CrossCore® Development Tools by Analog Devices

Release Notes for CrossCore Embedded Studio 2.6.0

Contents

1 Introduction

This document describes the changes for CrossCore Embedded Studio (CCES) 2.6.0. 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)
- 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 and ADSP-SC57x ARM Cortex-A core.
- Compilation using the GNU ARM toolchain for the ADuCM302x and ADuCM4x50 ARM Cortex-M cores.
- Debugging ADSP-SC58x, ADSP-SC57x, ADuCM302x and ADuCM4x50 via the IDE with GDB/OpenOCD.
- Development and debugging of Applications running under Linux on the ADSP-SC58x and ADSP-SC57x 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 and ADSP-SC57x 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

> **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.

To install CrossCore Embedded Studio, double-click `ADI_CrossCoreEmbeddedStudio-Rel2.6.0.exe`

To uninstall CrossCore Embedded Studio, open Control Panel / Programs and Features applet, and select to uninstall CrossCore Embedded Studio 2.6.0. 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. 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-x86-2.6.0.deb
```

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

```
sudo dpkg -r adi-cces-2.6.0
sudo dpkg -P adi-cces-2.6.0
sudo rm -rf /opt/analog/cces/2.6.0 (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/2.6.0/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 Eclipse Neon 4.6 and CDT 9.2 (CCES-15053)

CrossCore Embedded Studio (CCES) Eclipse platform has been updated to Eclipse 4.6 (Neon) and the version of the Eclipse C/C++ Development Tooling (CDT) has been updated to version 9.2.

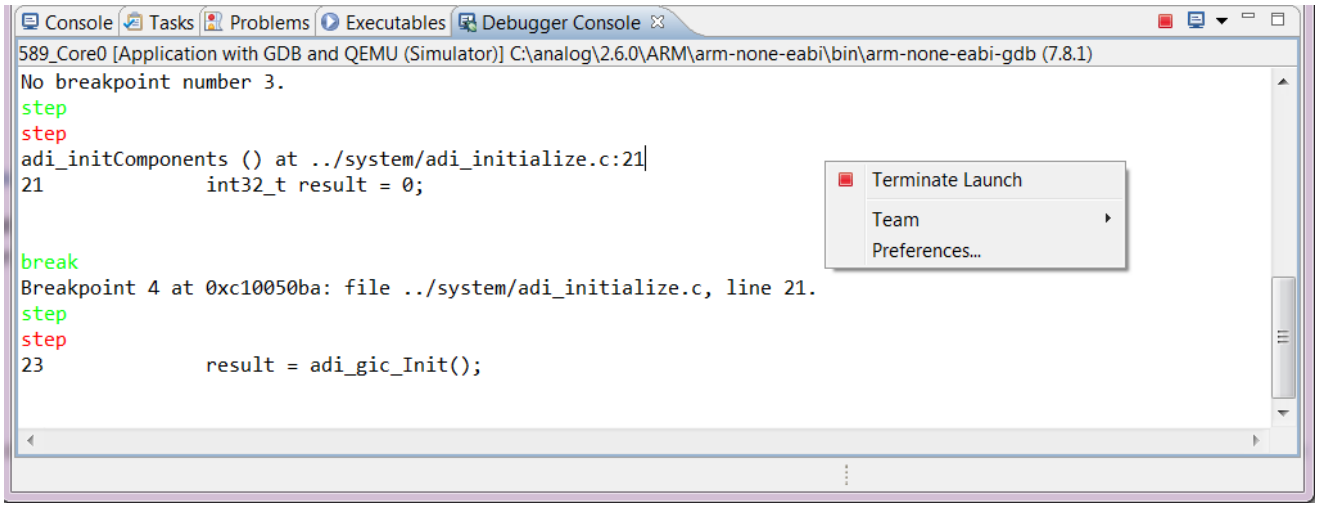
Eclipse 4.6 (Neon) and CDT 9.2 include many new features (new in CDT 9.x and new and Noteworthy for Eclipse 4.6), as well as API modifications that are not backward compatible.

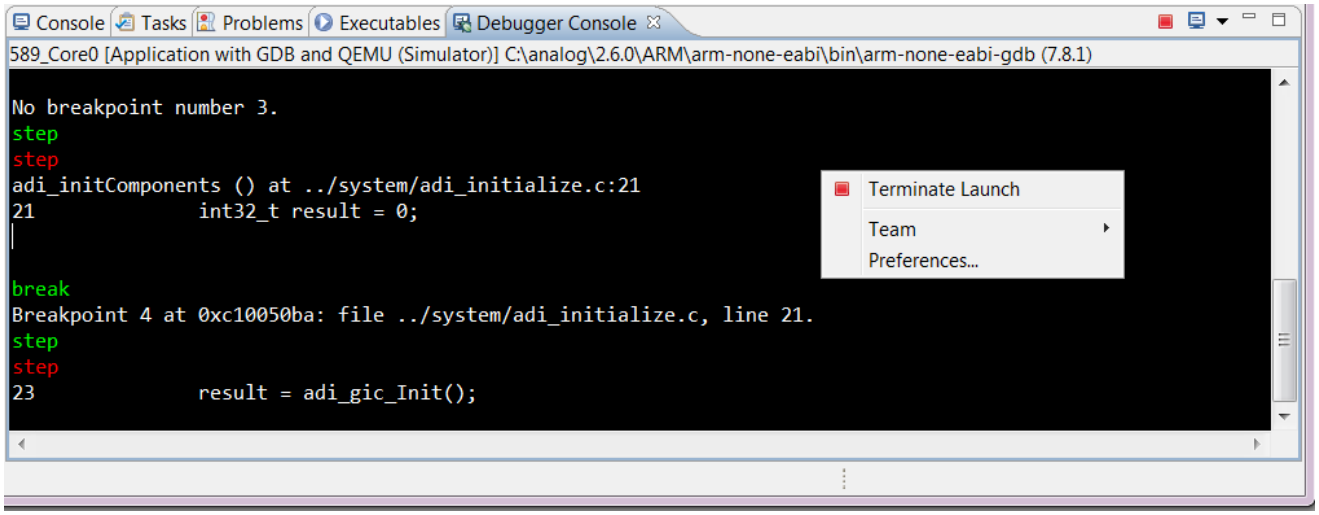
Due to the new Eclipse and CDT API modifications, it is recommended that you use the latest CrossCore Embedded Studio Micrium Add-in products at version 2.6.0 with CCES 2.6.0.

3.2 Full GDB Console

The GDB console has been moved from Console view to a new Debugger Console view, which is a full-featured GDB command-line console, its background defaults to white to better blend with the other views but can be set to black by using the console preferences.

When a debug session starts, the Debugger Console view will open automatically.

