Sourcery™ CodeBench Lite

AMD64 GNU/Linux

Sourcery CodeBench Lite 2018.11-5

Getting Started

Sourcery™ CodeBench Lite: AMD64 GNU/Linux: Sourcery CodeBench Lite 2018.11-5: Getting Started

Mentor Graphics®, Inc.

Mentor Graphics Corporation
8005 S.W. Boeckman Road,
Wilsonville, Oregon 97070-7777

Contacting Mentor Graphics Corporation
Telephone: 503.685.7000
Toll-Free Telephone: 800.592.2210
Support Center: support.mentor.com

Send Feedback on Documentation: supportnet.mentor.com/doc_feedback_form

Abstract

This guide explains how to install and use Sourcery CodeBench Lite, Mentor Graphics'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, libraries, and debugging tools.

Table of Contents

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. 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, "Installation and Configuration"	This chapter describes how to download, install and configure Sourcery CodeBench Lite. This section documents host system requirements and explains how to set up your environment so that you can build and debug applications.
Chapter 2, "Contents of this Release"	This chapter lists the components and features included with this release of Sourcery CodeBench Lite for AMD64 GNU/Linux.
Chapter 3, "Sourcery CodeBench Lite for AMD64 GNU/Linux"	This chapter contains information about using Sourcery CodeBench Lite that is specific to AMD64 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.
Appendix A, "Sourcery CodeBench Lite Release Notes"	This appendix contains information about changes in this release of Sourcery CodeBench Lite for AMD64 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:

`> 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.
\	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.

4. Pathname Conventions

This document uses *$INSTALLDIR* to refer to the absolute path (including any drive prefix on Microsoft Windows hosts) of your Sourcery CodeBench toolchain installation. The locations of important files are documented relative to this path.

As used in this guide, the *$INSTALLDIR* may not be the top-level installation directory when the Sourcery CodeBench toolchain is bundled with other Mentor Graphics products.

Chapter 1
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.

1.1. 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. In general, when you are developing applications to run on an embedded processor, the host and target systems are different. The target system for this version of Sourcery CodeBench is `x86_64-amd-linux-gnu`.

1.2. System Requirements

1.2.1. Host Operating System Requirements

This version of Sourcery CodeBench includes installation packages supporting the following host operating systems and architectures:

X86_64 Windows. 64-bit Microsoft Windows operating systems:

- Windows 7
- Windows 8
- Windows 10

X86_64 GNU/Linux. 64-bit operating system distributions:

- Ubuntu Desktop 14.04, 16.04
- Red Hat Enterprise Linux 7.1
- CentOS 6.8, 7.1

Sourcery CodeBench command-line tools may also be compatible with other GNU/Linux distributions and versions using GLIBC version 2.11 or later, but no support is provided on those host systems.

Note that some components and features of Sourcery CodeBench Lite for AMD64 GNU/Linux may not be supported by all host packages, or may have more restrictive host system requirements than those listed above.

1.2.2. Host Hardware Requirements

The amount of disk space required for a complete Sourcery CodeBench Lite installation directory depends on the host operating system and the features included with the Sourcery CodeBench package. 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. If you want to specify the scratch directory for installer extraction, set the `P2_INSTALLER_TEMP_PATH` environment variable. Otherwise the system default location is used for these temporary files.

1.2.3. Target System Requirements

See Chapter 3, “Sourcery CodeBench Lite for AMD64 GNU/Linux” for requirements that apply to the target system.

1.3. Downloading an Installer

If you have received Sourcery CodeBench Lite on a DVD or other physical media, then you do not need to download an installer. You may skip ahead to Section 1.4, “Installing Sourcery CodeBench Lite”.

You can download Sourcery CodeBench Lite from the Sourcery CodeBench web site[1]. This free version of Sourcery CodeBench, which is made available to the general public, does not include all the functionality of Mentor Graphics's product releases.

Choose the installer that corresponds to your host operating system. For Microsoft Windows systems, the Sourcery CodeBench installer is provided as a self-extracting archive with the `.exe` extension. For GNU/Linux systems, Sourcery CodeBench Lite is provided as an executable installer package with the `.bin` extension.

Save the installer file to a convenient location such as the desktop or your home directory.

1.4. 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.

1.4.1. Using the Sourcery CodeBench Lite Installer

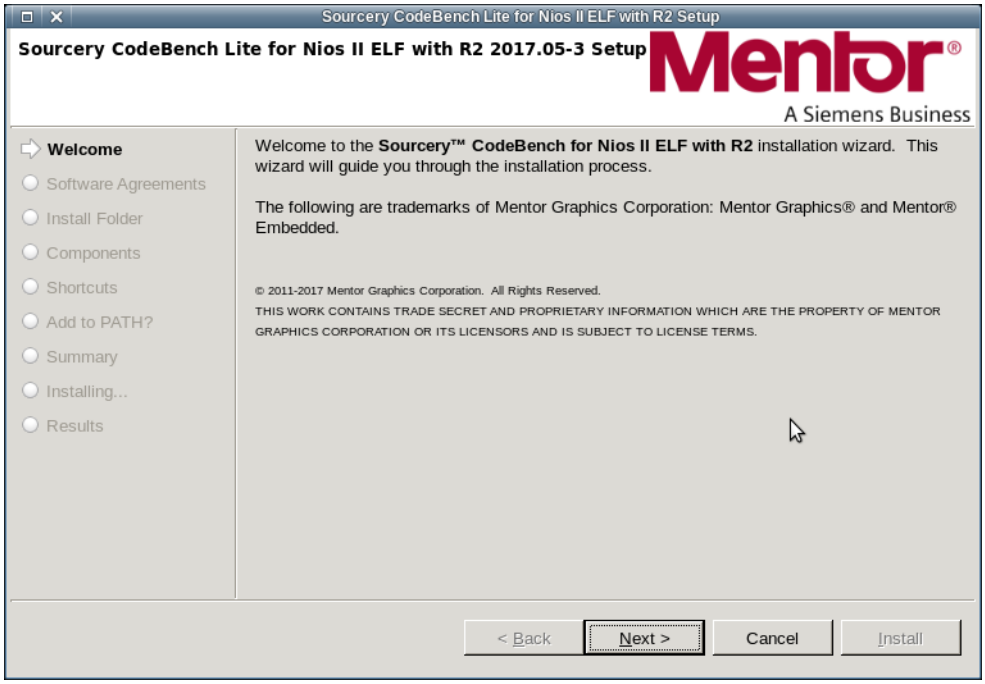
On Microsoft Windows hosts, start the installer by double-clicking on the `.exe` file.

On GNU/Linux hosts, start the graphical installer by invoking the executable shell script:

