CodeWarrior Development Studio for Advanced Packet Processing Simulator Getting Started Guide

Document Number: CWAPPSIMGS
Rev. 10.2, 01/2016

Contents

Section number **Title** **Page**

Chapter 1 Introduction

This document explains how to install the CodeWarrior Development Studio for Advanced Packet Processing (APP). Additionally, it describes how to use the CodeWarrior software to create, build, and debug a simple project.

This chapter contains:

- System requirements

1.1 System requirements

The table below lists the operating platform requirements for installing CodeWarrior Development Studio for Advanced Packet Processing software:

Table 1-1. System requirements

Processor	Windows® OS: Intel® Pentium® 4 processor, 2 GHz or faster, Intel® Xeon™, Intel® Core™, AMD Athlon™ 64, AMD Opteron™, or later Linux® OS: 1.8 GHz Intel® Pentium® class processor (or better). 64-bit host OS required.
Hardware	• CD-ROM drive for CD installation • Microsoft® Mouse compliant pointing device • Internet connectivity for web downloads and update access
Operating System	• Windows® 7 SP1 (32-bit and 64-bit) • Windows® 8 U1 (64-bit) • Windows® 10 (64-bit) • Ubuntu 12.04 (64-bit) • Ubuntu 14.04 (64-bit) • Fedora 20 (64-bit) • Fedora 21 (64-bit) • Open SUSE 13.2 (64-bit) • RedHat Enterprise Linux 5.8 (64-bit) • RedHat Enterprise Linux / CentOS 6.5 (64-bit) • RedHat Enterprise Linux / CentOS 6.6 (64-bit) • Mint 17.3 (64-bit)
Memory	At least 2 GB of RAM At least 3 GB of free disk space

Table 1-1. System requirements

	When the Simulator is installed on the same machine as the CodeWarrior software, the system requirements will change as follows: • Minimum RAM memory: 4 GB • Recommended RAM memory: 6 GB

NOTE

For more information regarding the simulator system requirements, see *QorIQ LS Series Simulator User Guide*.

NOTE

CodeWarrior installation on 64-bit Linux hosts requires presence of dependent 32-bit libraries in the system. For supported distributions, the installer detects missing dependencies and provides options to install them automatically. It is recommended (although not required) that all product maintenance operations are performed with administrative privileges. When running the installer in GUI mode with admin privileges, please use the graphical sudo command recommended by your Linux distribution (ie. gksu/gksudo).

Chapter 2
Installing CodeWarrior software and creating, building, and debugging a bareboard project

This section explains how to install the CodeWarrior Development Studio for Advanced Packet Processing software. Additionally, it describes how to use the software to create, build, and debug a demonstration multi-core processor project.

This chapter explains:

- Installing and registering CodeWarrior software
- Working with bareboard application

2.1 Installing and registering CodeWarrior software

This section provides the steps required to install the CodeWarrior software on both Windows and Linux operating systems.

Microsoft® Windows OS installation:

Administrator rights are required to install CodeWarrior on Microsoft Windows 7 operating system, since the installer copies files into the System and Program Files folders. CodeWarrior service packs are installed with the **Eclipse Updater**. The updater must also be run using administrator rights. To start the Eclipse Updater, select '**Help > Install new software**' in the menu.

The default CodeWarrior installation folder is `C:\Freescale`. To protect against malware, Windows operating system do not allow normal processes to change files in the *Program Files* folder. Therefore, you must have the administrator rights to install and run the CodeWarrior software from this location.

Your project workspace needs to be setup in any folder that you can fully access.

1. Run the installer.

   The install wizard appears.

2. Follow the wizard’s on-screen instructions to install the CodeWarrior software and the desired GCC toolchains.

   When installation completes, the **InstallShield Wizard Completed** page appears.

3. Click **Finish**.

   The wizard closes. A browser starts and displays the Documentation page. This page contains tabs that group the CodeWarrior documentation into categories.

