Cycle Model Studio

Version 11.5

Installation Guide

arm

Cycle Model Studio

Installation Guide

Copyright © 2020, 2021 Arm Limited or its affiliates. All rights reserved.

Release Information

Document History

Issue	Date	Confidentiality	Change
1102-00	15 January 2020	Non-Confidential	Release 11.2
1103-00	13 November 2020	Non-Confidential	Release 11.3
1103-01	20 January 2021	Non-Confidential	Documentation update
1104-00	12 February 2021	Non-Confidential	Release 11.4
1105-00	25 June 2021	Non-Confidential	Release 11.5

110 Fulbourn Road, Cambridge, England CB1 9NJ.

(LES-PRE-20349)

Confidentiality Status

This document is Non-Confidential. The right to use, copy and disclose this document may be subject to license restrictions in accordance with the terms of the agreement entered into by Arm and the party that Arm delivered this document to.

Unrestricted Access is an Arm internal classification.

Product Status

The information in this document is Final, that is for a developed product.

Web Address

developer.arm.com

Progressive terminology commitment

Arm values inclusive communities. Arm recognizes that we and our industry have used terms that can be offensive. Arm strives to lead the industry and create change.

This document includes terms that can be offensive. We will replace these terms in a future issue of this document.

If you find offensive terms in this document, please contact *terms@arm.com*.

Contents
Cycle Model Studio Installation Guide

Preface

This preface introduces the *Cycle Model Studio Installation Guide*.

It contains the following:

-

About this book

This document provides instructions for installing the Cycle Model Studio software, and includes information about system requirements, environment variables, and licensing.

Using this book

This book is organized into the following chapters:

Chapter 1 Introduction

This section describes system requirements for Cycle Model Studio, and information about analytics collected when using the product.

Chapter 2 Installing the Cycle Model Studio software

This section describes how to access, install, and validate Cycle Model Studio software.

Chapter 3 Licensing overview

This section describes licensing requirements for Cycle Model Studio, and how to use the license queuing feature.

Glossary

The Arm® Glossary is a list of terms used in Arm documentation, together with definitions for those terms. The Arm Glossary does not contain terms that are industry standard unless the Arm meaning differs from the generally accepted meaning.

See the *Arm Glossary* for more information.

Typographic conventions

italic

Introduces special terminology, denotes cross-references, and citations.

bold

Highlights interface elements, such as menu names. Denotes signal names. Also used for terms in descriptive lists, where appropriate.

`monospace`

Denotes text that you can enter at the keyboard, such as commands, file and program names, and source code.

<u>`mono`</u>`space`

Denotes a permitted abbreviation for a command or option. You can enter the underlined text instead of the full command or option name.

`monospace italic`

Denotes arguments to monospace text where the argument is to be replaced by a specific value.

`monospace bold`

Denotes language keywords when used outside example code.

`<and>`

Encloses replaceable terms for assembler syntax where they appear in code or code fragments. For example:

```
MRC p15, 0, <Rd>, <CRn>, <CRm>, <Opcode_2>
```

SMALL CAPITALS

Used in body text for a few terms that have specific technical meanings, that are defined in the *Arm® Glossary*. For example, IMPLEMENTATION DEFINED, IMPLEMENTATION SPECIFIC, UNKNOWN, and UNPREDICTABLE.

Feedback

Feedback on this product

If you have any comments or suggestions about this product, contact your supplier and give:

- The product name.
- The product revision or version.
- An explanation with as much information as you can provide. Include symptoms and diagnostic procedures if appropriate.

Feedback on content

If you have comments on content then send an e-mail to *errata@arm.com*. Give:

- The title *Cycle Model Studio Installation Guide*.
- The number 101106_1105_00_en.
- If applicable, the page number(s) to which your comments refer.
- A concise explanation of your comments.

Arm also welcomes general suggestions for additions and improvements.

———— **Note** ————

Arm tests the PDF only in Adobe Acrobat and Acrobat Reader, and cannot guarantee the quality of the represented document when used with any other PDF reader.

Other information

- *Arm® Developer*.
- *Arm® Documentation*.
- *Technical Support*.
- *Arm® Glossary*.

Chapter 1
Introduction

This section describes system requirements for Cycle Model Studio, and information about analytics collected when using the product.

It contains the following sections:

1.1 Intended audience

These instructions are intended for system administrators or other users familiar with shell commands and installation packages.

1.2 System requirements and prerequisites

This section describes space and platform requirements.

Disk space and memory requirements

- 3 GB of disk space for unpacked media.
- For RAM and working memory, we recommend a minimum of 2 GB RAM for running Cycle Model Studio. Some designs may require more or less memory. Indications that you may be running out of memory include unexplained errors during the compilation, such as: `g++: internal compiler error: Killed (program cc1plus)`. Contact support-esl@arm.com if you are experiencing problems.

