QUICKSTART – CODE COMPOSER STUDIO™

Stellaris® Development and Evaluation Kits for Code Composer Studio™

Stellaris Development and Evaluation Kits provide a low-cost way to start designing with Stellaris microcontrollers using Texas Instruments' Code Composer Studio development tools. The evaluation boards can function as either a complete evaluation target or as a debugger interface to any external Stellaris device.

Requirements

- You have a PC with a USB interface, running Microsoft® Windows XP (SP2 or greater) or Vista
- You have the Stellaris Evaluation Kit Documentation and Software DVD or the standalone Code Composer Studio DVD found in the Development Kit

CAUTION: There is a known electrical issue with the FT2232 device that is used in the on-board In Circuit Debug Interface (ICDI). Some USB hubs can cause the device to misbehave, with symptoms ranging from failed enumeration to corrupt data transfers. If you experience trouble when using the on-board ICDI, try connecting the USB cable directly to one of the USB ports on your PC or laptop.

Code Composer Studio

This quickstart shows you how to install the Code Composer Studio development tool and how to use it to build and run an example application on your Stellaris Evaluation or Development Board.

Step 1: Install Code Composer Studio

1. Insert the Evaluation Kit Documentation and Software DVD or the standalone Code Composer Studio DVD into the DVD-ROM drive of your computer. If Autoplay is enabled on your PC, the index.htm file automatically opens in your default web browser. If Autoplay is not enabled, use Windows Explorer to open the DVD manually.

2. With the Evaluation Kit DVD, click the Tools button and then click the Code Composer Studio logo to start the setup program.

   NOTE: The CCS installer executable relies on additional installer executables on the DVD. Your web browser must support running files without first saving them to your hard drive to successfully run the CCS installer by clicking the CCS logo. If your web browser does not support this feature, you will need to use Windows Explorer to browse to and run the CCS installer (setup_CCS_n.n.n.nnnnn.exe) from the \Tools\CCS\ directory on the DVD.

   With the standalone DVD, follow the installer dialog.

3. Follow the instructions in the Code Composer Studio installation program. Select to install the Platinum Edition or the Microcontroller Edition with the Stellaris Cortex-M3 MCUs ISA selected at a minimum.

Step 2: Install the StellarisWare® Package

A full set of C-based peripheral drivers is provided, covering all peripherals and functionality of the Stellaris devices. The StellarisWare package includes various example applications with project files for all major tool vendors that support Stellaris, including Code Composer Studio. To install StellarisWare components, follow these steps:

1. Navigate to the *Tools* tab on the Evaluation Kit Documentation and Software DVD or Development Kit Documentation and Software DVD.

   NOTE: If you are navigating the DVD using Windows Explorer (or a similar application), go to the Tools/StellarisWare/ directory.

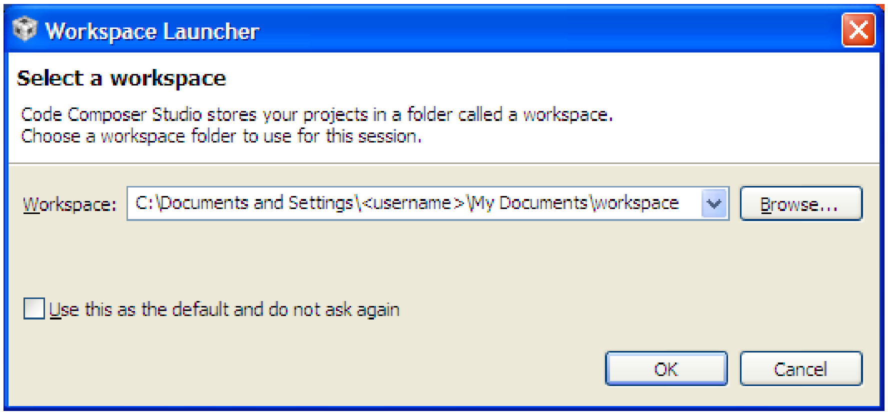
2. Click on the 'Install' link next in the StellarisWare section of the DVD and run the StellarisWare installer. If you prefer to manually install StellarisWare, the installer is a self-extracting zip file that is located in the Tools/StellarisWare directory. You can use a zip file extraction utility such as WinZip to manually extract the contents.

3. To view the StellarisWare documentation, navigate to the installation directory and click on the *Stellaris Peripheral Driver Library User's Guide* PDF.

NOTE: For the most recent version of the StellarisWare, check the www.ti.com/stellaris web site for the latest software updates.

Step 3: Start Code Composer Studio and Open a Workspace

1. Start the Code Composer Studio IDE by selecting it from the Windows Start menu or double-clicking the icon installed on your desktop. When the IDE loads, it asks you where to open the workspace folder:



Workspace Launcher

Select a workspace

Code Composer Studio stores your projects in a folder called a workspace.
Choose a workspace folder to use for this session.

Workspace: C:\Documents and Settings\<username>\My Documents\workspace

□ Use this as the default and do not ask again

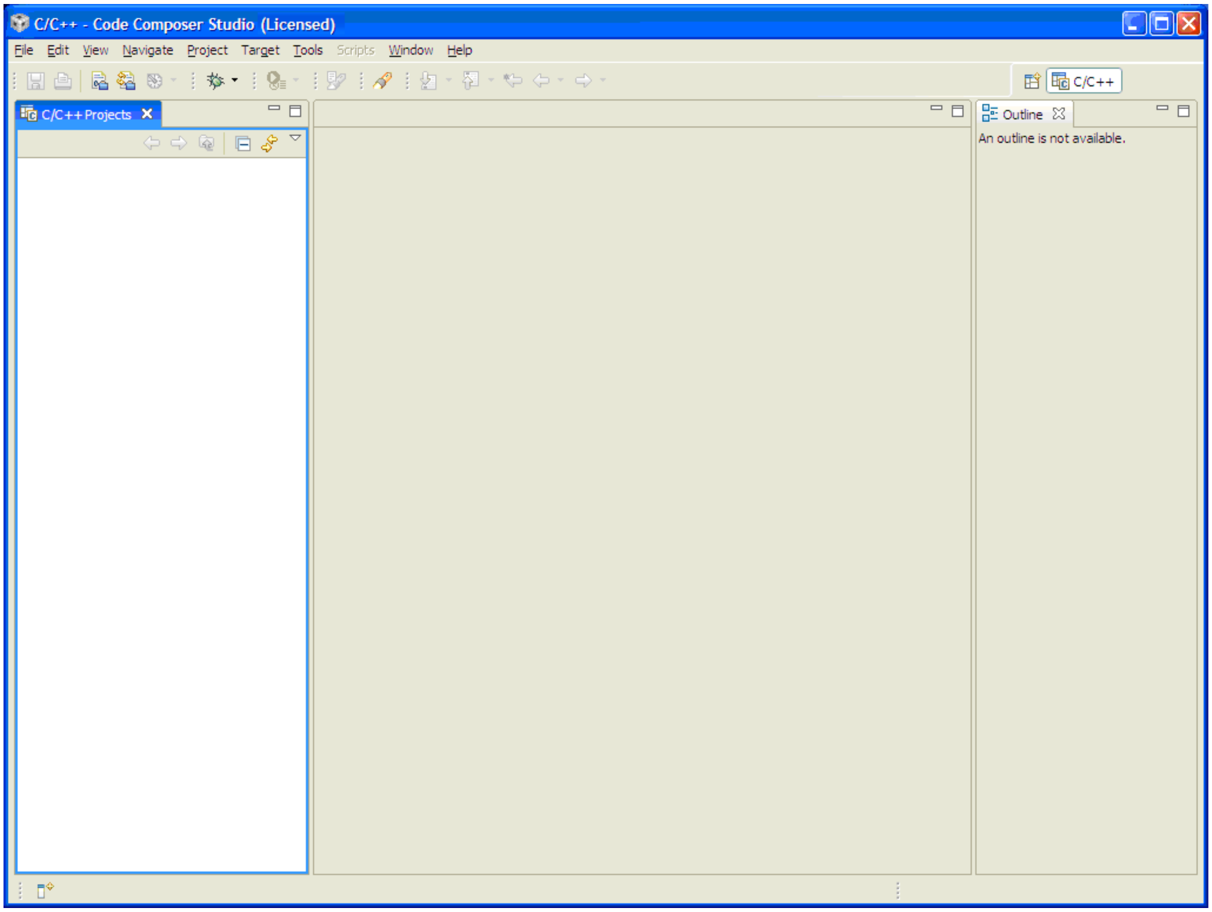
2. The Workspace launcher defaults to the following path:

```
C:\Documents and Settings\<username>\My Documents\workspace
```

Click OK to use this default workspace location.

Note: If you installed the Code Composer Studio IDE from the DVD, the license was automatically installed. If you downloaded the Code Composer Studio IDE, you may have to go through an extra licensing step.

