

# **μCUSB Host™ Stack for CrossCore® Embedded Studio Release 1.0.0 Release Notes**

## **What is μC/USB Host™ Stack for CrossCore® Embedded Studio**

The release of μC/USB Host™ Stack for CrossCore® Embedded Studio is the result of a partnership between Analog Devices and Micrium to provide a user-friendly programming environment for embedded applications that require USB Host functionality. To use the stack in an embedded application requires a minimum of the μC /USB Host™ Core for CrossCore® Embedded Studio product, along with one or more supported Host Class products described below.

The μC/USB Host Stack is provided with a hardware abstraction layer which is modified to support Blackfin processors with built-in USB Host controllers such as the ADSP-BF526, ADSP-BF527, ADSP-BF548 and ADSP-BF60x. Class drivers for the Host Mass Storage Class (MSC) and Human Interface Device (HID) class are offered.

The μC/USB-Host™ Stack uses a modular architecture with three software layers between the application and the hardware:

- The Host Class layer provides functionality to the host using one or more class drivers. Each class driver is responsible for class-specific requests and may provide an API for controlling some implementation features and for receiving/transmitting data.
- The Host Core layer controls data reception and transmission, and is responsible for hub requests (device connection, enumeration).
- The Host Controller layer interfaces with the USB Controller driver to process interrupts, notify the Host Core layer of bus events, and receive/transmit data.

## **Class Support**

### **μC/USB Host™ Class MSC for CrossCore® Embedded Studio**

The Mass Storage Class (MSC) enables an embedded target host to access files from a USB Flash Drive or similar device.

### **μC/USB Host™ Class HID for CrossCore® Embedded Studio**

The Human Interface Device Class (HID) enables an embedded target host as to communicate with both standard (e.g. keyboards) and vendor-specific HID devices.

# Getting Started with $\mu$ C/USB Host™ Stack for CrossCore® Embedded Studio

## Installation

CrossCore® Embedded Studio 1.0.3 or newer must be installed prior to installing any of the  $\mu$ C/USB Host™ Stack for CrossCore® Embedded Studio products. In addition,  $\mu$ C/USB Host™ Stack operation requires the support of an RTOS.  $\mu$ C/OS-III™ Real-Time Kernel for CrossCore® Embedded Studio is a separate product that may be purchased and installed in support of  $\mu$ C/USB Host Stack. All of the examples that are provided in the various  $\mu$ C/USB Host Stack products require  $\mu$ C/OS-III support.

In addition,  $\mu$ C/USB Host™ Class MSC for CrossCore® Embedded Studio requires the support of a file system.  $\mu$ C/FS™ File System for CrossCore® Embedded Studio is a separate product that may be purchased and installed in support of  $\mu$ C/USB Host Class MSC. See "Software Requirements" below for the compatible versions of  $\mu$ C/FS required.

As previously outlined, there are three software layers involved and each is delivered in the products as outlined below:

- The USB Host Controller driver is delivered with CrossCore® Embedded Studio 1.0.3 (CCES) or later.
- The Host Core layer product is licensed as a stand alone product and must be installed before any of the Class layer products can be used. It is recommended, though not required, that the Host Core product be installed before installing any of the Class layer products.
- The Class layer products are licensed and installed as individual products.

The following table summarizes the layers and associated products:

<b>Software Layer</b>	<b>Product</b>	<b>Notes</b>
Layer 1	USB Controller Driver	Provided with CCES 1.0.3 or later. No additional license required.
Layer 2	USB Host Core	Separate license required.
Layer 3	<b>USB Host Class Drivers</b>	
	$\mu$ C/USB Host™ Class MSC	Separate license required.
	$\mu$ C/USB Host™ Class HID	Separate license required.

