SYS/BIOS 6.35 Getting Started Guide

June 4, 2013

Read This First

About This Manual

This document describes the process of installing and using SYS/BIOS 6.35 with Code Composer Studio v5.3+.

Previous versions of SYS/BIOS were called DSP/BIOS. The new name reflects that the kernel can also be use on processors other than DSPs.

How to Use This Manual

This document provides installation instructions and step-by-step instructions for starting to use SYS/BIOS.

After you install SYS/BIOS, you might want to review the release notes in the installation before reading further.

After you have read this document, you should see the *SYS/BIOS 6.35 User's Guide* (SPRUEX3) and the online CDOC reference for more information.

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(){
      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.

Related Documentation from Texas Instruments

See Sections 2.2 and 2.3 in this document for a detailed list of other documentation resources.

Trademarks

The Texas Instruments logo and Texas Instruments are registered trademarks of Texas Instruments. Trademarks of Texas Instruments include: TI, Code Composer, Code Composer Studio, DSP/BIOS, SYS/BIOS, SPOX, TMS320, TMS320C54x, TMS320C55x, TMS320C62x, TMS320C64x, TMS320C67x, TMS320C28x, TMS320C5000, TMS320C6000 and TMS320C2000.

Windows is a registered trademark of Microsoft Corporation.

Linux is a registered trademark of Linus Torvalds.

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

Contents

Using SYS/BIOS with CCS

This chapter describes how to build SYS/BIOS examples and applications.

1.1 Installing SYS/BIOS

If you are installing Code Composer Studio (CCS), all the components you need in order to use SYS/BIOS 6.35 can be installed as part of the CCS installation. When you perform the CCS installation, make sure the checkboxes are selected to install XDCtools and SYS/BIOS 6.35 for your specific platform or alternately choose to install everything in the installation wizard.

If you are performing a separate installation of SYS/BIOS, be sure to read the release notes. On Windows, install SYS/BIOS in the top-level folder of the CCS installation. That is, install SYS/BIOS in the same folder that contains the "ccsv5" folder and other component installations. You will need to restart CCS after it discovers the new version of SYS/BIOS.

1.2 Creating SYS/BIOS Projects with the TI Resource Explorer

Follow these steps to use the TI Resource Explorer to create a CCS project that can use SYS/BIOS 6.35.

1) Open CCS.

2) If you do not see the TI Resource Explorer, make sure you are in the CCS Edit perspective and choose **View > TI Resource Explorer** from the menus.