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

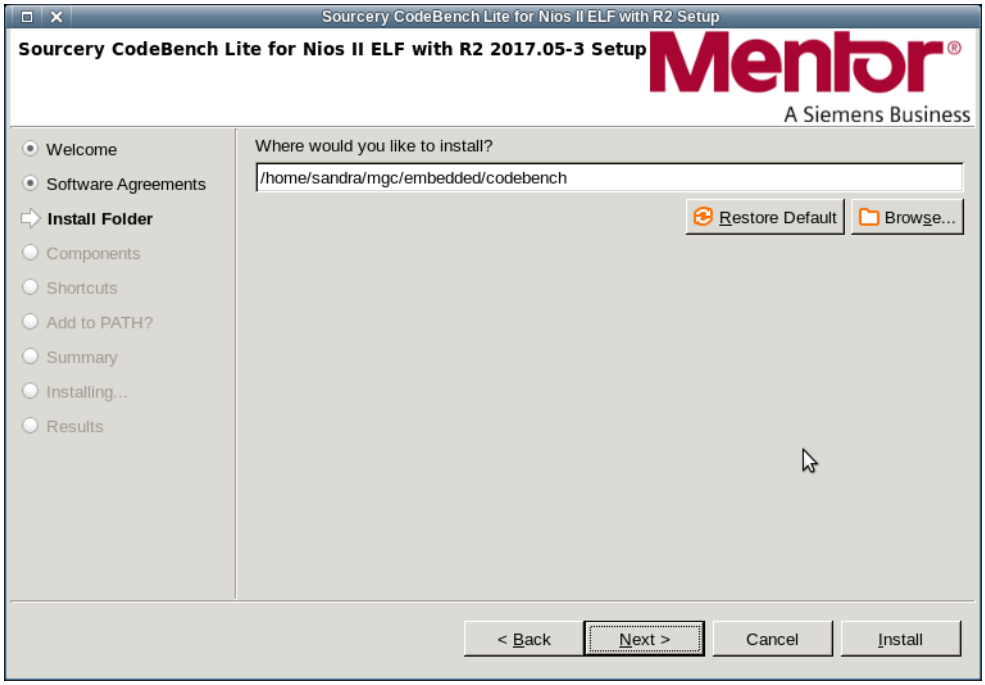
Follow the on-screen dialogs to install Sourcery CodeBench Lite. The installer is intended to be self-explanatory and on most pages the defaults are appropriate.

[1] http://go.mentor.com/codebench/



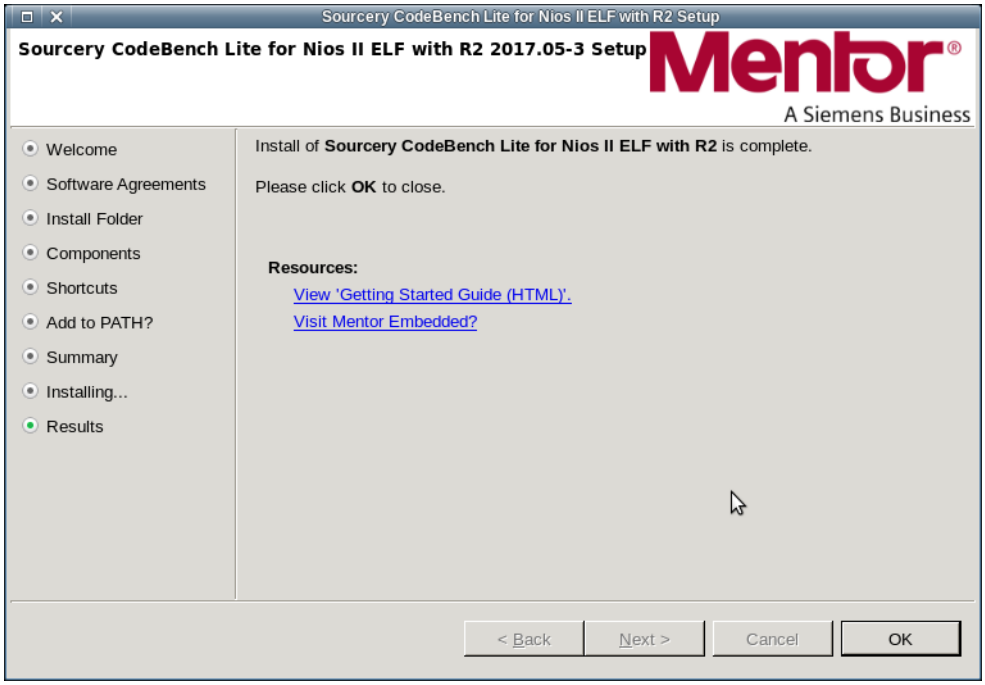
Running the Installer. The graphical installer guides you through the steps to install Sourcery CodeBench Lite.

You may want to change the install directory pathname. You can install multiple Sourcery CodeBench Lite products in the same directory, but the directory must not contain other files.



Choose Install Folder. Select the pathname for your install directory.

When the installer has finished, it asks if you want to launch a viewer for the Getting Started guide, or visit Mentor® Embedded.



Install Complete. You should see a screen similar to this after a successful install.

If you prefer, you can run the installer in console mode rather than using the graphical interface. To do this, invoke the installer with the `-console` command-line option. For example, on a Microsoft Windows host:

```
> /path/to/package.exe -console
```

On a GNU/Linux host, use a command similar to:

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

1.4.2. 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 `$HOME/CodeBench` subdirectory of your home directory and that the filename of the package you have downloaded is `/path/to/package.tar.bz2`. After installation the software will be in `$HOME/CodeBench/amd-2018.11`.

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/CodeBench
```

Change to the installation directory:

```
> cd $HOME/CodeBench
```

Unpack the package:

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

1.5. Installing Sourcery CodeBench Lite Updates

If you have an existing Sourcery CodeBench Lite for AMD64 GNU/Linux installation, when you start the graphical installer it asks you whether you want to upgrade your existing copy, or install the new version in a different location.

In some cases, the installer may be unable to update an older release due to incompatible changes in packaging. In this case you must either uninstall the previous release or choose a different installation directory.

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.

1.6. Setting up the Environment

1.6.1. Setting up the Environment on Microsoft Windows Hosts

1.6.1.1. Setting the PATH

In order to use the Sourcery CodeBench tools from the command line, you should add them to your PATH. The graphical installer for Sourcery CodeBench Lite normally does this setup for you, however it may not take effect until you next log in. This section describes how to set the PATH manually.

To set the PATH on a system running Microsoft Windows 7, from the desktop bring up the Start menu and right click on Computer. Select Properties and click on Advanced system settings. Go to the Advanced tab, then click on the Environment Variables button. Select the PATH variable and click Edit. Add the string *;$INSTALLDIR*\bin to the end, and click OK.

To set the PATH on a system running Microsoft Windows 8 or 10, open the System Control Panel. Click on Advanced system settings. Go to the Advanced tab, then click on the Environment Variables button. Select the PATH variable and click Edit. Add the string *;$INSTALLDIR*\bin to the end, and click OK.

You can verify that your PATH is set up correctly by starting a new cmd.exe shell and running:

```
> x86_64-amd-linux-gnu-gcc -v
```

Verify that the last line of the output contains: Sourcery CodeBench Lite 2018.11-5.

1.6.1.2. Working with Cygwin

Sourcery CodeBench Lite does not require Cygwin or any other UNIX emulation environment. You can use Sourcery CodeBench command-line tools directly from the Windows command shell. You can also use Sourcery CodeBench 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 CodeBench is not a Cygwin application, it does not, by default, recognize Cygwin paths.

If you are using Sourcery CodeBench from Cygwin, you should set the `CYGPATH` environment variable. If this environment variable is set, Sourcery CodeBench 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 CodeBench relies on the `cygpath` utility provided with Cygwin. You must provide Sourcery CodeBench with the full path to `cygpath` if `cygpath` is not in your `PATH`. For example:

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

directs Sourcery CodeBench 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.

1.6.2. Setting up the Environment on GNU/Linux Hosts

1.6.2.1. Setting the `PATH`

In order to use the Sourcery CodeBench tools from the command line, you should add them to your `PATH`. The graphical installer for Sourcery CodeBench Lite normally does this setup for you, however it may not take effect until you next log in. This section describes how to set the `PATH` manually.

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.

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

