



# **Release Notes for CrossCore Embedded Studio 2.5.1**

# 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	Pre-installed ADuCM302x CMSIS-Pack file has been updated	8
3.2	ADuCM4x50 CMSIS-Pack file is supported, but not pre-installed.	8
3.2.1	EZK license	8
3.2.2	Installing ADuCM4x50 support	8
3.2.3	Creating a new project, building and debugging with GDB and OpenOCD	9
3.3	OpenOCD console now shows its standard output in black and errors are shown in red (CCES-16581)	11
3.4	adi_initialize.c and adi_initialize.h are re-generated if they do not exist in the project (CCES-15211)	12
3.5	Register Browser (CCES-15067)	12
4	Changes That Might Impact Backwards Compatibility	13
4.1	ADuCM302x LD script now matches Linker configuration file supplied with the ADuCM302x BSP for IAR. Less memory is available by default (GLUEM-280)	13
5	Known Issues	14
5.1	Register View does not show peripheral registers (CCES-14240)	14
5.2	X11 forwarding is required to create or build projects from a command line with CCES on Linux remotely by SSH	14
5.3	For More Information	15

# 1 Introduction

This document describes the changes for CrossCore Embedded Studio (CCES) 2.5.1. 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)
- 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 [ADuCM302x](#) family of MCUs.

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

- Ubuntu 14.04 32-bit

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](#) and [ADuCM4x50](#) ARM Cortex-M cores.
- Debugging [ADSP-SC58x](#), [ADuCM302x](#) and [ADuCM4x50](#) 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](#) 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 [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)

## 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

### **i 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](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 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.

To install CrossCore Embedded Studio, double-click  
`ADI_CrossCoreEmbeddedStudio-Rel2.5.1.exe`

To uninstall CrossCore Embedded Studio, click the Start Menu / Analog Devices / CrossCore Embedded Studio 2.5.1 / Uninstall CrossCore Embedded Studio 2.5.1 shortcut

### 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. Post-install configuration may fail when installing via Ubuntu Software Center.

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

```
sudo dpkg -i adi-CrossCoreEmbeddedStudio-linux-x86-2.5.1.deb
```