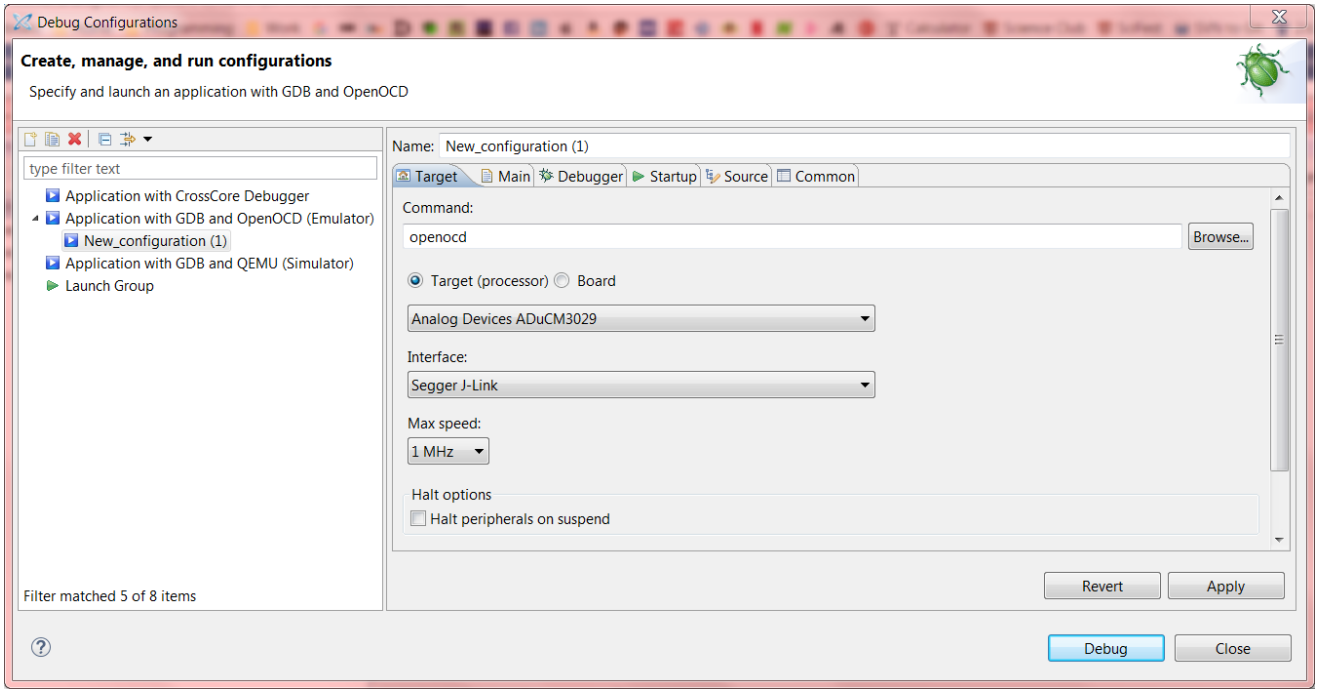




User can access the console preferences through the context-menu of the view, or under the GDB preference section.

3.3 Add J-Link Lite support to OpenOCD for Cortex-M (CCES-16524)

CrossCore Embedded Studio now has support for the J-Link Lite Emulator. To use your J-Link Lite to download and debug a program for Cortex-M parts, such as ADuCM3029 and ADuCM4050, select Segger J-Link as your Interface in your new Launch configuration.



Note: Windows Only

Installing UsbDk

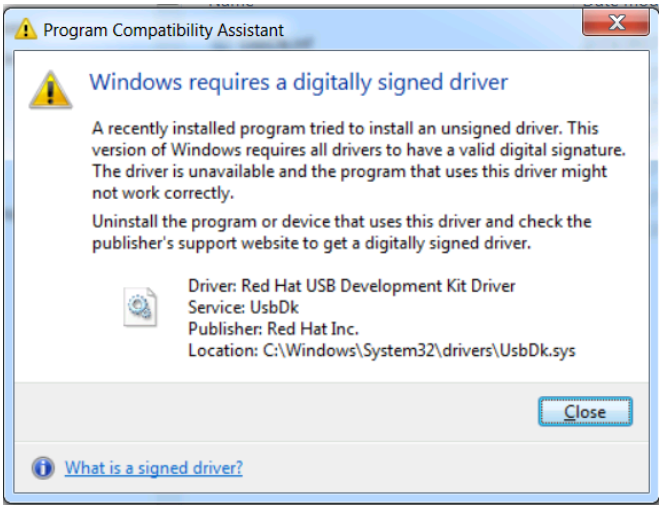
On Windows it is necessary to install UsbDk if you want to use J-Link Lite support with GDB and OpenOCD.

To install UsbDK, double-click UsbDk_1.0.17_x86.msi (on 32-bit Windows) or UsbDk_1.0.17_x64.msi (on 64-bit Windows). The files can be found in <Installation Directory>\Setup\UsbDk folder (by default, C:\Analog Devices\CrossCore Embedded Studio 2.6.0\Setup\UsbDk).

In order to install UsbDk on Windows 7 SP1, you must have the following Windows Update on your system,

https://technet.microsoft.com/en-us/library/security/3033929

You will get the following message during UsbDk installation if the above Windows Update is missing:



If you get the above error message, install the appropriate update for your operating system and reinstall UsbDk.

3.4 Faster ARM Debug performance for ADSP-SC5xx (CCES-14634)

The disassembly window for ARM updates faster when debugging in CCES for the ADSP-SC5xx processor

3.5 Improved Breakpoint support when core is running (CCES-17013)

Breakpoints cannot be added or removed while the core is running. There is now notification, and the breakpoint now remains visible in the Breakpoints tab.

In previous releases, if attempting to tick or untick a breakpoint in the Breakpoints tab while a core is running, the breakpoint would be added or removed from Breakpoints tab, but the breakpoint was not actually added or removed and there was no notification.

3.6 Improved Memory Browser when switching formats for SHARC and SHARC+ Cores (CCES-2712)

When switching formats from 8 or 16 bit Hex to 32 bit Hex in the Memory Browser using 2.5.1 and earlier, the Memory Browser would pad values with zeroes for the ADSP-SC5xx /ADSP-21xxx processors. The Memory Browser now correctly displays the 32 bit Hex format.

3.7 Improved Register Browser (CCES-13107, CCES-15067, CCES-14478, CCES-16427, CCES-16604, CCES-16688)

3.7.1 The IDDE has improvements for finding registers and when creating custom register browser tabs.

CCES introduced a cache and refined the infrastructures to sharply shorten the time to retrieve the register info, and also provided a better filter functionality for user to locate the target register group using the whole or part name or description of the register group, register or bit field. Besides that, the trigger of filtering in the edit pane was changed from ANY letter to ENTER only, this can save time when user wants to input a long key word, and also the UI is less likely to be blocked if there are huge numbers of filtered items.

CCES also introduced a filter to make finding the register group easier when you create a new register tab. You can match register group with the input filter text using the name of register group, register, bit field, or all of them, and you can choose to match with their description too, excluding register group which has no description for CrossCore debug target.

3.7.2 Register Browser updates are now faster for ADSP-SC5xx, ADSP-215xx, and ADSP-BF70x

Opening large register browser windows and stepping is now faster for ADSP-SC5xx /ADSP-215xx and ADSP-BF70x processors.

3.8 Plot View configuration dialogs(CCES-2953)

The IDDE has fixed some UI issue for the Plot View configuration dialogs.