```
> x86_64-amd-linux-gnu-gcc -v
```

Verify that the last line of the output contains: `Sourcery CodeBench Lite 2018.11-5`.

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 *`$INSTALLDIR`*`/share/doc/amd-x86_64-amd-linux-gnu/man`.

1.7. 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 removes all files and the installation directory, and it may remove files you have altered.

1.7.1. Using the Uninstaller on Microsoft Windows Hosts

You should use the provided uninstaller to remove a Sourcery CodeBench Lite installation originally created by the graphical installer. Start the graphical uninstaller by invoking `uninstall.exe` located in your installation directory, or use the uninstall shortcut created during installation. 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 `uninstallc.exe` found in your Sourcery CodeBench Lite installation directory with the `-nosplash -install.console` command-line option.

To uninstall third-party drivers bundled with Sourcery CodeBench Lite, first disconnect the associated hardware device. Then use `Uninstall a program` to remove the drivers separately. Depending on the device, you may need to reboot your computer to complete the driver uninstall.

1.7.2. Using the Uninstaller on GNU/Linux Hosts

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 `uninstall` binary 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` binary with the `-install.console` command-line option.

1.7.3. 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 2
Contents of this Release

This chapter lists the components and features included with this release of Sourcery CodeBench Lite for AMD64 GNU/Linux.

2.1. Included Components and Features

This section briefly lists the important components and features included in Sourcery CodeBench Lite for AMD64 GNU/Linux, and tells you where you may find further information about these features.

Component	Version	Notes
GNU programming tools		
GNU Compiler Collection	8.2	Separate manual included.
GNU Binary Utilities	2.31	Includes assembler, linker, and other utilities. Separate manuals included.
Debugging support and simulators		
GNU Debugger	8.2	Separate manual included.
GDB Server	N/A	Included with GDB. See Section 3.4, “Using GDB Server for Debugging”.
Target libraries		
GNU C Library	2.28	Separate manual included.
Linux Kernel Headers	4.18.5	
OpenMP	N/A	
Other utilities		
GNU Make	N/A	Build support on Windows hosts.

Chapter 3
Sourcery CodeBench Lite for AMD64 GNU/Linux

This chapter contains information about features of Sourcery CodeBench Lite that are specific to AMD64 GNU/Linux targets. You should read this chapter to learn how to best use Sourcery CodeBench Lite on your target system.

3.1. Target Library Configurations

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 3.3, "Using Sourcery CodeBench Lite on GNU/Linux Targets" for instructions on how to install and use these support files on your target GNU/Linux system. You can find the sysroot directories provided with Sourcery CodeBench in `$INSTALLDIR/x86_64-amd-linux-gnu/libc`. In the tables below, the dynamic linker pathname is given relative to the corresponding sysroot.

3.1.1. Included Libraries

The following library configurations are available in Sourcery CodeBench Lite for AMD64 GNU/Linux.

AMD Family 16H (Puma) - GLIBC, 64-bit	
Command-line option(s):	default
Sysroot subdirectory:	`./`
Dynamic linker:	`lib64/ld-linux-x86-64.so.2`

AMD Family 15H (Steamroller) - GLIBC, 64-bit	
Command-line option(s):	`-march=bdver3`
Sysroot subdirectory:	`bdver3/`
Dynamic linker:	`lib64/ld-linux-x86-64.so.2`

3.1.2. Library Selection

A given multilib may be compatible with additional processors and build options beyond those listed above. However, even if a particular set of command-line options produces code compatible with one of the provided multilibs, those options may not be sufficient to identify the intended library to the linker. For example, on some targets, specifying only a processor option on the command line may imply architecture features or floating-point support for compilation, but not for library selection. The details of the mapping from command-line options to multilibs are target-specific and quite complex. In some cases, you may need to supply different options for linking than for compilation to select the appropriate multilib.

GCC can tell you which multilib corresponds to a given set of link options if you add the `-print-multi-directory` option to your other command-line options. For example:

```
> x86_64-amd-linux-gnu-gcc -print-multi-directory options...
```

The output of this command is a directory name for the multilib, which you can look up in the tables given previously.

3.2. Target Kernel Requirements

The GNU C library supplied with this version of Sourcery CodeBench Lite requires that Linux kernel version 3.2.0 or later be installed on the target in order to run applications.

3.3. Using Sourcery CodeBench Lite on GNU/Linux Targets

In order to run and debug programs produced by Sourcery CodeBench on your GNU/Linux target system, 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 `$INSTALLDIR/x86_64-amd-linux-gnu/libc`. The sysroot consists of the contents of the `etc`, `lib`, `sbin`, and `usr` directories. There may be other directories in `$INSTALLDIR/x86_64-amd-linux-gnu/libc` that contain additional sysroots customized for particular combinations of command-line compiler flags, or *multilibs*. Refer to Section 3.1, "Target Library Configurations" for a list of the included multilibs in this version of Sourcery CodeBench Lite, and the corresponding sysroot directory pathnames.

Note for Windows Host Users

The sysroots provided in Windows host packages for Sourcery CodeBench are not directly usable on the GNU/Linux target because of differences between the Windows and GNU/Linux file systems. Some files that are hard links, or copies, in the sysroot as installed on the Windows file system should be symbolic links on the GNU/Linux target. Additionally, some files in the sysroot that should be marked executable on the GNU/Linux target are not marked executable on Windows. If you intend to use the sysroot provided with Sourcery CodeBench on a Windows host system as the basis for your GNU/Linux target filesystem, you must correct these issues after copying the sysroot to the target.

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 GNU/Linux root filesystem from scratch. If your target board already has a 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.

3.3.1. 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 3.1, "Target 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 3.4.2, "Setting the Sysroot in the Debugger" for further discussion of this issue.

Otherwise, you must copy files from the appropriate sysroot subdirectory in *`$INSTALLDIR`*`/x86_64-amd-linux-gnu/libc`. 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`.

3.3.2. 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.

Follow these steps:

1. First find the correct sysroot, dynamic linker, and library subdirectory for your selected multilib. Refer to Section 3.1, "Target Library Configurations". In the following steps, *`sysroot`* is the absolute path to the directory on the target where you have installed the sysroot corresponding to your selected multilib.

2. When invoking `x86_64-amd-linux-gnu-gcc` to link your executable, include the command-line options:

```
-Wl,-rpath=sysroot/lib64:sysroot/usr/lib64 \
-Wl,--dynamic-linker=sysroot/lib64/ld-linux-x86-64.so.2
```

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.

3.3.3. 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. First find the correct sysroot, dynamic linker, and library subdirectory for your selected multilib. Refer to Section 3.1, “Target Library Configurations”. In the following steps, *`sysroot`* is the absolute path to the directory on the target where you have installed the sysroot corresponding to your selected multilib.

3. On the target system, invoke the dynamic linker with your executable as:

   ```
   > sysroot/lib64/ld-linux-x86-64.so.2 \
     --library-path sysroot/lib64:sysroot/usr/lib64 \
     /path/to/your-executable
   ```

   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.

3.4. Using GDB Server for Debugging

The GDB server utility provided with Sourcery CodeBench Lite can be used to debug a 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.

3.4.1. 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 3.1, “Target Library Configurations” for details.

You must copy the sysroot to your target system as described in Section 3.3.1, “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 3.3.2, “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 3.3.3, “Specifying the Sysroot Location at Runtime”. In other words, the command to invoke `gdbserver` in this case would be similar to:

```
> sysroot/lib64/ld-linux-x86-64.so.2 \
  --library-path sysroot/lib64:sysroot/usr/lib64 \
  sysroot/usr/lib64/bin/gdbserver :10000 \
  program arg1 arg2 ...
```

3.4.2. Setting the Sysroot in the Debugger

When debugging programs using shared libraries, GDB reads the files for the dynamic linker and libraries so that it has access to the symbol table and debug information they contain. GDB's default behavior when debugging with GDB server is to read the library files on demand from the remote target. This can be problematical because a slow connection to the target, or a slow target filesystem, can cause lengthy delays on debugger startup, even when no code in shared libraries is actually being debugged. Moreover, shared libraries installed on the target are often stripped, while unstripped copies of the libraries and/or separate library debug information may be available on the host. For these reasons, it is recommended that you always configure GDB to map target library pathnames to equivalent host pathnames prior to establishing a connection to GDB server.

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.

As another special case, sometimes the target pathnames are valid on the host system. You can provide an identity mapping for library pathname lookup by specifying the `set sysroot` command alone, without an argument.

Refer to Section 3.3.1, “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.

3.4.3. 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.

3.5. Using OpenMP

Sourcery CodeBench Lite for AMD64 GNU/Linux includes the GNU OpenMP library (libgomp). This is an API that supports multi-platform shared-memory parallel programming.

To compile programs that use OpenMP features, use the `-fopenmp` command-line option. For more information about OpenMP, see `http://www.openmp.org/`.

To run programs that use OpenMP features, you need to tell the binary where to find the libgomp runtime library:

```
> export LD_LIBRARY_PATH=$INSTALLDIR/x86_64-amd-linux-gnu/lib64
```

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 x86_64-amd-linux-gnu, as indicated by the `x86_64-amd-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:

```
> x86_64-amd-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 `x86_64-amd-linux-gnu-gcc` with `x86_64-amd-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 AMD64 GNU/Linux” for specific instructions.

