



Release Notes for CrossCore Embedded Studio 2.11.0

1 Table of Contents

1	Table of Contents.....	2
2	Introduction	3
3	New and noteworthy	4
3.1	ADSP-SC598 Family Processors Support.....	4
3.1.1	Evaluation Boards.....	4
3.1.2	CrossCore Embedded Studio	4
3.1.3	Using CrossCore Launch Group for ADSP-SC598 Family Applications (Heterogeneous Debug)	5
3.1.4	Support for Peripheral Register View When Debugging Cortex-A55	5
3.1.5	System Services and Device Drivers.....	5
3.2	Updated make version.....	6
4	Changes That Might Impact Backwards Compatibility.....	7
4.1	Ubuntu 16.04 Support is Removed	7
4.2	C++ applications on Cortex-A55 core must use cached memory	7
4.3	arm-none-eabi-ldr no longer available.....	7
5	Known Issues.....	8
5.1	System Services and Device Drivers Known Issues	8
5.2	Spurious "Prefer GDB Command" Warning.....	8
5.3	Register Read Warning When in Wrong Exception Level	8
5.4	OpenOCD does not support some of the Memory Browser functionality.....	8
5.5	OpenOCD (using ICE-1000/2000) does not support ADuCM3029 EZ-Kit Debugging.....	8
5.6	Peripherals View is not yet working using the Launch Group Debugging.....	9
5.7	adi_fatal_error Details are not Output to the Console using GDB and OpenOCD.....	9
5.8	ADSP-SC598 MCAP1 support between Cortex-A55 and SHARC Cores is not Working	9

2 Introduction

This document describes the changes for [CrossCore Embedded Studio \(CCES\) 2.11.0](#). You can find the release notes for older releases in the docs sub-directory of your CCES installation as well as an Installation Guide which will help you install this release.

For information on Linux please refer to the general Linux documentation [Linux for ADSP-SC5xx Processors \[Analog Devices Wiki\]](#). Linux information for ADSP-SC59x will be provided in a separate document once it is released. Other useful links for CrossCore and Linux Development are:

- <http://www.analog.com/cces-quickstart>
- [EngineerZone > Processors and DSP > Software and Development Tools > CrossCore Embedded Studio and Add-ins](#)
- [EngineerZone > Processors and DSP > Software and Development Tools > CrossCore Embedded Studio and Add-ins > tags > CCES](#)
- [EngineerZone > Processors and DSP > Software and Development Tools > CrossCore Embedded Studio and Add-ins > tags > CCES 2.11.0](#)
- [How to debug SHARC cores in CCES while running Linux](#)
- [Configuring System Memory for the ADSP-SC5xx When Using Linux and SHARC Applications](#)

3 New and noteworthy

3.1 ADSP-SC598 Family Processors Support

CCES 2.11.0 provides support for a new family of SHARC+ Dual-Core DSP with Arm Cortex-A55 parts. These newly supported parts are:

Non-Automotive Models	Automotive models
ADSP-SC595 ADSP-SC596 ADSP-SC598	ADSP-SC595W ADSP-SC596W ADSP-SC598W

More information about the new parts can be found at <https://www.analog.com/en/products/ADSP-SC598.html>.


3.1.1 Evaluation Boards


Two evaluation boards are provided for support of the new parts.


- EV-SC598-SOM (System-on-Module)
- EV-SOMCRR-EZKIT (Carrier)

The examples BSP for these is version 2.0.0 or newer and is available to download from <https://www.analog.com/EV-SC598-SOM>.

3.1.2 CrossCore Embedded Studio

 The ARM Cortex-A55 core 0 uses a different toolchain to that used by Cortex-A5. When porting a Cortex-A5 project to Cortex-A55, a new Cortex-A55 project should be created and source files and project options migrated across to the new project.

 The ARM Cortex-A55 core 0 is a 64-bit processor so pointers, int and long types are all 64-bits wide unlike the SHARC+ core where these are 32-bit types.

 Debugging on the Cortex-A55 must be done using GDB and OpenOCD (Emulator) as the CrossCore debugger does not support the Cortex-A55.

CrossCore Embedded Studio includes usual support for the new parts, such as:

- CCES project build and debug capabilities.

- Initcodes, SHARC preloads, and OpenOCD board and processor configuration scripts.
- System services (SSL) and device drivers (DD) 3.0
- Functional and cycle-accurate simulators.

3.1.3 Using CrossCore Launch Group for ADSP-SC598 Family Applications (Heterogeneous Debug)

CrossCore Launch Groups have been added to allow you to launch OpenOCD and CrossCore Debugger sessions for the ADSP-SC598 family parts. See the following help topic for more information:

- CrossCore® Embedded Studio 2.11.0 > Integrated Development Environment > Debugging Targets > Debugging ADSP-SC5xx SHARC+ and ARM Projects > Using CrossCore Launch Group for ADSP-SC598 Family Applications (Heterogeneous Debug)

3.1.4 Support for Peripheral Register View When Debugging Cortex-A55

Peripheral Registers on the ADSP-SC598 family parts Cortex-A55 core using OpenOCD debug sessions can be viewed using the Peripherals view (Window → Show View → Peripherals for new CCES workspaces, Window → Show View → Other → Debug/Peripherals for old CCES workspaces). The Peripherals view is supported in standalone OpenOCD debug sessions only - it is not currently supported in CrossCore Launch Groups which are used in heterogeneous debugging.