3.9 New/Improved Functionality for Watchdog and General Purpose Timers for ADSP-SC57x/ADSP-2157x Functional Simulator

The ADSP-SC57x/ADSP-2157x functional simulator has new/improved functionality:

General Purpose Timer – Trigger toggle functionality added, additional TRU connections for same.

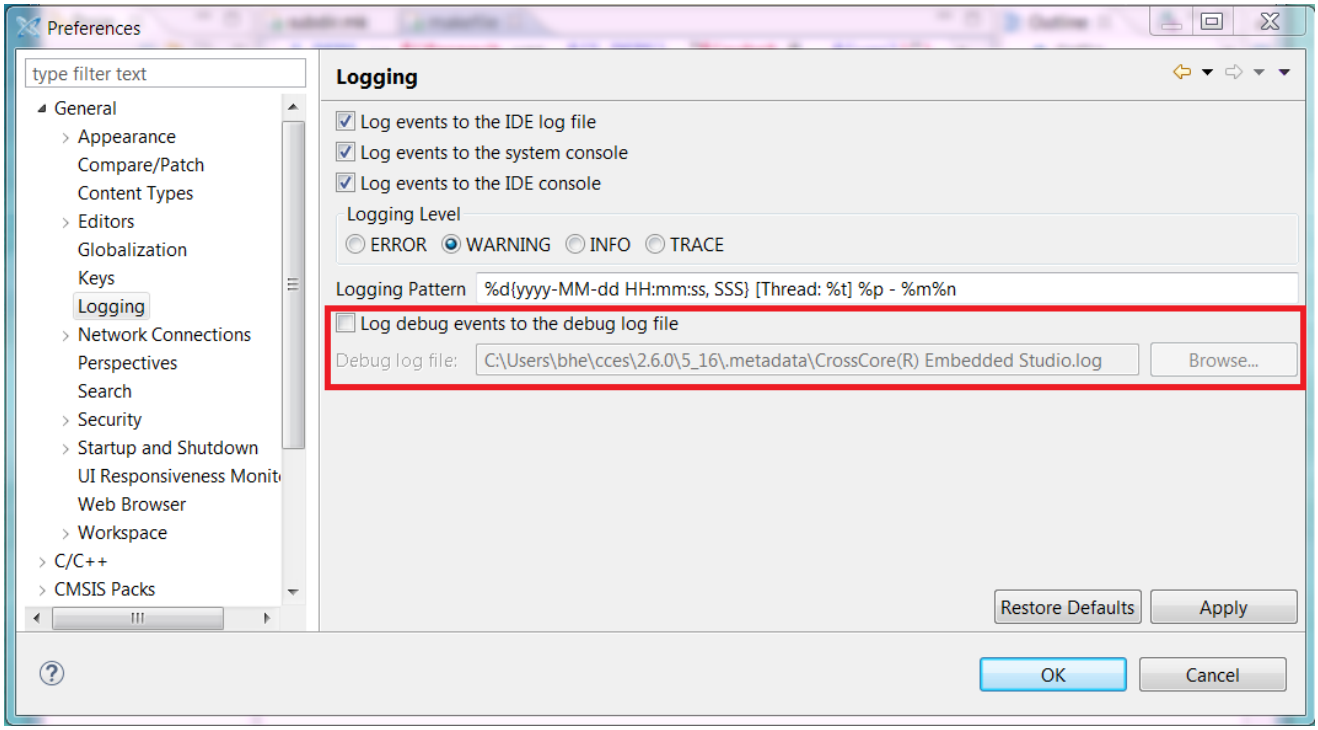
Watchdog Timer – Window feature addition, instance count increased to 3 from 2, connection to TRU.

3.10 Improved Stepping with ADSP-SC5xx/ADSP-215xx Functional Simulator (CCES-12353)

It is now possible to step in core 2 of an ADSP-SC5xx/ADSP-215xx functional simulator session and maintain the focus on core 2.

3.11 Logging is off by default (CCES-17024)

Logging debug events to the debug log file is now disabled by default, you can go to `Window > Preferences > General > Logging` to enable it.



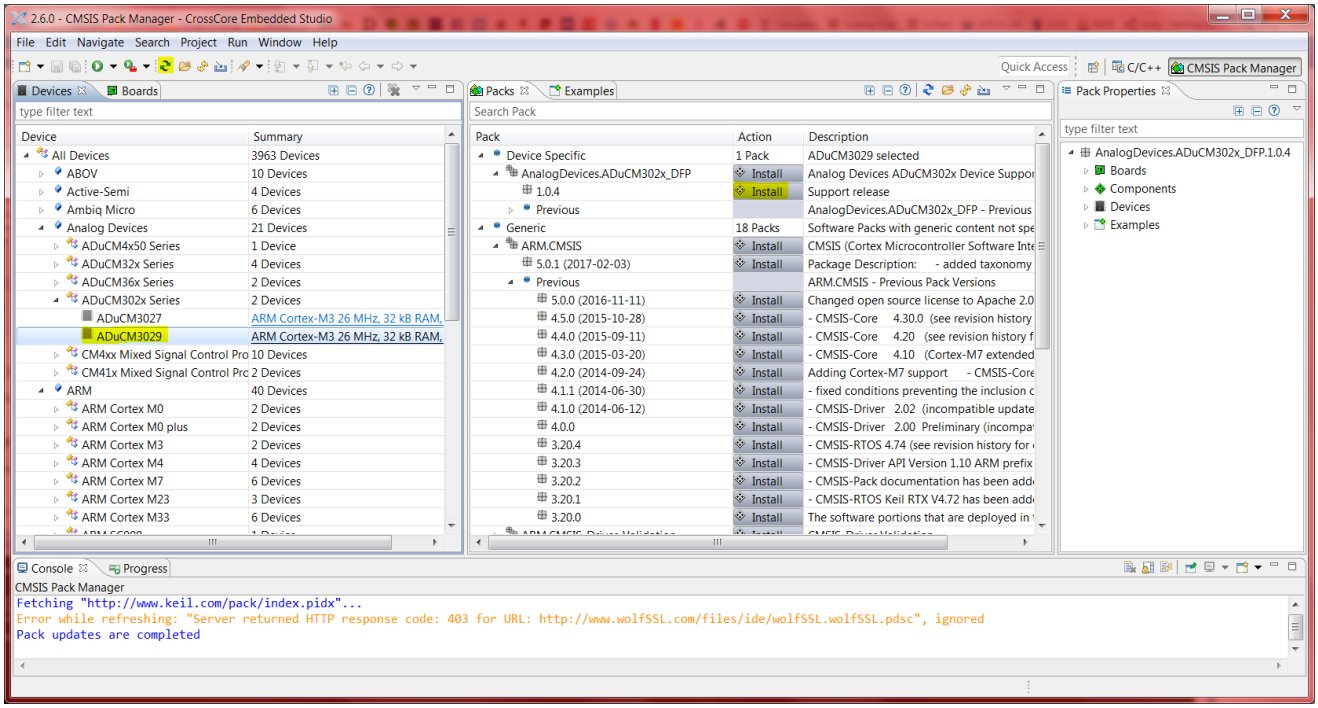
3.12 OpenOCD support for ADuCM360 (CCES-16894)

You are now able to load and debug ADuCM360 programs using GDB and OpenOCD on an ADICUP360 with CMSIS-DAP.