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

```
sudo dpkg -r adi-cces-2.5.1
sudo dpkg -P adi-cces-2.5.1
sudo rm -rf /opt/analog/cces/2.5.1 (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/2.5.1/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 Pre-installed ADuCM302x CMSIS-Pack file has been updated

CCES 2.5.1 comes pre-installed with the ADuCM302x 1.0.4 CMSIS-Pack file.

#### Future release

The ADuCM302x CMSIS-Pack file will not be pre-installed in the next major release of CCES. The pack file can be installed via the Keil CMSIS-Pack Index or by downloading it from the ADI website and installed from your local download. Please consult the CCES Online Help for more information on how to install CMSIS-Pack files.

### 3.2 ADuCM4x50 CMSIS-Pack file is supported, but not pre-installed.

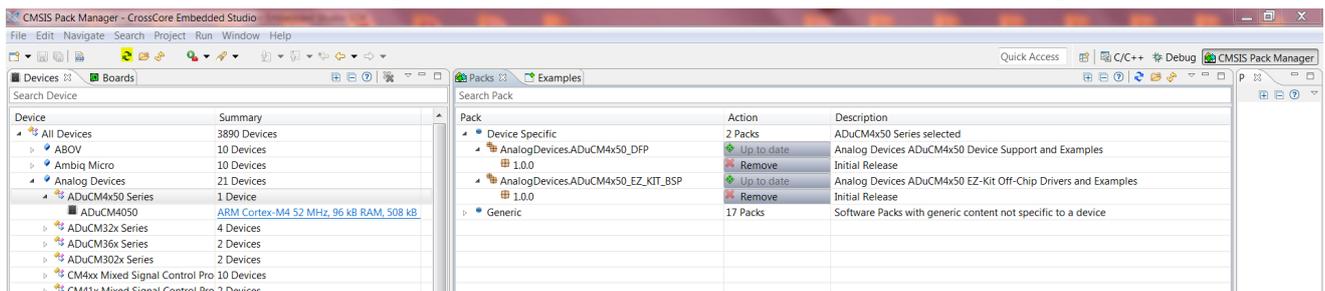
#### 3.2.1 EZK license

An EZK license is provided with each ADuCM4x50 EZ-Kit and the serial number can be used to activate your copy of CrossCore Embedded Studio (CCES).

As with the EZK license for ADuCM302x, this license will allow you to create and build ARM projects for Cortex-M processors and debug your applications with GDB and OpenOCD. The EZK license will not allow you to build or debug applications using the CrossCore toolchains and CrossCore Debugger.

#### 3.2.2 Installing ADuCM4x50 support

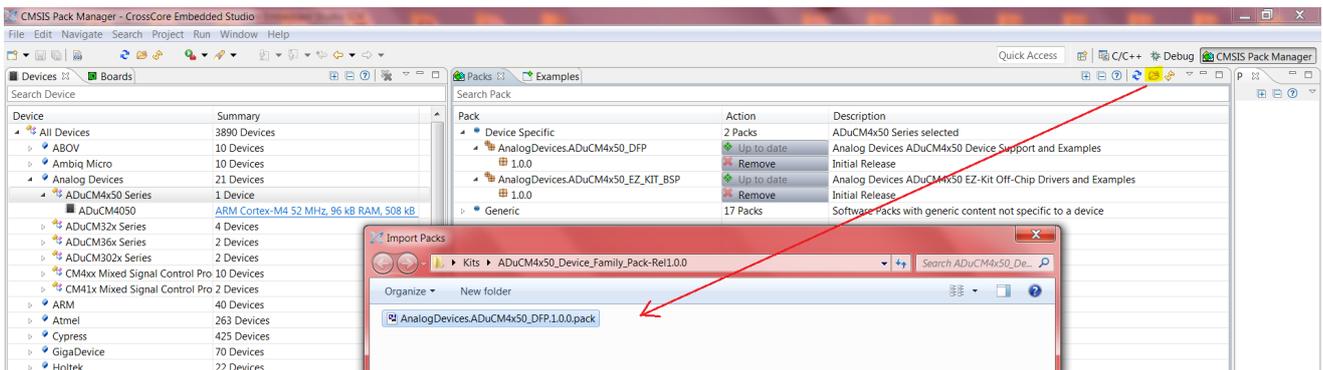
ADuCM4x50 CMSIS-Pack files can be installed by opening the CMSIS Pack Manager perspective in CCES and selecting Check for Updates on Web. ADuCM4050 will be available to install by selecting it from the Devices View, choosing the Device Family Pack (DFP) and/or Board Support Pack (BSP) in the Packs View, and clicking the Install Action.