To run your program on a 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:

```
> x86_64-amd-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 AMD64 GNU/Linux target. Follow the instructions in Chapter 3, “Sourcery CodeBench Lite for AMD64 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[1]. 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. 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 *$INSTALLDIR*`/share/doc/amd-x86_64-amd-linux-gnu/`.

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 $INSTALLDIR/share/doc/amd-x86_64-amd-linux-gnu/man/man1
```

Then you can invoke `man` as:

```
> man ./x86_64-amd-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 1.6, “Setting up the Environment” for instructions. Then you can invoke `man` with just the command name rather than a pathname.

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.

[1] https://sourcery.mentor.com/GNUToolchain/

Appendix A
Sourcery CodeBench Lite Release Notes

This appendix contains information about changes in this release of Sourcery CodeBench Lite for AMD64 GNU/Linux. You should read through these notes to learn about new features and bug fixes.

A.1. Changes in Sourcery CodeBench Lite for AMD64 GNU/Linux

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

A.1.1. Changes in Sourcery CodeBench Lite 2018.11-5

GCC version 8.2. Sourcery CodeBench Lite for AMD64 GNU/Linux is now based on GCC version 8.2, with additional bug fixes from the GCC 8 release branch up to SVN revision 265154. For more information about changes from GCC version 7 that was included in previous releases, see `https://gcc.gnu.org/gcc-8/changes.html`.

GNU C Library version 2.28. Sourcery CodeBench Lite for AMD64 GNU/Linux now includes the GNU C Library version 2.28, git revision 5a74abda201907cafbdabd1debf98890313ff71e. For more information about changes, see `https://www.gnu.org/software/libc/`.

A.1.2. Changes in Sourcery CodeBench Lite 2018.11-2

GCC version 8.2. Sourcery CodeBench Lite for AMD64 GNU/Linux is now based on GCC version 8.2, with additional bug fixes from the GCC 8 release branch up to SVN revision 264434. For more information about changes from GCC version 7 that was included in previous releases, see `https://gcc.gnu.org/gcc-8/changes.html`.

Binutils update. The binutils package has been updated to version 2.31.51 from the FSF trunk, git revision 5ea637fe9afa547309f46a79106eb4e11fff30c9. This update includes numerous bug fixes.

GNU C Library version 2.28. Sourcery CodeBench Lite for AMD64 GNU/Linux now includes the GNU C Library version 2.28, git revision a55e109709af55e6ed67d3f9536cac5d929c982e. For more information about changes, see `https://www.gnu.org/software/libc/`.

`pthread_cleanup_push` compiler warnings. Previously, use of the `pthread_cleanup_push` macro from `pthread.h` sometimes caused warnings when compiled with optimization enabled. The macro has been modified to avoid those warnings.

Linux kernel headers update. Linux kernel header files have been updated to version 4.18.5.

GDB update. The version of GDB has been updated to 8.2, git revision 5ea637fe9afa547309f46a79106eb4e11fff30c9. This update adds numerous bug fixes and features. Refer to `http://www.gnu.org/software/gdb/news` for more information.

A.1.3. Changes in Sourcery CodeBench Lite 2017.11-8

GCC version 7.2. Sourcery CodeBench Lite for AMD64 GNU/Linux is now based on GCC version 7.2, with additional bug fixes from the GCC 7 release branch up to SVN revision 253840. For more information about changes from GCC version 6 that was included in previous releases, see `https://gcc.gnu.org/gcc-7/changes.html`.

GNU C Library version 2.26. Sourcery CodeBench Lite for AMD64 GNU/Linux now includes the GNU C Library version 2.26, git revision f725563967c1f277e0f02bb1516fe9ebfa4737bf. For more information about changes, see `https://www.gnu.org/software/libc/`.

A.1.4. Changes in Sourcery CodeBench Lite 2017.11-5

64-bit host systems required. Sourcery CodeBench Lite for AMD64 GNU/Linux now requires a 64-bit host operating system. 32-bit versions of Microsoft Windows and GNU/Linux are no longer supported as hosts for running Sourcery CodeBench Lite for AMD64 GNU/Linux. For more information about the currently supported host systems, refer to Section 1.2.1, "Host Operating System Requirements".

GCC version 7.2. Sourcery CodeBench Lite for AMD64 GNU/Linux is now based on GCC version 7.2. For more information about changes from GCC version 6 that was included in previous releases, see `https://gcc.gnu.org/gcc-7/changes.html`.

Binutils update. The binutils package has been updated to version 2.28.51 from the FSF trunk, git revision c56a60013009d83391281ab6d4019a143d517b09. This update includes numerous bug fixes.

GNU C Library version 2.26. Sourcery CodeBench Lite for AMD64 GNU/Linux now includes the GNU C Library version 2.26, git revision 37d4262a7a35886cf8ac856457bbad8c0498c8d6. For more information about changes, see `https://www.gnu.org/software/libc/`.

`AF_BUS` support removed. Support for the `AF_BUS` address family has been removed from the GNU C Library in this version of Sourcery CodeBench Lite. This extension was formerly provided for compliance with old versions of the GENIVI specification that are now obsolete.

Linux kernel headers update. Linux kernel header files have been updated to version 4.13.

GDB update. The version of GDB has been updated to 8.0.1, git revision c56a60013009d83391281ab6d4019a143d517b09. This update adds numerous bug fixes and features. Refer to `http://www.gnu.org/software/gdb/news` for more information.

A.1.5. Changes in Older Releases

For information about changes in older releases of Sourcery CodeBench Lite for AMD64 GNU/Linux, please refer to the Getting Started guide packaged with those releases.

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.

The Mentor Graphics License is available in Section B.1, “Sourcery CodeBench Lite License Agreement”.

B.1. Sourcery CodeBench Lite License Agreement

Sourcery CodeBench Lite is licensed under the Mentor Graphics **End-User License Agreement with Embedded Software Supplement.** If you have a separate signed or shrinkwrap agreement (as applicable) with Mentor Graphics related to your use of Sourcery CodeBench Lite, your order is subject to the terms of that agreement. If you do not, the following terms apply, unless otherwise specifically agreed to in writing by an authorized representative of Mentor Graphics. The terms of this appendix supplement, but do not replace or amend, the terms of your separate agreement with Mentor Graphics. Accordingly, to the extent the following terms and conditions conflict with such separate agreement, the terms and conditions of the separate agreement shall control.

The latest version of the License Agreement is available on-line at https://www.mentor.com/terms_conditions/embedded-supplement-eula.

B.1.1. End-User License Agreement with Embedded Software Supplement

IMPORTANT INFORMATION

USE OF ALL SOFTWARE IS SUBJECT TO LICENSE RESTRICTIONS. CAREFULLY READ THIS LICENSE AGREEMENT BEFORE USING THE PRODUCTS. USE OF SOFTWARE INDICATES CUSTOMER'S COMPLETE AND UNCONDITIONAL ACCEPTANCE OF THE TERMS AND CONDITIONS SET FORTH IN THIS AGREEMENT. ANY ADDITIONAL OR DIFFERENT PURCHASE ORDER TERMS AND CONDITIONS SHALL NOT APPLY.

END-USER LICENSE AGREEMENT ("Agreement")

This is a legal agreement concerning the use of Software (as defined in Section 2) and hardware (collectively "Products") between the company acquiring the Products ("Customer"), and the Mentor Graphics entity that issued the corresponding quotation or, if no quotation was issued, the applicable local Mentor Graphics entity ("Mentor Graphics"). Except for license agreements related to the subject matter of this license agreement which are physically signed by Customer and an authorized representative of Mentor Graphics, this Agreement and the applicable quotation contain the parties' entire understanding relating to the subject matter and supersede all prior or contemporaneous agreements. If Customer does not agree to these terms and conditions, promptly return or, in the case of Software received electronically, certify destruction of Software and all accompanying items within five days after receipt of Software and receive a full refund of any license fee paid.

1. ORDERS, FEES AND PAYMENT.

1.1. To the extent Customer (or if agreed by Mentor Graphics, Customer's appointed third party buying agent) places and Mentor Graphics accepts purchase orders pursuant to this Agreement (each an "Order"), each Order will constitute a contract between Customer and Mentor Graphics, which shall be governed solely and exclusively by the terms and conditions of this Agreement, any applicable addenda and the applicable quotation, whether or not those documents are referenced on the Order. Any additional or conflicting terms and conditions appearing on an Order or presented in any electronic portal or automated order management system, whether or not required to be electronically accepted, will not be effective unless agreed in writing and physically signed by an authorized representative of Customer and Mentor Graphics.