   You have successfully installed CodeWarrior Development Studio for Advanced Packet Processing.

Linux OS installation

The CodeWarrior installer must run from a root account to install any required dependencies. If running from a non-root account, the installer will warn you regarding any un-fulfilled product dependencies but will continue to run. You will have to manually add the dependencies by referring to README.txt in the layout.

CodeWarrior service packs are installed with the **Eclipse Updater**. To start the Eclipse Updater select '**Help > Install new software**' in the menu.

Eclipse needs read/write access to the installation folder. Make sure the eclipse installation folder has the appropriate permissions for all users. Make sure your project workspace has read and write permissions.

1. If you have a CodeWarrior Development Studio installation CD, insert it into the Linux host computer’s CD-ROM drive and mount the CD-ROM media on the Linux file system.
2. On the host computer, open a new terminal window.

   A shell session starts.

3. If you have an instalation CD, change the working directory to the CD-ROM mount directory.

   NOTE

   See README.txt file in the mount directory. This file contains installation instructions of different Linux distributions.

4. If you have an archive instead of an installation CD, unpack it and change the working directory to the unpacked folder.
5. If you have logged in with a non-root account, issue the command `xhost +`.
6. Issue the command `./setuplinux.sh`.

The install wizard starts and displays its welcome page.

7. Follow the wizard's on-screen instructions to install the CodeWarrior software and the desired GCC toolchains.

   When the software installation completes, the wizard displays its installation summary page.

8. Click **Finish**.

   The CodeWarrior software installation is now complete.

Licensing

Both Windows and Linux installers generate an Evaluation license for the Architect edition that is valid for 15 days. The license is generated regardless of other product versions that may have been installed on the same host. The certificate generated is valid only on the machine where the product has been installed. During the installation, the user is informed that the evaluation key is node-locked and number of days the key is valid. Node-locking element is by default Ethernet ID. In case no Ethernet ID is found, the disk ID is used.

NOTE

For details regarding the simulator license setup, see *Simulator License Setup Guide*.

2.2 Working with bareboard application

This section explains the steps required to create, build, and debug an APP bareboard project.

2.2.1 Creating, building bareboard project

To create and build a bareboard simulator project, follow the steps listed below:

1. Launch the simulator.

   NOTE

   The simulator runs only on a Linux x86 64-bit host.

   If you are using a Windows-hosted CodeWarrior, follow the steps below:
   If you have installed the Simulator package on a remote Linux64 host during the installation of the CodeWarrior software, directly skip to Step b.

a. Get simulator from layout if you have copied the Simulator package on your local machine during installation.

```
<Layout>/Common/CCSSim/LS_SIM_<version>.tgz
```

Move the file to the `Linux x86_64` machine and unzip it.

b. In a console, navigate to the `linux64` folder and depending on the simulation variant you want to use, run the following command:

```
./start_sim_bare_metal -simconfig=ls2085a
 or
./start_sim_bare_metal -simconfig=ls2045a
```

c. Wait for the simulator to start (look for `Simulation thread for device 0 started` message).

If you are using a Linux-hosted CodeWarrior, follow the steps below:

a. On the Linux machine hosting CodeWarrior software, simulator is already unpacked under the `<Layout>/Common/CCSSim` folder.

b. In a console, navigate to the `<Layout>/Common/CCSSim` folder and depending on the simulation variant you want to use, run the following command.

```
./start_sim_bare_metal -simconfig=ls2085a
or
./start_sim_bare_metal -simconfig=ls2045a
```

c. Wait for the simulator to start (look for "`C?:D0:N0:0:Connect: Connection 0 to reuse Device 0, use count is 1 C?:D0:N0:1:Connect: Connection 0 to Device 0 CONFIGURATION:* waiting for Clients (On Port=40969) ...`" message).

2. Launch the CodeWarrior IDE.

On Windows:

- Select **Start > All Programs > Freescale CodeWarrior > CW4NET {$CW Version} > CodeWarrior for APP**.