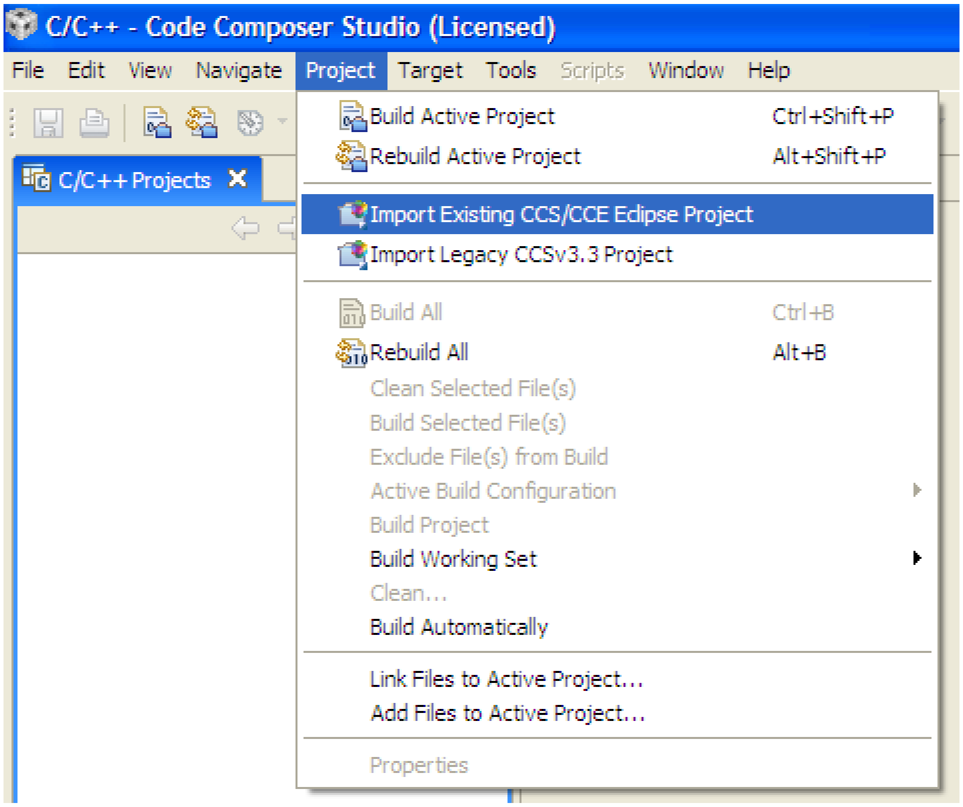
3. The Code Composer Studio IDE may now open with the welcome page. If so, close out the welcome page by clicking the 'X' on the tab. You should now have an empty workspace.


C/C++ - Code Composer Studio (Licensed)
File Edit View Navigate Project Target Tools Scripts Window Help
C/C++
C/C++ Projects
Outline

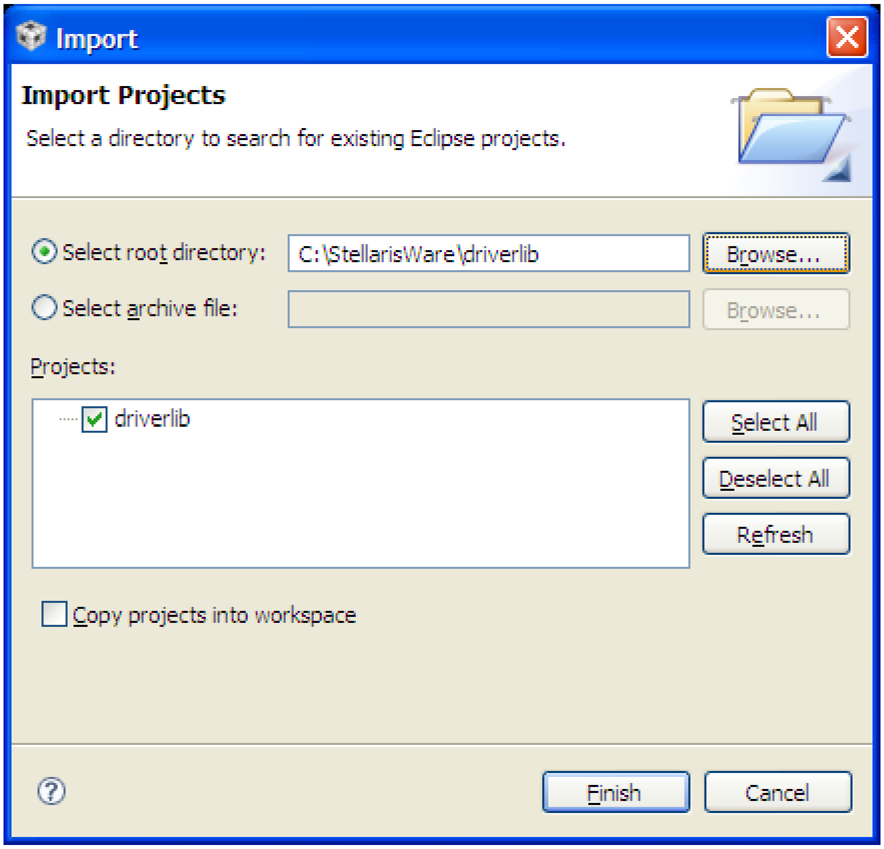
An outline is not available.

Step 4: Import Libraries

1. Select the "Import Existing CCS/CCE Eclipse Project" option from the Project menu in the IDE.


C/C++ - Code Composer Studio (Licensed)
File
Edit
View
Navigate
Project
Target
Tools
Scripts
Window
Help
C/C++ Projects
Build Active Project Ctrl+Shift+P
Rebuild Active Project Alt+Shift+P
Import Existing CCS/CCE Eclipse Project
Import Legacy CCSv3.3 Project
Build All Ctrl+B
Rebuild All Alt+B
Clean Selected File(s)
Build Selected File(s)
Exclude File(s) from Build
Active Build Configuration
Build Project
Build Working Set
Clean...
Build Automatically
Link Files to Active Project...
Add Files to Active Project...
Properties

2. The “Import” dialog box appears. Browse to the root directory of driver library (`StellarisWare\driverlib`). Be sure that “driverlib” is selected in the listed projects and click “Finish.”



Import

Import Projects

Select a directory to search for existing Eclipse projects.

● Select root directory: C:\StellarisWare\driverlib

○ Select archive file:

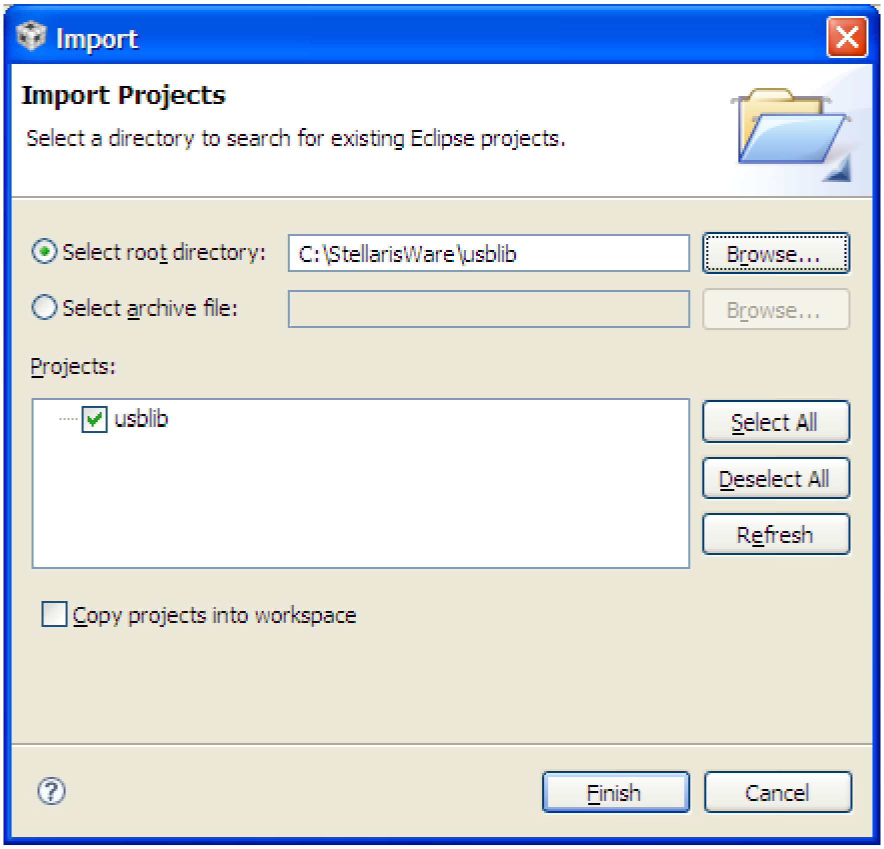
Projects:

☑ driverlib

□ Copy projects into workspace

IMPORTANT: If you have installed the StellarisWare Firmware Development package for a board that does not use the USB library or Graphics library, the following steps do not apply. If the directories mentioned below do not exist in your StellarisWare installation, then cancel out of the import projects dialog.

3. Select the "Import Existing CCS/CCE Eclipse Project" option from the Project menu in the IDE again. Browse to the root directory of USB library (`StellarisWare\usblib`). Be sure that "usblib" is selected in the projects list and click "Finish."



Import

Import Projects

Select a directory to search for existing Eclipse projects.

● Select root directory: C:\StellarisWare\usblib

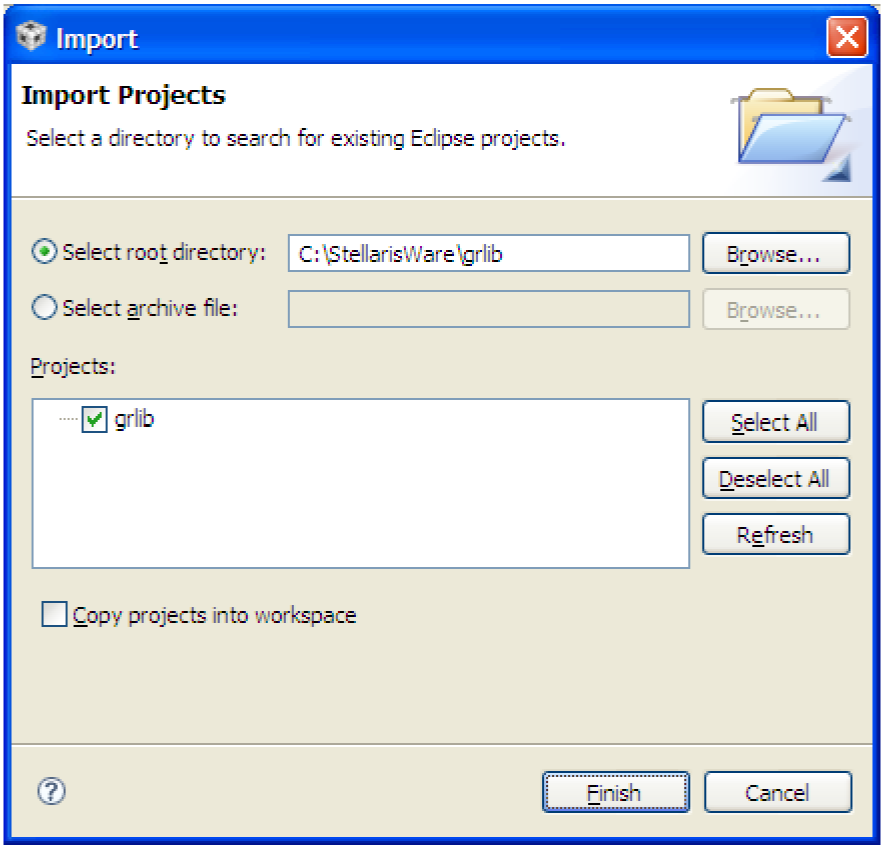
○ Select archive file:

Projects:

☑ usblib

□ Copy projects into workspace

4. Select the “Import Existing CCS/CCE Eclipse Project” option from the Project menu in the IDE again. Browse to the root directory of the graphics library (`StellarisWare\grlib`). Be sure that “grlib” is selected in the projects list and click “Finish.”