1.2. Amounts invoiced will be paid, in the currency specified on the applicable invoice, within 30 days from the date of such invoice. Any past due invoices will be subject to the imposition of interest charges in the amount of one and one-half percent per month or the applicable legal rate currently in effect, whichever is lower. Prices do not include freight, insurance, customs duties, taxes or other similar charges, which Mentor Graphics will state separately in the applicable invoice. Unless timely provided with a valid certificate of exemption or other evidence that items are not taxable, Mentor Graphics will invoice Customer for all applicable taxes including, but not limited to, VAT, GST, sales tax, consumption tax and service tax. Customer will make all payments free and clear of, and without reduction for, any withholding or other taxes; any such taxes imposed on payments by Customer hereunder will be Customer's sole responsibility. If Customer appoints a third party to place purchase orders and/or make payments on Customer's behalf, Customer shall be liable for payment under Orders placed by such third party in the event of default.

1.3. All Products are delivered FCA factory (Incoterms 2010), freight prepaid and invoiced to Customer, except Software delivered electronically, which shall be deemed delivered when made available to Customer for download. Mentor Graphics retains a security interest in all Products delivered under this Agreement, to secure payment of the purchase price of such Products, and Customer agrees to sign any documents that Mentor Graphics determines to be necessary or convenient for use in filing or perfecting such security interest. Mentor Graphics' delivery of Software by electronic means is subject to Customer's provision of both a primary and an alternate e-mail address.

2. GRANT OF LICENSE. The software installed, downloaded, or otherwise acquired by Customer under this Agreement, including any updates, modifications, revisions, copies, documentation, setup files and design data ("Software") are copyrighted, trade secret and confidential information of Mentor Graphics or its licensors, who maintain exclusive title to all Software and retain all rights not expressly granted by this Agreement. Except for Software that is embeddable ("Embedded Software"), which is licensed pursuant to separate embedded software terms or an embedded software supplement, Mentor Graphics grants to Customer, subject to payment of applicable license fees, a nontransferable, nonexclusive license to use Software solely: (a) in machine-readable, object-code form (except as provided in Subsection 4.2); (b) for Customer's internal business purposes; (c) for the term of the license; and (d) on the computer hardware and at the site authorized by Mentor Graphics. A site is restricted to a one-half mile (800 meter) radius. Customer may have Software temporarily used by an employee for telecommuting purposes from locations other than a Customer office, such as the employee's residence, an airport or hotel, provided that such employee's primary place of employment is the site where the Software is authorized for use. Mentor Graphics' standard policies and programs, which vary depending on Software, license fees paid or services purchased, apply to the following: (a) relocation of Software; (b) use of Software, which may be limited, for example, to execution of a single session by a single user on the authorized hardware or for a restricted period of time (such limitations may be technically implemented through the use of authorization codes or similar devices); and (c) support services provided, including eligibility to receive telephone support, updates, modifications, and revisions. For the avoidance of doubt, if Customer provides any feedback or requests any change or enhancement to Products, whether in the course of receiving support or consulting services, evaluating Products, performing beta testing or otherwise, any inventions, product improvements, modifications or developments made by Mentor Graphics (at Mentor Graphics' sole discretion) will be the exclusive property of Mentor Graphics.

3. BETA CODE.

3.1. Portions or all of certain Software may contain code for experimental testing and evaluation (which may be either alpha or beta, collectively "Beta Code"), which may not be used without Mentor Graphics' explicit authorization. Upon Mentor Graphics' authorization, Mentor Graphics grants to Customer a temporary, nontransferable, nonexclusive license for experimental use to test and evaluate the Beta Code without charge for a limited period of time specified by Mentor Graphics.

Mentor Graphics may choose, at its sole discretion, not to release Beta Code commercially in any form.

3.2. If Mentor Graphics authorizes Customer to use the Beta Code, Customer agrees to evaluate and test the Beta Code under normal conditions as directed by Mentor Graphics. Customer will contact Mentor Graphics periodically during Customer's use of the Beta Code to discuss any malfunctions or suggested improvements. Upon completion of Customer's evaluation and testing, Customer will send to Mentor Graphics a written evaluation of the Beta Code, including its strengths, weaknesses and recommended improvements.

3.3. Customer agrees to maintain Beta Code in confidence and shall restrict access to the Beta Code, including the methods and concepts utilized therein, solely to those employees and Customer location(s) authorized by Mentor Graphics to perform beta testing. Customer agrees that any written evaluations and all inventions, product improvements, modifications or developments that Mentor Graphics conceived or made during or subsequent to this Agreement, including those based partly or wholly on Customer's feedback, will be the exclusive property of Mentor Graphics. Mentor Graphics will have exclusive rights, title and interest in all such property. The provisions of this Subsection 3.3 shall survive termination of this Agreement.

4. RESTRICTIONS ON USE.

4.1. Customer may copy Software only as reasonably necessary to support the authorized use. Each copy must include all notices and legends embedded in Software and affixed to its medium and container as received from Mentor Graphics. All copies shall remain the property of Mentor Graphics or its licensors. Except for Embedded Software that has been embedded in executable code form in Customer's product(s), Customer shall maintain a record of the number and primary location of all copies of Software, including copies merged with other software, and shall make those records available to Mentor Graphics upon request. Customer shall not make Products available in any form to any person other than Customer's employees and on-site contractors, excluding Mentor Graphics competitors, whose job performance requires access and who are under obligations of confidentiality. Customer shall take appropriate action to protect the confidentiality of Products and ensure that any person permitted access does not disclose or use Products except as permitted by this Agreement. Customer shall give Mentor Graphics written notice of any unauthorized disclosure or use of the Products as soon as Customer becomes aware of such unauthorized disclosure or use. Customer acknowledges that Software provided hereunder may contain source code which is proprietary and its confidentiality is of the highest importance and value to Mentor Graphics. Customer acknowledges that Mentor Graphics may be seriously harmed if such source code is disclosed in violation of this Agreement. Except as otherwise permitted for purposes of interoperability as specified by applicable and mandatory local law, Customer shall not reverse- assemble, disassemble, reverse-compile, or reverse-engineer any Product, or in any way derive any source code from Software that is not provided to Customer in source code form. Log files, data files, rule files and script files generated by or for the Software (collectively "Files"), including without limitation files containing Standard Verification Rule Format ("SVRF") and Tcl Verification Format ("TVF") which are Mentor Graphics' trade secret and proprietary syntaxes for expressing process rules, constitute or include confidential information of Mentor Graphics. Customer may share Files with third parties, excluding Mentor Graphics competitors, provided that the confidentiality of such Files is protected by written agreement at least as well as Customer protects other information of a similar nature or importance, but in any case with at least reasonable care. Customer may use Files containing SVRF or TVF only with Mentor Graphics products. Under no circumstances shall Customer use Products or Files or allow their use for the purpose of developing, enhancing or marketing any product that is in any way competitive with Products, or disclose to any third party the results of, or information pertaining to, any benchmark.