  The **Workspace Launcher** dialog appears.

On Linux:

- Open a new terminal window and change the working directory to: `{$CWInstall}/CW_APP/`, where `{$CWInstall}` is the path to your CodeWarrior installation.

- Issue the command `./fsl_eclipse.sh`

  The **Workspace Launcher** dialog appears.

a. If you wish to change the location of your project's workspace, click **Browse** to select the new path.

The **Select Workspace Directory** dialog appears.

b. Select the required folder. Alternatively, to create a new workspace directory:

- On Windows, click **Make New Folder**.
- On Linux, click **Create Folder**.

c. Click **OK**.

The **Select Workspace Directory** dialog closes.

d. Click **OK** to store the project at the specified location.

The CodeWarrior IDE launches and displays the **Welcome** page.

NOTE

The **Welcome** page is displayed when the CodeWarrior IDE is launched for the first time. You can always return to the Welcome page by selecting **Help > Welcome** from the CodeWarrior IDE menu bar.

e. Click **Go to the workbench**, on the Welcome page.

The workbench window appears.

3. Create a new project.

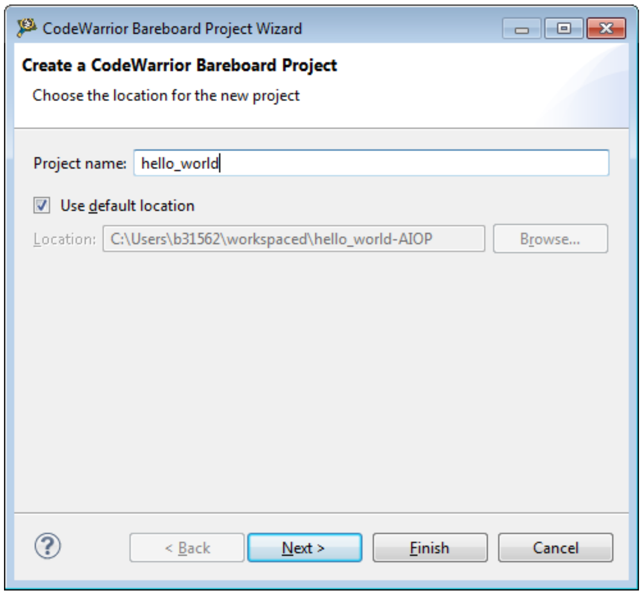
a. From CodeWarrior IDE menu bar, select **File < New < CodeWarrior Bareboard Project Wizard**.

The **CodeWarrior Bareboard Project Wizard** launches and the **Create a CodeWarrior Bareboard Project** page appears.

b. In the **Project name** text box, type `hello_world`.

NOTE

The **Location** text box shows the default workspace location. To change this location, deselect the **Use default location** checkbox and click **Browse** to select a new location. Ensure to append the name of your project to the end of selected path. Until you do so, the message `"An existing directory cannot be specified for the project location."` appears at the top of the current wizard page, and you cannot go to the next page.