Please make sure to close CrossCore Embedded Studio before proceeding with the installation. If CCES is left open during the installation, it will have to be restarted after installing the  $\mu$ C/USB Host Stack products in order for the changes to take effect, and for  $\mu$ C/USB Host to be available. All of the  $\mu$ C/USB Host Stack,  $\mu$ C/FS File System and uC/OS-III products install the following common products:

- **$\mu$ C/LIB** . This software is always installed into Common Program Files directory. This location is determined by the %CommonProgramFiles(x86)% environment variable in 64-bit operating systems or by %CommonProgramFiles% in 32-bit operating systems.
- **$\mu$ C/CPU**. This software is always installed into Common Program Files directory. This location is determined by the %CommonProgramFiles(x86)% environment variable in 64-bit operating systems or by %CommonProgramFiles% in 32-bit operating systems.

The default location for the installation of the  $\mu$ C/USB Host Add-in products is under C:\Analog Devices, e.g. C:\Analog Devices\ uCUSB\_Host\_Core-Rel1.0.0. Should you wish to use a different location Analog Devices strongly recommends installing the  $\mu$ C/USB Host Stack products outside of the Program Files directory to prevent possible permission issues related to UAC (User Access Control). If you have already installed the product under Program Files then we recommend that you uninstall it and re-install it in a different location.

Note: Multiple versions of the  $\mu$ C/USB Host Stack can be installed on the same system. Only a single instance of a specific version of the product can be installed on a system.

## License Checking

The installation process checks for a separate license for each of the  $\mu$ C/USB Host Stack products. If a valid license is not detected, the installer will start the Manage Licenses utility for entering and activating a license. The installer will fail in a non-interactive mode when valid license is not present.

## Installation Logging

The installer does not create a log file by default. If you encounter installation issues, you can generate an installation log file by running the installer from the command prompt.

Change to the directory containing downloaded installer executable and run the following from the command prompt to install the Host Core layer product:

```
ADI_uCUSB_Host_Core-Rel1.0.0.exe /v"/!*"v c:\temp\installer.log"
```

Similarly, the Class layer products may also be installed from the command line as follows

<b>Class Layer Product</b>	<b>Command Line Executable Name</b>
<b><math>\mu</math>C/USB Host™ Class</b>	ADI_uCUSBH_Class_MSC-Rel1.0.0.exe /v"/!*"v

<b>MSC</b>	c:\temp\installer.log
<b>µC/USB Host™ Class</b>	ADI_uCUSBH_Class_HID-Rel1.0.0.exe /v"/l*v
<b>HID</b>	c:\temp\installer.log

## License

The installation process checks for a valid license for each of the µC/USB Host™ Stack products as listed below. Refer to the Licensing Guide in your CCES installation which can also be found in <http://www.analog.com/CrossCoreLicensingGuide>.

### **µC/USB Host™ Stack for CrossCore® Embedded Studio Products**

µC/USB Host™ Core for CrossCore® Embedded Studio Products

µC/USB Host™ Class MSC for CrossCore® Embedded Studio Products

µC/USB Host™ Class HID for CrossCore® Embedded Studio Products

## Support and Assistance

There are several options for contacting support:

- Submit your questions online at: <http://www.analog.com/support>
- E-mail your Processor and DSP software and development tools questions from within CrossCore Embedded Studio.

Go to "Help->E-mail Support...". This will create a new e-mail addressed to [processor.tools.support@analog.com](mailto:processor.tools.support@analog.com), and will automatically attach your CrossCore Embedded Studio version information (ProductInfo.html).

- E-mail your Processors and DSP applications and processor questions to:
  - [processor.support@analog.com](mailto:processor.support@analog.com)

## Supported Processors

Support is provided, in all of the µC/USB Host™ products, for ADSP-BF52x, ADSP-BF54x, ADSP-BF60x.

## Software Requirements

### Tools

µC/USB Host™ Stack for CrossCore® Embedded Studio 1.0.0 requires the installation of CrossCore® Embedded Studio 1.0.3 or later.