4.2. If any Software or portions thereof are provided in source code form, Customer will use the source code only to correct software errors and enhance or modify the Software for the authorized

use, or as permitted for Embedded Software under separate embedded software terms or an embedded software supplement. Customer shall not disclose or permit disclosure of source code, in whole or in part, including any of its methods or concepts, to anyone except Customer's employees or on-site contractors, excluding Mentor Graphics competitors, with a need to know. Customer shall not copy or compile source code in any manner except to support this authorized use.

4.3. Customer agrees that it will not subject any Product to any open source software ("OSS") license that conflicts with this Agreement or that does not otherwise apply to such Product.

4.4. Customer may not assign this Agreement or the rights and duties under it, or relocate, sublicense, or otherwise transfer the Products, whether by operation of law or otherwise ("Attempted Transfer"), without Mentor Graphics' prior written consent and payment of Mentor Graphics' then-current applicable relocation and/or transfer fees. Any Attempted Transfer without Mentor Graphics' prior written consent shall be a material breach of this Agreement and may, at Mentor Graphics' option, result in the immediate termination of the Agreement and/or the licenses granted under this Agreement. The terms of this Agreement, including without limitation the licensing and assignment provisions, shall be binding upon Customer's permitted successors in interest and assigns.

4.5. The provisions of this Section 4 shall survive the termination of this Agreement.

5. SUPPORT SERVICES. To the extent Customer purchases support services, Mentor Graphics will provide Customer with updates and technical support for the Products, at the Customer site(s) for which support is purchased, in accordance with Mentor Graphics' then current End-User Support Terms located at https://www.mentor.com/support/en/legal.

6. OPEN SOURCE SOFTWARE. Products may contain OSS or code distributed under a proprietary third party license agreement, to which additional rights or obligations ("Third Party Terms") may apply. Please see the applicable Product documentation (including license files, header files, readme files or source code) for details. In the event of conflict between the terms of this Agreement (including any addenda) and the Third Party Terms, the Third Party Terms will control solely with respect to the OSS or third party code. The provisions of this Section 6 shall survive the termination of this Agreement.

7. LIMITED WARRANTY.

7.1. Mentor Graphics warrants that during the warranty period its standard, generally supported Products, when properly installed, will substantially conform to the functional specifications set forth in the applicable user manual. Mentor Graphics does not warrant that Products will meet Customer's requirements or that operation of Products will be uninterrupted or error free. The warranty period is 90 days starting on the 15th day after delivery or upon installation, whichever first occurs. Customer must notify Mentor Graphics in writing of any nonconformity within the warranty period. For the avoidance of doubt, this warranty applies only to the initial shipment of Software under an Order and does not renew or reset, for example, with the delivery of (a) Software updates or (b) authorization codes or alternate Software under a transaction involving Software re-mix. This warranty shall not be valid if Products have been subject to misuse, unauthorized modification, improper installation or Customer is not in compliance with this Agreement. MENTOR GRAPHICS' ENTIRE LIABILITY AND CUSTOMER'S EXCLUSIVE REMEDY SHALL BE, AT MENTOR GRAPHICS' OPTION, EITHER (A) REFUND OF THE PRICE PAID UPON RETURN OF THE PRODUCTS TO MENTOR GRAPHICS OR (B) MODIFICATION OR REPLACEMENT OF THE PRODUCTS THAT DO NOT MEET THIS LIMITED WARRANTY. MENTOR GRAPHICS MAKES NO WARRANTIES WITH RESPECT TO: (A) SERVICES; (B) PRODUCTS PROVIDED AT NO CHARGE; OR (C) BETA CODE; ALL OF WHICH ARE PROVIDED "AS IS."

7.2. THE WARRANTIES SET FORTH IN THIS SECTION 7 ARE EXCLUSIVE. NEITHER MENTOR GRAPHICS NOR ITS LICENSORS MAKE ANY OTHER WARRANTIES EXPRESS,

IMPLIED OR STATUTORY, WITH RESPECT TO PRODUCTS PROVIDED UNDER THIS AGREEMENT. MENTOR GRAPHICS AND ITS LICENSORS SPECIFICALLY DISCLAIM ALL IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT OF INTELLECTUAL PROPERTY.

8. LIMITATION OF LIABILITY. TO THE EXTENT PERMITTED UNDER APPLICABLE LAW, IN NO EVENT SHALL MENTOR GRAPHICS OR ITS LICENSORS BE LIABLE FOR INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES (INCLUDING LOST PROFITS OR SAVINGS) WHETHER BASED ON CONTRACT, TORT OR ANY OTHER LEGAL THEORY, EVEN IF MENTOR GRAPHICS OR ITS LICENSORS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. IN NO EVENT SHALL MENTOR GRAPHICS' OR ITS LICENSORS' LIABILITY UNDER THIS AGREEMENT EXCEED THE AMOUNT RECEIVED FROM CUSTOMER FOR THE HARDWARE, SOFTWARE LICENSE OR SERVICE GIVING RISE TO THE CLAIM. IN THE CASE WHERE NO AMOUNT WAS PAID, MENTOR GRAPHICS AND ITS LICENSORS SHALL HAVE NO LIABILITY FOR ANY DAMAGES WHATSOEVER. THE PROVISIONS OF THIS SECTION 8 SHALL SURVIVE THE TERMINATION OF THIS AGREEMENT.

9. THIRD PARTY CLAIMS.

9.1. Customer acknowledges that Mentor Graphics has no control over the testing of Customer's products, or the specific applications and use of Products. Mentor Graphics and its licensors shall not be liable for any claim or demand made against Customer by any third party, except to the extent such claim is covered under Section 10.

9.2. In the event that a third party makes a claim against Mentor Graphics arising out of the use of Customer's products, Mentor Graphics will give Customer prompt notice of such claim. At Customer's option and expense, Customer may take sole control of the defense and any settlement of such claim. Customer WILL reimburse and hold harmless Mentor Graphics for any LIABILITY, damages, settlement amounts, costs and expenses, including reasonable attorney's fees, incurred by or awarded against Mentor Graphics or its licensors in connection with such claims.

9.3. The provisions of this Section 9 shall survive any expiration or termination of this Agreement.

10. INFRINGEMENT.

10.1. Mentor Graphics will defend or settle, at its option and expense, any action brought against Customer in the United States, Canada, Japan, or member state of the European Union which alleges that any standard, generally supported Product acquired by Customer hereunder infringes a patent or copyright or misappropriates a trade secret in such jurisdiction. Mentor Graphics will pay costs and damages finally awarded against Customer that are attributable to such action. Customer understands and agrees that as conditions to Mentor Graphics' obligations under this section Customer must: (a) notify Mentor Graphics promptly in writing of the action; (b) provide Mentor Graphics all reasonable information and assistance to settle or defend the action; and (c) grant Mentor Graphics sole authority and control of the defense or settlement of the action.

10.2. If a claim is made under Subsection 10.1 Mentor Graphics may, at its option and expense: (a) replace or modify the Product so that it becomes noninfringing; (b) procure for Customer the right to continue using the Product; or (c) require the return of the Product and refund to Customer any purchase price or license fee paid, less a reasonable allowance for use.

10.3. Mentor Graphics has no liability to Customer if the action is based upon: (a) the combination of Software or hardware with any product not furnished by Mentor Graphics; (b) the modification of the Product other than by Mentor Graphics; (c) the use of other than a current unaltered release of Software; (d) the use of the Product as part of an infringing process; (e) a product that Customer

makes, uses, or sells; (f) any Beta Code or Product provided at no charge; (g) any software provided by Mentor Graphics' licensors who do not provide such indemnification to Mentor Graphics' customers; (h) OSS, except to the extent that the infringement is directly caused by Mentor Graphics' modifications to such OSS; or (i) infringement by Customer that is deemed willful. In the case of (i), Customer shall reimburse Mentor Graphics for its reasonable attorney fees and other costs related to the action.

