



Release Notes for CrossCore Embedded Studio 2.8.0

Contents

1	Introduction	3
1.1	Supported Operating Systems	3
1.2	System Requirements	4
1.3	Obtaining Technical Support	5
2	Installing CrossCore Embedded Studio	6
2.1	Installing CrossCore Embedded Studio on Windows	6
2.2	Installing CrossCore Embedded Studio on Linux	6
2.2.1	Different users sharing the same CCES license on Linux	7
2.2.2	OpenOCD needs to be run as sudo on Linux	7
3	New and Noteworthy	8
3.1	Support for ADSP-BF71x Blackfin Processors	8
3.2	Eclipse Upgraded to Version 4.7	8
3.3	Intel Hex and BIN File Format Support	8
3.4	CMSIS Pack File Support Improvements	8
3.5	Improved Support for ARM Mbed OS Export Creating CrossCore Embedded Studio projects	9
3.6	Add-in Initialization Source File Generation	9
3.7	New support for SHARC/SHARC+ Arithmetic Exception Interrupt Handlers	10
3.8	Command Line Device Programmer (CLDP) No Longer Requires a License	10
3.9	Headless Builder Support for Saving File Specific Tool Options in JSON Files	11
4	Changes That Might Impact Backwards Compatibility	12
4.1	New ea2570 assembler warning for silicon anomaly 15000033	12
5	Known Issues	13
5.1	CCES Internal Build does not generate input-file.txt	13
5.2	µC/OS-II® applications trigger asynchronous abort exception for ADSP-SC58x and ADSP-SC57x parts.	13
5.3	ADSP-SC57x/ADSP-2157x Processors Running at 500 MHz	14
5.4	DDR configuration register settings for ADSP-SC5xx and ADSP-215xx processors have been modified	14
5.5	Additional Known Issues	14

1 Introduction

This document describes the changes for CrossCore Embedded Studio (CCES) 2.8.0. You can find the release notes for older releases in the docs sub-directory of your CCES installation.

1.1 Supported Operating Systems

Notes for Windows Users

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

- Windows 7 Professional, Enterprise, or Ultimate (32 and 64-bit, SP1 or later)
- Windows 8.1 Pro or Enterprise (32 and 64-bit)
- Windows 10 Pro or Enterprise (32 and 64-bit)

Notes for Linux Users

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 such as the [ADuCM360](#), [ADuCM302x](#) and [ADuCM4x50](#) family of MCUs.

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

- Ubuntu 14.04 32-bit
- Ubuntu 16.04 32-bit

The following features are available and supported:

- Compilation using the GNU toolchain for the ARM Cortex-A core on [ADSP-SC57x](#) and [ADSP-SC58x](#) processors.
- Compilation using the GNU ARM toolchain for the [ADuCM360](#), [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 running under Linux on the ARM Cortex-A core on [ADSP-SC57x](#) and [ADSP-SC58x](#) processors.
- Development and debugging of bare-metal applications on the [ADuCM360](#), [ADuCM302x](#) and [ADuCM4x50](#) ARM Cortex-M cores.

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 the ARM Cortex-A core on [ADSP-SC57x](#) and [ADSP-SC58x](#) processors)
- Use of CrossCore Embedded Studio Add-Ins other than the [Linux Add-In](#)
- Debugging an Application using the native CrossCore Debugger

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
- 4 GB RAM; 8GB 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. 8GB 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
- 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

2 Installing CrossCore Embedded Studio

2.1 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.

Caution (Windows 8.1 users)

Prior to installation: Ensure your machine is up-to-date with relevant Windows updates from Microsoft. CrossCore Embedded Studio relies upon the Microsoft Universal C Runtime from VisualStudio 2015, and this can silently fail to install if your machine is out of date. For more details, refer to [Update for Universal C Runtime in Windows](#) on Microsoft's web site.

To install CrossCore Embedded Studio, double-click
ADI_CrossCoreEmbeddedStudio-Rel2.8.0.exe

To uninstall CrossCore Embedded Studio, open Control Panel / Programs and Features applet, and select to uninstall CrossCore Embedded Studio 2.8.0. You may need to delete the installation directory to clean up any leftover files.

2.2 Installing CrossCore Embedded Studio on Linux

 **Note:** Linux Only

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.

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

```
sudo dpkg -i adi-CrossCoreEmbeddedStudio-linux-x86-2.8.0.deb
```

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

```
sudo dpkg -r adi-cces-2.8.0  
sudo dpkg -P adi-cces-2.8.0  
sudo rm -rf /opt/analog/cces/2.8.0 (to clean up any leftover files)
```

2.2.1 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`.

2.2.2 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 permissions to access your USB device. You can set-up the necessary permissions when installing CCES on Linux by selecting 'Configure OpenOCD permissions' option on the installation dialog or afterwards by running `sudo sh /opt/analog/cces/2.8.0/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.

