses/gpl.html</a>
GNU Binary Utilities	GNU General Public License 3.0 <a href="http://www.gnu.org/licenses/gpl.html">http://www.gnu.org/licenses/gpl.html</a>
GNU Debugger	GNU General Public License 3.0 <a href="http://www.gnu.org/licenses/gpl.html">http://www.gnu.org/licenses/gpl.html</a>
GNU C Library	GNU Lesser General Public License 2.1 <a href="http://www.gnu.org/licenses/old-licenses/lgpl-2.1.html">http://www.gnu.org/licenses/old-licenses/lgpl-2.1.html</a>
Linux Kernel Headers	GNU General Public License 2.0 <a href="http://www.gnu.org/licenses/old-licenses/gpl-2.0.html">http://www.gnu.org/licenses/old-licenses/gpl-2.0.html</a>
GNU Make	GNU General Public License 2.0 <a href="http://www.gnu.org/licenses/old-licenses/gpl-2.0.html">http://www.gnu.org/licenses/old-licenses/gpl-2.0.html</a>
GNU Core Utilities	GNU General Public License 2.0 <a href="http://www.gnu.org/licenses/old-licenses/gpl-2.0.html">http://www.gnu.org/licenses/old-licenses/gpl-2.0.html</a>

The CodeSourcery License is available in Section B.2, “Sourcery CodeBench Software License Agreement”.

### Important

Although some of the licenses that apply to Sourcery CodeBench 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 CodeBench Lite. You can develop proprietary applications and libraries with Sourcery CodeBench Lite.

Sourcery CodeBench Lite may include some third party example programs and libraries in the `share/sourceryg++-mips-mentor-linux-gnu-examples` subdirectory. These examples are not covered by the Sourcery CodeBench Software License Agreement. To the extent permitted by law, these examples are provided by CodeSourcery as is with no warranty of any kind, including implied warranties of merchantability or fitness for a particular purpose. Your use of each example is governed by the license notice (if any) it contains.

## B.2. Sourcery CodeBench™ Software License Agreement

1. **Parties.** The parties to this Agreement are you, the licensee (“You” or “Licensee”) and Mentor Graphics. If You are not acting on behalf of Yourself as an individual, then “You” means Your company or organization.
2. **The Software.** The Software licensed under this Agreement consists of computer programs and documentation referred to as Sourcery CodeBench™ Lite Edition (the “Software”).
3. **Definitions.**
  - 3.1. **Mentor Graphics Proprietary Components.** The components of the Software that are owned and/or licensed by Mentor Graphics and are not subject to a “free software” or “open source” license, such as the GNU Public License. The Mentor Graphics Proprietary Components of the Software include, without limitation, the Sourcery CodeBench Installer, any Sourcery CodeBench Eclipse plug-ins, the CodeSourcery C Library (CSLIBC), and any Sourcery CodeBench Debug Sprite. For a complete list, refer to the *Getting Started Guide* included with the 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.
  - 3.4. **Redistributable Components.** The Mentor Graphics Proprietary Components that are intended to be incorporated or linked into Licensee object code developed with the Software. The Redistributable Components of the Software include, without limitation, CSLIBC and the CodeSourcery Common Startup Code Sequence (CS3). For a complete list, refer to the *Getting Started Guide* included with the distribution.
4. **License Grant to Proprietary Components of the Software.** You are granted a non-exclusive, royalty-free license (a) to install and use the Mentor Graphics Proprietary Components of the Software, (b) to transmit the Mentor Graphics Proprietary Components over an internal computer network, (c) to copy the Mentor Graphics Proprietary Components for Your internal use only, and (d) to distribute the Redistributable Component(s) in binary form only and only as part of Licensee object code developed with the Software that provides substantially different functionality than the Redistributable Component(s).
5. **Restrictions.** You may not: (i) copy or permit others to use the Mentor Graphics Proprietary Components of the Software, except as expressly provided above; (ii) distribute the Mentor Graphics Proprietary Components of the Software to any third party, except as expressly provided above; or (iii) reverse engineer, decompile, or disassemble the Mentor Graphics 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 Mentor Graphics. Sourcery CodeBench includes components provided under various different licenses. The *Getting Started Guide* provides an overview of which license applies to different components, and, for compon-



ents subject to the Eclipse Public License, contains information on how to obtain the source code. Definitive licensing information for each “free software” or “open source” component is available in the relevant source file.

7. **Mentor Graphics 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 Mentor Graphics trademark, whether registered or unregistered, including without limitation, CodeSourcery™, Sourcery CodeBench™, the CodeSourcery crystal ball logo, or the Sourcery CodeBench 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 Mentor Graphics trademarks in the resulting binary.
8. **Term and Termination.** This Agreement shall remain in effect unless terminated pursuant to this provision. Mentor Graphics 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 Mentor Graphics 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 Mentor Graphics, 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.** Mentor Graphics owns and/or has licensed the Mentor Graphics 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 Mentor Graphics Proprietary Components of the Software are protected by United States copyright laws and international treaty provisions. Mentor Graphics 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.** MENTOR GRAPHICS AND ITS LICENSORS PROVIDE THE SOFTWARE “AS-IS” AND PROVIDED WITH ALL FAULTS. MENTOR GRAPHICS DOES NOT MAKE ANY WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. MENTOR GRAPHICS 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 MENTOR GRAPHICS

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 MENTOR GRAPHICS 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 MENTOR GRAPHICS' 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 s'y 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 Mentor Graphics and shall not have the power or authority to bind or obligate Mentor Graphics in any manner to any third party.
21. **Force Majeure.** Neither Mentor Graphics 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.

## B.3. Attribution

This version of Sourcery CodeBench Lite may include code based on work under the following copyright and permission notices:

### B.3.1. Android Open Source Project

```
/*
 * Copyright (C) 2008 The Android Open Source Project
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * * Redistributions of source code must retain the above copyright
 *   notice, this list of conditions and the following disclaimer.
 * * Redistributions in binary form must reproduce the above copyright
 *   notice, this list of conditions and the following disclaimer in
 *   the documentation and/or other materials provided with the
 *   distribution.
 *
 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
 * "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
```

## Sourcery CodeBench Lite Licenses

---

\* LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS  
\* FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE  
\* COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,  
\* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING,  
\* BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS  
\* OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED  
\* AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,  
\* OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT  
\* OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF  
\* SUCH DAMAGE.  
\*/