## Software Products

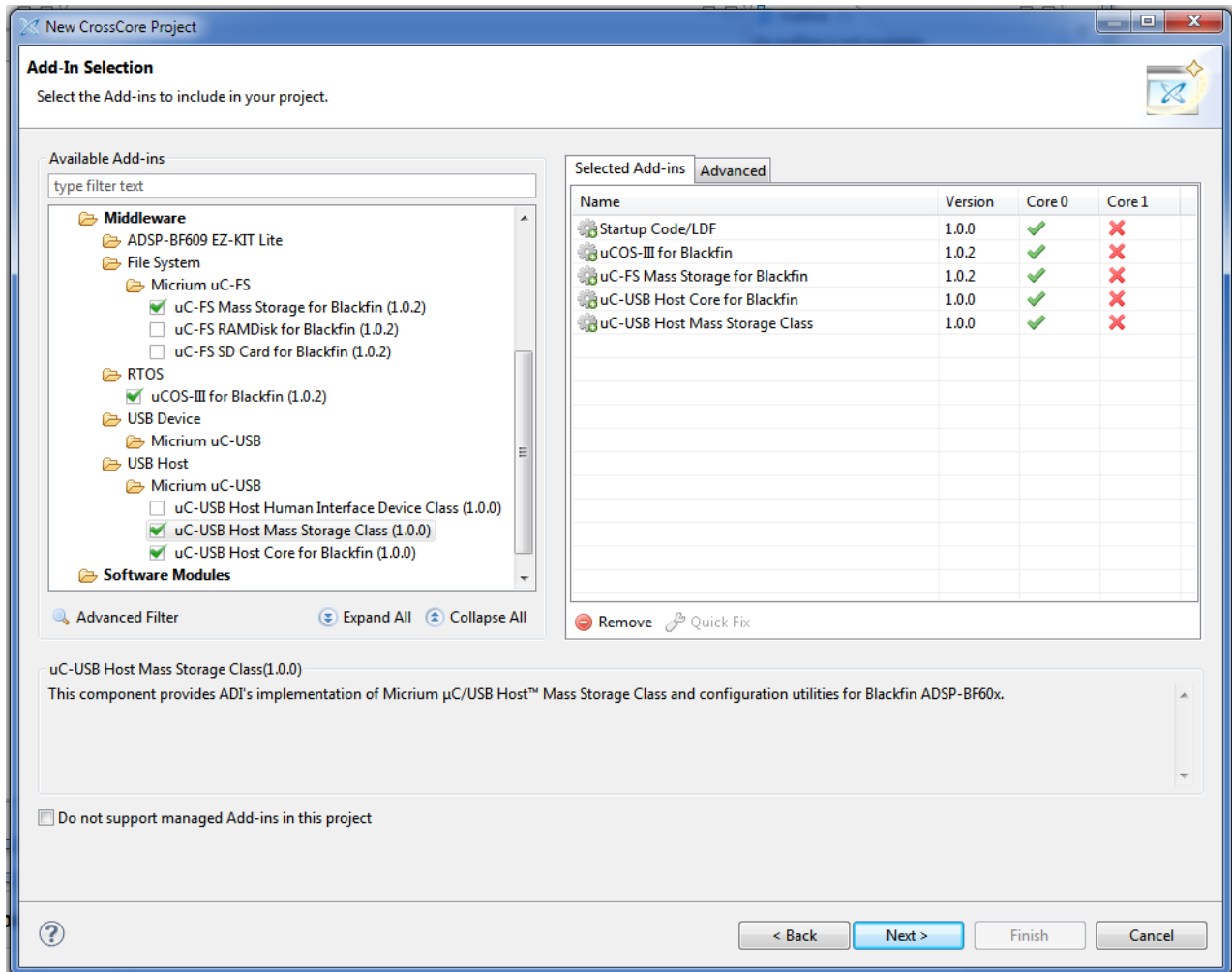
$\mu$ C/USB Host™ Stack for CrossCore® Embedded Studio requires installation of  $\mu$ C/OS-III™ Real Time Kernel for CrossCore® Embedded Studio 1.0.2.