10.4. THIS SECTION 10 IS SUBJECT TO SECTION 8 ABOVE AND STATES THE ENTIRE LIABILITY OF MENTOR GRAPHICS AND ITS LICENSORS, AND CUSTOMER'S SOLE AND EXCLUSIVE REMEDY, FOR DEFENSE, SETTLEMENT AND DAMAGES, WITH RESPECT TO ANY ALLEGED PATENT OR COPYRIGHT INFRINGEMENT OR TRADE SECRET MISAPPROPRIATION BY ANY PRODUCT PROVIDED UNDER THIS AGREEMENT.

11. TERMINATION AND EFFECT OF TERMINATION.

11.1. If a Software license was provided for limited term use, such license will automatically terminate at the end of the authorized term. Mentor Graphics may terminate this Agreement and/or any license granted under this Agreement immediately upon written notice if Customer: (a) exceeds the scope of the license or otherwise fails to comply with the licensing or confidentiality provisions of this Agreement, or (b) becomes insolvent, files a bankruptcy petition, institutes proceedings for liquidation or winding up or enters into an agreement to assign its assets for the benefit of creditors. For any other material breach of any provision of this Agreement, Mentor Graphics may terminate this Agreement and/or any license granted under this Agreement upon 30 days written notice if Customer fails to cure the breach within the 30 day notice period. Termination of this Agreement or any license granted hereunder will not affect Customer's obligation to pay for Products shipped or licenses granted prior to the termination, which amounts shall be payable immediately upon the date of termination.

11.2. Upon termination of this Agreement, the rights and obligations of the parties shall cease except as expressly set forth in this Agreement. Upon termination of this Agreement and/or any license granted under this Agreement, Customer shall ensure that all use of the affected Products ceases, and shall return hardware and either return to Mentor Graphics or destroy Software in Customer's possession, including all copies and documentation, and certify in writing to Mentor Graphics within ten business days of the termination date that Customer no longer possesses any of the affected Products or copies of Software in any form.

12. EXPORT. The Products provided hereunder are subject to regulation by local laws and European Union ("E.U.") and United States ("U.S.") government agencies, which prohibit export, re-export or diversion of certain products, information about the products, and direct or indirect products thereof, to certain countries and certain persons. Customer agrees that it will not export or re- export Products in any manner without first obtaining all necessary approval from appropriate local, E.U. and U.S. government agencies. If Customer wishes to disclose any information to Mentor Graphics that is subject to any E.U., U.S. or other applicable export restrictions, including without limitation the U.S. International Traffic in Arms Regulations (ITAR) or special controls under the Export Administration Regulations (EAR), Customer will notify Mentor Graphics personnel, in advance of each instance of disclosure, that such information is subject to such export restrictions.

13. U.S. GOVERNMENT LICENSE RIGHTS. Software was developed entirely at private expense. The parties agree that all Software is commercial computer software within the meaning of the applicable acquisition regulations. Accordingly, pursuant to U.S. FAR 48 CFR 12.212 and DFAR 48 CFR 227.7202, use, duplication and disclosure of the Software by or for the U.S. government or a U.S. government subcontractor is subject solely to the terms and conditions set forth in this Agreement, which shall supersede any conflicting terms or conditions in any government order document, except for provisions which are contrary to applicable mandatory federal laws.

14. THIRD PARTY BENEFICIARY. Mentor Graphics Corporation, Mentor Graphics (Ireland) Limited, Microsoft Corporation and other licensors may be third party beneficiaries of this Agreement with the right to enforce the obligations set forth herein.

15. REVIEW OF LICENSE USAGE. Customer will monitor the access to and use of Software. With prior written notice and during Customer's normal business hours, Mentor Graphics may engage an internationally recognized accounting firm to review Customer's software monitoring system and records deemed relevant by the internationally recognized accounting firm to confirm Customer's compliance with the terms of this Agreement or U.S. or other local export laws. Such review may include FlexNet (or successor product) report log files that Customer shall capture and provide at Mentor Graphics' request. Customer shall make records available in electronic format and shall fully cooperate with data gathering to support the license review. Mentor Graphics shall bear the expense of any such review unless a material non-compliance is revealed. Mentor Graphics shall treat as confidential information all information gained as a result of any request or review and shall only use or disclose such information as required by law or to enforce its rights under this Agreement. The provisions of this Section 15 shall survive the termination of this Agreement.

16. CONTROLLING LAW, JURISDICTION AND DISPUTE RESOLUTION. The owners of certain Mentor Graphics intellectual property licensed under this Agreement are located in Ireland and the U.S. To promote consistency around the world, disputes shall be resolved as follows: excluding conflict of laws rules, this Agreement shall be governed by and construed under the laws of the State of Oregon, U.S., if Customer is located in North or South America, and the laws of Ireland if Customer is located outside of North or South America or Japan, and the laws of Japan if Customer is located in Japan. All disputes arising out of or in relation to this Agreement shall be submitted to the exclusive jurisdiction of the courts of Portland, Oregon when the laws of Oregon apply, or Dublin, Ireland when the laws of Ireland apply, or the Tokyo District Court when the laws of Japan apply. Notwithstanding the foregoing, all disputes in Asia (excluding Japan) arising out of or in relation to this Agreement shall be resolved by arbitration in Singapore before a single arbitrator to be appointed by the chairman of the Singapore International Arbitration Centre ("SIAC") to be conducted in the English language, in accordance with the Arbitration Rules of the SIAC in effect at the time of the dispute, which rules are deemed to be incorporated by reference in this section. Nothing in this section shall restrict Mentor Graphics' right to bring an action (including for example a motion for injunctive relief) against Customer in the jurisdiction where Customer's place of business is located. The United Nations Convention on Contracts for the International Sale of Goods does not apply to this Agreement.

17. SEVERABILITY. If any provision of this Agreement is held by a court of competent jurisdiction to be void, invalid, unenforceable or illegal, such provision shall be severed from this Agreement and the remaining provisions will remain in full force and effect.

18. MISCELLANEOUS. This Agreement contains the parties' entire understanding relating to its subject matter and supersedes all prior or contemporaneous agreements. Any translation of this Agreement is provided to comply with local legal requirements only. In the event of a dispute between the English and any non-English versions, the English version of this Agreement shall govern to the extent not prohibited by local law in the applicable jurisdiction. This Agreement may only be modified in writing, signed by an authorized representative of each party. Waiver of terms or excuse of breach must be in writing and shall not constitute subsequent consent, waiver or excuse.

Rev. 170330, Part No. 270943 (with ESS)

EMBEDDED SOFTWARE SUPPLEMENT

This Embedded Software Supplement ("Supplement") is attached to and made a part of Mentor Graphics End-User License Agreement, which may be viewed at www.mentor.com/eula ("Agreement"), and governs the licensing of Embedded Software to Customer by the Mentor Graphics entity that issued the corresponding quotation ("Mentor Graphics"). Any conflict between the terms of this

Supplement and those of the Agreement as it relates to Customer's use of Embedded Software shall be resolved in favor of this Supplement.

1. Introduction. Unless otherwise noted, the capitalized terms used in this Supplement shall have the same meanings as set forth in the Agreement.

2. Definitions. As used in this Supplement and the applicable quotation ("Quotation"), these terms shall have the following meanings:

2.1. "Customer's Product" means Customer's product identified by a unique stock keeping unit or "SKU" in the Quotation that is developed, manufactured, branded and shipped solely by Customer or an authorized manufacturer or subcontractor on behalf of Customer to Customer's customer, whether End-Users or intermediate manufacturers;

