

TI-RTOS 2.00 for C6000

Getting Started Guide



Literature Number: SPRUHU2
April 2014

Preface	3
1 About TI-RTOS	4
1.1 What is TI-RTOS?	4
1.2 What are the TI-RTOS Components?.....	5
1.3 How Can I Find Example Projects?	6
1.4 What Compilers and Targets are Supported?	7
1.5 For More Information	7
2 Installing TI-RTOS	9
2.1 System Requirements	9
2.2 Installing Code Composer Studio	10
2.3 Installing TI-RTOS in Code Composer Studio	10
2.4 Installing TI-RTOS as a Standalone Product	11
3 Examples for TI-RTOS	12
3.1 Creating Example Projects Using the TI Resource Explorer	13
4 Configuring TI-RTOS	16
4.1 Starting the Configuration Tool.....	17
4.2 Configuring TI-RTOS	18
4.3 Configuring Components of TI-RTOS.....	18
Index	19

Read This First

About This Manual

This manual describes TI-RTOS for C6000. The version number as of the publication of this manual is v2.00.

Notational Conventions

This document uses the following conventions:

- Program listings, program examples, and interactive displays are shown in a special typeface. Examples use a bold version of the special typeface for emphasis.

Here is a sample program listing:

```
#include <xdc/runtime/System.h>
int main(void)
{
    System_printf("Hello World!\n");
    return (0);
}
```

- Square brackets ([and]) identify an optional parameter. If you use an optional parameter, you specify the information within the brackets. Unless the square brackets are in a **bold** typeface, do not enter the brackets themselves.

Trademarks

Registered trademarks of Texas Instruments include Stellaris and StellarisWare. Trademarks of Texas Instruments include: the Texas Instruments logo, Texas Instruments, TI, TI.COM, C2000, C5000, C6000, Code Composer, Code Composer Studio, Concerto, controlSUITE, DSP/BIOS, MSP430, MSP430Ware, SPOX, Sitara, TI-RTOS, Tiva, TivaWare, TMS320, TMS320C5000, TMS320C6000, and TMS320C2000.

ARM is a registered trademark, and Cortex is a trademark of ARM Limited.

Windows is a registered trademark of Microsoft Corporation.

Linux is a registered trademark of Linus Torvalds.

IAR Systems and IAR Embedded Workbench are registered trademarks of IAR Systems AB:

All other brand or product names are trademarks or registered trademarks of their respective companies or organizations.

April 11, 2014

About TI-RTOS

This chapter provides an overview of TI-RTOS for C6000.

Topic	Page
1.1 What is TI-RTOS?	4
1.2 What are the TI-RTOS Components?	5
1.3 How Can I Find Example Projects?	6
1.4 What Compilers and Targets are Supported?	7
1.5 For More Information	7

1.1 What is TI-RTOS?

TI-RTOS is a scalable embedded tools ecosystem for TI devices. It scales from a real-time multitasking kernel (SYS/BIOS) to a complete RTOS solution including additional middleware components. By providing essential system software components that are pre-tested and pre-integrated, TI-RTOS enables you to focus on differentiating your application.



TI-RTOS is *not* installed automatically as part of the Code Composer Studio v6.0 installation. You can install TI-RTOS from the CCS App Center (choose **View > CCS App Center** in CCS). Choose the version of TI-RTOS for your device family. If you use devices in multiple families, you can install multiple TI-RTOS versions. See [Section 2.3](#) for details.

If you do not use CCS, you can [download and install TI-RTOS as a standalone product](#) (see [Section 2.4](#)).

TI-RTOS is provided with full source code and requires no up-front or runtime license fees.

1.2 What are the TI-RTOS Components?

TI-RTOS contains its own source files, pre-compiled libraries (both instrumented and non-instrumented), and examples. Additionally, TI-RTOS contains a number of components within its "products" subdirectory. The components of TI-RTOS for C6000 are as follows.