3) Expand the SYS/BIOS item in the tree to show **SYS/BIOS >** *device* **>** *board*, where *device > board* is your platform.
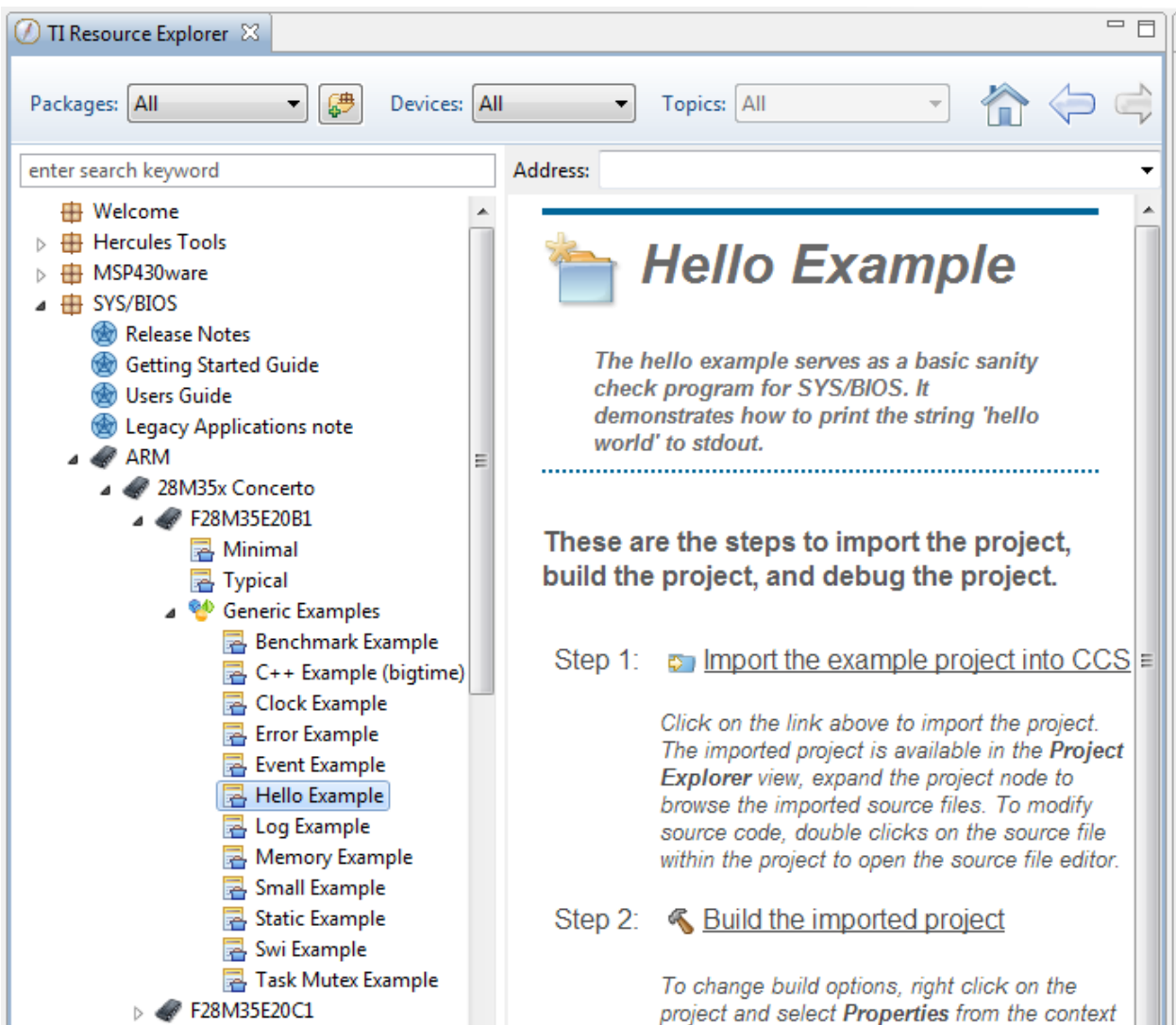
4) Select the example you want to create. A description of the selected example is shown at the top of the page to the right of the example list. To get started

with SYS/BIOS, you can choose one of the Generic Examples, such as the **Log Example** or **Task Mutex Example**.

5) When you are ready to create your own application project, you might choose the "Minimal" or "Typical" example depending on how memory-limited your target is. For some device families, device-specific SYS/BIOS templates are also provided. (If you have other software components that use SYS/BIOS, such as IPC, you can choose a template provided for that component.)

6) 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.

Step 1: 📂 Import the example project into CCS

7) 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.

8) The page shown when you select an example in the TI Resource Explorer provides additional links to perform common actions with that example. You'll see a green checkmark next to the steps that you have performed.

9) Use the **Step 2** link in the TI Resource Explorer 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

10) Use the **Step 3** link in the TI Resource Explorer 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, choose **View > Target Configurations** in CCS and find the *.ccxml file for your project. Double-click on the file to open the Target Configuration File editor. Change the **Connection** as needed, and click **Save**.)

Step 3: 🛠 Debugger Configuration

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

Step 4: 🐞 Debug the imported project

1.3 SYS/BIOS Examples

The following SYS/BIOS examples are available for most targets:

- Benchmark Example
- C++ Example (bigtime)
- Clock Example
- Error Example
- Event Example
- Hello Example
- Log Example
- Memory Example
- Small Example
- Static Example
- Swi Example
- Task Mutex Example

For some device families, device-specific SYS/BIOS templates are also provided. (If you have other software components that use SYS/BIOS, such as IPC, you can choose a template provided for that component.)

When you are ready to create your own application project, you might choose the "Minimal" or "Typical" example depending on how memory-limited your target is.

1.4 Working with CCS Build Settings

After you have created a project that contains a SYS/BIOS configuration file, you can change the properties of the project in CCS by right-clicking the project name and choosing **Properties**. Choose the "CCS General" category and then the "RTSC" tab. You can change the settings you made on the "RTSC Configuration Settings" page of the project creation wizard. See Section 1.2.

In the "CCS General" category of the Properties dialog, the **General** tab applies to compiler settings, and the **RTSC** tab applies to the "configuro" utility provided by XDCtools to process the .cfg file.