$\mu$ C/USB Host™ Class MSC for CrossCore® Embedded Studio requires installation of  $\mu$ C/FS™ .File System for CrossCore® Embedded Studio 1.0.2.

## Getting started with a project that uses uC/USB Host Stack

### Adding uC/USB Host Stack to a project

Every CrossCore Embedded Studio project contains a System Configuration file called system.svc which is located in the root of the project. The file is the IDE's interface for managing the various pre-written software components used in the "system" implemented by a project. Double-clicking any system.svc file in a navigation view opens that file in the System Configuration Utility which allows you to see the add-ins that you currently have in your project. Clicking on Add and selecting one of the listed Add-ins from the Middleware section under the USB Host category adds the selected product to your project. The following screenshot shows the equivalent dialog for the creation of a new USB Host MSC CrossCore project. Please note that you only need to add the components to a single core.



Please note that  **$\mu$ C-USB Host Core for Blackfin 1.0.0** is required for all other products (as previously discussed in this release note). Therefore, when you add in any one of the Class layer products, the  $\mu$ C/USB Host Core product will be automatically added in also.

If an RTOS has not been added in, when you select "Next" in the Add-In dialog, you will be presented with a warning screen indicating that an RTOS product does not yet exist in your application. You will not be able to proceed unless you also select **uCOS-III for Blackfin 1.0.2** as an additional Add-In.

Finally, a  $\mu$ C/USB Host project requires the use of external memory, so the **Startup Code/LDF** Add-in is required and the appropriate settings made to enable external memory:

## Startup Code/LDF

### Code Generation Options

These options control the generation of the Startup Code and LDF

Startup Code  
Interrupt  
LDF

### LDF Configuration

#### Stack and Heaps

Name	Heap Id	Memory
System stack	N/A	L1 internal memory
System heap	0	L1 internal memory

Add... Edit... Remove...

#### External Memory

Use external memory (SDRAM)

Size of external memory (in MegaBytes):

128

Partition external memory:

default

The  $\mu$ C/USB product Add-ins generate code for initializing the  $\mu$ C/USB Host Stack. To ensure timely initialization, when system components are configured the IDE adds any required code to a global C function named `adi_initComponents()` in `system/adi_initialize.c`. A call to this function will be added to the `main()` function when the  $\mu$ C/USB Host and RTOS components are added.

### Notes:

- Please refer to the  $\mu$ C/OS-III Release Notes for RTOS related information.
- Please refer to the  $\mu$ C/FS Release Notes for File System related information.