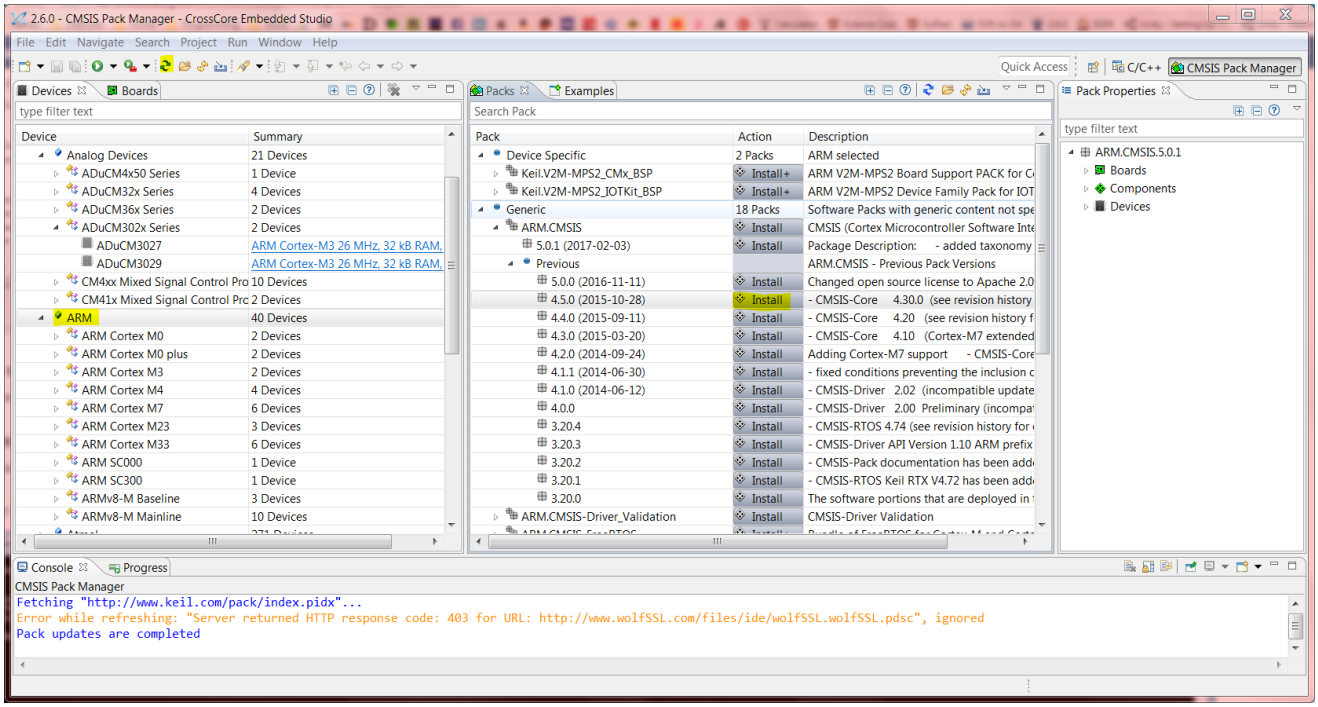
3.13 Pre-installed ADuCM302x Device Family Pack (DFP) and ARM 4.5.0 CMSIS Pack file have been removed (CCES-16721)

CrossCore Embedded Studio 2.6.0 no longer comes with the ADuCM302x Device Family Pack (DFP) and ARM 4.5.0 CMSIS Pack file pre-installed.

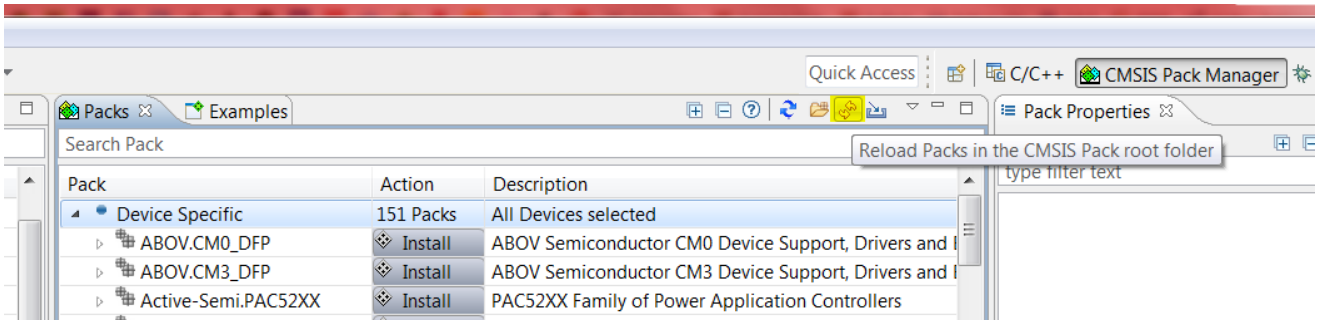
To install the ADuCM302x 1.0.x DFP, switch to the CMSIS Pack Manager perspective, select *Check for Updates on Web* (blue arrows on the toolbar), choose Analog Devices and ADuCM302x Series in the Devices View, select the ADuCM302x CMSIS Pack version (e.g. 1.0.4) from the Packs View, and click Install.



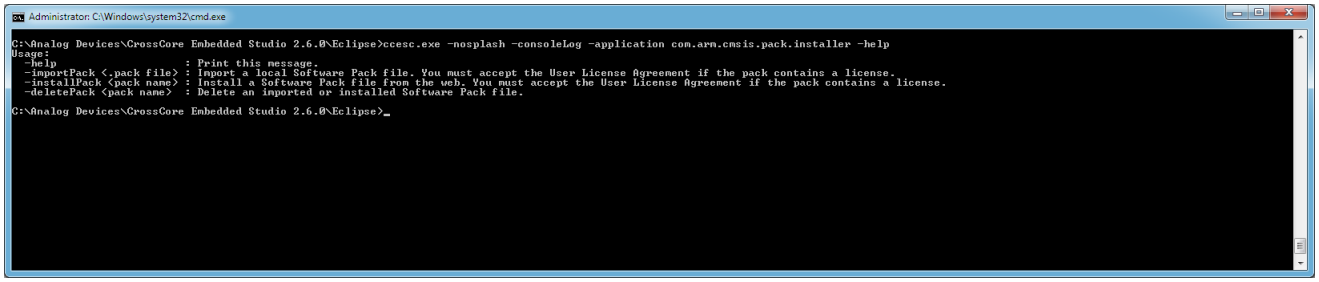
To install the ARM 4.5.0 CMSIS Pack, switch to the CMSIS Pack Manager perspective, select *Check for Updates on Web* (if you have not already done so), choose ARM in the Devices View, Generic and ARM.CMSIS from the Packs View, select 4.5.0 version, and click Install.



After installing your CMSIS Pack files, you may need to click *Reload Packs in the CMSIS Pack root folder* button on the toolbar for the pack information to appear.



3.14 Support for installing and uninstalling CMSIS-Pack files from a command line (CCES-16633)



CrossCore Embedded Studio 2.6.0 and later provides a new command line application to install and remove CMSIS pack files without the GUI, similar to the Command-Line Builder.