Table 1–1. TI-RTOS Components

TI-RTOS Component	Name	PDF Documentation Location
TI-RTOS Kernel	SYS/BIOS	<i>SYS/BIOS (TI-RTOS Kernel) User's Guide</i> -- SPRUJEX3
TI-RTOS Instrumentation	UIA	<i>System Analyzer User's Guide</i> -- SPRUH43
TI-RTOS Networking	NDK	<i>TI Network Developer's Kit (NDK) Guide</i> -- SPRU523 <i>TI Network Developer's Kit (NDK) API Reference</i> -- SPRU524
TI-RTOS Interprocessor Communication	IPC	<i>IPC User's Guide</i> on Texas Instruments Wiki
TI-RTOS File System	FatFS	<i>TI-RTOS User's Guide</i> -- SPRUHD4

The components in the "products" subdirectory are:

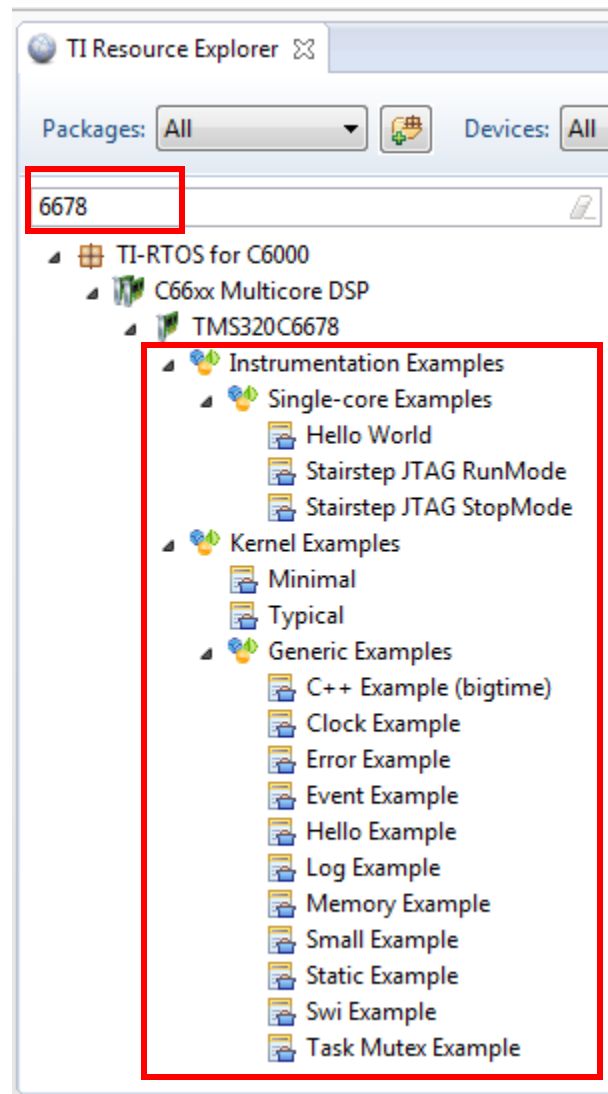
- **TI-RTOS Kernel — SYS/BIOS.** SYS/BIOS is a scalable real-time kernel. It is designed to be used by applications that require real-time scheduling and synchronization or real-time instrumentation. It provides preemptive multi-threading, hardware abstraction, real-time analysis, and configuration tools. SYS/BIOS is designed to minimize memory and CPU requirements on the target.
- **TI-RTOS Instrumentation — UIA.** The Unified Instrumentation Architecture (UIA) provides target content that aids in the creation and gathering of instrumentation data (for example, Log data).
- **TI-RTOS Interprocessor Communication — IPC.** IPC contains packages that are designed to allow communication between processors in a multi-processor environment and communication to peripherals. This communication includes message passing, streams, and linked lists. These work transparently in both uni-processor and multi-processor configurations.
- **TI-RTOS Networking — NDK.** The Network Developer's Kit (NDK) is a platform for development and demonstration of network enabled applications on TI embedded processors.
- **XDCtools.** This core component provides the underlying tooling for configuring and building TI-RTOS and its components. XDCtools is installed as part of CCSv6. If you install TI-RTOS outside CCS, a compatible version of XDCtools is installed automatically. XDCtools is installed in a directory at the same level as TI-RTOS, not in the "products" directory of the TI-RTOS installation.

1.3 How Can I Find Example Projects?

TI-RTOS and its components provide numerous examples that you can import using the **TI Resource Explorer** in Code Composer Studio (CCS). These examples use TI-RTOS and its components and have all the settings needed for your device. Expand the tree in the TI Resource Explorer to see the examples that are available for your device.

- **Instrumentation Examples** are UIA examples.
- **Kernel Examples** are the SYS/BIOS examples.