Alternatively, the ADuCM4x50 Device Family Pack (DFP) and Board Support Pack (BSP) files can be installed using a local download.

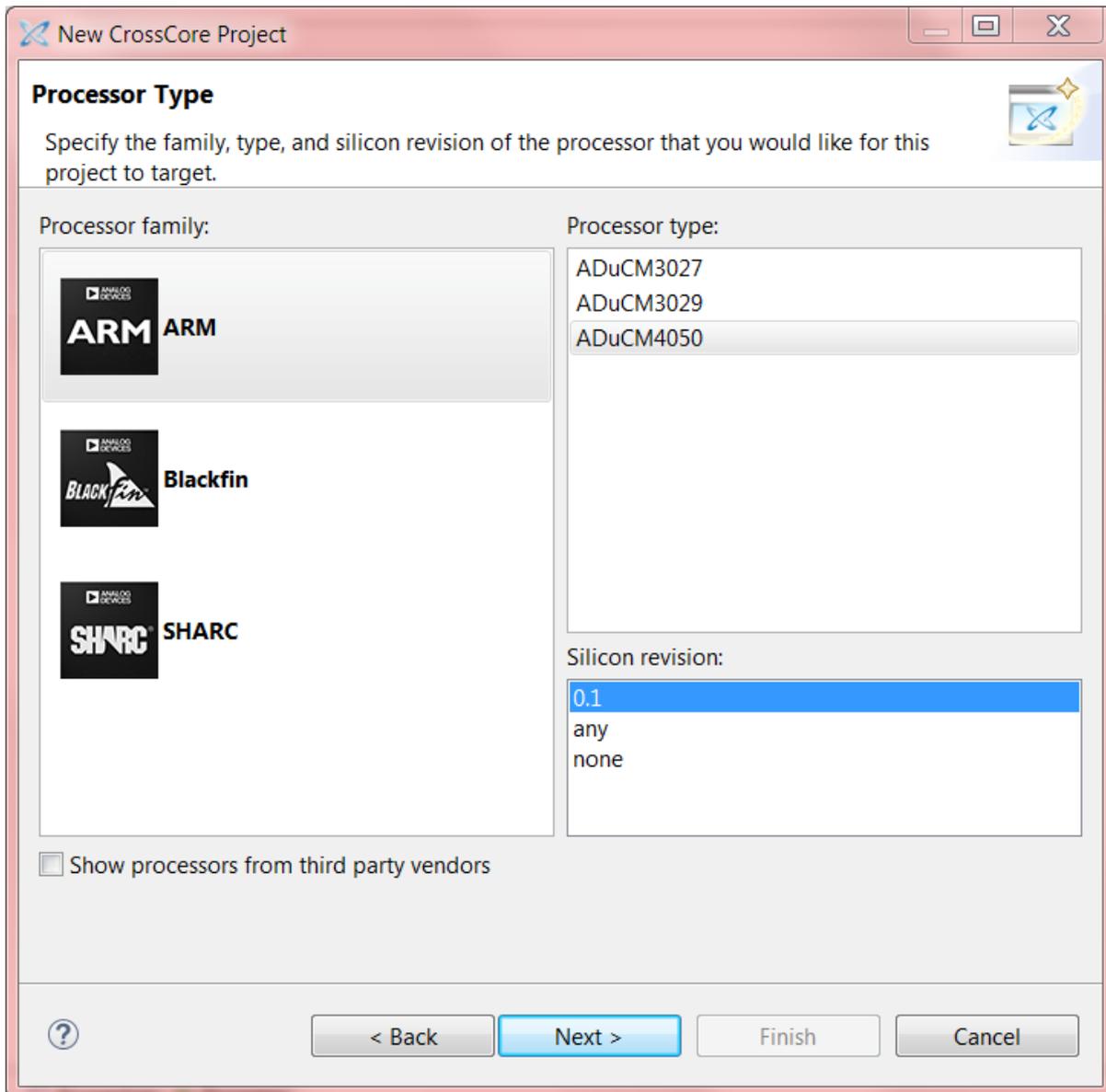
After downloading the .pack files from the ADI website, select the Import Packs button in the CMSIS Pack Manager's Packs View, choose the .pack file as shown in the screenshot below, and click Open.

You will be prompted to accept a license agreement and, after agreeing to it, the CMSIS-Pack file will be installed into CrossCore Embedded Studio's CMSIS-Pack installation directory (C:\Analog Devices\CrossCore Embedded Studio 2.5.1\ARM\packs\AnalogDevices).

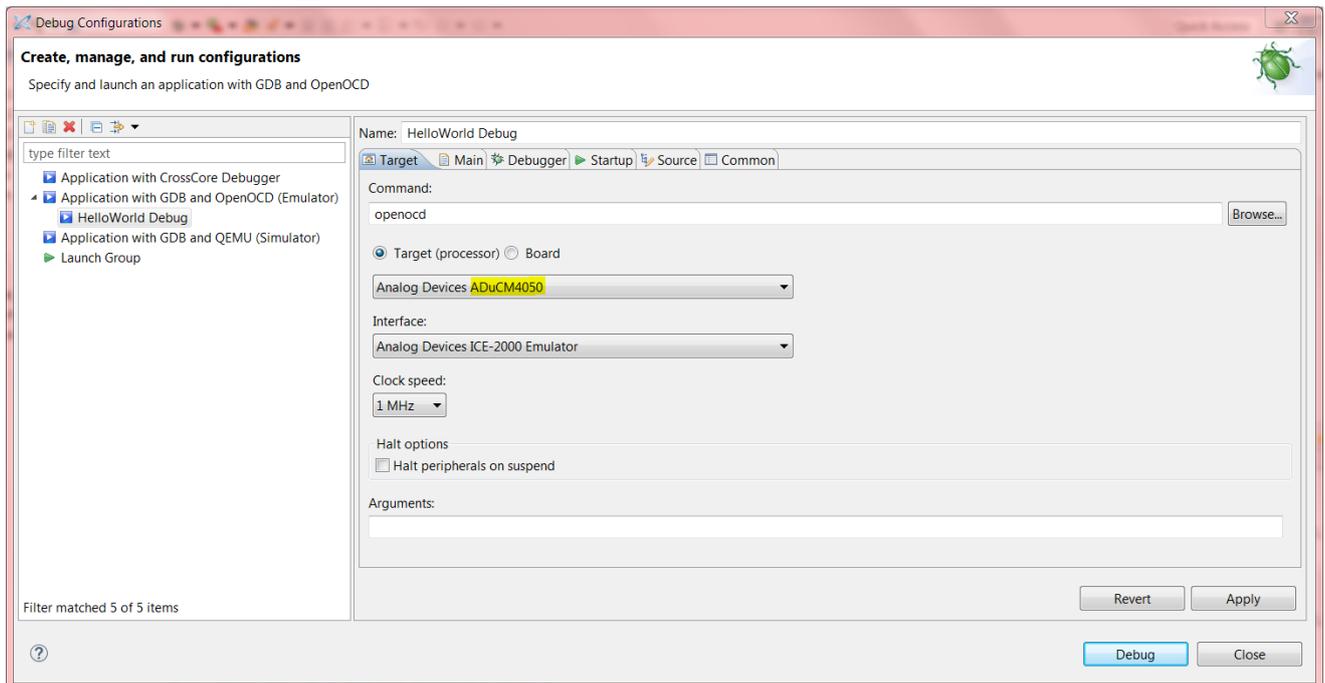


### 3.2.3 Creating a new project, building and debugging with GDB and OpenOCD

A project for ADuCM4050 can be created using the New CrossCore Project Wizard.



ADuCM4050 programs can be downloaded and debugged using GDB and OpenOCD.



For more information, please read the ADuCM4x50 Device Family Pack (DFP) for CCES Users Guide available in the CMSIS-Pack installation folder: C:\Analog Devices\CrossCore Embedded Studio 2.5.1 \ARM\packs\AnalogDevices\ADuCM4x50\_DFP\1.0.0\Documents.

### 3.3 OpenOCD console now shows its standard output in black and errors are shown in red (CCES-16581)

The console output from OpenOCD has been stream-lined so that its standard text output will appear in black while errors will be shown in red.

The output from OpenOCD can be seen in the Console View by selecting the OpenOCD process in the Debug View or by switching to the OpenOCD Console using the "Display Selected Console" option in the Console View itself.