3 New and Noteworthy

3.1 Support for ADSP-BF71x Blackfin Processors

CrossCore Embedded Studio 2.8.0 adds support for the ADSP-BF71x family of A²B connected Blackfin processors:

- ADSP-BF715
- ADSP-BF716
- ADSP-BF718
- ADSP-BF719

For details on these new parts please visit the [Blackfin product page](#) or [contact your local ADI sales representative](#).

3.2 Eclipse Upgraded to Version 4.7

CrossCore Embedded Studio's Eclipse Platform has been upgraded to [Oxygen \(4.7\)](#) and its [C/C++ Development Tooling \(CDT\)](#) has been upgraded to 9.3.

- [New and Noteworthy in Eclipse 4.7](#)
- [New and Noteworthy in CDT 9.3](#)

3.3 Intel Hex and BIN File Format Support

You are now able to generate an Intel Hex or BIN Artifact when building a project. You can change the Artifact type by visiting your project's Tools Settings under the Build Artifact tab.

Intel Hex and BIN files can be used with the CrossCore Serial Flash Programmer and third party flash programming utilities.

3.4 CMSIS Pack File Support Improvements

Installing a CMSIS pack will now also install the CMSIS packs that it depends on, if they are missing from your installation.

If your Cortex-M project uses a CMSIS pack and it is not installed, a missing CMSIS pack dialog will appear providing more details of the missing CMSIS packs and how to install them.

3.5 Improved Support for ARM Mbed OS Export Creating CrossCore Embedded Studio projects

The ARM mbed-os [command line interface](#) can now export mbed projects to the CrossCore Embedded Studio IDE. Exporting from the command line will generate two files: `README.md` and `cces.json`. `README.md` contains directions on how to generate and build the CrossCore Embedded Studio project using headless tools from the command line. Headless tools accepts the `cces.json` input file to generate the project. Once generated, you can import the project into the CrossCore Embedded Studio IDE to build, develop and debug your application.

Export from the command line

You can export mbed-os projects from your local machine using the mbed python-based [command line interface](#) with the `export` command from within your mbed-os project's directory:

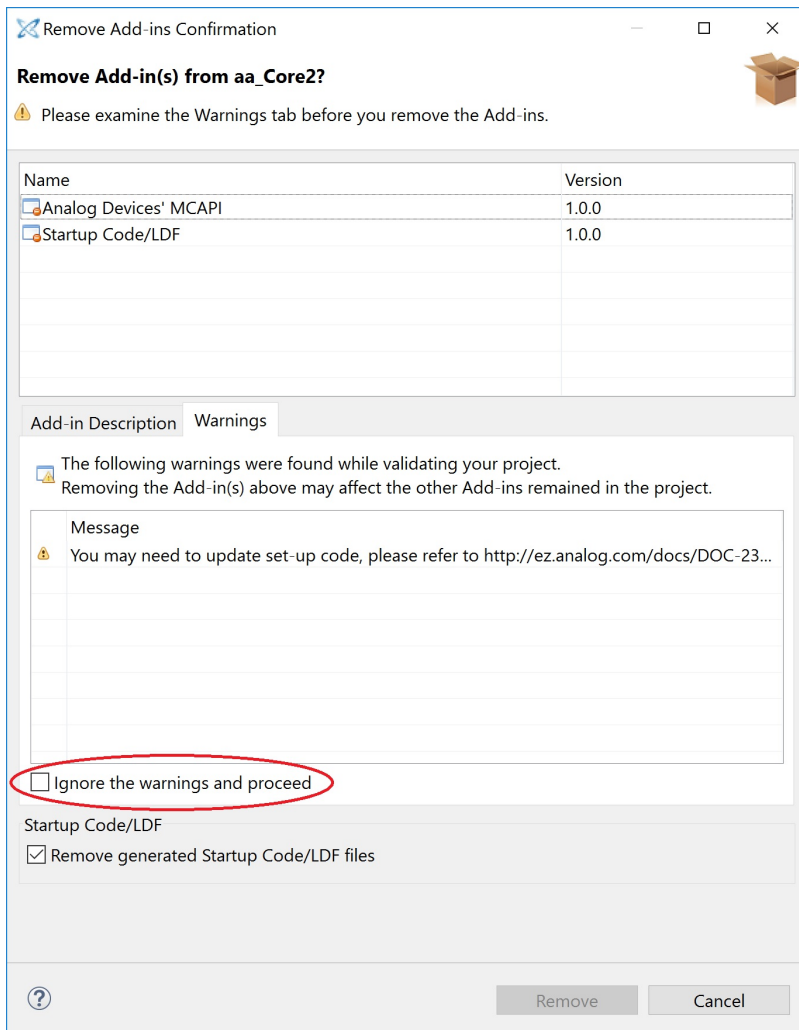
```
> mbed export -i cces -m EV_COG_AD4050LZ
```

3.6 Add-in Initialization Source File Generation

CrossCore Embedded Studio will no longer generate the `adi_initialize` header and C source files when you opt not to add support for Add-ins in your project or you do not select an Add-in during New Project creation.