Import Projects

Select a directory to search for existing Eclipse projects.

● Select root directory: C:\StellarisWare\grlib

○ Select archive file:

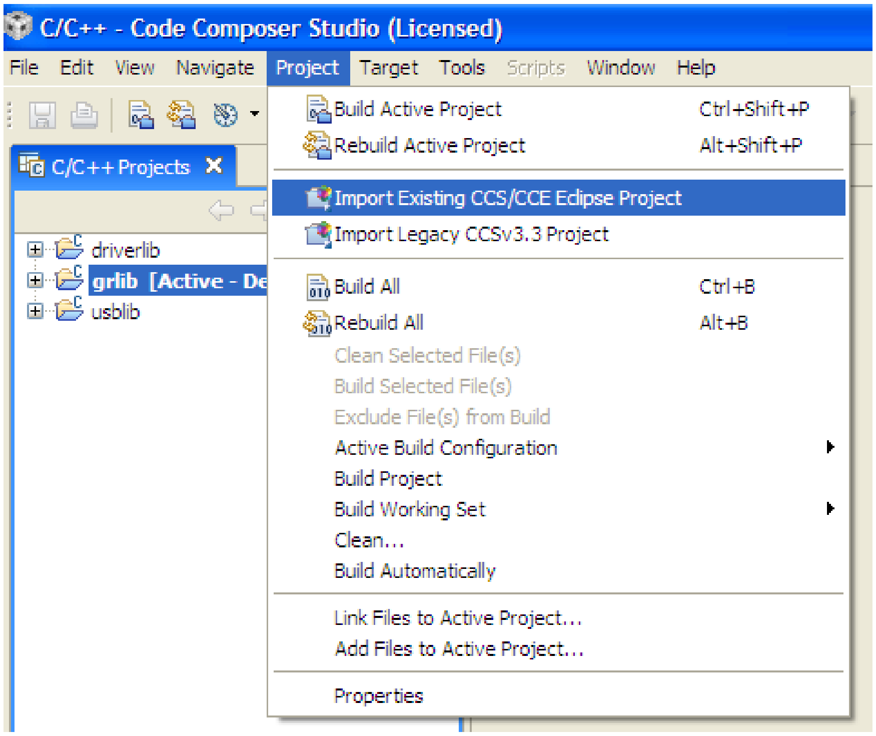
Projects:

☑ grlib

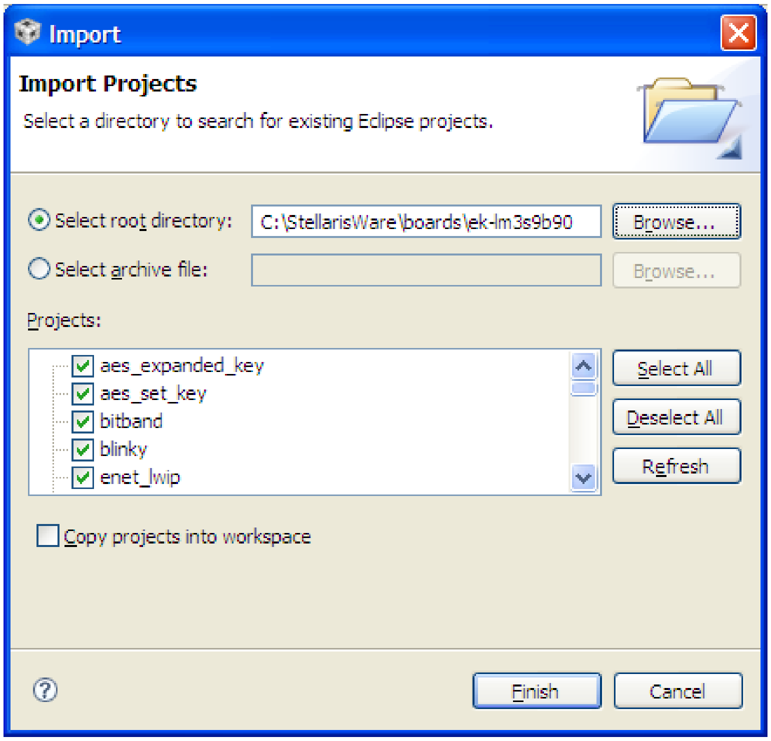
□ Copy projects into workspace

Step 5: Import Board Examples

1. Select the “Import Existing CCS/CCE Eclipse Project” option from the Project menu in the IDE.


C/C++ - Code Composer Studio (Licensed)
File Edit View Navigate Project Target Tools Scripts Window Help
C/C++ Projects
driverlib
grlib [Active - De
usblib
Build Active Project Ctrl+Shift+P
Rebuild Active Project Alt+Shift+P
Import Existing CCS/CCE Eclipse Project
Import Legacy CCSv3.3 Project
Build All Ctrl+B
Rebuild All Alt+B
Clean Selected File(s)
Build Selected File(s)
Exclude File(s) from Build
Active Build Configuration
Build Project
Build Working Set
Clean...
Build Automatically
Link Files to Active Project...
Add Files to Active Project...
Properties

2. The “Import” dialog box appears. Browse to the root directory for your chosen board. The example screen shot below has the EK-LM3S9B90 board as the chosen board. (`StellarisWare\boards\ek-lm3s9b90`). Be sure that the examples are selected in the listed projects and click “Finish.”



Import Projects

Select a directory to search for existing Eclipse projects.

● Select root directory: C:\StellarisWare\boards\ek-lm3s9b90

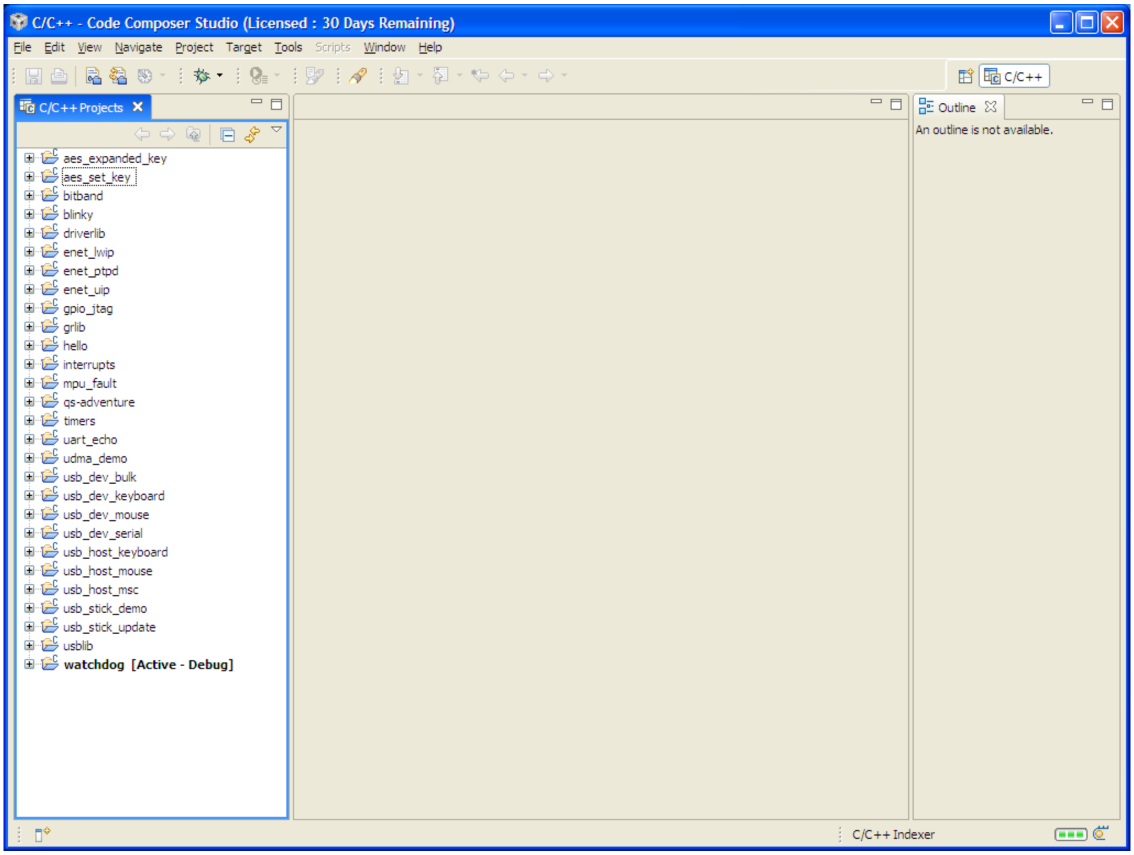
○ Select archive file:

Projects:

