



## **Release Notes for CrossCore Embedded Studio 2.4.0 on Linux**

# Contents

1	Introduction	3
1.1	Supported Linux Distributions	3
1.2	System Requirements	3
1.3	Obtaining Technical Support	3
2	Using CrossCore Embedded Studio on Linux (Ubuntu 14.04 32-bit)	5
2.1	Installation	5
2.1.1	Installing CrossCore Embedded Studio on 32-bit operating system	5
2.1.2	Uninstalling CrossCore Embedded Studio	5
2.2	Licenses	5
2.2.1	Different users sharing the same CCES license (license.dat file)	5
2.3	Supported Features	6
2.4	Known Issues	6
2.4.1	OpenOCD needs to be run as sudo on Linux	6
2.4.2	X11 forwarding is required to create or build projects from a command line with CCES on Linux remotely by SSH	7
2.4.3	Other Known Issues	7

# 1 Introduction

This document describes the changes for CrossCore Embedded Studio (CCES) 2.4.0 on Linux hosts.

CrossCore Embedded Studio 2.4.0 for Linux supports Linux application development on the ARM Cortex-A5 cores of the ADSP-SC5xx family of Digital Signal Processors from Analog Devices and bare-metal application development on the ARM Cortex-M cores of the ADuCM302x family from Analog Devices. General bare-metal application development for ADSP-SC5xx is not recommended nor is there support for architectures other than ARM Cortex-A and Cortex-M in this release (e.g. SHARC, SHARC+ or Blackfin).

## 1.1 Supported Linux Distributions

This release of CCES is supported on the following Linux distributions:

- Ubuntu 14.04 32-bit

## 1.2 System Requirements

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

- 2 GHz single core processor; 3.3GHz dual core or better recommended
- 1 GB RAM; 4GB or more recommended
- 2 GB available disk space
- One open USB port

### Note

A faster disk drive or SSD decreases the build time, especially for a large amount of source files. 4GB of RAM or more will substantially increase the performance of the IDE.

## 1.3 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](mailto:processor.tools.support@analog.com)
- E-mail your questions about processors and processor applications to [processor.support@analog.com](mailto: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

## 2 Using CrossCore Embedded Studio on Linux (Ubuntu 14.04 32-bit)

### 2.1 Installation

#### Caution

It is strongly recommended to use the command prompt to install CrossCore Embedded Studio. Post-install configuration may fail when installing via Ubuntu Software Center.

#### 2.1.1 Installing CrossCore Embedded Studio on 32-bit operating system

Install CrossCore Embedded Studio by running the following command from the command prompt:

```
sudo dpkg -i adi-CrossCoreEmbeddedStudio-linux-x86-2.4.0.deb
```

#### 2.1.2 Uninstalling CrossCore Embedded Studio

Uninstall CrossCore Embedded Studio by running the following commands from the command prompt:

```
sudo dpkg -r adi-cces-2.4.0
sudo dpkg -P adi-cces-2.4.0
sudo rm -rf /opt/analog/cces/2.4.0 (to clean up any leftover files)
```

### 2.2 Licenses

#### 2.2.1 Different users sharing the same CCES license (license.dat file)

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.

## 2.3 Supported Features

This release of CrossCore Embedded Studio for Linux has been provided to support the Linux Add-In for CrossCore Embedded Studio and support bare-metal development on Cortex-M processors (ADuCM302x).

The following features are available and supported:

- Compilation using the GNU toolchain for the ADSP-SC58x ARM Cortex-A core.
- Compilation using the GNU ARM toolchain for the ADuCM302x ARM Cortex-M cores.
- Debugging ADSP-SC58x and ADuCM302x via the IDE with GDB/OpenOCD.
- Development and debugging of Applications running under Linux on the ADSP-SC58x ARM Cortex-A core.
- Development and debugging of bare-metal applications on the ADuCM302x ARM Cortex-M core.

The following features are only supported via the Windows version of CrossCore Embedded Studio:

- Development, simulation and debug of Blackfin processors
- Development, simulation and debug of SHARC processors (excluding ADSP-SC58x ARM core)
- Use of CrossCore Embedded Studio Add-Ins other than the Linux Add-In
- Debugging an Application using the CrossCore Debugger (TPSDK)

## 2.4 Known Issues

### 2.4.1 OpenOCD needs to be run as sudo on Linux

In order to debug an Application with GDB and OpenOCD (Emulator) on Linux, OpenOCD needs to have permission to access your USB device. You can set-up the necessary permission when installing CCES on Linux or afterwards by running `sudo sh /opt/analog/cces/2.4.0/Setup/setup_openocd_permissions.sh`.

If you debug an Application with GDB and OpenOCD (Emulator) using the IDE and OpenOCD fails, because it cannot access your USB device, a dialog will appear with a message telling you that you can run the `setup_openocd_permissions.sh` script.

If you start CCES with sudo permission, then there should be no problems with OpenOCD accessing your USB device.

## 2.4.2 X11 forwarding is required to create or build projects from a command line with CCES on Linux remotely by SSH

If you want to use the CrossCore Embedded Studio headless tools application to create or build projects from a command line on Linux remotely by SSH, then you will need to setup:

1. X11 forwarding needs to be enabled on both the client side and the server side.
2. Have X Server (e.g. [Xming](#)) setup on your client side.

To make sure your server side has enabled X11 forwarding, check if its `/etc/ssh/sshd_config` contains:

```
X11Forwarding yes
X11DisplayOffset 10
```

If you use Putty as client, enable the X11 forwarding option by checking Configuration > Connection > SSH > X11 > Enable X11 forwarding and adding an X display location (e.g. localhost:0.0).

If you use Cygwin as client, enable the X11 forwarding by the following commands:

```
export DISPLAY=localhost:0.0
ssh -XY username@remote_server_ip
```

## 2.4.3 Other Known Issues

For the latest anomalies please consult our [Software and Tools Anomalies Search](#) page.