If your C source code and your SYS/BIOS configuration file are stored in separate projects, note that your build settings for both projects must match or be compatible. If you change the build settings for a configuration project, you should also change the build settings for application projects that use that configuration.

Note that if there is any platform-specific configuration in your .cfg file, you much change those settings in addition to any changes you make to the CCS Build Settings.

For more details, see Section 2.3 of the *SYS/BIOS User's Guide.*

1.5   Using XGCONF to Configure SYS/BIOS Modules and Objects

XGCONF is a tool that allows you to graphically create and view SYS/BIOS configuration scripts. (If you have used DSP/BIOS 5.x, it is somewhat similar to the DSP/BIOS Configuration Tool.)

To open XGCONF, right-click on a *.cfg file in the "C/C++ Projects" view, and select **Open with > XGCONF**.
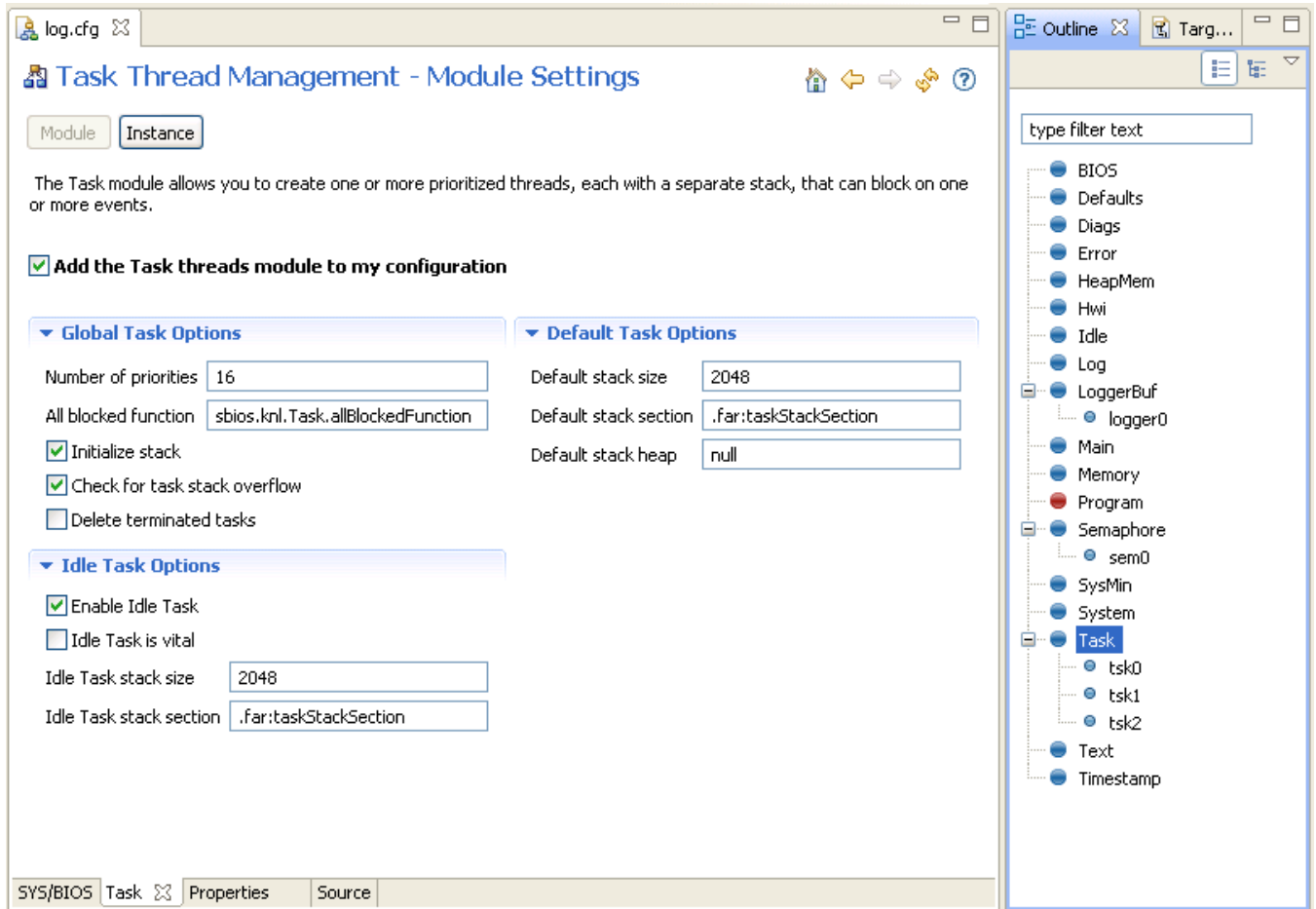
For more details, see Section 2.2 of the *SYS/BIOS User's Guide*.
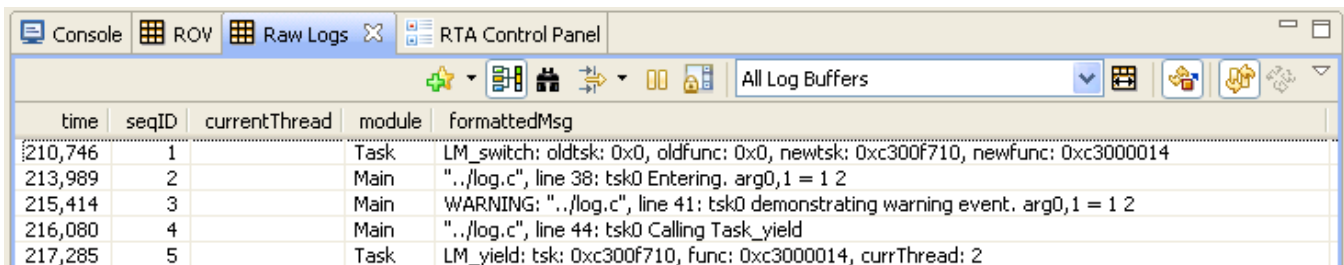
1.6 Building a SYS/BIOS Project