Follow the steps in [Section 3.1](#) to import, build, and run these examples.



1.4 What Compilers and Targets are Supported?

The following code generation tool (compilers and linkers) version is supported. The version listed is recommended because it was used to build the TI-RTOS libraries and to perform testing. More recent versions are expected to be compatible.

Texas Instruments: TMS320C6000 CodeGen Tools v7.4.7

The configuration uses a "target" specification during the build. This specification is sometimes called the "RTSC target." The targets supported are:

- ti.targets.elf.C66
- ti.targets.elf.C66_big_endian
- ti.targets.elf.C674

1.5 For More Information

To see release notes for a component, go to the subdirectory for that component within the TI-RTOS products directory. For example, `C:\ti\tirtos_c6000_2_###_###_###\products\bios_6_40_###_###` contains release notes for SYS/BIOS.

To see user guide PDFs and other documentation for a component, go to the "docs" subdirectory within the directory that contains the release notes.

To learn more about TI-RTOS and its components, refer to the following documentation:

- **TI-RTOS**
 - [TI-RTOS User's Guide \(SPRUHD4\)](#)
 - [TI-RTOS on the Texas Instruments Wiki](#)
 - [TI-RTOS forum on TI's E2E Community](#)
 - [TI-RTOS Porting Guide](#)
 - [Embedded Software Download Page](#)
- **Code Composer Studio (CCS)**
 - [CCS online help](#)
 - [CCSv6 on the Texas Instruments Wiki](#)
 - [Code Composer forum on TI's E2E Community](#)
- **SYS/BIOS**
 - [SYS/BIOS User's Guide \(SPRUEX3\)](#)
 - SYS/BIOS online reference (also called "CDOC"). Choose **Help > Help Contents** and expand the **TI-RTOS** item for your device family. Choose the **Config API Reference (cdoc)** item and see help under the ti.sysbios and xdc.runtime modules.
 - [SYS/BIOS on the Texas Instruments Wiki](#)
 - [TI-RTOS forum on TI's E2E Community](#)
 - [SYS/BIOS 6.x Product Folder](#)

- **XDCtools**
 - XDCtools online reference. Open from CCS help or run `<xdc_install>/docs/xdctools.chm`.
 - [RTSC-Pedia Wiki](#)
 - [TI-RTOS forum on TI's E2E Community](#)
- **IPC**
 - [IPC User's Guide \(SPRUG06\)](#)
 - IPC online API reference. Run `<tirtos_install>/products/ipc_#_##_##_###/docs/doxygen/index.html`.
 - IPC online configuration reference. Choose **Help > Help Contents** and expand the **TI-RTOS** item for your device family. Choose the **Config API Reference (cdoc)** item and see help under the ti.ipc module.
- **NDK**
 - [NDK User's Guide \(SPRU523\)](#)
 - [NDK Programmer's Reference Guide \(SPRU524\)](#)
 - NDK online API reference. Run `<tirtos_install>/products/ndk_#_##_##_###/docs/doxygen/html/index.html`.
 - NDK online configuration reference. Choose **Help > Help Contents** and expand the **TI-RTOS** item for your device family. Choose the **Config API Reference (cdoc)** item and see help under the ti.ndk module.
 - [NDK on the Texas Instruments Wiki](#)
 - [TI-RTOS forum on TI's E2E Community](#)
- **UIA**
 - [System Analyzer User's Guide \(SPRUH43\)](#)
 - UIA online reference. Choose **Help > Help Contents** and expand the **TI-RTOS** item for your device family. Choose the **Config API Reference (cdoc)** item and see help under the ti.uia module.
 - [System Analyzer on the Texas Instruments Wiki](#)
- **FatFS API**
 - [Open source documentation](#)
 - [FatFS for SYS/BIOS wiki page](#)
 - SYS/BIOS online reference (also called "CDOC").
Open from CCS help or run `<tirtos_install>/products/bios_#_##_##_###/docs/cdoc/index.html`.
Navigate to the ti.sysbios.fatfs.FatFS module topic in the SYS/BIOS API reference documentation.
- **General microcontroller information**
 - [Microcontrollers forum on TI's E2E Community](#)

Installing TI-RTOS

This chapter covers the steps to install TI-RTOS within Code Composer Studio or as a standalone software product.

