



Installation Guide for CrossCore Embedded Studio

Contents

1	Introduction	4
2	System Hardware Requirements	5
3	Supported Operating Systems	6
4	Installing CrossCore Embedded Studio on Windows	7
5	Installing CrossCore Embedded Studio on Linux	8
5.1	Different users sharing the same CCES license on Linux	9
6	Installing SHARC+ Linux Command-Line Tools	10
6.1	Supported Parts	10
6.2	Installing Using the CrossCore Embedded Studio IDE	10
6.3	Installing Manually From the Command Line	11
7	Obtaining Technical Support	12

Copyright Information

© 2025 Analog Devices, Inc., ALL RIGHTS RESERVED. This document may not be reproduced in any form without prior, express written consent from Analog Devices, Inc.

Disclaimer

Analog Devices, Inc. reserves the right to change this product without prior notice. Information furnished by Analog Devices is believed to be accurate and reliable. However, no responsibility is assumed by Analog Devices for its use; nor for any infringement of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under the patent rights of Analog Devices, Inc.

Trademark and Service Mark Notice

The Analog Devices logo, CrossCore, A²B, EngineerZone, EZ-Board, EZ-KIT Lite and VisualDSP++ are registered trademarks of Analog Devices, Inc. Blackfin, Blackfin+, SHARC, SHARC+ and SigmaStudio are trademarks of Analog Devices, Inc. All other brand and product names are trademarks or service marks of their respective owners.

1 Introduction

This document describes the installation procedures for CrossCore Embedded Studio under both Windows and Linux as well as minimum hardware requirements and supported versions of each operating system. You can find the release notes for older releases in the Docs sub-directory of your CCES installation.

2 System Hardware Requirements

Verify that your PC has these minimum requirements for the CCES installation:

- 3.3GHz dual core or better recommended
- 8 GB RAM; 16GB or more recommended
- 11 GB available disk space

Note

A faster disk drive or SSD decreases the build time, especially for a large amount of source files. 8GB of RAM or more will substantially increase the performance of the IDE.

3 Supported Operating Systems

Notes for Windows Users

The following versions of Windows are supported for this release of CCES:

- Windows 10 Pro or Enterprise (32-bit and 64-bit)
- Windows 11 Pro or Enterprise (64-bit)

Notes for Linux Users

The following Linux distributions are supported for this release of CCES:

- Ubuntu 20.04 64-bit
- Ubuntu 22.04 64-bit

4 Installing CrossCore Embedded Studio on Windows

 **Note:** Windows Only

Caution

Windows users may experience User Access Control (UAC) related errors if the software is installed into a protected location, such as Program Files or Program Files (x86). We recommend installing the software in a non-UAC-protected location.

To install CrossCore Embedded Studio, double-click on ADI_CrossCoreEmbeddedStudio-Rel3.0.3.exe

To uninstall CrossCore Embedded Studio, open Control Panel / Programs and Features applet and select to uninstall CrossCore Embedded Studio 3.0.3. You may need to delete the installation directory to clean up any leftover files.

5 Installing CrossCore Embedded Studio on Linux

 **Note:** Linux Only

The following features are available and supported by this installation:

- Compilation using the Xtensa toolchain for the SHARC-FX core of ADSP-SC83x and ADSP-2183x processors.
- Compilation using the GNU ARM toolchain for the ARM Cortex-M core of ADSP-SC83x processors.
- Compilation using the GNU toolchain for the ARM Cortex-A core on [ADSP-SC57x](#), [ADSP-SC58x](#) and [ADSP-SC59x](#) processors.
- Compilation using the GNU ARM toolchain for the [ADuCM36x](#), [ADuCM302x](#) and [ADuCM4x50](#) ARM Cortex-M cores.
- Debugging [ADSP-SC5xx](#), [ADuCM360](#), [ADuCM302x](#) and [ADuCM4x50](#) via the IDE with GDB /OpenOCD.
- Development and debugging of Applications for ADSP-SC83x and ADSP-2183x processors.
- Development and debugging of Applications running under Linux on the ARM Cortex-A core on [ADSP-SC57x](#), [ADSP-SC58x](#) and [ADSP-SC59x](#) processors.
- Development and debugging of bare-metal applications on the [ADuCM360](#), [ADuCM302x](#) and [ADuCM4x50](#) ARM Cortex-M cores.

The following features are available after installing *SHARC Linux Command-Line Tools* though *Help>Install New Software...*:

- Building and command-line execution via the functional simulator (but not cycle-accurate simulation, or debugging within the IDE) of [SHARC+](#) processors.

The following features are *only* supported under Windows:

- Development, simulation and debug of [SHARC](#) processors (excluding the ARM Cortex-A core on [ADSP-SC57x](#), [ADSP-SC58x](#) and [ADSP-SC59x](#) processors)
- Use of CrossCore Embedded Studio Add-Ins other than the Startup Code/LSP add-in for SHARC-FX and the System Services and Device Driver add-ins
- Debugging an Application using the native CrossCore Debugger

Caution

It is strongly recommended to use the command prompt to install CrossCore Embedded Studio. The installation may not work properly when using Ubuntu Software and/or Ubuntu Software Center.