Supported operating systems and required packages

The supported Linux operating systems are:

- Red Hat Enterprise Linux 7.0 (64-bit)

You must install certain additional packages and group packages:

──────── **Note** ────────

If you are using a package manager other than yum, refer to its documentation for instructions on installing the required additional packages.

──────────────────────

1. Add the following line to `/etc/yum.conf`.

```
multilib_policy=all
```
2. Execute the following command to install the necessary group packages:

```
yum groupinstall "Additional Development" "Compatibility Libraries" "Development tools" "Perl Support"
```
3. Execute the following command to install additional required font packages:

```
yum install xorg-x11-fonts-75dpi xorg-x11-fonts-100dpi
```
4. Execute the following command to install additional required packages:

```
yum install libXext libXext-devel libXrender libXrender-devel glibc-devel
```

SystemC integration

To integrate a Cycle Model into a SystemC development environment, you need a supported version of SystemC. See the *Cycle Model Studio SystemC Integration Application Note*) (101198) .

FSDB file compatibility

To view FSDB dump files created by Cycle Models, Synopsys Verdi 2017.03-SP1 or later is required.

1.3   Simulation dependencies and requirements

The compiled Cycle Model has certain dependencies in order to simulate properly.

- Cycle Models must be able to find `libstdc++.so` from GCC 4.8.3 or later. Add the directory that contains `libstdc++.so` to the `LD_LIBRARY_PATH` environment variable.
- If you are compiling custom code and using a third-party simulation tool, use the GCC version provided by the simulation tool. The GCC version provided by the simulation tool must be GCC 4.8.3 or later.
- Ensure only a single GCC version is included within your environment to avoid library conflicts.

1.4 Data collection in Cycle Model Studio

Arm periodically collects anonymous information about the usage of our products to understand and analyze what components or features you are using, with the goal of improving our products and your experience with them. Product usage analytics contain information such as system information, settings, and usage of specific features of the product. They do not include any personal information.

Host information includes:

- Operating system name, version, and locale.
- Number of CPUs.
- Amount of physical memory.
- Screen resolution.
- Processor and GPU type.

——————— **Note** ———————

To disable analytics collection for all tools running in the environment, set the environment variable `ARM_DISABLE_ANALYTICS` to any value, including 0 or an empty string. This setting is not saved in persistent storage. It must be reset at subsequent invocations of the tool.

———————————————————

Chapter 2
Installing the Cycle Model Studio software

This section describes how to access, install, and validate Cycle Model Studio software.

It contains the following sections:

2.1 Accessing the Cycle Model Studio installation package

The Cycle Model Studio installation package is available for download for Arm account holders. It includes the Cycle Model Compiler and the Cycle Model Studio GUI.

Access the Cycle Model Studio installation package from the Downloads page on the Support section of the Arm IP Exchange web site (*https://ipx.arm.com/*). Copy the file to your host machine.

The installation package is available as a tarball: `Arm-CycleModel-release-<version>.tgz`.

2.2 Installing Cycle Model Studio

Extract the contents of the installation package, and set required environment variables.

Extract the contents of the installation package

To install Cycle Model Studio software:

1. Create a directory where you want to install the software:

   ```
   mkdir installation directory
   ```

2. Change the working directory to the installation directory:

   ```
   cd installation directory
   ```

3. Untar the Cycle Model Studio installation package:

   ```
   tar xzf Arm-CycleModel-release-vversion.tgz
   ```

Set the License environment variable

Prior to running Cycle Model Studio or simulations using Cycle Models, you must set the Arm-specific license environment variable `ARMLMD_LICENSE_FILE`. This environment variable offers the best performance, although you may use the standard FlexNet license variable `LM_LICENSE_FILE` instead.

- For Linux csh shell:

  ```
  setenv ARMLMD_LICENSE_FILE licenseFile
  ```

  where *licenseFile* is either a license file or *socket@hostname*. For example:

  ```
  setenv ARMLMD_LICENSE_FILE 7275@licserver
  ```

- For Linux Bourne shell:

  ```
  ARMLMD_LICENSE_FILE=licenseFileexport ARMLMD_LICENSE_FILE
  ```

  where *licenseFile* is either a license file or *socket@hostname*. For example:

  ```
  ARMLMD_LICENSE_FILE=7275@licserver
  export ARMLMD_LICENSE_FILE
  ```

Set the Home and Path environment variables

Source the Cycle Model Studio setup script to set the CARBON_HOME and PATH environment variables:

- Bourne shell - `source CMS install path/etc/setup.sh`
- C-Shell - `source CMS install path/etc/setup.csh`

Cycle Model Studio users may find it convenient to insert one of these command lines into their login files. As a root user Administrator, you can insert the appropriate command line into the global logins of all users who require access to Cycle Model Studio.