Topic	Page
2.1 System Requirements	9
2.2 Installing Code Composer Studio	10
2.3 Installing TI-RTOS in Code Composer Studio	10
2.4 Installing TI-RTOS as a Standalone Product	11

2.1 System Requirements

The Microsoft Windows version of TI-RTOS can be installed on systems running Windows 7, Windows Vista, or Windows XP (SP2 or SP3).

The Linux version of TI-RTOS can be installed on systems that are running Linux RedHat v4 and higher or Ubuntu v10.04 and higher.

Separate versions of TI-RTOS are available for various Texas Instruments device families.

In order to install TI-RTOS, you must have at least 1 GB of free disk space. (If you have not yet installed Code Composer Studio, you will also need at least 4 GB of disk space for that installation.)

2.2 Installing Code Composer Studio

TI-RTOS 2.0 is used in conjunction with Code Composer Studio 6.0 or higher. CCS is available for Microsoft Windows and Linux.

For Windows installations, we recommend that you install CCS in the default installation directory of `c:\ti`. If you install in `c:\Program Files` (or `c:\Program Files (x86)` with Windows 7), you are likely to run into problems related to Windows security permissions.

Note: Do not install CCS in a location that contains any spaces in the full path. For example, CCS should not be installed in `c:\Program Files`. Makefiles may not function correctly with directory paths that include spaces.

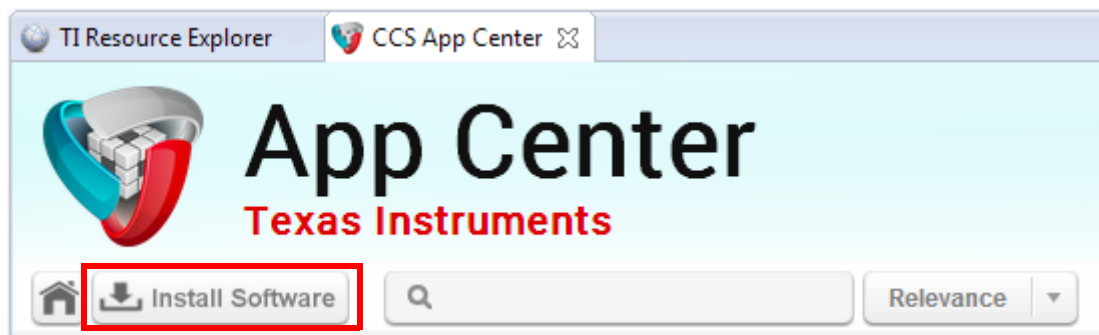
To install CCS 6.0, go to the ["Download CCS" page on the Texas Instruments wiki](#) and follow a link to download the software for your license type. For multi-user licenses, see the [CCS product page](#).

Run the installer, and answer the prompts as appropriate.

2.3 Installing TI-RTOS in Code Composer Studio

TI-RTOS is *not* installed automatically as part of the Code Composer Studio v6.0 installation. Instead, you install it through the CCS App Center. Follow these steps to install TI-RTOS:

1. Run CCS v6.0 or higher.
2. Choose **View > CCS App Center** in CCS.
3. Select the version of TI-RTOS for your device family. If you use devices from multiple families, you can select multiple TI-RTOS versions.
4. Click the **Install Software** button near the top of the App Center view.



5. Answer the prompts as necessary to complete the TI-RTOS installation.
6. Restart CCS in order for TI-RTOS and its components to be available.

2.4 Installing TI-RTOS as a Standalone Product

If you do not use Code Composer Studio, you can install TI-RTOS as a standalone product.

1. [Download the Windows or Linux installer](#) for TI-RTOS for the device family you use. For example, `tirtos_setupwin32_c6000_2_##_##_##.exe` or `tirtos_setuplinux_c6000_2_##_##_##.bin`.
2. Run the downloaded file to install TI-RTOS. You can install TI-RTOS in a standalone directory. Installing in a directory path that contains spaces, such as `C:\Program Files (x86)`, is not recommended.

Note: TI-RTOS installs the core functionality of the XDCtools component if you have not already installed the necessary version as part of a CCS installation. TI-RTOS places XDCtools in a separate directory at the same level where you install TI-RTOS. For example, if the TI-RTOS installation directory is located in `C:\ti\tirtos_c6000_2_##_##_##`, the XDCtools directory will be in `C:\ti\xdctools_3_30_##_##_core`.