Some CrossCore Embedded Studio (CCES) components require 32-bit support, so you will need to enable that first. Installing the CrossCore Embedded Studio package will fail otherwise.

```
sudo dpkg --add-architecture i386
sudo apt-get update
```

To install CrossCore Embedded Studio run the following command from the command prompt:

```
sudo apt-get install ./adi-cces-linux-amd64-3.0.3.deb
```

To uninstall CrossCore Embedded Studio run the following commands from the command prompt:

```
sudo apt-get purge adi-cces-3.0.3
sudo rm -rf /opt/analog/cces/3.0.3 (to clean up any leftover files)
```

If you want to use the tools from the command line (e.g. via make, cmake or similar) you should add the following directories to your path, whereby \$CCES_BASE needs to be replaced with the CCES install location, which normally is /opt/analog/cces/<version>.

- `$CCES_BASE : $CCES_BASE/Xtensa/XtensaTools/bin :$CCES_BASE/ARM/gcc-arm-embedded/bin:$CCES_BASE/ARM/arm-none-eabi/bin:$CCES_BASE/ARM/aarch64-none-elf/bin`

5.1 Different users sharing the same CCES license on Linux

Many users can share a single valid license.dat file on a system by creating a symbol link to the valid license.dat in their own home directory (~/.analog/cces).

The user who installed license should ensure that the appropriate directory and file permissions are set-up to allow other users to access the valid license.dat.

6 Installing SHARC+ Linux Command-Line Tools

The SHARC+ toolchains are available as an optional extra installation, through *Help>Install New Software...*

Please note:

- These toolchains are made available in order to support Linux-hosted Continuous Integration /Continuous Deployment environments.
- These toolchains do *not* contain support for cycle-accurate simulation, or debugging SHARC applications within the IDE on Linux hosts.
- The appropriate Linux-hosted version of CrossCore Embedded Studio must be installed first, then the *SHARC Linux Command-line Tools* can be installed.
- There is typically a delay of 1–2 weeks between the release of the Windows-hosted CCES product and the corresponding *SHARC Linux Command-line Tools*.

There are two ways to install the SHARC toolchains, described below.

6.1 Supported Parts

The supported processors are all ADSP-2156x, ADSP-2157x, ADSP-2158x, ADSP-2159x, ADSP-SC57x, ADSP-SC58x parts and ADSP-SC59x.

Earlier SHARC processors are not supported.

6.2 Installing Using the CrossCore Embedded Studio IDE

In these steps, replace "x.y.z" with the appropriate version number for your CrossCore Embedded Studio installation.

1. If you are using docker, make sure you create a container that exports its display to your local machine.
2. Open the IDE by executing the following command:
`/opt/analog/cces/x.y.z/Eclipse/cces &`
3. Go to *Help > Install New Software...*
4. Click the down arrow to the right of the *Work with:* entry box
5. Choose *CrossCore Embedded Studio Software and Documentation* - <http://www.analog.com/static/ccesupdatesite>

6. From the resulting list of Available Software, expand the *CrossCore Toolchain Linux Support* category
7. Choose *SHARC and Blackfin Linux Command-Line Tools* , version x.y.z.<timestamp>
8. Click *Next*>
9. Click *Finish*
10. Close CrossCore Embedded Studio

6.3 Installing Manually From the Command Line

In these steps, replace "x.y.z" with the appropriate version number for your CrossCore Embedded Studio installation.

1. Download the JAR package from http://www.analog.com/static/ccesupdatesite/blackfin_sharc_linux/x.y.z-SNAPSHOT/plugins/com.analog.crosscore.incubation.blackfin_sharc_linux.stage_x.y.z.<timestamp>.jar
 - The <timestamp> can be obtained by finding the installation first using CCES following the steps above to step (7). You can also do this using CCES on a Windows machine.

2. Make sure the unzip utility is available:

```
sudo apt install unzip
```

3. Extract the payload tarball from the JAR:

```
unzip com.analog.crosscore.*.jar support_files.tar.gz
```

4. Unpack the tarball into the CCES install:

```
sudo tar xvf support_files.tar.gz -C /opt/analog/cces/x.y.z
```

7 Obtaining Technical Support

You can reach Analog Devices software and tools technical support in the following ways:

- Post your questions in the [software and development tools support community](#) at [EngineerZone[®]](#)
- E-mail your questions about software and development tools directly from CrossCore Embedded Studio by choosing Help > Email Support or directly to processor.tools.support@analog.com
- E-mail your questions about processors and processor applications to processor.support@analog.com
- Submit your questions to technical support directly via <http://www.analog.com/support>
- Contact your [Analog Devices sales office](#) or authorized distributor