If you have opted not to add support for Add-ins or you do not have an Add-ins in your project, then the `adi_initialize` header and C source files will no longer be re-generated when you open your project.

If you opt to remove all installed Add-ins in your project, there will be a warning message and you need to check the checkbox to finish the operation as following snapshot:



3.7 New support for SHARC/SHARC+ Arithmetic Exception Interrupt Handlers

New low-level RTL based support for defining handlers that can clear bits in STKYx and STKYy has been added. See SHARC and SHARC+ Arithmetic Exception Interrupt Handling in the CCES 2.8.0 help for more information.

3.8 Command Line Device Programmer (CLDP) No Longer Requires a License

In CCES 2.8.0, the need for a CCES license has been removed from the CLDP. Users can now use the CLDP without having to have a separate license for each machine it is used on. A full CCES install is still needed in order to use the CLDP however.

3.9 Headless Builder Support for Saving File Specific Tool Options in JSON Files

The headless builder now supportss file specific tool options in both directions. You can store the file specific options into JSON file by use of `-command projectinfo`. Conversely, you can create new project and set file specific options using the JSON file by `-command projectcreate` as well.

4 Changes That Might Impact Backwards Compatibility

4.1 New ea2570 assembler warning for silicon anomaly 15000033

The SHARC assembler for ADSP-214xx parts has been enhanced to raise a warning for assembly that may need to be modified to workaround silicon anomaly 15000033. For further information see the CCES help topic for ea2570

5 Known Issues

5.1 CCES Internal Build does not generate input-file.txt

When using Internal Builder for Blackfin or SHARC, the linker won't generate `input-file.txt` to contain object files like External Builder does, the object files will be appended to the command directly.

5.2 μ C/OS-II® applications trigger asynchronous abort exception for ADSP-SC58x and ADSP-SC57x parts.

The CCES 2.8.0 supplied initcodes and preloads for ADSP-SC58x and ADSP-SC57x parts now write to the DMC_CPHY_CTL registers as part of the Hardware Reference Manual recommended DMC (Dynamic Memory Controllers) initialization for DDR3 mode. These DMC_CPHY_CTL registers writes are done on core 0 and trigger a false data abort exception - this is silicon anomaly 20000091 (DMC_CPHY_CTL access may cause unexpected results). Applications using version 2.7.0 or earlier of the μ C/OS-II® Real-Time Kernel for CrossCore® Embedded Studio will hit this problem when using the new preloads and initcodes.

An example of the fatal error output seen in the CCES console when this problem occurs is shown below.

```
A non-recoverable error or exception has occurred.
Description:   Data Fault Exception - caused by attempting to access
General Type:  RunTimeError
Specific Type: ExceptAbrtData
Error Message: If this is a synchronous fault, address 0x0c87ba25 hel
Error PC:      0x8901657d
```

Update to μ C/OS-II® Real-Time Kernel for CrossCore® Embedded Studio version 2.8.0 (or newer) to avoid this problem.

5.3 ADSP-SC57x/ADSP-2157x Processors Running at 500 MHz

The ADSP-SC57x/ADSP-2157x family processors can have a maximum core clock frequency (CCLK) of 500 MHz. The ADSP-SC573 EZ-KIT and their pre-built initcode and preload executable support in CCES are configured instead for 450 MHz. When not using an EZ-KIT and developing for ADSP-SC58x/ADSP-2158x processors, the core clock frequency can be set to 500 MHz by modifying and rebuilding the preloads and initcodes. Refer to the `readme.txt` of an appropriate preload or initcode project for an explanation of how to set the core clock frequency. The ADSP-SC57x preload and initcode projects are located in the `SHARC/ldr/init_code/SC57x_Init` folder.

5.4 DDR configuration register settings for ADSP-SC5xx and ADSP-215xx processors have been modified

The values of the certain DDR configuration registers in the preload and initcode files have been updated to resolve JEDEC compliance issues with DDR memory on ADSP-SC5xx EZ-Kits:

- For ADSP-SC58x/2158x parts that support DDR3 memory, the PADCTL2 and MR registers have been updated
- For ADSP-SC58x/2158x parts that support DDR2 memory, the PADCTL2 register has been updated
- For the ADSP-SC57x/2157x, the PADCTL2, MR and MR1 registers have been updated.

5.5 Additional Known Issues

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