Examples for TI-RTOS

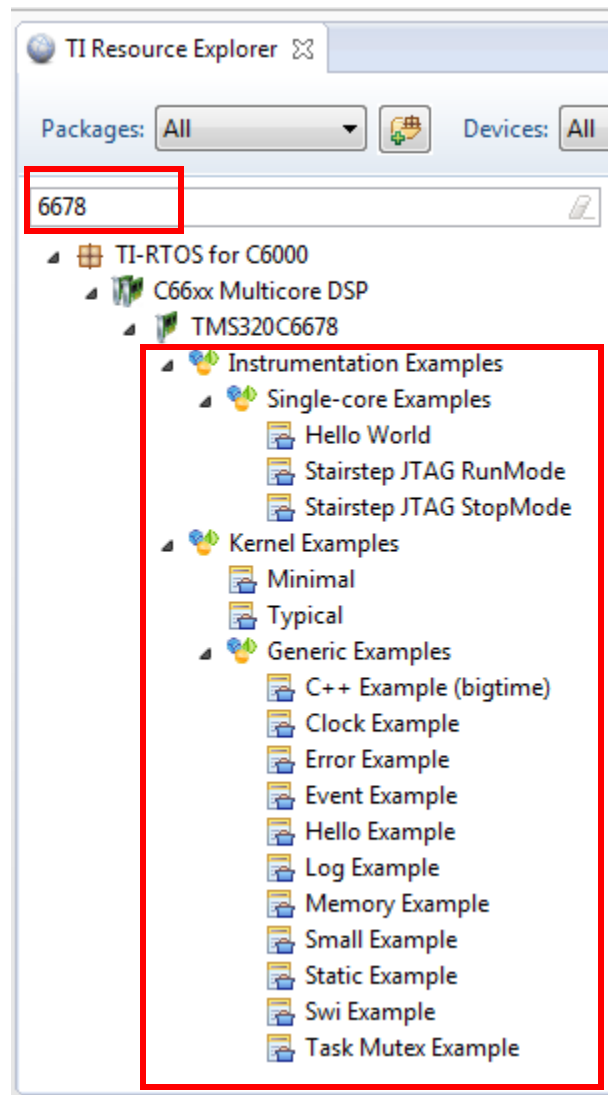
TI-RTOS comes with a number of examples that illustrate on how to use the individual components. This chapter explains how to create and use these examples.

Topic	Page
3.1 Creating Example Projects Using the TI Resource Explorer	13

3.1 Creating Example Projects Using the TI Resource Explorer



You can use the **TI Resource Explorer** in Code Composer Studio (CCS) to create example projects that use TI-RTOS and its components and have all the settings needed for your device. Follow these steps:

1. Open CCS. If you do not see the TI Resource Explorer, make sure you are in the **CCS Edit** perspective and choose **View > Resource Explorer (Examples)** from the menus.
2. Type the name or part of the name of your device in the **enter search keyword** field to hide all the examples that don't apply to your device.
3. Expand the tree until you see the examples for your device. Any **Driver Examples** listed are TI-RTOS driver examples. Any **Instrumentation Examples** listed are UIA examples. The **Kernel Examples** are the SYS/BIOS examples.



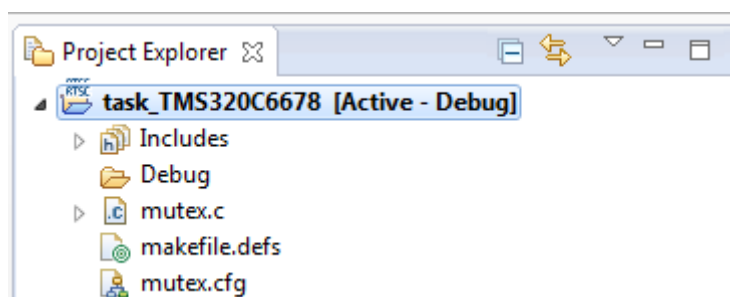
4. Select the example you want to create. A description of the selected example is shown to the right of the example list.

- Click the **Step 1** link in the right pane of the TI Resource Explorer to **Import the example project into CCS**. This adds a new project to your Project Explorer view. Once you have completed a step for a particular example and device, a green checkmark will be shown next to that step.