CodeWarrior Bareboard Project Wizard

Create a CodeWarrior Bareboard Project

Choose the location for the new project

Project name: hello_world

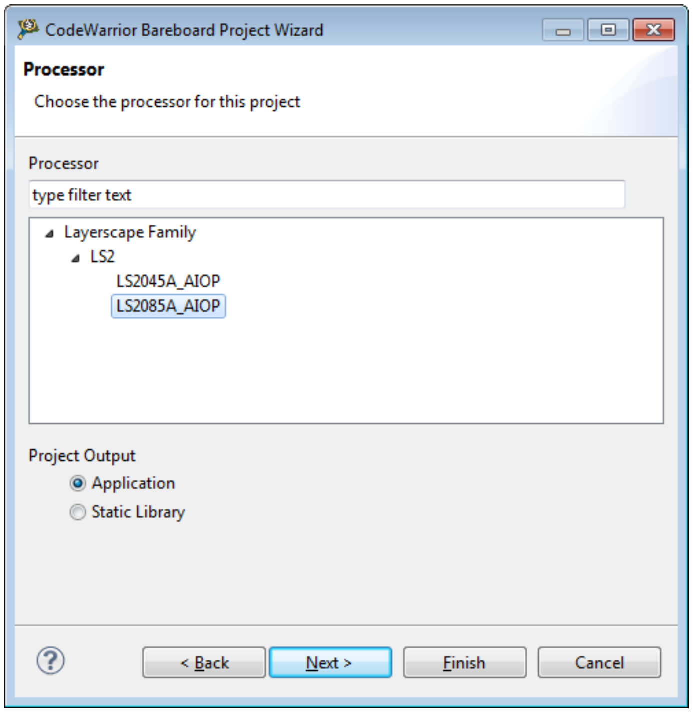
[✓] Use default location

Location: C:\Users\b31562\workspaced\hello_world-AIOP

Figure 2-1. Create a CodeWarrior Bareboard Project page

c. Click **Next**.

The **Processor** page appears.


CodeWarrior Bareboard Project Wizard

Processor

Choose the processor for this project

Processor

type filter text

Layerscape Family
LS2
LS2045A_AIOP
LS2085A_AIOP

Project Output

[✓ Application]
Static Library

Figure 2-2. Processor page

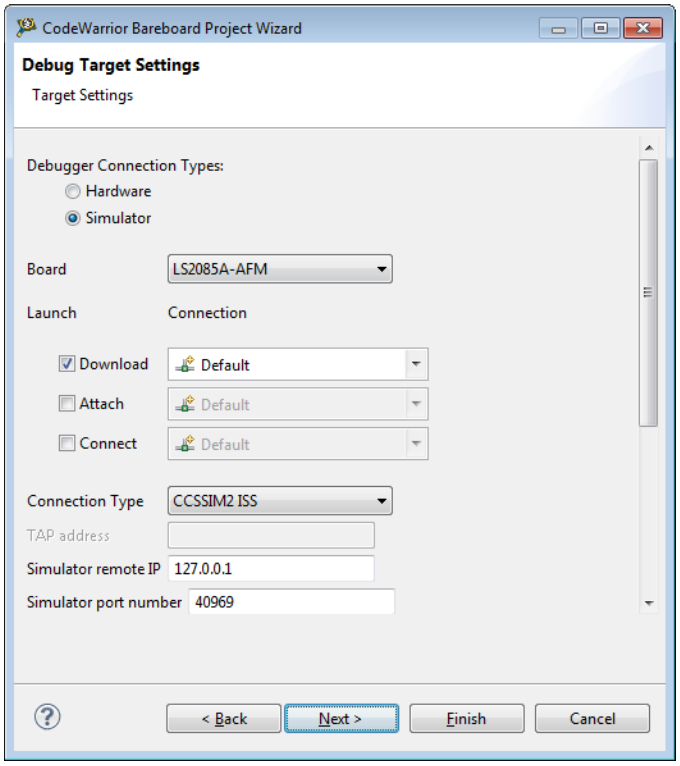
d. Select a target processor for the new project, from the **Processor** list.
e. Select **Application** from the **Project Output** group, to create an application with .elf extension, that includes information required to debug the project.
f. Click **Next**.

The **Debug Target Settings** page appears.

g. Select a connection type option, from the **Debugger Connection Types** group.
h. Select the simulator, you plan to use, from the **Board** dropdown list.

NOTE

Hardware or simulators that support the target processor selected on the **Processors** page are only available for selection.