There are two methods of installing CMSIS pack files:

1. Import a pack file from a local directory.
2. Install a pack file from the web. The pack indexes to search are specified by the CMSIS Packs preferences in the IDE. By default, this searches Keil's pack index at http://www.keil.com/pack/index.pidx.

The CMSIS pack command line installer will return a non-zero error code in the event of an error or failure.

3.14.1 Windows

To run the CMSIS pack command line installer in Windows, run the `ccesc.exe` executable from the `CrossCore Embedded Studio 2.6.0\Eclipse` installation directory with the appropriate arguments. For example, to install the ADuCM4x50 DFP version 1.0.0:

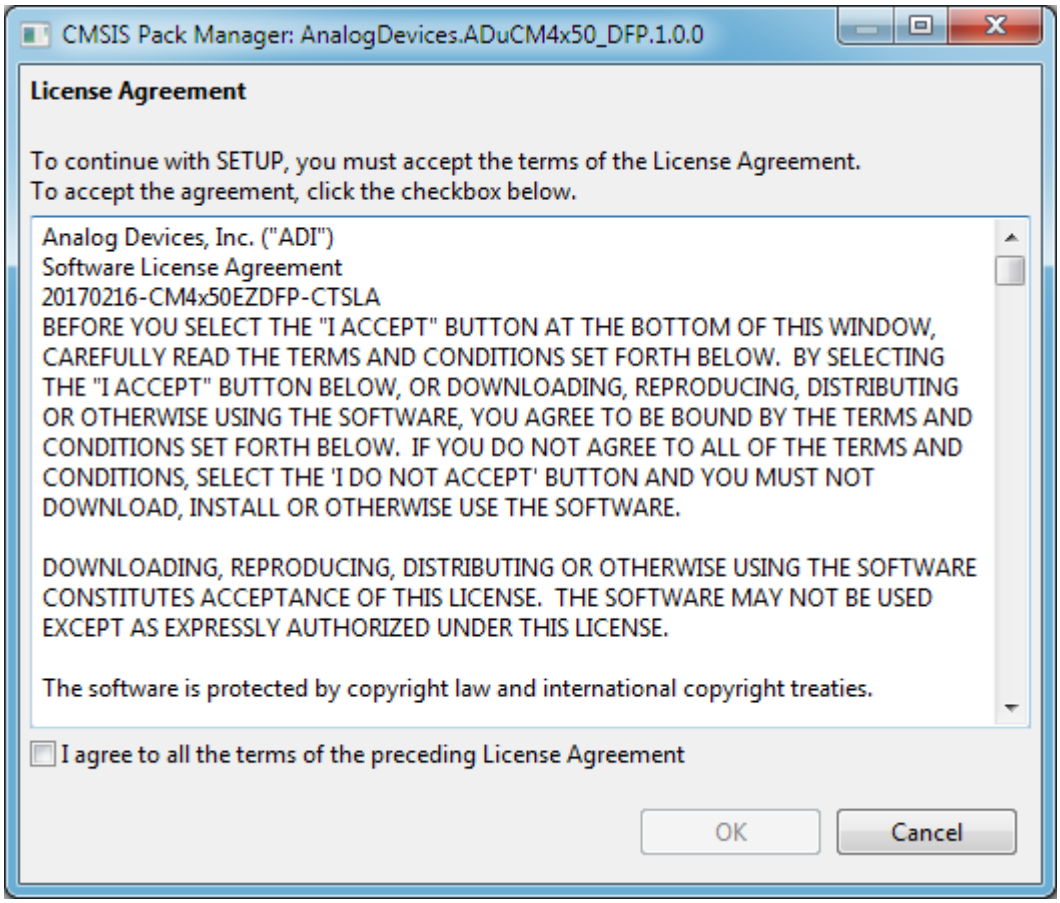
```
ccesc.exe -nosplash -consoleLog -application com.arm.cmsis.pack.
installer -installPack AnalogDevices.ADuCM4x50_DFP.1.0.0
```

3.14.2 Linux

To run the CMSIS pack command line installer in Linux, run the `cces` executable from the `cces/2.6.0/Eclipse` installation directory with the appropriate arguments. For example, to install the ADuCM4x50 DFP version 1.0.0:

```
cces -nosplash -consoleLog -application com.arm.cmsis.pack.installer
-installPack AnalogDevices.ADuCM4x50_DFP.1.0.0
```

If the CMSIS pack contains a User License Agreement, you will be presented with the License Agreement dialog:



You must accept the license agreement to complete the setup.

3.15 ADSP-SC58x/ADSP-2158x Processors Running at 500 MHz

The ADSP-SC58x/ADSP-2158x family processors can have a maximum core clock frequency (CCLK) of 500 MHz. The ADSP-SC589 EZ-KIT and ADSP-SC584 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-SC589 and ADSP-SC584 preload and initcode projects are located in the `SHARC/ldr/init_code/SC58x_Init` folder.

3.16 New Initcode and Preload projects for ADSP-2157x and ADSP-2158x (CCES-16635)

New `215xx_preload_Core1` and `215xx_init_Core1` SHARC+ projects have been added. These may be configured by adding defines of a suitable selection of the available `CONFIG_` macros into the config.h source to build preload and init code executables for ADSP-2157x or ADSP-2158x based custom hardware. The new projects are located in the `SHARC/ldr/init_code/215xx_Init` folder. See the "ADSP-SC5xx/ADSP-215xx Loader Collateral" CCES help topic for further information.

3.17 SHARC+ fir_interp reimplemented to use SIMD (CCES-16824)

The SHARC+ parts libdsp defined FIR interpolation filter function, `fir_interp`, has been updated to use SIMD when possible resulting in a performance improvement of 38%. To achieve this performance improvement you need to use an even `interp_index` parameter value and not use the `-no-simd` switch.

3.18 New SHARC+ fatal error added that detects startup reentry due to a jump or call of 0x0. (CCES-16767)

The CCES supplied and generated SHARC+ parts EMUI interrupt vector code has been modified to include a jump to new fatal error support function called `_adi_bad_reset_detected`. This change has been added to detect an application error that results in calling a NULL function pointer or jumping to address 0x0. An example

of the CCES debug console output for this error when it occurs is shown below. Previously this same application error would have resulted in a "HeapSetupFailure" fatal error.