Step 1:  [Import the example project into CCS](#) 

*Click on the link above to import the project. The imported project is available in the **Project Explorer** view, expand the project node to browse the imported source files. To modify source code, double clicks on the source file within the project to open the source file editor.*

- The project created will have a name with the format `<example_name>_<board>`. You can expand the project to view or change the source code and configuration file.





- The page shown when you select an example in the TI Resource Explorer provides additional links to perform common actions with that example.
- Use the **Step 2** link when you are ready to build the project. If you want to change any build options, right click on the project and select **Properties** from the context menu. For example, you can change compiler, linker, and RTSC (XDCtools) options.

Step 2:  [Build the imported project](#) 

*To change build options, right click on the project and select **Properties** from the context menu. To build the project, select the link above, or select the **Build** toolbar button, or select the **Project | Build Project** menu item.*


- Use the **Step 3** link to change the connection used to communicate with the board. The current setting is shown in the TI Resource Explorer page for the selected example. (If you want to use a simulator instead of a hardware connection, double-click the *.ccxml file in the targetConfigs folder of the project to open the Target Configuration File editor. Change the **Connection** as needed, and click **Save**.)

Step 3:  [Debugger Configuration](#) 

*Connection: **Spectrum Digital XDSPRO USB Emulator***
Click on the link above to change the device connection. Additionally, this option is also available in the project properties.

- You will see the Debugger Configuration dialog. Choose an emulator from the list.

11. Use the **Step 4** link to launch a debug session for the project and switch to the **CCS Debug Perspective**.

Step 4:  [Debug the imported project](#)



*Click on the link above to launch a debug session for the **UART Echo** project and switch to the **CCS Debug Perspective**. Additionally, these are other methods to start a project debug session. Select the project in the **Project Explorer** view and click on the bug toolbar button. To relaunch a previous debug session, click on the small arrow beside the bug toolbar button and select one of the debug session from the history.*

Configuring TI-RTOS

This chapter describes how to configure TI-RTOS and its components for use by your application.

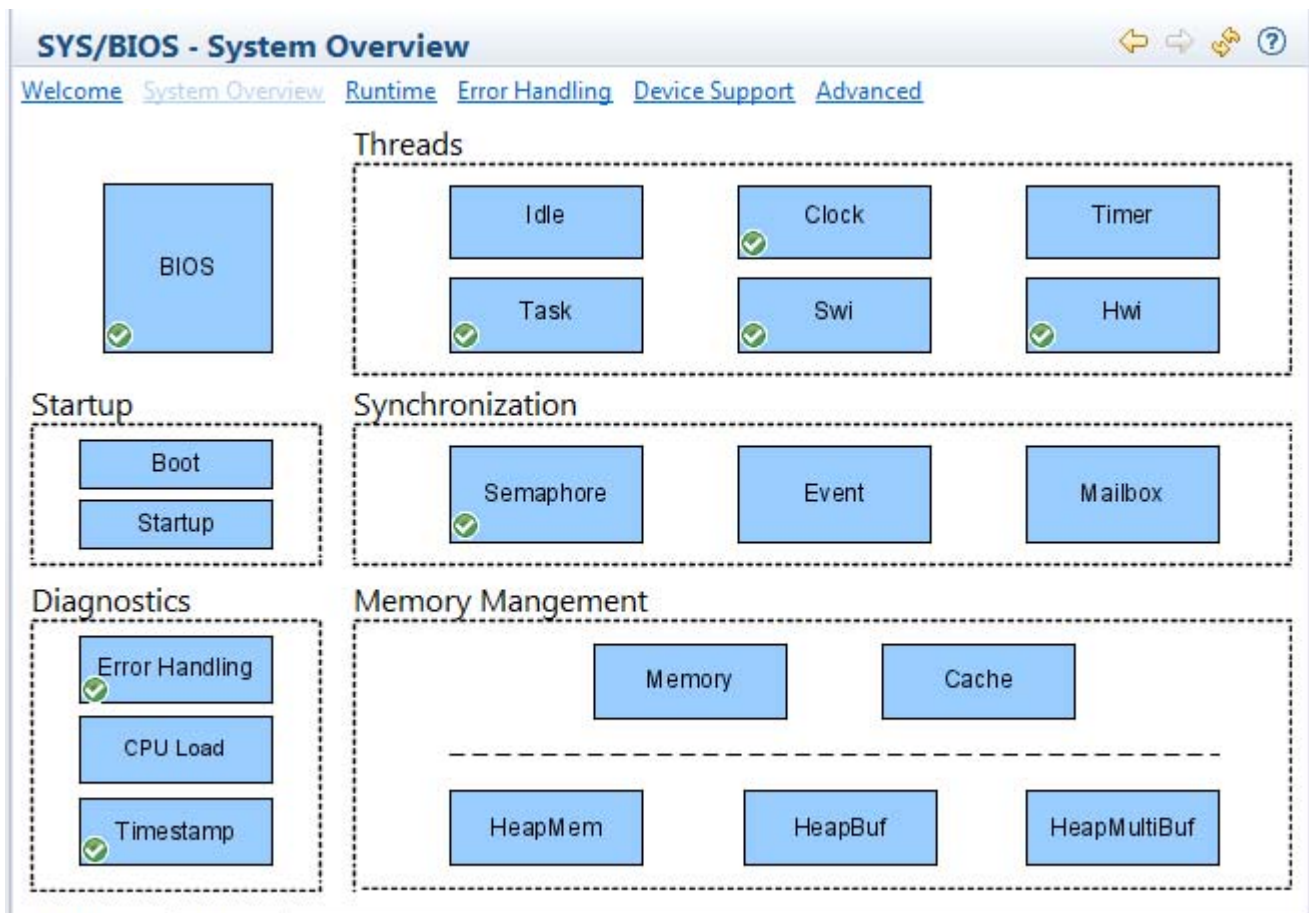
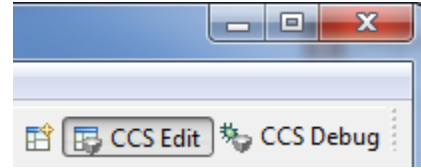
Topic	Page
4.1 Starting the Configuration Tool	17
4.2 Configuring TI-RTOS.	18
4.3 Configuring Components of TI-RTOS	18