☑ aes_expanded_key
☑ aes_set_key
☑ bitband
☑ blinky
☑ enet_lwip

□ Copy projects into workspace

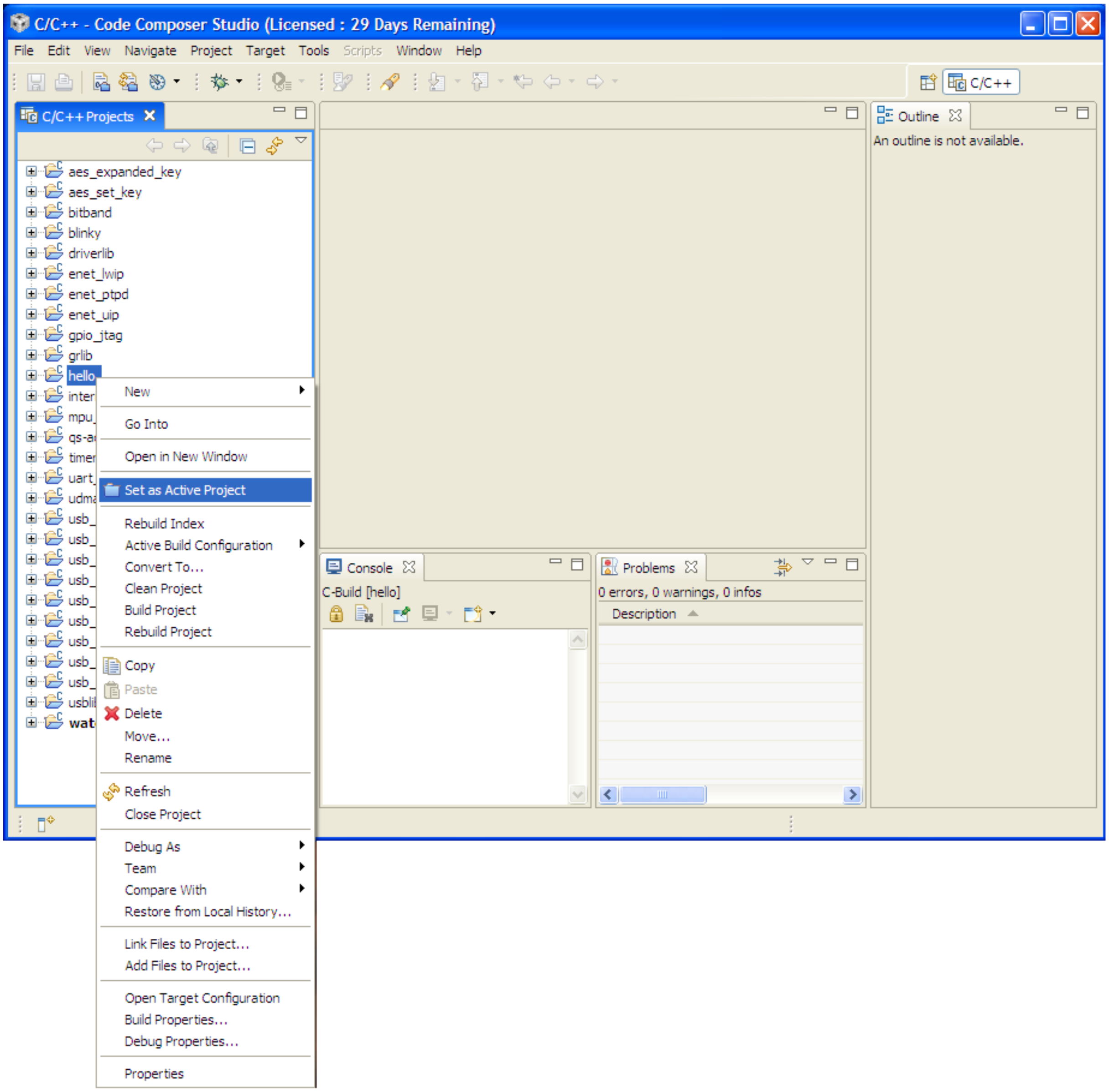
3. All of the projects now show in the Projects Explorer Window.


C/C++ - Code Composer Studio (Licensed : 30 Days Remaining)
File Edit View Navigate Project Target Tools Scripts Window Help
C/C++
C/C++ Projects
Outline
An outline is not available.
C/C++ Indexer

aes_expanded_key
aes_set_key
bitband
blinky
driverlib
enet_lwip
enet_ptpd
enet_uip
gpio_jtag
grlib
hello
interrupts
mpu_fault
qs-adventure
timers
uart_echo
udma_demo
usb_dev_bulk
usb_dev_keyboard
usb_dev_mouse
usb_dev_serial
usb_host_keyboard
usb_host_mouse
usb_host_msc
usb_stick_demo
usb_stick_update
usblib
watchdog [Active - Debug]

Step 6: Building and Debugging a Project

1. Right click on the hello project and select “Set as Active Project.”

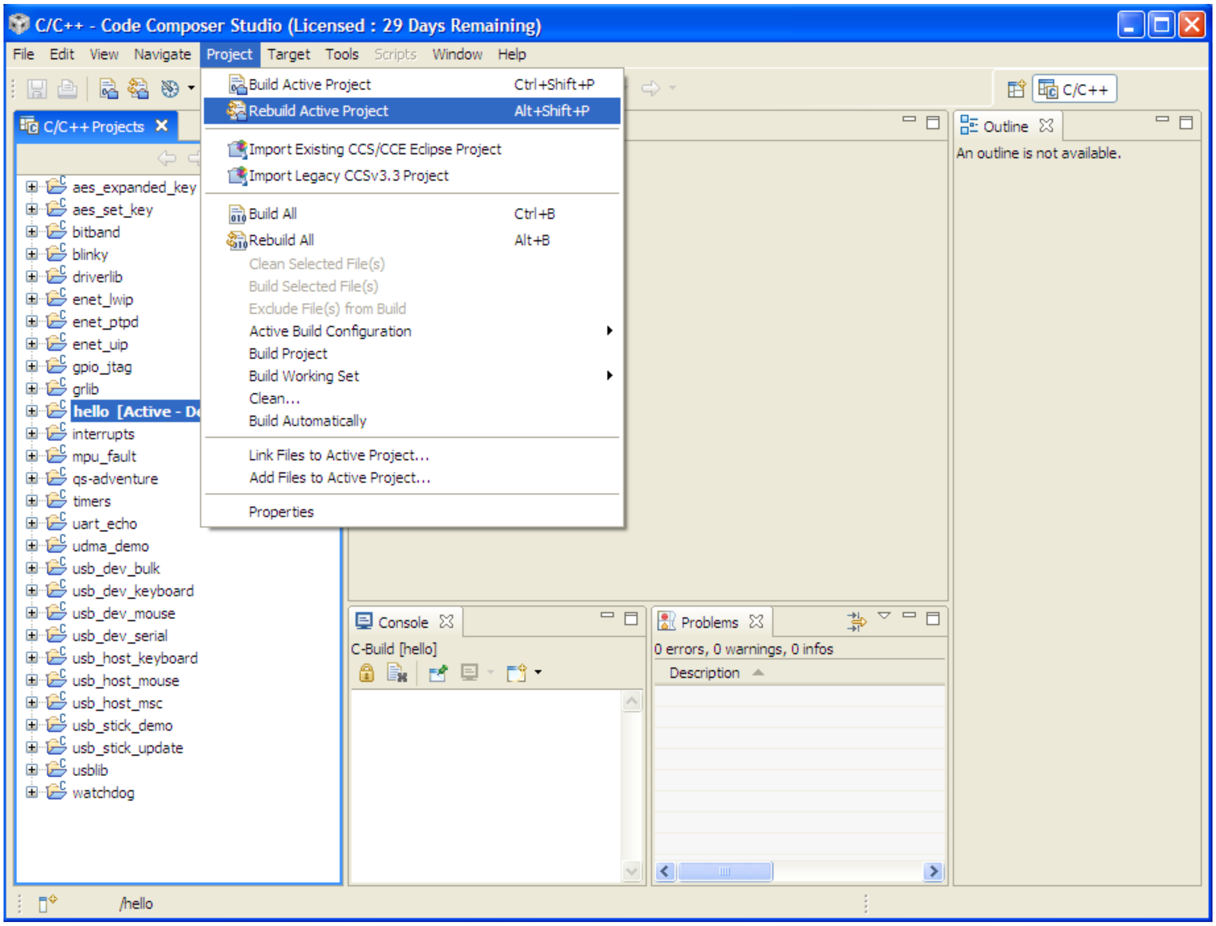


An outline is not available.

C-Build [hello]

0 errors, 0 warnings, 0 infos

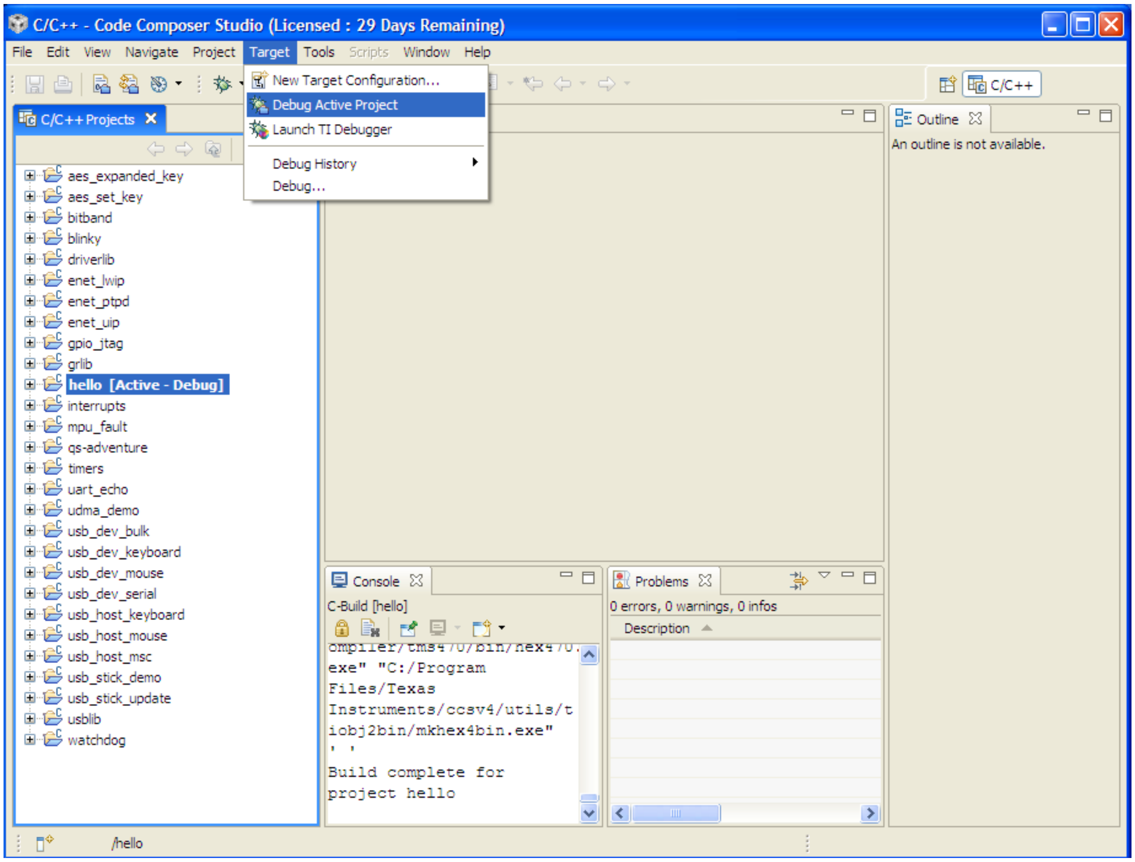
2. Click on the Project pull-down menu and select “Rebuild Active Project.” The build may take a few moments. As the project builds, messages scroll by in the console window. When the build is complete, the words “Build complete for project hello” appear in the console window.