To build your project, click on the project "bioslog [Active – Debug]" so it is highlighted and then choose **Project > Build Active Project** from the CCS menus.

You will see messages about the build's progress in a Console window. The "Invoking: XDCtools" section of the messages provides information about processing of the SYS/BIOS configuration file. For SYS/BIOS projects, this is performed before the compiler and linker are run.

For more details, see Section 2.3 of the *SYS/BIOS User's Guide*.

1.7 Running a SYS/BIOS Project

1) To run your project, choose **Run > Debug** from the CCS menus. If this is the first time you are debugging a project for your target, you may need to set up a CCS Target Configuration. See the CCS help or Section 2.3.4 of the *SYS/BIOS User's Guide* for details.

2) In the Debug perspective, open the Runtime Object Viewer (ROV) tool by choosing T**ools > RTOS Object View (ROV)**. Also open the Raw Logs view by choosing **Tools > RTOS Analyzer > RTA (Legacy) > Raw Logs**. These tools allow you to see the activity of XDCtools and SYS/BIOS modules.

3) Set some breakpoints in the log.c source file. (You can do this by right-clicking on a line and choosing **New Breakpoint > Breakpoint**.) For example, set a breakpoint on the last line of each function in log.c.

4) Run the application.

5) In the Raw Logs window, you can see the informational, warning, and error messages sent by the calls to Log module APIs in log.c. The messages that begin with "LM" are diagnostics provided by XDCtools. Messages that begin with "WARNING" come from calls to Log_warning2. Messages that begin with "ERROR" come from calls to Log_error2. Messages that begin with "../log.c" come from calls to Log_info0 and Log_info2 (depending on the number of arguments).

time	seqID	currentThread	module	formattedMsg
210,746	1		Task	LM_switch: oldtsk: 0x0, oldfunc: 0x0, newtsk: 0xc300f710, newfunc: 0xc3000014
213,989	2		Main	"../log.c", line 38: tsk0 Entering. arg0,1 = 1 2
215,414	3		Main	WARNING: "../log.c", line 41: tsk0 demonstrating warning event. arg0,1 = 1 2
216,080	4		Main	"../log.c", line 44: tsk0 Calling Task_yield
217,285	5		Task	LM_yield: tsk: 0xc300f710, func: 0xc3000014, currThread: 2

6) In the ROV window, expand the tree to see the ti.sysbios.knl.Task module. The right pane shows a list of the Task threads in the application. As you advance from breakpoint to breakpoint, you see the run mode of the threads change.

For more details, see Chapter 8 of the *SYS/BIOS User's Guide.*

More About SYS/BIOS

This chapter provides more information about beginning to use SYS/BIOS.