CodeWarrior Bareboard Project Wizard

Debug Target Settings

Target Settings

Debugger Connection Types:
- ○ Hardware
- ● Simulator

Board: LS2085A-AFM

Launch	Connection
☑ Download	Default
☐ Attach	Default
☐ Connect	Default

Connection Type: CCSSIM2 ISS

TAP address:

Simulator remote IP: 127.0.0.1

Simulator port number: 40969

Figure 2-3. Debug Target Settings

i. Select the launch configurations and the corresponding connection to be included in the project.
j. Select the interface to communicate with the hardware, from the **Connection Type** drop-down list.
k. Enter the IP address of the TAP device in the **TAP address** text box. This option is currently disabled and cannot be edited.
l. Specify the IP address of the remote Linux 64-bit host, CCSSIM2 is started on, in the **Simulator remote IP** text box.
m. Specify the port number that the debugger will use to communicate with the simulator launched on the remote Linux host, in the **Simulator port number** text box.
n. Click **Next**.

The **Build Settings** page appears.

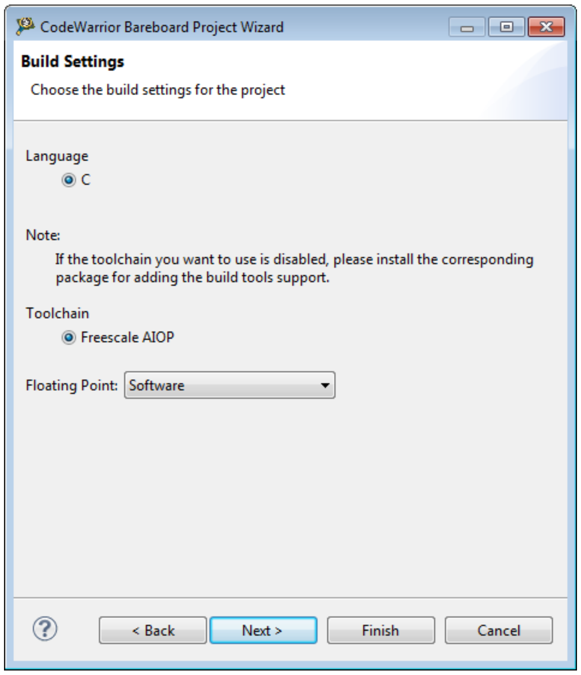
o. Select a programming language, from the **Language** group.

   The language you select determines the libraries that are linked with your program and the contents of the main source file that the wizard generates.

p. Select the processor used by the new project, from the **Build Tools Architecture** group.

q. Select a toolchain from the **Toolchain** group.

   Selected toolchain sets up the default compiler, linker, and libraries used to build the new project. Each toolchain generates code targeted for a specific platform.


CodeWarrior Bareboard Project Wizard

Build Settings

Choose the build settings for the project

Language

[✓ C]

Note:

If the toolchain you want to use is disabled, please install the corresponding package for adding the build tools support.

Toolchain

[✓ Freescale AIOP]

Floating Point: Software

Figure 2-4. Build Settings page

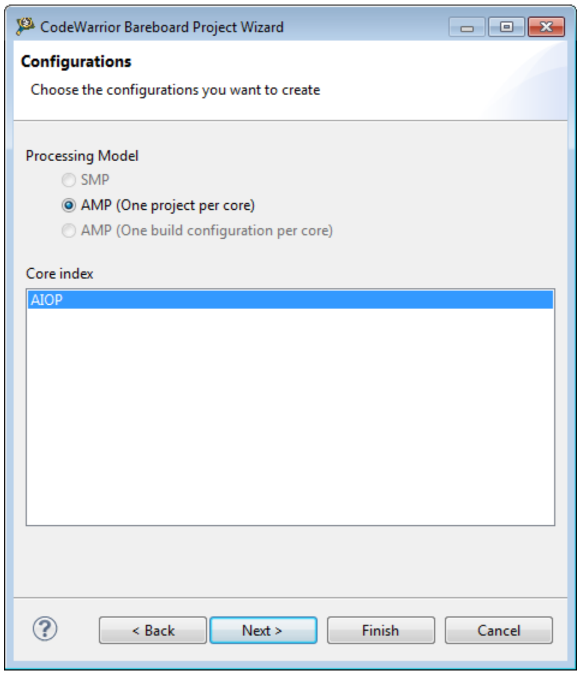
r. Select an option from the **Floating Point** drop-down list, to prompt the compiler to handle the floating-point operations by generating instructions for the selected floating-point unit.

s. Click **Next**.