An outline is not available.

0 errors, 0 warnings, 0 infos

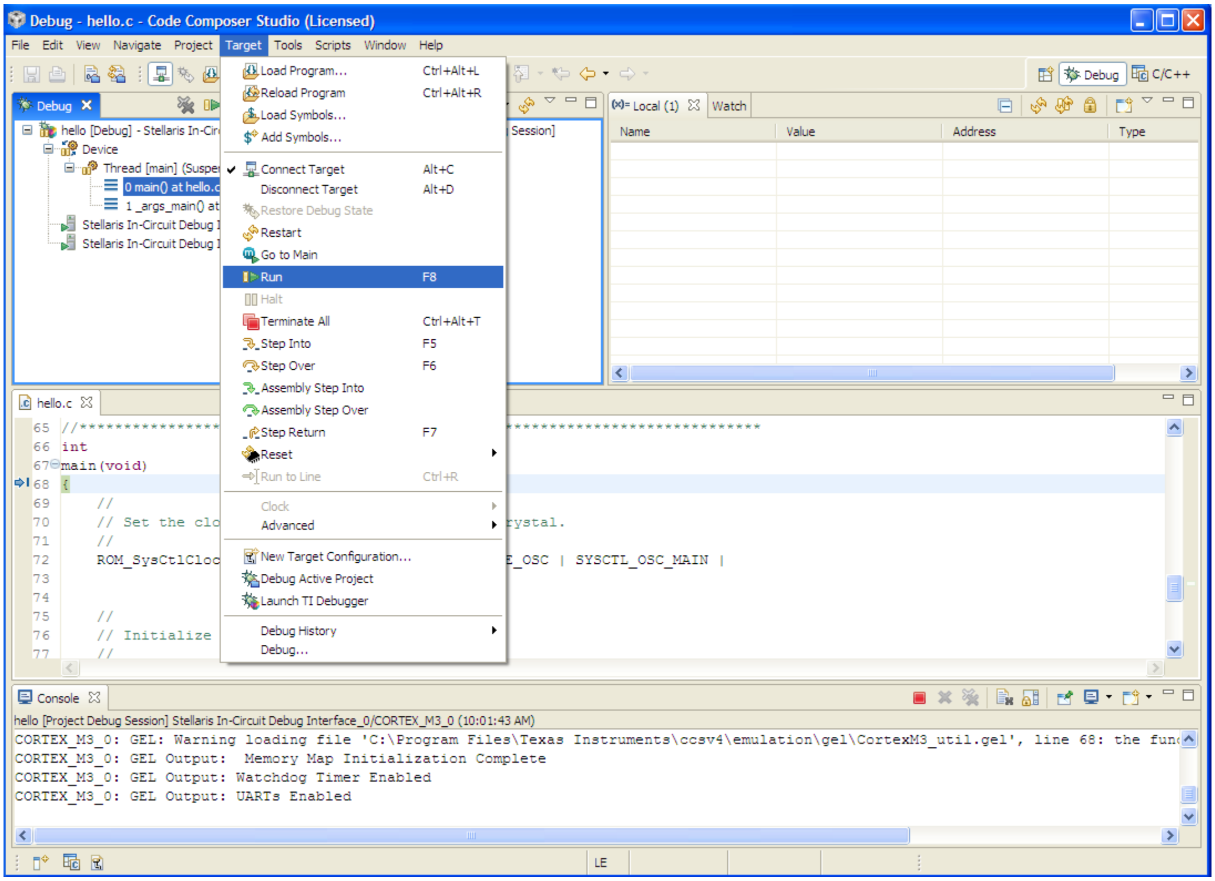
3. Click on the Target pull-down menu and select “Debug Active Project.” Alternatively, you can simply click the “Debug Launch” icon on the toolbar.



```
ompiler/tms470/bin/hex470.
exe" "C:/Program
Files/Texas
Instruments/ccsv4/utils/t
iobj2bin/mkhex4bin.exe"
' '
Build complete for
project hello
```

0 errors, 0 warnings, 0 infos

4. The Code Composer Studio debugger automatically connects to your evaluation board, programs the Flash memory, and runs to the beginning of the `main()` function. From here, you can examine and modify memory, program variables and processor registers, set breakpoints, step, and perform other typical debugging activities. To run the program, select “Run” from the Target pull-down menu.



```
//****************                               *****************************
int
main(void)
{
    //
    // Set the clo                                rystal.
    //
    ROM_SysCtlCloc                                _OSC | SYSCTL_OSC_MAIN |


    //
    // Initialize
    //
```

```
hello [Project Debug Session] Stellaris In-Circuit Debug Interface_0/CORTEX_M3_0 (10:01:43 AM)
CORTEX_M3_0: GEL: Warning loading file 'C:\Program Files\Texas Instruments\ccsv4\emulation\gel\CortexM3_util.gel', line 68: the func
CORTEX_M3_0: GEL Output:  Memory Map Initialization Complete
CORTEX_M3_0: GEL Output: Watchdog Timer Enabled
CORTEX_M3_0: GEL Output: UARTs Enabled
```

Step 7: Build and Run Additional Example Programs

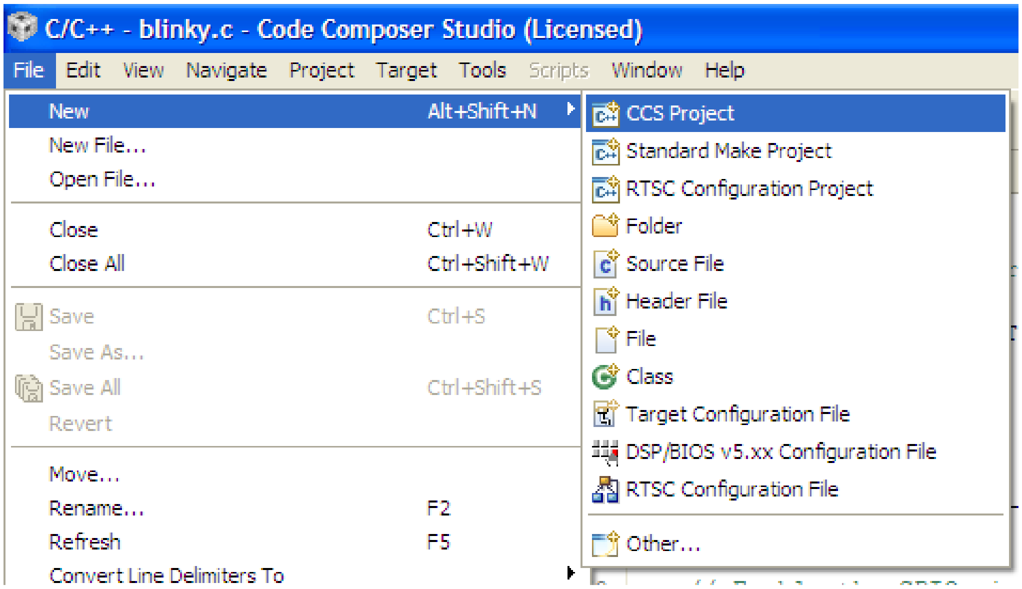
There are several additional example projects listed in the workspace. If you would like to build another example project, follow the above instructions. The quickstart application that came preloaded on the evaluation board is the qs-xxxxx project listed with the examples.