### **3.4 adi\_initialize.c and adi\_initialize.h are re-generated if they do not exist in the project (CCES-15211)**

The CCES Add-in framework in CCES 2.5.1 will re-generate the System configuration's Initialization adi\_initialize.c and adi\_initialize.h source files when they're used in the project but were not found (i.e. deleted or were not added to source control).

These files can still be re-generated manually by right-clicking on the /system folder in your project and selecting Generate Add-in Initialization Source Files from the context menu.

### **3.5 Register Browser (CCES-15067)**

Improvements have been made to speed up the Register Browser.

## **4 Changes That Might Impact Backwards Compatibility**

### **4.1 ADuCM302x LD script now matches Linker configuration file supplied with the ADuCM302x BSP for IAR. Less memory is available by default (GLUEM-280)**

The default linker configuration file provided with the ADuCM302x BSP for IAR locates variables in SRAM regions to be retained during hibernation. In CCES, variables are located in non-retained SRAM.

The mapping of code and data in the CCES (GCC) ADuCM302x LD script now matches the Linker configuration file provided with the ADuCM302x BSP for IAR thereby avoiding confusion when porting an application set-up to use low power from one toolchain to another.

## 5 Known Issues

### 5.1 Register View does not show peripheral registers (CCES-14240)

If you are debugging a Cortex-M program (ADuCM302x/ADuCM4x50) with GDB and OpenOCD, then you will need to add an entry to the Expressions View in order to view the values of peripherals registers. CCES does not provide a Debug View to view peripheral registers.

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

 **Note:** Linux Only

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
```

## 5.3 For More Information

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