2.1 Using SYS/BIOS as a Standalone Product

You can also install SYS/BIOS for use outside of CCS.

In order to use SYS/BIOS, you must also install XDCtools. See the *XDCtools Getting Started Guide* for installation instructions. XDCtools provides tools to build projects that contain the configuration files used by SYS/BIOS.

To install SYS/BIOS as standalone software, follow these steps:

1) Place the distribution file into a temporary location.

2) Double-click on the distribution file to start the installation process. The installation directory, *<bios_install_dir>*, can be anywhere on your system, but make sure there are no spaces in the full path to *<bios_install_dir>*.

3) View documentation for the SYS/BIOS packages at *<bios_install_dir>*/docs.

You can build SYS/BIOS applications, including the *.cfg configuration file, on the command line on Windows or Linux using makefiles. When you use a makefile to build, you can compile and link with the TI Code Generation Tools, the GCC compiler for your target, or other command-line compilers.

For more details, see Section 2.3.3 of the *SYS/BIOS User's Guide*. For sample files to download, see the SYS/BIOS with GCC (http://processors.wiki.ti.com/index.php/SYS/BIOS_with_GCC) topic on the TI Embedded Processors Wiki.

2.2 Learning More

To learn more about using SYS/BIOS, see the following:

- *SYS/BIOS Release Notes.* In the top-level SYS/BIOS installation directory, or choose **Help > Help Contents** in CCS and expand the **SYS/BIOS** item.

- *SYS/BIOS User's Guide* (SPRUEX3). In <bios_install_dir>/docs, or choose **Help > Help Contents** in CCS and expand the **SYS/BIOS** item. Also available on the Texas Instruments website (http://www.ti.com/lit/pdf/spruex3).

- *Migrating a DSP/BIOS 5 Application to SYS/BIOS 6* (SPRAAS7). In <bios_install_dir>/docs/ Bios_Legacy_App_Note.pdf, or choose **Help > Help Contents** in CCS and expand the **SYS/BIOS** item. Also available on the Texas Instruments website (http://www.ti.com/lit/pdf/spraas7).

- *SYS/BIOS API Reference* (also called "CDOC"). Run <bios_install_dir>/docs/cdoc/index.html, or choose **Help > Help Contents** in CCS and expand the **SYS/BIOS** item.

- SYS/BIOS main page (http://processors.wiki.ti.com/index.php/Category:SYSBIOS) on the TI Embedded Processors Wiki. This page provides links to many resources for using SYS/BIOS.

- SYS/BIOS quick starts (http://processors.wiki.ti.com/index.php/SYS/BIOS_Quick_Starts) provide target-specific examples of how to create, build, and run sample projects.

- BIOS forum (http://e2e.ti.com/support/embedded/f/355.aspx) on TI's E2E Community.

- SYS/BIOS Product Folder (http://focus.ti.com/docs/toolsw/folders/print/dspbios6.html) on ti.com

- Embedded Software Download Page (http://software-dl.ti.com/dsps/dsps_public_sw/sdo_sb/targetcontent/index.html)

To learn more about using Code Composer Studio (CCS), see the following:

- *CCS online help.* Choose **Help > Help Contents** in CCS.

- CCSv5 main page (http://processors.wiki.ti.com/index.php/Category:Code_Composer_Studio_v5) on the TI Embedded Processors Wiki

- Code Composer forum (http://e2e.ti.com/support/development_tools/code_composer_studio/f/81.aspx) on TI's E2E Community

2.3 About XDCtools

XDCtools is a set of tools that enable the creation and use of software packages. It allows products to be developed as reusable packages. (RTSC is the underlying standard that is implemented by XDCtools.)

XDCtools defines the configuration language used by SYS/BIOS (and other products that use XDCtools). For SYS/BIOS, the configuration scripts are stored in .cfg and other files.

To learn more about using XDCtools, see the following:

- *XDCtools API Reference* (also called "CDOC"). Run <xdctools_install_dir>/docs/xdctools.chm, or choose **Help > Help Contents** in CCS and expand the **XDCtools** item.

- RTSC-Pedia Wiki (http://rtsc.eclipse.org/docs-tip)

- BIOS forum (http://e2e.ti.com/support/embedded/f/355.aspx) on TI's E2E Community

- Embedded Software Download Page (http://software-dl.ti.com/dsps/dsps_public_sw/sdo_sb/targetcontent/index.html)