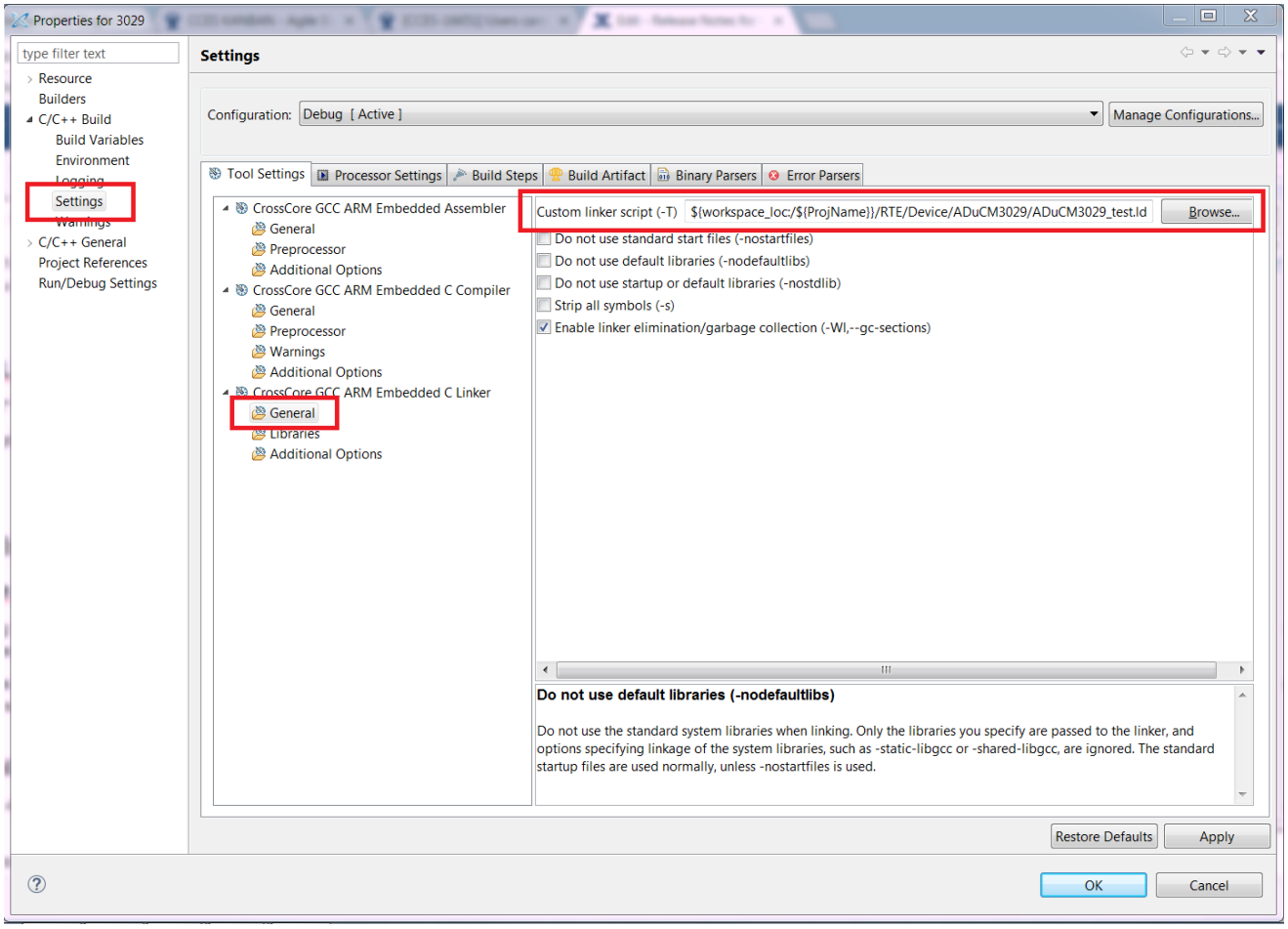
```
A non-recoverable error or exception has occurred.
 Description: A bad reset has been detected, possibly due to a jump to a NULL code

 General Type: RunTimeError
 Specific Type: BadResetDetected
 Error Message: If executing a C/C++ call through a function pointer, the return ac

 Error PC: 0x001c0bd1
```

3.19 Users can overwrite the .LD script that is copied into the project by the Startup CMSIS component (CCES-16651)

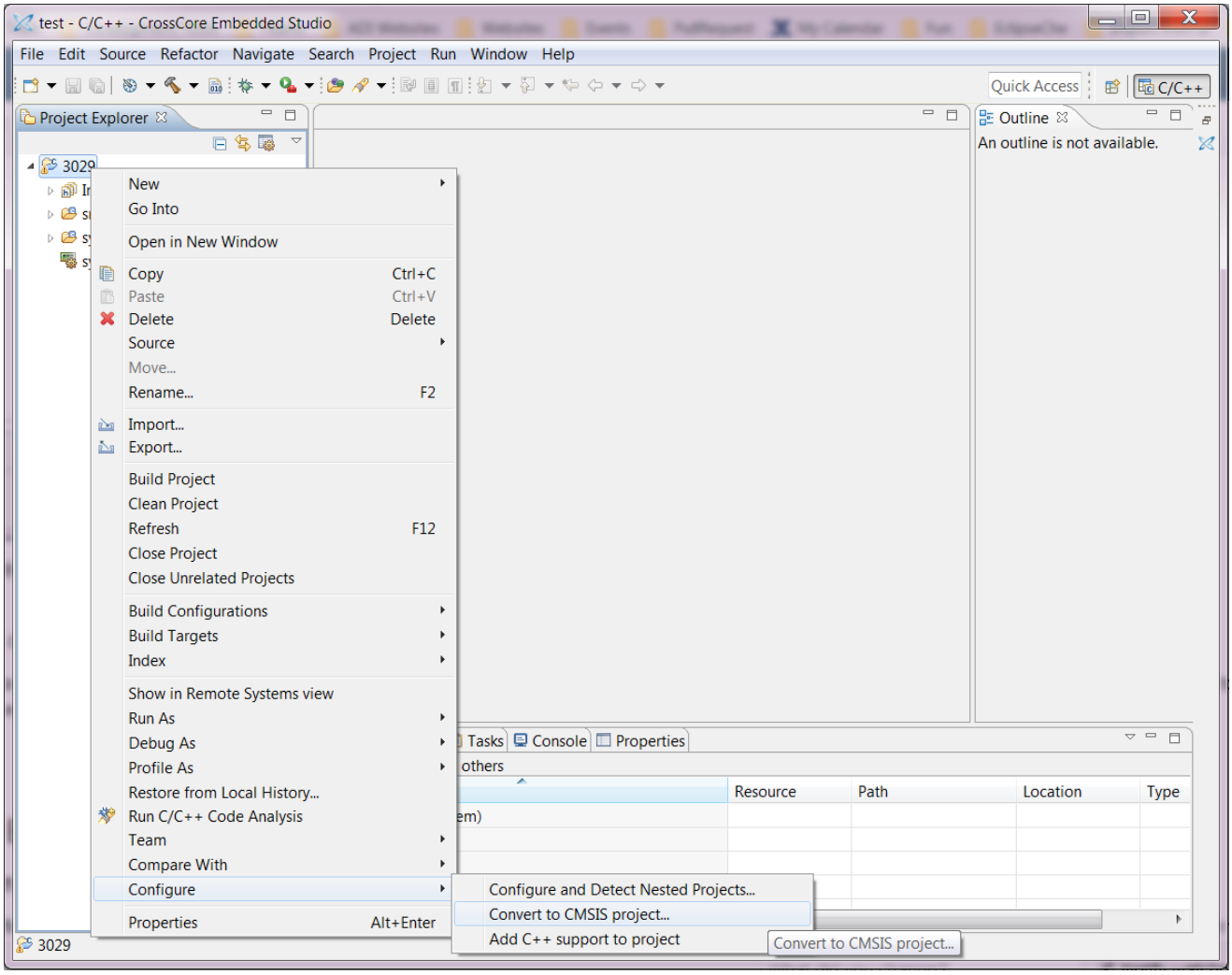
You are now able to specify "-T" pointing to your customized linker script from Tool Settings without overwritten by Startup CMSIS component.