3.1.5 System Services and Device Drivers

The following Device Drivers and System Services modules are supported in this release for ADSP-SC598 family of parts.

Device drivers	System Services
<ul style="list-style-type: none"> • ASRC • CAN-FD (automotive parts only) • CRC • Crypto (PKA, PKTE ,TRNG) • EMSI • EMAC • FIR • HADC • EPPI • IIR • Link Port • MLB (automotive parts only) • OSPI • Rotary Counter • SPDIF Rx • SPDIF Tx • SPI • SPORT • TMU • TWI • UART 	<ul style="list-style-type: none"> • ARM-Generic Timer • DAI • DMCPM • DMA (MDMA, EMDMA) • GPIO • L2CTL • MEPU • OTP • PCG • PDM • PMU • PWR • RCU • SMPU • SPU • STDIO • SWU • TMR • TRU • WD

For more information regarding device drivers and system services see the following CCES help topics:

- [CrossCore® Embedded Studio 2.11.0 > System Run-Time Documentation > Device Drivers User Guide](#)
- [CrossCore® Embedded Studio 2.11.0 > System Run-Time Documentation > System Services and Device Drivers](#)

3.2 Updated make version

The version of GNU make supplied in CCES has been updated from 3.811 to 4.3.

4 Changes That Might Impact Backwards Compatibility

4.1 Ubuntu 16.04 Support is Removed

CCES on Linux now only supports Ubuntu 20.04 with 32-bit compatibility libraries installed. Support for Linux Ubuntu 16.04 has been removed in CCES 2.11.0. See the Installation Guide for more details.

4.2 C++ applications on Cortex-A55 core must use cached memory

On the Cortex-A55 core, when using some C++ language features with runtime support, such as exceptions, an application will contain compiler-generated data that is used when an event is taken. This data must be located in cached memory, else the application will not be able to perform atomic accesses to the data and execution will fail. By default, the linker `ld` scripts are set up to place all data in cached memory. We recommend only moving user data to uncached memory (by using `section` attributes in your code), and leaving all default sections mapped to their original locations. This means that you will not inadvertently move such compiler-generated data tables to uncached memory.

4.3 `arm-none-eabi-ldr` no longer available

The `arm-none-eabi-ldr` utility is no longer available in CCES. To create `ldr` files, you should use the `elfloader` utility. This is also now available from the command line when using CCES on Linux.

5 Known Issues

5.1 System Services and Device Drivers Known Issues

- For EMSI driver, predefined non-blocking transfer and blocking transfer API supports (≈32MB) of data transfer in a single API call. For Non-Blocking open-ended transfers, 64MB of data transfer in a single API call is supported. EMSI `adi_ems_i_SetPriority()` API is setting eMMC QoS Priority to 15 which should have been set as 12.
- The PVP driver does not work for ADSP-BF609 family of parts. Debug configuration PVP driver will trigger an assert at runtime and in Release configuration memory corruption will occur.

5.2 Spurious "Prefer GDB Command" Warning

When connecting to a GDB with OpenOCD debug session in CCES you'll see a warning in the OpenOCD output:

```
Warn: Prefer GDB command "target extended-remote 3333" instead of "target remote 3333"
```

This warning does not affect the actual execution, and can be ignored.

5.3 Register Read Warning When in Wrong Exception Level

When reading a register in the wrong exception level for Cortex-A55, a warning is printed out but the read is not performed.

5.4 OpenOCD does not support some of the Memory Browser functionality

When using the memory browser on OpenOCD the functions to refresh view, fill memory, dump memory and the go to symbol are unavailable and greyed out.

5.5 OpenOCD (using ICE-1000/2000) does not support ADuCM3029 EZ-Kit Debugging

Getting packs update using CMSIS Pack Manager has no problem but debugging a project or example from CMSIS Pack Manager with ICE-1000 or ICE-2000 emulators for ADuCM3029 EZ-Kit will generate the following error message :

```
Error: Can't change session's transport after the initial selection was made
C:/Analog Devices/CrossCore Embedded Studio 2.11.0/ARM/packs/AnalogDevices/ADuCM302x_DFP/3.2.0/openocd/scripts//target/aducm302x.tcl:16: Error:
in procedure 'script'
at file "embedded:startup.tcl", line 26
at file "C:/Analog Devices/CrossCore Embedded Studio 2.11.0/ARM/packs/AnalogDevices/ADuCM302x_DFP/3.2.0/openocd/scripts/target/aducm3029.cfg", line 6
at file "C:/Analog Devices/CrossCore Embedded Studio 2.11.0/ARM/packs/AnalogDevices/ADuCM302x_DFP/3.2.0/openocd/scripts//target/aducm302x.tcl", line 16
```


5.6 Peripherals View is not yet working using the Launch Group Debugging

Registers are not displayed in the peripheral view when using OpenOCD through Launch Group debugging.

5.7 adi_fatal_error Details are not Output to the Console using GDB and OpenOCD

The CrossCore debugger used for Sharc and Blackfin cores has a feature to output useful fatal error details but this is not happening using GDB and OpenOCD for Cortex-A5 and Cortex-A55.

5.8 ADSP-SC598 MCAPI support between Cortex-A55 and SHARC Cores is not Working

Multicore Communication API (MCAPI) support between SHARC+ cores 1 and 2 is working well using the CrossCore Debugger but is not fully functional between the Cortex-A55 and SHARC core using the CrossCore Launch Debug configuration.