### Configuration

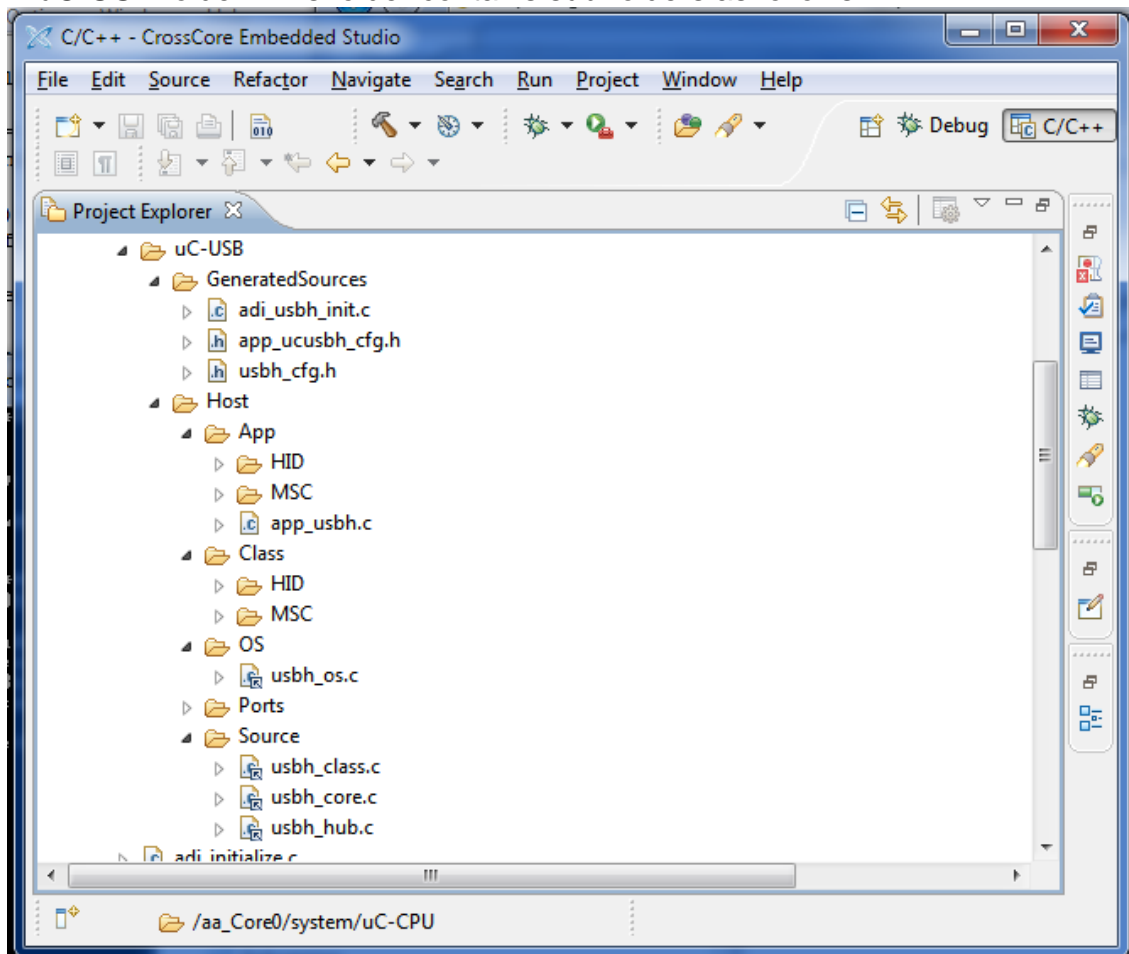
$\mu$ C/USB Host Stack application developers traditionally configure applications by creating header files which contain a long list of macro definitions.  $\mu$ C/USB Host Stack for CrossCore Embedded Studio provides a more intuitive configuration mechanism by

providing a tab in the System Configuration utility, which can be accessed by double-clicking the `system.svc` file and selecting the  $\mu$ C/USB Host tab. Filling in all the required fields in the configuration tab generates the appropriate files, `app_ucusbh_cfg.h` and `usbh_cfg.h` located within the project under `system/uc-USB/GeneratedSources`.

## $\mu$ C/USB Host Stack project structure

When adding  $\mu$ C/USB Host Stack to a CrossCore Embedded Studio project all the  $\mu$ C/USB Host Stack specific files get placed in the system folder. Please do not change this organization. In the system folder the following structure gets created

- A uC-USB folder. This folder contains sub-folders as follows



- A uC-CPU folder. This folder contains any sources and header files which are required by Micrium  $\mu$ C/CPU software.  $\mu$ C/CPU provides a processor-independent interface to the supported processors and toolchains that is used in all Micrium products.
- A uC-LIB folder. This folder contains any sources and header files which are required by Micrium  $\mu$ C/LIB software.  $\mu$ C/LIB provides a clean and organized



implementation of some of the most common standard library functions, macros and constants.  $\mu$ C/LIB is required by many Micrium products including  $\mu$ C/USB.