4.1 Starting the Configuration Tool

This section shows how to open XGCONF and view the System Overview. For details about how to use XGCONF, see Chapter 2 of the [SYS/BIOS User's Guide \(SPRUEX3\)](#).

To use CCS to open the graphical tool for editing configuration files (XGCONF), follow these steps:

1. Make sure you are in the **C/C++** perspective of CCS. If you are not in that perspective, click the C/C++ icon to switch back.
2. Double-click on the *.cfg configuration file for a TI-RTOS example project in the **Project Explorer** tree. (See [Section 3.1](#) if you need to create an example project.) While XGCONF is opening, the CCS status bar shows that the configuration is being processed and validated.
3. When XGCONF opens, you see the **Welcome** sheet for SYS/BIOS. The Welcome sheet provides links to documentation resources.
4. Click the **System Overview** link to see a diagram of the modules available through SYS/BIOS. A green check mark indicates which modules are being used by the application.



5. Click a blue box in the System Overview to go to the configuration page for a module.

Note: If the configuration is shown in a text editor instead of XGCONF, close the text editor window. Then, right-click on the .cfg file and choose **Open With > XGCONF**. If you are comfortable editing configuration scripts with a text editor, you can do that. However, you should not have the file open in both types of editor at the same time.

4.2 Configuring TI-RTOS

When you open a configuration file in XGCONF, you see a list of Available Products.

The SysCallback module lets you configure the functions that handle System output—for example, System_printf() and System_abort(). This module handles transmissions to System output only; it does not handle responses received. See the chapter on "TI-RTOS Utilities" in the *TI-RTOS User's Guide* (SPRUHD4) for more about the SysCallback module.

Other SystemSupport implementations are provided with XDCtools.

- **SysMin** stores System_printf() strings in an internal buffer in RAM. SysMin requires RAM, so it not ideal for devices with minimal RAM.
- **SysStd** writes System_printf() strings to STDOUT (the CCS Console window). By default, SysStd allows System_printf() to be called from Tasks only (not Swis or hardware interrupts); it can be modified to allow calls from Swis and Hwis, but this impacts real-time performance.

4.3 Configuring Components of TI-RTOS

For information about configuring individual sub-components of TI-RTOS, see the documentation for that component. Chapter 2 of the [SYS/BIOS User's Guide \(SPRUJEX3\)](#) provides information about using XGCONF.