   The **Configurations** page appears.

t. Select a processing model option from the **Processing Model** group.

u. Select the processor core that executes the project, from the **Core index** list.


CodeWarrior Bareboard Project Wizard

Configurations

Choose the configurations you want to create

Processing Model

- ○ SMP
- ● AMP (One project per core)
- ○ AMP (One build configuration per core)

Core index

AIOP

Figure 2-5. Configurations page

v. To debug programs on a multi-core processor, you can select different core index for creating separate projects for each core.

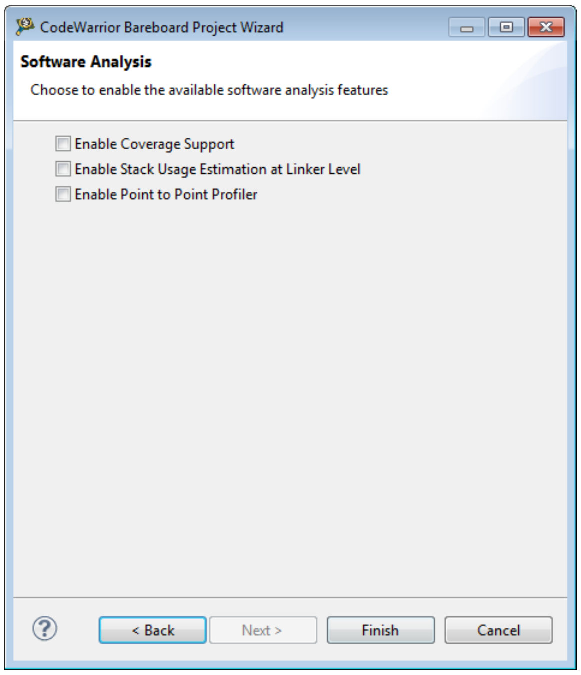
w. Click **Next**.

The **Software Analysis** page appears.

x. If you plan to analyze the source code, that runs on a AIOP or MC core, check **Enable Coverage Support**.

y. If you plan to generate the stack usage information in the linker map file, check **Enable Stack Usage Estimation at Linker Level**.

z. If you plan to enable performance gathering, check **Enable Point to Point Profiler**.



CodeWarrior Bareboard Project Wizard

Software Analysis

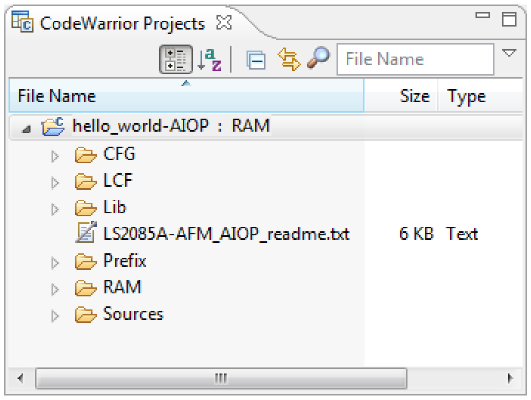
Choose to enable the available software analysis features

- [ ] Enable Coverage Support
- [ ] Enable Stack Usage Estimation at Linker Level
- [ ] Enable Point to Point Profiler

Figure 2-6. Software Analysis

aa. Click **Finish**.

The wizard creates a simulator project according to your specifications. You can access the project from the **CodeWarrior Projects** view on the Workbench.