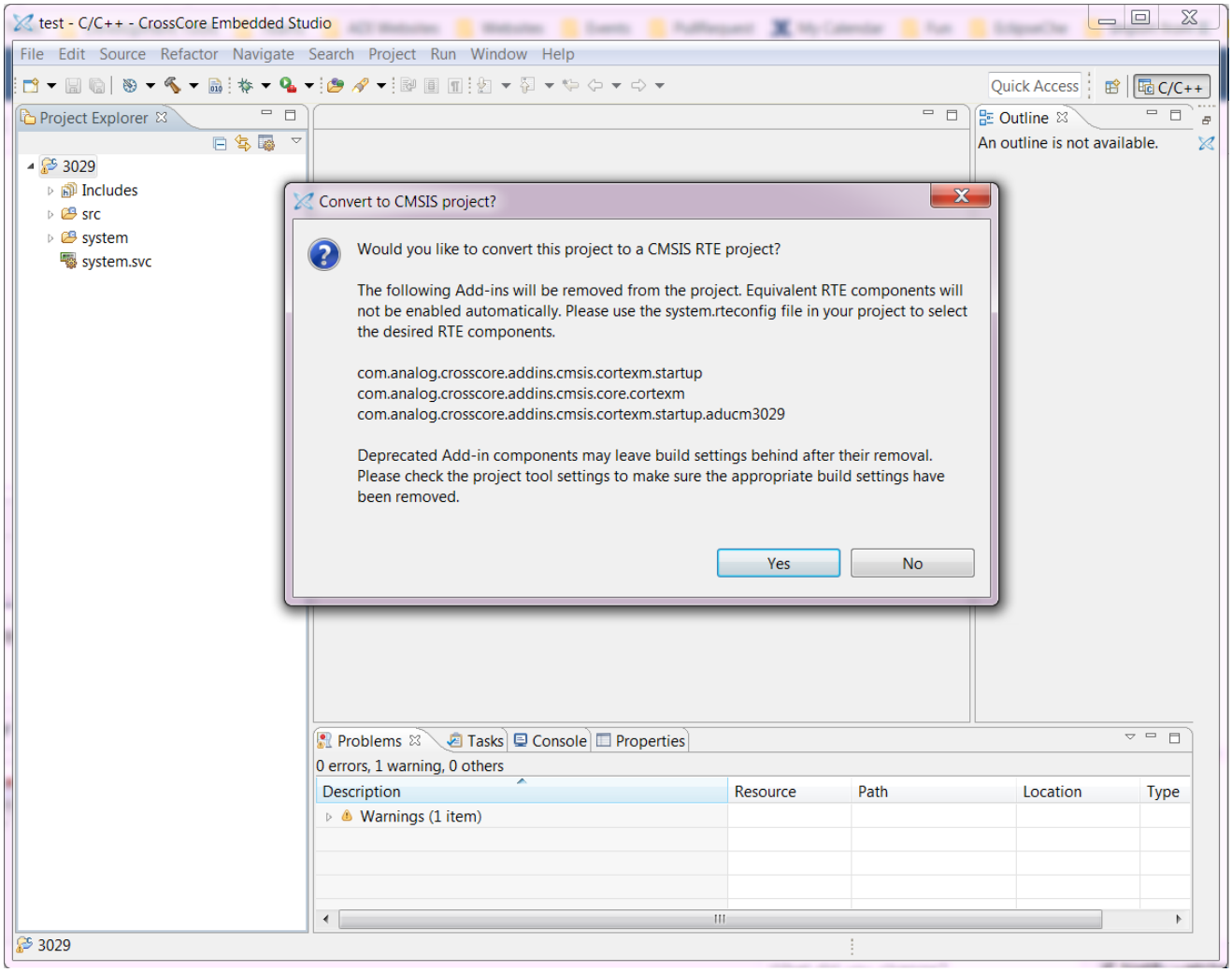
Note: If leave "-T" empty, after refreshing the project, the default .LD script will be added back by the Startup CMSIS component.

3.20 Configure a project to use CMSIS components (CCES-16652)

If you has created an project for ADuCM302x processor in CCES 2.4.0, you can now configure it to use CMSIS components by right-clicking the project and selecting `Configure > Convert to CMSIS project.`



There will be a popup to warn you that some deprecated Add-ins will be removed, equivalent RTE components will not be enabled automatically so you need select them in the `system.rteconfig` file, refer to ADuCM302x Device Family Pack User's Guide regarding adding RTE configurations. You also need to check the project tool settings to make sure appropriate build settings.


test - C/C++ - CrossCore Embedded Studio
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
C/C++
Project Explorer
3029
Includes
src
system
system.svc
Outline
An outline is not available.
Convert to CMSIS project?
Would you like to convert this project to a CMSIS RTE project?
The following Add-ins will be removed from the project. Equivalent RTE components will not be enabled automatically. Please use the system.rteconfig file in your project to select the desired RTE components.
com.analog.crosscore.addins.cmsis.cortexm.startup
com.analog.crosscore.addins.cmsis.core.cortexm
com.analog.crosscore.addins.cmsis.cortexm.startup.aducm3029
Deprecated Add-in components may leave build settings behind after their removal. Please check the project tool settings to make sure the appropriate build settings have been removed.
Yes
No
Problems
Tasks
Console
Properties
0 errors, 1 warning, 0 others
Description
Resource
Path
Location
Type
Warnings (1 item)
3029

4 Changes That Might Impact Backwards Compatibility

4.1 Constellation Add-in products support in CCES 2.6.0 (Eclipse Neon) (CCES-15053)

CrossCore Embedded Studio 2.6.0 requires that you use the latest Micrium Add-in products at version 2.6.0. For more information on how to obtain the latest Add-in product versions, visit the µC/OS-II, µC/OS-III, µC/FS, lwIP, µC/USB Device or µC/USB Host Add-in web pages.