Within XGCONF, you can see the full file path to the version of the component being used by hovering your mouse cursor over a component in the "Other Products" list in the **Available Products** area.

The CCS online help lets you access the CDOC reference help for various components. For information about properties you can configure and instances you can create, scroll down in CDOC to see the sections with gray outlines. (You can use the "Configuration settings" link at the top of each page to skip to the configuration information.)

Index

A

App Center 10
Available Products list 18

C

CCS
 creating a project 6, 13
 installation 10
 other documentation 7
CCS App Center 4, 10
components 5
Concerto
 other documentation 8
configuration 16
 graphical editor 17

D

disk space 9
documentation 7

E

examples 12

F

FatFs API, documentation 8
forum 7

I

installation
 CCS 10
 directory 10
IPC 5
 other documentation 8

N

NDK 5

other documentation 8

P

products directory 5

R

Resource Explorer 6, 13

S

SYS/BIOS 5
 other documentation 7
SysCallback module, configuration 18
System Overview configuration 17
system requirements 9

T

target configuration file 14
TI Resource Explorer 6, 13

U

UIA 5
 other documentation 8

W

wiki 7

X

XDCtools 5
 other documentation 8
XGCONF
 configuring other components 18
 configuring TI-RTOS modules 18
 starting 17

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, enhancements, improvements and other changes to its semiconductor products and services per JESD46, latest issue, and to discontinue any product or service per JESD48, latest issue. Buyers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All semiconductor products (also referred to herein as “components”) are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its components to the specifications applicable at the time of sale, in accordance with the warranty in TI's terms and conditions of sale of semiconductor products. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by applicable law, testing of all parameters of each component is not necessarily performed.

TI assumes no liability for applications assistance or the design of Buyers' products. Buyers are responsible for their products and applications using TI components. To minimize the risks associated with Buyers' products and applications, Buyers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI components or services are used. Information published by TI regarding third-party products or services does not constitute a license to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of significant portions of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI components or services with statements different from or beyond the parameters stated by TI for that component or service voids all express and any implied warranties for the associated TI component or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Buyer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products, and any use of TI components in its applications, notwithstanding any applications-related information or support that may be provided by TI. Buyer represents and agrees that it has all the necessary expertise to create and implement safeguards which anticipate dangerous consequences of failures, monitor failures and their consequences, lessen the likelihood of failures that might cause harm and take appropriate remedial actions. Buyer will fully indemnify TI and its representatives against any damages arising out of the use of any TI components in safety-critical applications.

In some cases, TI components may be promoted specifically to facilitate safety-related applications. With such components, TI's goal is to help enable customers to design and create their own end-product solutions that meet applicable functional safety standards and requirements. Nonetheless, such components are subject to these terms.

No TI components are authorized for use in FDA Class III (or similar life-critical medical equipment) unless authorized officers of the parties have executed a special agreement specifically governing such use.

Only those TI components which TI has specifically designated as military grade or “enhanced plastic” are designed and intended for use in military/aerospace applications or environments. Buyer acknowledges and agrees that any military or aerospace use of TI components which have not been so designated is solely at the Buyer's risk, and that Buyer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI has specifically designated certain components as meeting ISO/TS16949 requirements, mainly for automotive use. In any case of use of non-designated products, TI will not be responsible for any failure to meet ISO/TS16949.

Products

Audio	www.ti.com/audio
Amplifiers	amplifier.ti.com
Data Converters	dataconverter.ti.com
DLP® Products	www.dlp.com
DSP	dsp.ti.com
Clocks and Timers	www.ti.com/clocks
Interface	interface.ti.com
Logic	logic.ti.com
Power Mgmt	power.ti.com
Microcontrollers	microcontroller.ti.com
RFID	www.ti-rfid.com
OMAP Mobile Processors	www.ti.com/omap
Wireless Connectivity	www.ti.com/wirelessconnectivity

Applications

Automotive and Transportation	www.ti.com/automotive
Communications and Telecom	www.ti.com/communications
Computers and Peripherals	www.ti.com/computers
Consumer Electronics	www.ti.com/consumer-apps
Energy and Lighting	www.ti.com/energy
Industrial	www.ti.com/industrial
Medical	www.ti.com/medical
Security	www.ti.com/security
Space, Avionics and Defense	www.ti.com/space-avionics-defense
Video & Imaging	www.ti.com/video
TI E2E Community	e2e.ti.com