Creating a New Project

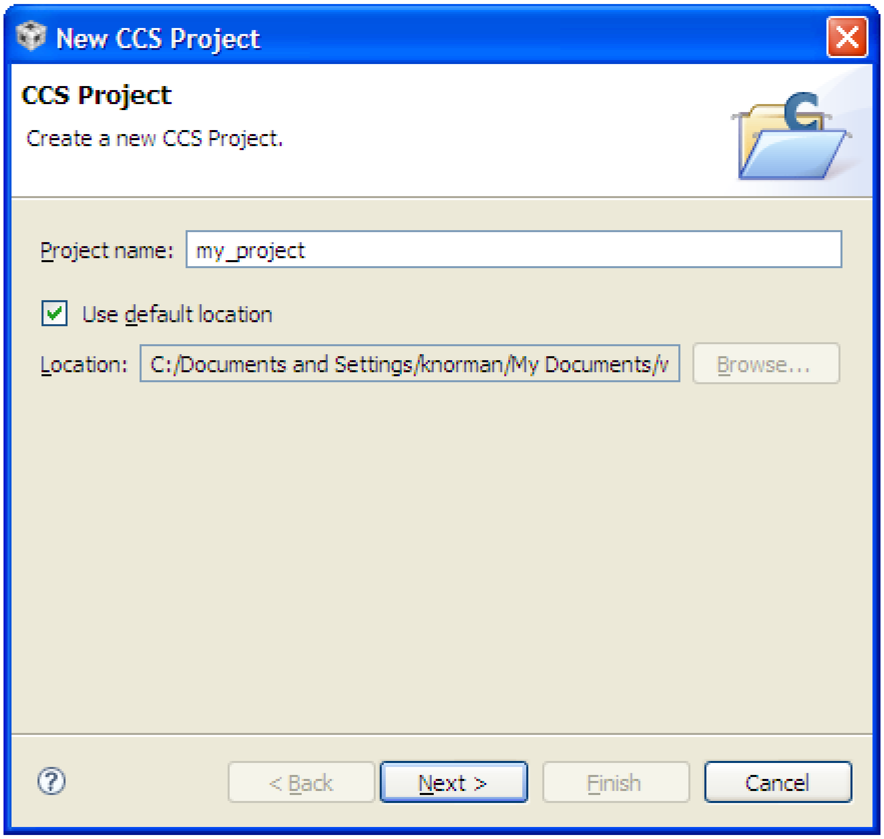
Once you have gone through the StellarisWare example applications, you may want to create your own project to start development. While you can always start with an existing, simple project, sometimes you may want to start fresh.

The example provided below creates a fresh project, copies code from an existing project, and builds the new project.

1. To add a new project to your workspace, go to File > New > CCS Project.



2. Code Composer Studio prompts you with a dialog box asking you to name the project and specify the location of the project. Name your project, specify the location, and click “Next.”



CCS Project

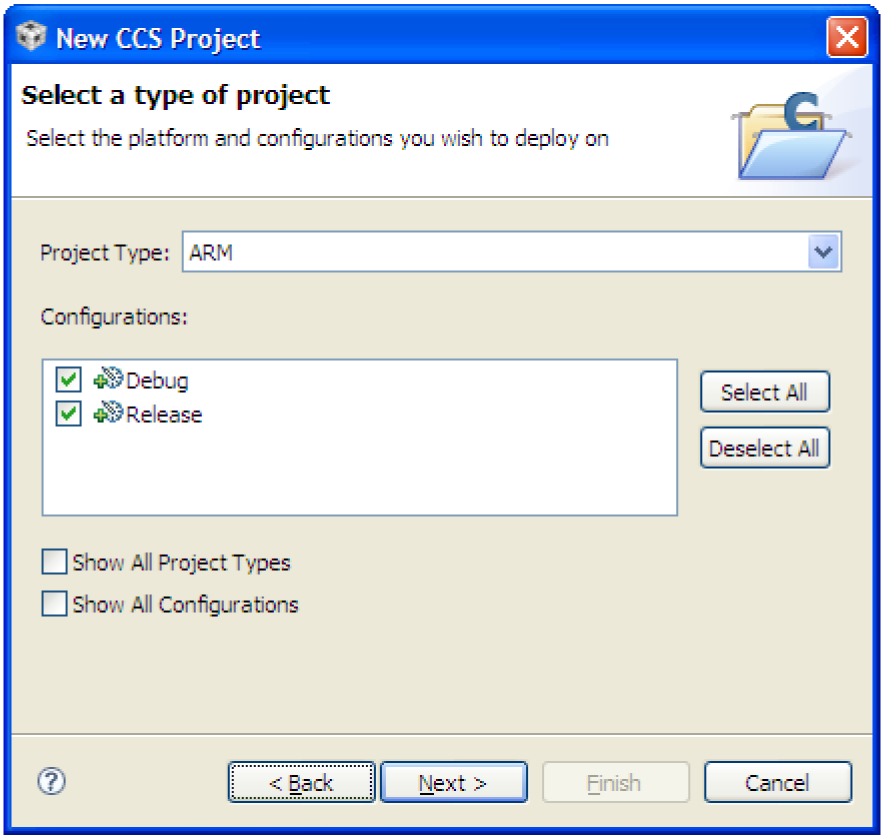
Create a new CCS Project.

Project name: my_project

☑ Use default location

Location: C:/Documents and Settings/knorman/My Documents/v

3. The next dialog box asks for the project type and configurations. Select “ARM” as the project type, check both the “Debug” and “Release” configurations, and click “Next.”



New CCS Project

Select a type of project

Select the platform and configurations you wish to deploy on

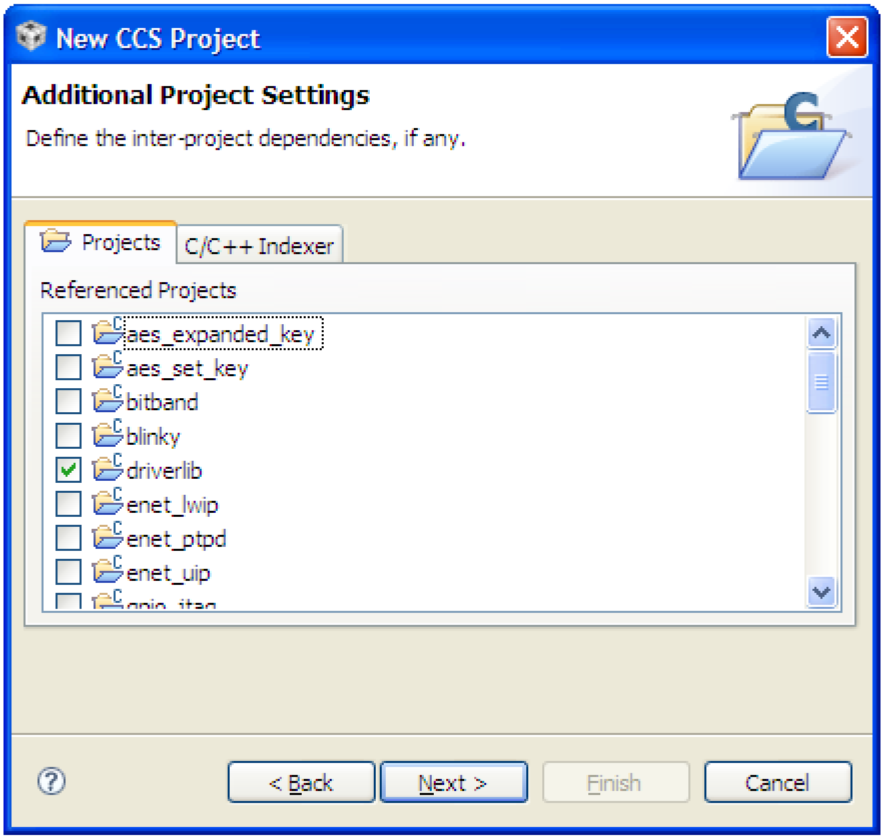
Project Type: ARM

Configurations:

☑ Debug
☑ Release

□ Show All Project Types
□ Show All Configurations

4. The next dialog box allows you to define any inter-project dependencies. If your project will be using driverlib, usblib, or grlib, now is a good time to define that dependency.



New CCS Project

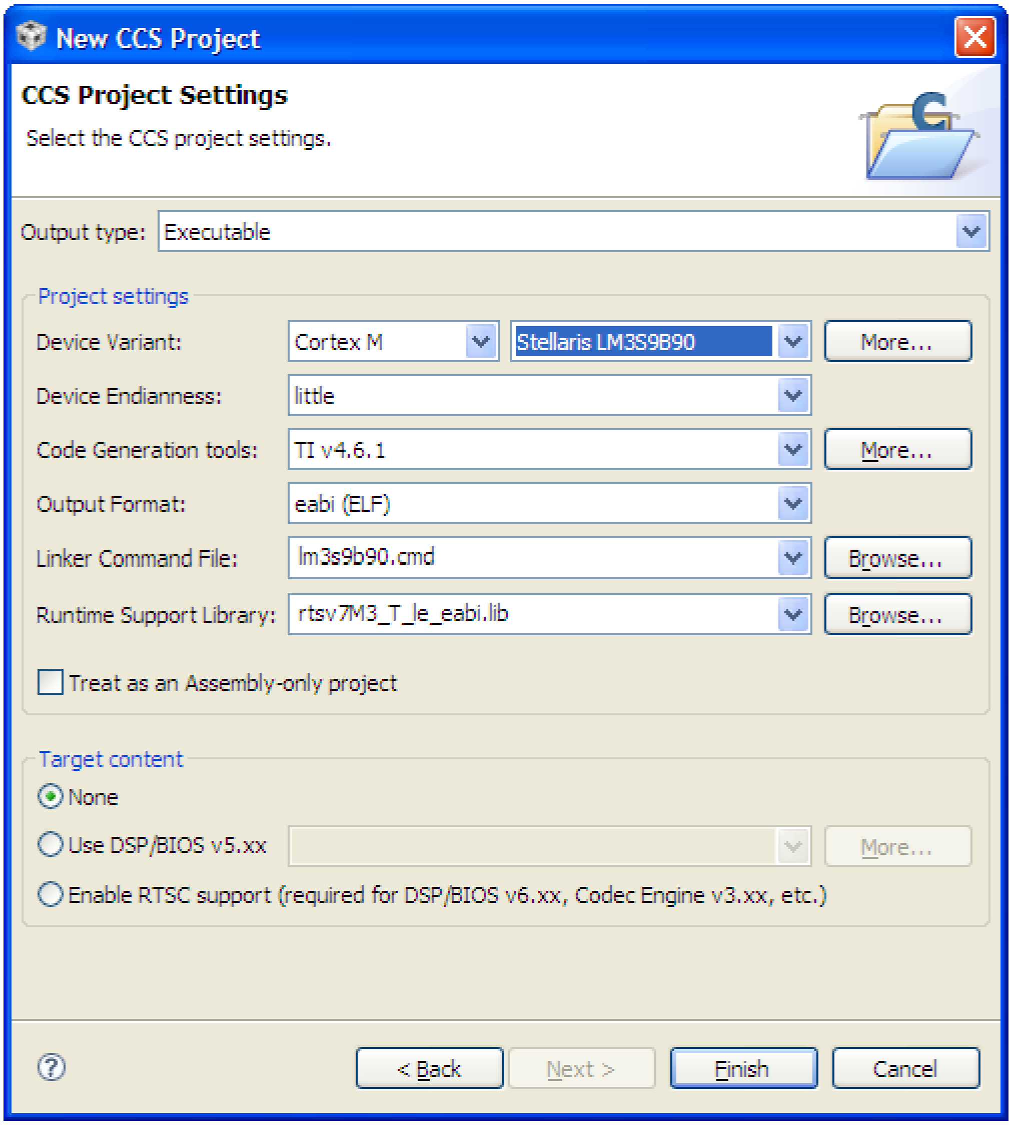
Additional Project Settings

Define the inter-project dependencies, if any.