2.3 Installed tools and files

The Cycle Model Studio is installed under a single directory structure. This file system must be visible to all systems that run Cycle Model Studio, or multiple installation areas must exist.

The root directory is `$CARBON_HOME`.

bin/
Cycle Model Studio executables

etc/
Setup scripts

examples/
Example designs

include/
API header files

lib/
Libraries

license_terms/
License files

Linux/
Third-party executables for the Linux platform (GCC), and appropriate libraries

Linux64/
Third-party executables for the Linux 64-bit platform (GCC), and appropriate libraries

makefiles/
Makefiles

README
Cycle Model Studio readme file

userdoc/
All Cycle Model Studio documentation

util/
Files for Cycle Models Studio utilities.

2.4 Validating the installation

Ensure that the installation of Cycle Model Studio has been successful by running one of the examples.

To test the Cycle Model Studio installation, you can run a Verilog example:

1. Launch Cycle Model Studio by entering `modelstudio` at the command line.

   ```
   ./bin/modelstudio
   ```

   The Cycle Model Studio GUI launches.
2. Quit Cycle Model Studio.
3. Return to the terminal window and copy the example files into your local work directory:

   ```
   cp -r $CARBON_HOME/examples/twocounter ./twocounter
   ```
4. Change to your work directory:

   ```
   cd twocounter
   ```
5. Run the Makefile in the twocounter directory:

   ```
   make
   ```

The results of the example are output to the `twocounter.out` file:

```
0: clk1=1 reset1=1 clk2=1 reset2=1 out1=0 out2=0
100: clk1=1 reset1=1 clk2=1 reset2=1 out1=0 out2=0
200: clk1=1 reset1=1 clk2=1 reset2=1 out1=0 out2=0
300: clk1=1 reset1=1 clk2=1 reset2=1 out1=0 out2=0
400: clk1=1 reset1=1 clk2=1 reset2=1 out1=0 out2=0
500: clk1=1 reset1=1 clk2=1 reset2=1 out1=0 out2=0
. . .
```

If the example runs without error, then Cycle Model Studio software has been installed properly.

Chapter 3
Licensing overview

This section describes licensing requirements for Cycle Model Studio, and how to use the license queuing feature.

It contains the following sections:

3.1 Required licenses

Arm Cycle Model products are licensed using the FlexNet license manager Version 11.13.

You are required to have a license for Cycle Model Studio. Licenses are available on the Arm Support site (*https://developer.arm.com/support*). Registration and login are required.

A license server must be available on your network.

Contact Arm Techical Support (support-esl@arm.com) if you have any questions.

3.2 License queuing

License queuing is a Cycle Model Studio feature that allows a client to request a checkout of a license feature without failing if none are available. Instead, if no license feature is available, the client waits until the feature becomes available.

——————— **Note** ———————

Cycle Model Compiler supports two methods of enabling license queueing: setting the `CM_ENABLE_LICENSE_Q` environment variable as described in this section, or by specifying the command line option `-licq` at compilation time. See the *Cycle Model Compiler User Manual* (101050) for more information.

————————————————

Enabling license queuing

To enable license queuing for compilation:
1. Set the environment variable `CM_ENABLE_LICENSE_Q` in your shell.
2. Compile.

For example:

```
$ setenv CM_ENABLE_LICENSE_Q 1
$ cbuild foo.v
```

Checking the License Queue

To check the status of the Cycle Model Runtime license queue, use the Flexera command `lmstat -f feature`. For more information about monitoring and managing the License Queue, see the Flexera documentation.

The following example checks the Cycle Model Runtime feature license status:

```
lmstat -f CM_ARM_Runtime
lmstat - Copyright (c) 1989-2015 Flexera Software LLC. All Rights Reserved.
Flexible License Manager status on Mon MM/DD/YYYY hh:mm

License server status: 3737@us-license-server
License file(s) on us-license-server: my_license_file.lic:

your_machine: license server UP (MASTER) v11.13.1

Vendor daemon status (on us-license-server):

armlmd: UP v11.13.1
Feature usage info:

Users of CM_ARM_Runtime:  (Total of 1 license issued;  Total of 1 license in use)

"CM_ARM_Runtime" v1.0, vendor: armlmd, expiry: 1-jan-0
floating license

User us-license-server /dev/pts/51 (v1.0) (us-license-server/3737 101), start Mon MM/DD hh:mm
User us-license-server /dev/pts/51 (v1.0) (us-license-server/3737 201) queued for 1 license
```