- A uC-Common folder. This folder contains sources and headers which are common to several Micrium products but that are not part of any Micrium product themselves. These include `app_cfg.h` which is needed by all Micrium applications.

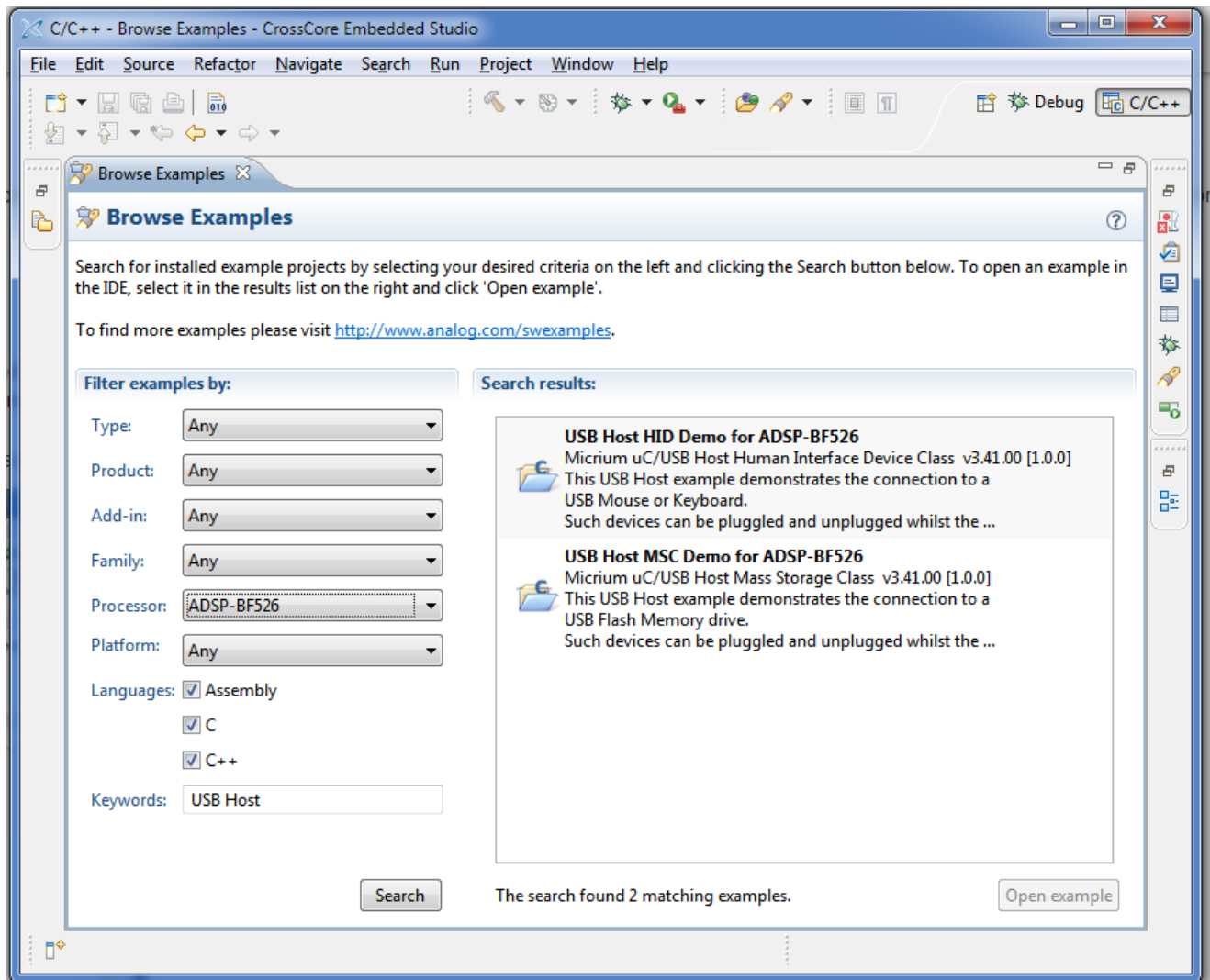
## Examples

There are examples for both  $\mu$ C/USB Host™ Class MSC and  $\mu$ C/USB Host™ Class HID products, available for ADSP-BF526, ADSP-BF527, ADSP-BF548 and ADSP-BF609 development boards. The example can be built for both Debug and Release configurations.