Referenced Projects

- □ aes_expanded_key
- □ aes_set_key
- □ bitband
- □ blinky
- ☑ driverlib
- □ enet_lwip
- □ enet_ptpd
- □ enet_uip
- □ gpio_jtag

5. On the next dialog, select the appropriate Stellaris device, little endian, the TI code generation tool, ELF output format, and the rtsv7M3_T_le_eabi.lib runtime library. Click “Finish.”



CCS Project Settings

Select the CCS project settings.

Output type: Executable

Project settings

Device Variant: Cortex M | Stellaris LM3S9B90

Device Endianness: little

Code Generation tools: TI v4.6.1

Output Format: eabi (ELF)

Linker Command File: lm3s9b90.cmd

Runtime Support Library: rtsv7M3_T_le_eabi.lib

□ Treat as an Assembly-only project

Target content

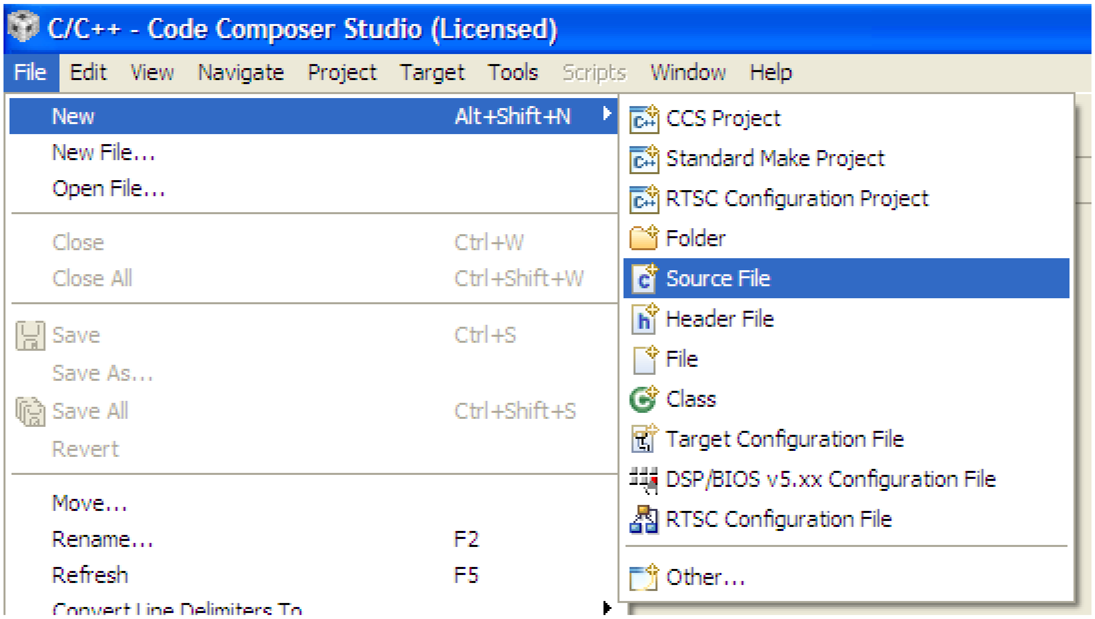
● None

○ Use DSP/BIOS v5.xx

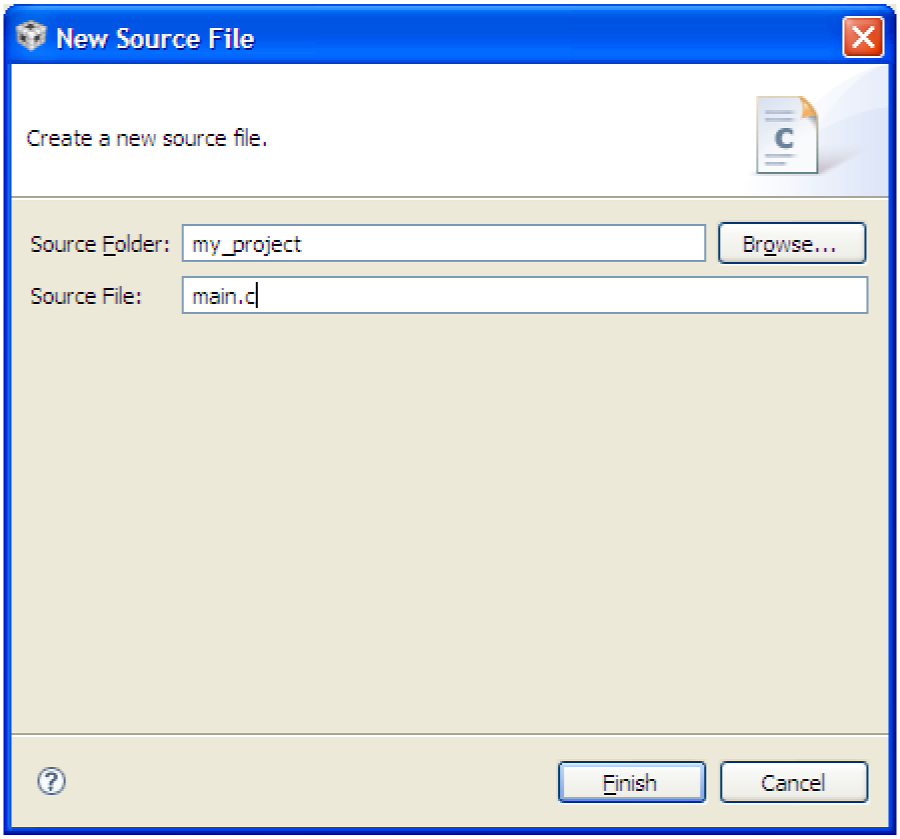
○ Enable RTSC support (required for DSP/BIOS v6.xx, Codec Engine v3.xx, etc.)

6. A new project has now been created, but the project does not include any source code. The next step is to add some startup code to the project. Copy the startup_ccs.c file from an existing example directory in StellarisWare (such as \StellarisWare\boards\<board>\blinky) to your new project directory created in step 2 above.

7. Create a new C source file by going to File > New > Source File.



8. In the dialog box that pops up, name the file main.c and click “Finish.”

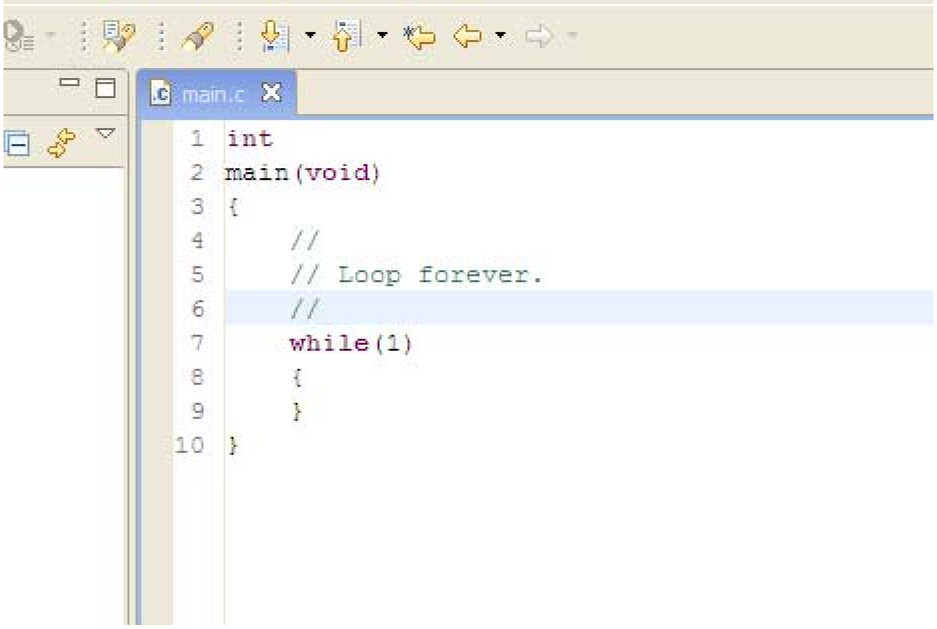


Create a new source file.

Source Folder: my_project

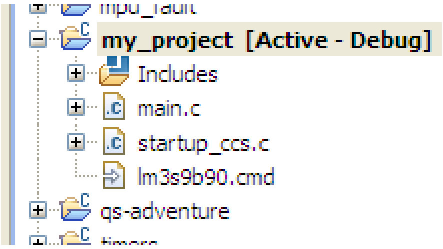
Source File: main.c

9. Add a main() function to main.c. Save the file. See the screenshot below.



```
int
main(void)
{
    //
    // Loop forever.
    //
    while(1)
    {
    }
}
```

10. Both files, main.c and startup_ccs.c, should have automatically been added to the project. If not, right-click the project, select “Add Files to Project” and browse to the files.