CodeWarrior Projects

File Name	Size	Type
hello_world-AIOP : RAM		
CFG		
LCF		
Lib		
LS2085A-AFM_AIOP_readme.txt	6 KB	Text
Prefix		
RAM		
Sources		

Figure 2-7. CodeWarrior Projects view

4. Build the program.
   a. Select the newly created project in the **CodeWarrior Projects** view.
   b. Select **Project > Build Project** to build the project. Alternatively, right-click the project in the **CodeWarrior Projects** view and select **Build Project** from the context-menu that appears.

   The IDE compiles the project's source code files and links resulting object code into an ELF-format executable file.

2.2.2 Debugging bareboard project

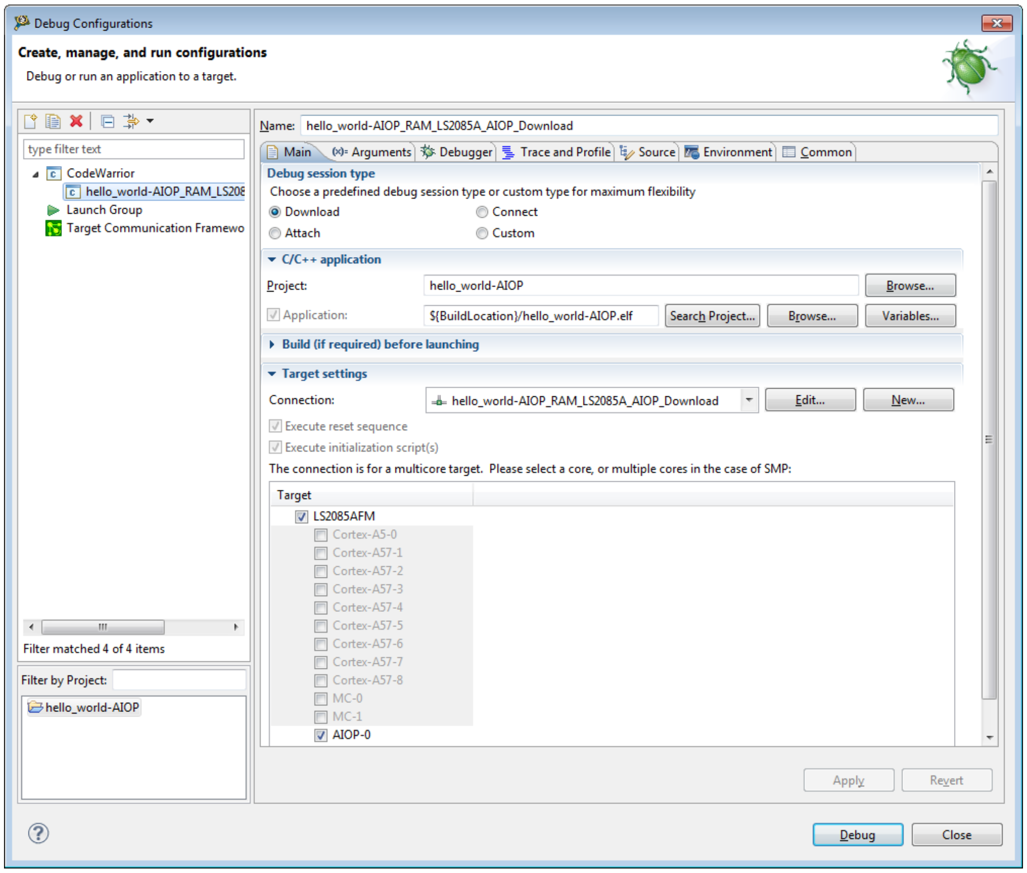
To debug a CodeWarrior bareboard project:

1. Prepare to debug the program.
   a. Launch the CodeWarrior IDE.
   b. Select **Run > Debug Configurations**.

      The **Debug Configurations** dialog appears.

   c. Expand **CodeWarrior**.

      Select the launch configuration with the name `${ProjectName}`.