### Location

In order to locate the  $\mu$ C/USB Host Stack examples, you can do the following:

- Open CrossCore Embedded Studio's (CCES) Example Browser, which can be found in CrossCore Embedded Studio under the Help menu. You may then perform one of the following steps:
  - In the Product Pull-Down select the USB Product that you have licensed and installed
  - In the Keyword textbox insert the keyword "USB"
- The result of either of these filters will be a list of USB Host examples in the Search results panel. The results of browsing with the "USB Host" keywords for the ADSP-BF526 processor are shown below



After locating an example of interest, double-clicking on the project in the search results pane will result in the example being copied to the current workspace and imported into the CCES Project Explorer.

## MISRA-C Support

MISRA C is a software development standard for the C programming language developed by the Motor Industry Software Reliability Association (MISRA). Its aims are to facilitate code safety, portability, and reliability in the context of embedded systems, specifically those systems programmed in ANSI C. The compiler detects violations of the MISRA rules at compile-time, link-time, and run-time.

As of this release a list of rules that  $\mu$ C/USB Host Stack breaks is not available. The USB Controller driver, provided by Analog Devices, suppresses all MISRA rules.

# **μC/USB Host™ Stack for CrossCore® Embedded Studio RTOS Requirements**

μC/USB Host Stack for CrossCore Embedded Studio requires the presence of an RTOS, although not necessarily the μC/OS-III Real-Time Kernel for CrossCore Embedded Studio product. When running in a μC/OS-III application, μC/USB Host Stack requires multiple μC/OS-III objects like semaphores and task-specific registers slots.

Removing any of the μC/OS-III functionality that is required by a μC/USB Host application could cause link errors.

Note that adding μC/USB Host to a project which already has μC/OS-III may require changes to some RTOS settings. Please see the MSC example configuration for some recommended settings.

## **Common Micrium Components**

There are several CrossCore® Embedded Studio add-ins based on Micrium's products which share common components. To ensure that the same version of these components is used by all the add-ins that require them, these components are installed in a common location which is distinct from the add-in install folders. These common components are

- μC/CPU which is installed in %COMMONPROGRAMFILES%\Analog Devices\uC-CPU 1.0.3. This installation includes μC/CPU 1.29.02.
- μC/LIB which is installed in %COMMONPROGRAMFILES%\Analog Devices\uC-LIB 1.0.3. This installation includes μC/LIB 1.37.01.

The documentation for these components can be found in CrossCore® Embedded Studio Help under Micrium μC/OS-III™ 1.0.2 > μC/OS-III for CCES Configuration > μC/OS-III Tab > Components Shared by Add-ins.

## **Known issues with μC/USB Host™ Stack for CrossCore® Embedded Studio**

These are the currently known problems which affect μC/USB Host™ Stack Class MSC for CrossCore® Embedded Studio.

- Processor Data cache is disabled in the examples; if enabled, the demo application may not work.

- With the device driver built into the CCES 1.0.3 device driver libraries, multi-packet DMA (DMA Mode 1) is used for transferring data on the ADSP-BF526 (si- rev 0.2) and ADSP-BF609 platforms; Single packet DMA (DMA Mode 0) is used for ADSP-BF527 and ADSP-BF548.
- While the USB host mode device driver for ADSP-BF609 supports hubs (see CCES 1.0.3 Release notes for further details) the host MSC application applies tests only to one connected USB MSC device.