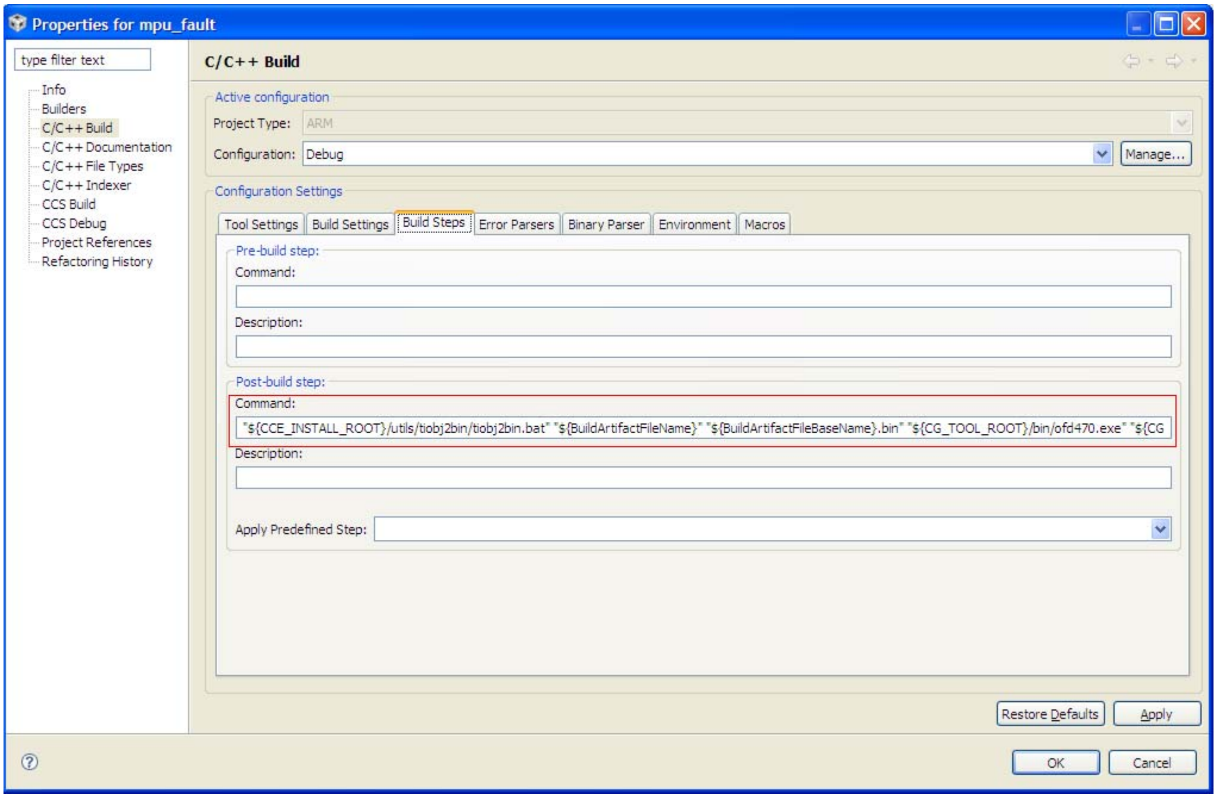
```
mpu_fault
my_project [Active - Debug]
    Includes
    main.c
    startup_ccs.c
    lm3s9b90.cmd
qs-adventure
timers
```

11. Build the new project by selecting Project > Rebuild Active Project.

With your project created, all you really need to do is add your own code. Use the existing StellarisWare board examples as a reference.

NOTE: To set up your project to output a binary file (.bin) to be used with tools such as LM Flash Programmer, copy the post-build step from an existing StellarisWare example project. See the following screenshot.

QUICKSTART – CODE COMPOSER STUDIO™



Project Type: ARM

Configuration: Debug

Post-build step:

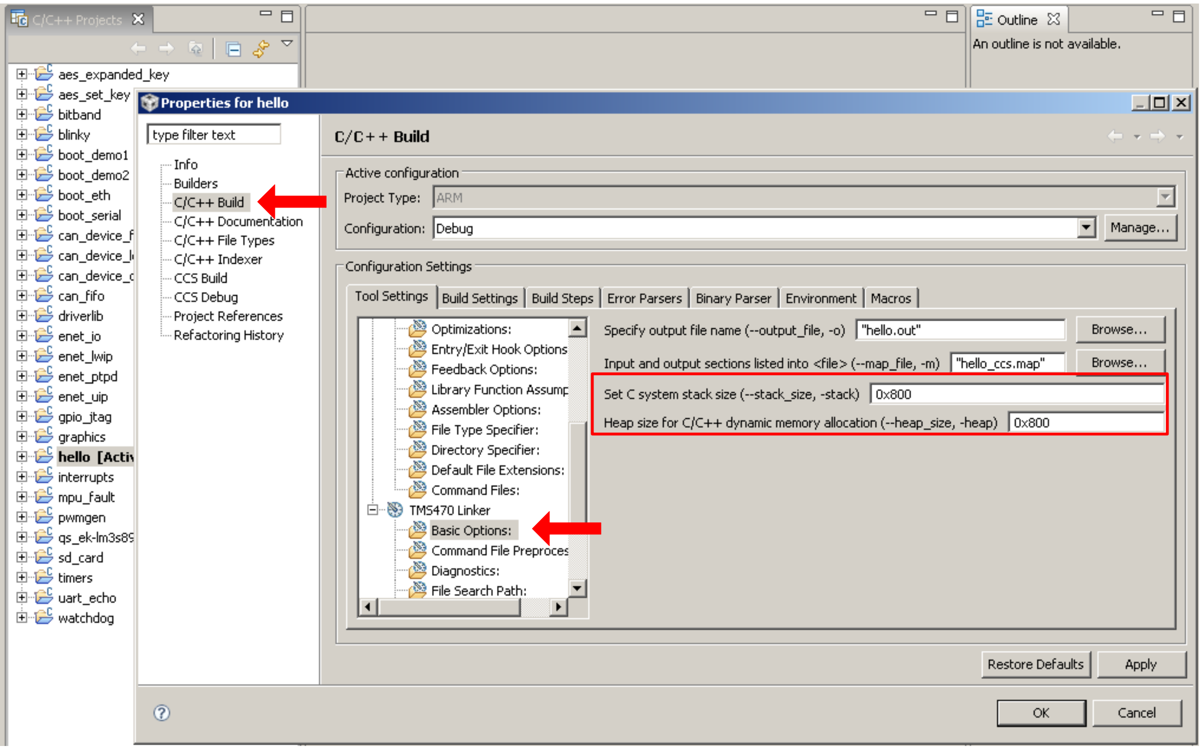
Command:

```
"${CCE_INSTALL_ROOT}/utils/tiobj2bin/tiobj2bin.bat" "${BuildArtifactFileName}" "${BuildArtifactFileBaseName}.bin" "${CG_TOOL_ROOT}/bin/ofd470.exe" "${CG
```

Description:

Apply Predefined Step:

NOTE: If you use standard C I/O functions such as `printf` or `fread`, you must increase the stack and heap allocations for your project. Right-click on your project and then select Properties to open a dialog containing the project options. Select C/C++ Build in the pane on the left side of the dialog box. This displays the build options. Under the Linker node in the Tool Settings tab, select Basic Options from the scroll list. Enter 0x800 for both the system stack and the heap. See the following screenshot.



Project Type: ARM

Configuration: Debug

Specify output file name (--output_file, -o) "hello.out"

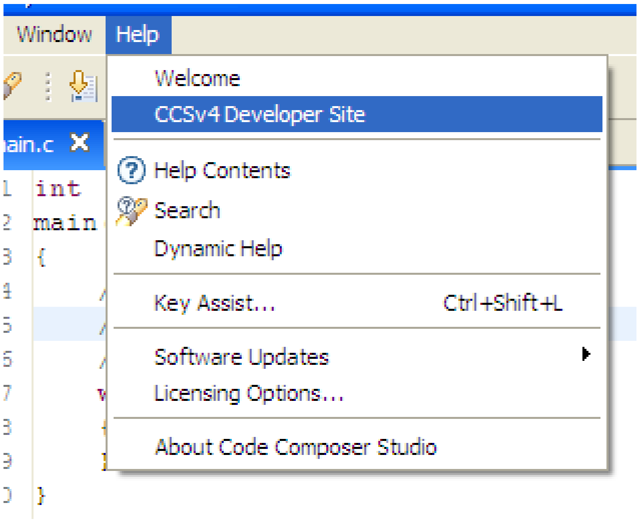
Input and output sections listed into <file> (--map_file, -m) "hello_ccs.map"

Set C system stack size (--stack_size, -stack) 0x800

Heap size for C/C++ dynamic memory allocation (--heap_size, -heap) 0x800

Conclusion

You have now installed the Code Composer Studio development tools and used them to build and load an example application on your Stellaris Evaluation Board. From here, you can experiment with the debugger or start creating your own application using the example projects as a reference. For further information on Code Composer Studio, go to the CCS Developer Site.



QUICKSTART – CODE COMPOSER STUDIO™

References

The following references are included on the Stellaris Evaluation Kit Documentation and Software DVD and are also available for download at www.ti.com/stellaris:

- *Stellaris Evaluation Kit User's Manual*
- *StellarisWare Software*, Order Number SW-LM3S
- *StellarisWare Peripheral Driver Library User’s Guide*, Order Number SW-DRL-UG

In addition, the following website may be useful:

- Code Composer Studio website at http://www.ti.com/ccstudio

Texas Instruments
108 Wild Basin Rd., Suite 350
Austin, TX 78746
http://www.ti.com/stellaris

IMPORTANT NOTICE

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		**Applications**	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DLP® Products	www.dlp.com	Communications and Telecom	www.ti.com/communications
DSP	dsp.ti.com	Computers and Peripherals	www.ti.com/computers
Clocks and Timers	www.ti.com/clocks	Consumer Electronics	www.ti.com/consumer-apps
Interface	interface.ti.com	Energy	www.ti.com/energy
Logic	logic.ti.com	Industrial	www.ti.com/industrial
Power Mgmt	power.ti.com	Medical	www.ti.com/medical
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
RFID	www.ti-rfid.com	Space, Avionics & Defense	www.ti.com/space-avionics-defense
RF/IF and ZigBee® Solutions	www.ti.com/lprf	Video and Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless-apps

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265