Debug Configurations

Create, manage, and run configurations

Debug or run an application to a target.

type filter text

- CodeWarrior
  - hello_world-AIOP_RAM_LS208
- Launch Group
- Target Communication Framewo

Filter matched 4 of 4 items

Filter by Project:

- hello_world-AIOP

Name: hello_world-AIOP_RAM_LS2085A_AIOP_Download

Main | Arguments | Debugger | Trace and Profile | Source | Environment | Common

Debug session type

Choose a predefined debug session type or custom type for maximum flexibility

[✓ Download] Connect
Attach Custom

C/C++ application

Project: hello_world-AIOP

[✓] Application: ${BuildLocation}/hello_world-AIOP.elf

Build (if required) before launching

Target settings

Connection: hello_world-AIOP_RAM_LS2085A_AIOP_Download

[✓] Execute reset sequence

[✓] Execute initialization script(s)

The connection is for a multicore target. Please select a core, or multiple cores in the case of SMP:

Target
[✓] LS2085AFM
[ ] Cortex-A5-0
[ ] Cortex-A57-1
[ ] Cortex-A57-2
[ ] Cortex-A57-3
[ ] Cortex-A57-4
[ ] Cortex-A57-5
[ ] Cortex-A57-6
[ ] Cortex-A57-7
[ ] Cortex-A57-8
[ ] MC-0
[ ] MC-1
[✓] AIOP-0

Figure 2-8. Launch configuration

   d. Click the **Edit** from the **Connection** group.

      The **Properties for <connection>** dialog appears.

   e. From the **Target** drop-down list to select your target.
   f. Select or create a Target configuration.

- You can select one of the predefined target configurations. For example, you can select LS2085A_Simulator, LS1043A_Simulator, if the CodeWarrior software and Simulator run on the same machine. This means, you can select the LS2085A target configuration if you use `./start_sim_bare_metal -simconfig=ls2085a`
- You can create a new target configuration if the CodeWarrior software and Simulator run on different machines. To create a new configuration, click **Add**.
  The **Target Connection Configurator** dialog appears.

NOTE

If you have selected a predefined target configuration, navigate to **step j. Click OK.**

g. Configure the **Connection Server** and **Port Number**.
   - New target configuration name - NewConfiguration(1).
   - IP address of the host where Simulator was started - 127.0.0.1
   - Default port number used by ccssim2 - 40969.
   - Timeout: 10 (default value)

h. Click **OK**.

   The **Target Connection Configurator** dialog closes.

i. Select the target connection configuration you just created.

j. Click **OK**.

   The **Target Connection Configurations** dialog closes.

k. Click the **Debugger** tab.

l. Select the core to debug (the default value should be "CortexA57#0").

m. Click the **Startup** tab.

n. Click **Apply** in the **Debug Configurations** dialog.

   The IDE saves your settings.

2. Debug the program.
   a. Click **Debug**.
   b. The IDE switches to the **Debug** perspective. The debugger downloads your program to the target board and halts execution at the first statement of `main()`.
   c. Click a thread in the **Debug** view.

      The program counter icon (on the marker bar) points to the next statement to be executed.

   d. In the **Debug** view, click **Step Over**.

The debugger executes the current statement and halts at the next statement.

3. Set breakpoint and execute the program up to the breakpoint.
    a. In the editor area, scroll to a statement.
    b. Double-click the marker bar next to the statement.

        A breakpoint indicator appears next to the statement.

    c. In the **Debug** view, click **Resume**.

        The debugger executes all statements up to, but not including the breakpoint statement.

4. Control the program:
    a. In the **Debug** view (top-left of perspective), click **Step Over**.

        The debugger executes the current statement and halts at the next statement.

    b. In the **Debug** view, click **Resume**.
    c. In the **Debug** view, click **Terminate**.

        The program terminates and the debug session ends.

5. Select **File > Exit**.

    The CodeWarrior IDE window closes.

How to Reach Us:

Home Page:
freescale.com

Web Support:
freescale.com/support

Document Number CWAPPSIMGS
Revision 10.2, 01/2016