2.2. "Developer" means a unique user, as identified by a unique user identification number, with access to Embedded Software at an authorized Development Location. A unique user is an individual who works directly with the Embedded Software in source code form, or creates, modifies or compiles Software that ultimately links to the Embedded Software in Object Code form and is embedded into Customer's Product;

2.3. "Development Location" means the location where Software may be used as authorized in the Quotation;

2.4. "Development Tools" means the Software that may be used by Customer for creating, editing, compiling, debugging or prototyping Customer's Product;

2.5. "Embedded Software" means Software that is embeddable;

2.6. "End-User" means consumers that receive Customer's Products after such products have been fully developed and marketed;

2.7. "Executable Code" means a compiled program translated into a machine- readable format that can be loaded into memory and run by a Processor;

2.8. "Linkable Object Code" or "Object Code" means linkable code resulting from the translation, processing, or compiling of Source Code by a computer into machine-readable format;

2.9. "Mentor Embedded Linux" or "MEL" means Mentor Graphics' tools, Source Code, and recipes for building Linux systems;

2.10. "Open Source Software" or "OSS" means software subject to an open source license which requires as a condition for redistribution of such software, including modifications thereto, that the: (i) redistribution be in source code form or be made available in source code form; (ii) redistributed software be licensed to allow the making of derivative works; or (iii) redistribution be at no charge;

2.11. "Processor" means the specific microprocessor to be used with Software and implemented in Customer's Product;

2.12. "Proprietary Components" means the components of the Products that are owned and/or licensed by Mentor Graphics and are not subject to an OSS license, as more fully set forth in the product documentation provided with the Products;

2.13. "Redistributable Components" means those components that are intended to be incorporated or linked into Customer's Linkable Object Code developed with Software, as more fully set forth in the documentation provided with the Products;

2.14. "Software" means Mentor Graphics software programs, Embedded Software and/or Development Tools, including any updates, modifications, revisions, copies, documentation and design data that are licensed under this Supplement;

2.15. "Source Code" means software in a form in which the program logic is readily understandable by a human being;

2.16. "Sourcery CodeBench Software" means Mentor Graphics' Development Tool for C/C++ embedded application development;

2.17. "Sourcery VSIPL++" is Software providing C++ classes and functions for writing embedded signal processing applications designed to run on one or more processors; and

2.18. "Subsidiary" means any corporation more than 50% owned by Customer, excluding Mentor Graphics competitors. Customer agrees to fulfill the obligations of such Subsidiary in the event of default. To the extent Mentor Graphics authorizes any Subsidiary's use of Products under this Supplement, Customer agrees to ensure such Subsidiary's compliance with the terms of this Supplement, the Quotation and the Agreement and will be liable for any breach by a Subsidiary.

3. Grant of License. Subject to the payment of the applicable license fees, Mentor Graphics grants to Customer, a nontransferable, nonexclusive license to use Software as described in the Quotation or, if no license is specified in the Quotation, Mentor Graphics grants to Customer a nontransferable, nonexclusive license to use Software solely for Customer's internal business purposes. The limited licenses granted under the Quotation or this Supplement shall continue until the expiration date of term-licensed Products or termination in accordance with the termination provision of the Agreement, whichever occurs first. Mentor Graphics does NOT grant Customer any right to (a) sublicense or (b) use Software, beyond the scope specified in the Quotation or this Supplement.

4. Support Services.

4.1. Except as described in Sections 4.2, 4.3 and 4.4 below, and unless otherwise specified in the Quotation, to the extent Customer purchases support services, Mentor Graphics will provide Customer updates and technical support for the number of Developers at the Development Location(s) for which support is purchased in accordance with Mentor Graphics' then-current End- User Support Terms located at https://www.mentor.com/support/en/legal.

4.2. To the extent Customer purchases support services for Sourcery CodeBench Software, Mentor Graphics will provide Customer updates and technical support for the number of Developers at the Development Location(s) for which support is purchased in accordance with Mentor Graphics' then-current Sourcery CodeBench Software Support Terms located at http://www.mentor.com/codebench-support-legal.

4.3. To the extent Customer purchases support services for Sourcery VSIPL++, Mentor Graphics will provide Customer updates and technical support for the number of Developers at the Development Location(s) for which support is purchased solely in accordance with Mentor Graphics' then-current Sourcery VSIPL++ Support Terms located at http://www.mentor.com/vsipl-support-legal.

4.4. To the extent Customer purchases support services for MEL, Mentor Graphics will provide Customer updates and technical support for the number of Developers at the Development Location(s) for which support is purchased solely in accordance with Mentor Graphics' then-current MEL Support Terms located at http://www.mentor.com/terms_conditions/mel-software-license.

5. Effect of Termination. In the event of termination, Customer shall discontinue its use of the Product in accordance with the termination provisions of the Agreement, except to the extent that an Open Source Software license permits Customer's continued use of any Open Source Software

portion or component of a Product. Further, upon termination for Customer's breach, an End- User may continue its use and/or distribution of Customer's Product, provided that: (a) the End-User was licensed according to the terms of this Supplement or the Agreement, if applicable to such End-User; and (b) such End-User is not in breach of its agreement with Customer or Mentor Graphics, if applicable, nor a party to Customer's breach.

6. Open Source Software. Customer agrees that it will not subject any Product provided by Mentor Graphics to any OSS license that does not otherwise apply to such Product. Products may contain OSS or code distributed under a proprietary license agreement to which additional rights or obligations ("Third Party Terms") may apply. Please see applicable software product documentation, including but not limited to license notice files, header files or source code for further details. In the event of conflict between the terms of this Supplement, the Quotation, the Agreement and the Third Party Terms, the Third Party Terms will control solely with respect to the OSS or third party code. The provisions of this Section 6 shall survive the termination of the Agreement and this Supplement.

Rev. 170330, Part No. 270943 (with EULA)

B.2. Licenses and Third-Party Information for Sourcery CodeBench Lite Components

The table below lists the major components of Sourcery CodeBench Lite for AMD64 GNU/Linux and the license terms which apply to each of these components.

Mentor Graphics Proprietary Components. 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, and any Sourcery CodeBench Debug Sprite.

Open-Source 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 http://www.gnu.org/licenses/gpl.html
GNU Binary Utilities	GNU General Public License 3.0 http://www.gnu.org/licenses/gpl.html
GNU Debugger	GNU General Public License 3.0 http://www.gnu.org/licenses/gpl.html
GNU C Library	GNU Lesser General Public License 2.1 http://www.gnu.org/licenses/old-licenses/lgpl-2.1.html
Linux Kernel Headers	GNU General Public License 2.0 http://www.gnu.org/licenses/old-licenses/gpl-2.0.html
GNU Make	GNU General Public License 3.0 http://www.gnu.org/licenses/gpl.html

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 *$INSTALLDIR*/share/sourceryg++-x86_64-amd-linux-gnu-examples. These examples are not covered by the Sourcery CodeBench Software License Agreement. To the extent permitted by law, these examples are provided by Mentor Graphics 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.2.1. Third-Party Information

Additional attribution and license information for third-party software bundled with Mentor Graphics Proprietary Components can be found in *$INSTALLDIR*/share/doc/amd-x86_64-amd-linux-gnu/legal/.