4.2 Linked objects in input-file.txt are not sorted when the folder has number in its name (CCES-16712)

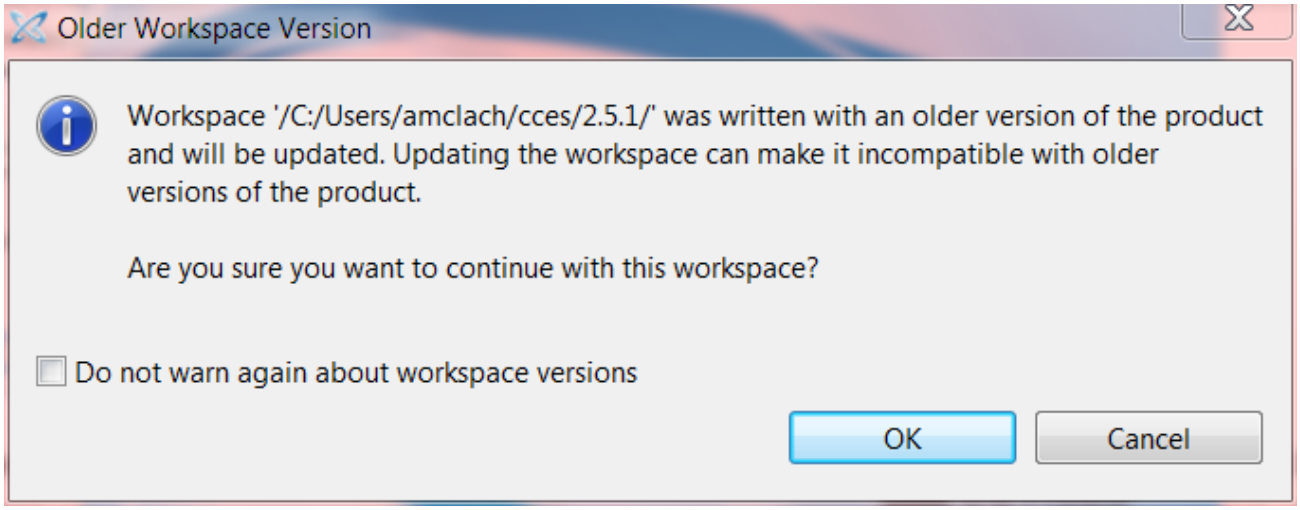
CCES links an application by placing the object file names into an input file. Since CCES 2.5.1, the file is ordered alphabetically. However, if the folder has a number in its name (e. g. ADuCM3029), then the object file(s) are not sorted. Starting with this release, files in folders with a number will also be sorted alphabetically.

You may have a project that happens to work, because of the order in which the objects are placed in the input data file, and you may find that your project does not build successfully in later releases of CCES. In such a case, you should review your project set-up to see if there was a problem that already existed but did not show up until the order of the objects on the linker command line was changed.

4.3 Your CCES workspace will be upgraded

Due to the move from Eclipse 4.4 to 4.6 it may not be possible to use the same workspace with CCES 2.5.1 and earlier versions of CCES. When you point CCES 2.6.0 to a workspace created by CCES 2.5.1 or earlier, you will see the following message.

We recommend keeping separate workspaces for multiple versions of CCES.



4.4 If project artifact type is set to Loader File and artifact name is not ${ProjName}, then the executable will not be populated in a launch configuration (CCES-15590)

This fix will cause a build failure for some ADSP-SCxxx example projects when their artifact type is set to Loader File and their artifact name is not set to ${ProjName}. If you encounter a build failure, visit the tools settings and navigate to the CrossCore SHARC Loader/Executable Files configuration. The Executable files need to be updated to the correct file name, which is identical to the LDR file name that is specified in the Artifact Name entry box.

4.5 L1DataBCacheEnabledWhenUsedForData Fatal Error before main() using ADSP-BF531

An application run on ADSP-BF531 built using a definition of the Blackfin CPLB control variable `__cplb_ctrl` to enable DATAB caching will result in a fatal error before main() when using CCES 2.6.0. The error seen in the CCES console will look like the following one:

```
A non-recoverable error or exception has occurred.
  Description:   L1 data B cache is enabled when it is used for data
  General Type:  RunTimeError
  Specific Type: L1DataBCacheEnabledWhenUsedForData
  Error PC:      0xffa1015c
```

The reason for the error is that ADSP-BF531 parts do not have L1 data B memory so DATAB caching cannot be enabled. Previous versions of CCES did not detect this error when they should have done. Fix the problem by not using `CPLB_ENABLE_DCACHE2` in the definition of `__cplb_ctrl` in applications run on ADSP-BF531 parts.

4.6 Blackfin and SHARC compiler changes for C++ 11 implicit exception specification of destructor functions

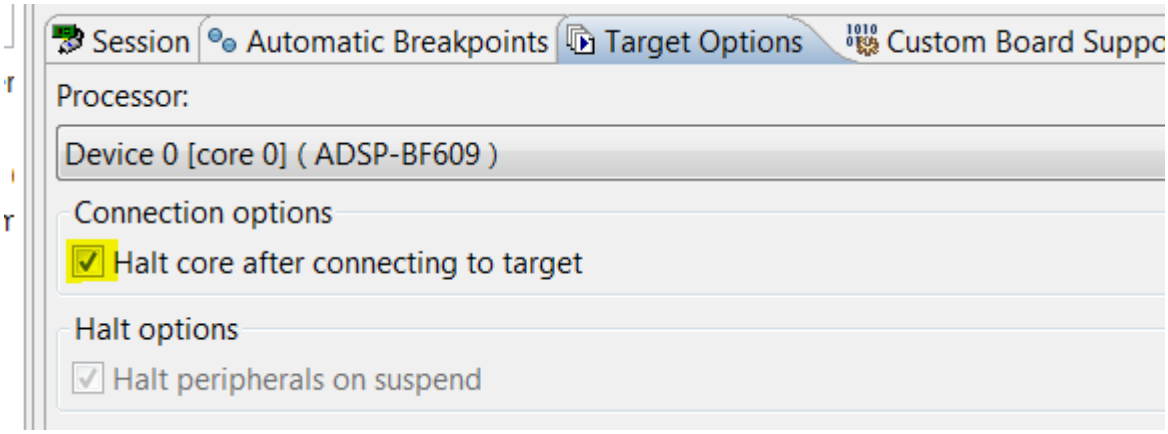
The Blackfin and SHARC C++ compilers have had improvements to reduce the size of applications using C++11 exceptions handling. Part of these changes mean the compiler now more strictly adheres to the C++11 standard regarding use of a default exception-specification of `noexcept(true)` for destructor functions that don't call functions or only call functions that don't allow exceptions to be thrown. Destructor functions that may throw exceptions that do not have an exception-specification in their declaration should therefore be updated to include an exceptions specification. For example:

```
class c {
  public:
    // --snip--
    ~c()
#if __cplusplus >= 201103L
      noexcept(false)  // Exception-specification allowing
exceptions required for destructors that throw for C++ 11.
#endif
    {
      if (err)
        throw -1;
    }
  private:
    bool err;
    // --snip--
};

c c_inst;
```

4.7 BF609 core(s) are told to halt incorrectly when using a multi-processor set-up (CCES-16625)

Prior to CrossCore Embedded Studio 2.6.0, a launch configuration for ADSP-BF609 was not able to halt core 0 correctly. If you create a new launch configuration for ADSP-BF609 with CCES 2.6.0 and you only supply a program to load on core 1, then it will now fail to load unless you supply a program to load on core 0. Alternatively you can visit the Target Options tab and check the *Halt core after connecting to target* option.



4.8 Need to update Custom Linker Script (-T) option after converting CCES 2.4.0 project to use CMSIS components (CCES-17358)

After converting the project and adding Startup component to the project, you need to manually point the "-T" to the LD script in your project that is copied by the Startup component from Tool Settings (the General section of Linker option). Take ADuCM3029 for example, pointing `Custom linker script (-T)` to `${workspace_loc:/${ProjName}}/RTE/Device/ADuCM3029/ADuCM3029.ld`.

5 Known Issues

5.1 For more information

For the latest anomalies please consult our Software and Tools Anomalies Search page.