Freescale Semiconductor
Application Note

Document Number: AN5048

Using an External GCC Toolchain with CodeWarrior for PowerArchitecture

1. Introduction

This document explains how to use an external GNU compiler collection (GCC) toolchain with CodeWarrior for PowerArchitecture. This process is only applicable to the Linux version of CodeWarrior.

This document provides steps to:

- Build the toolchain supplied with Freescale Linux SDK
- Customize a stationary Linux project to work with an SDK standalone toolchain
- Build the project using an external toolchain

Contents

2. Preliminary background

CodeWarrior for PowerArchitecture includes the Freescale PPC or GCC binary toolchain. If you are developing a Linux user space application with CodeWarrior, then you are recommended to use the toolchain supplied with the Freescale Linux SDK.

3. Installing SDK standalone toolchain

You can use the standalone toolchain provided in SDK to build a Linux user space application with CodeWarrior. To build and install the standalone toolchain with Yocto, perform these steps:

```
$ cd build_<machine>_release
$ bitbake fsl-toolchain
$ cd build_<machine>_release/tmp/deploy/sdk
$ ./fsl-networking-eglibc-<host-system>-<core>-toolchain-
<release>.sh
```

NOTE	The default installation path for the standalone toolchain is: `/opt/fsl-networking/`. You need to specify this path while installing the standalone toolchain. For additional information about building and installing the standalone toolchain with Yocto, see **SDK Knowledge Center**.

See Change toolchain for using SDK standalone toolchain as the default build tool in CodeWarrior.

4. Working with a PowerArchitecture Linux application project

This section contains the following subsections:

- Create a stationary project for Linux application
- Change toolchain
- Verify build settings
- Build project using an external toolchain

4.1. Create a stationary project for Linux application

To create a PowerArchitecture stationary project for Linux application, follow these steps:

1. Start CodeWarrior for PowerArchitecture.
2. Choose **File > New > CodeWarrior Linux Project Wizard** from the CodeWarrior IDE menu

bar. **CodeWarrior Linux Project Wizard** starts.

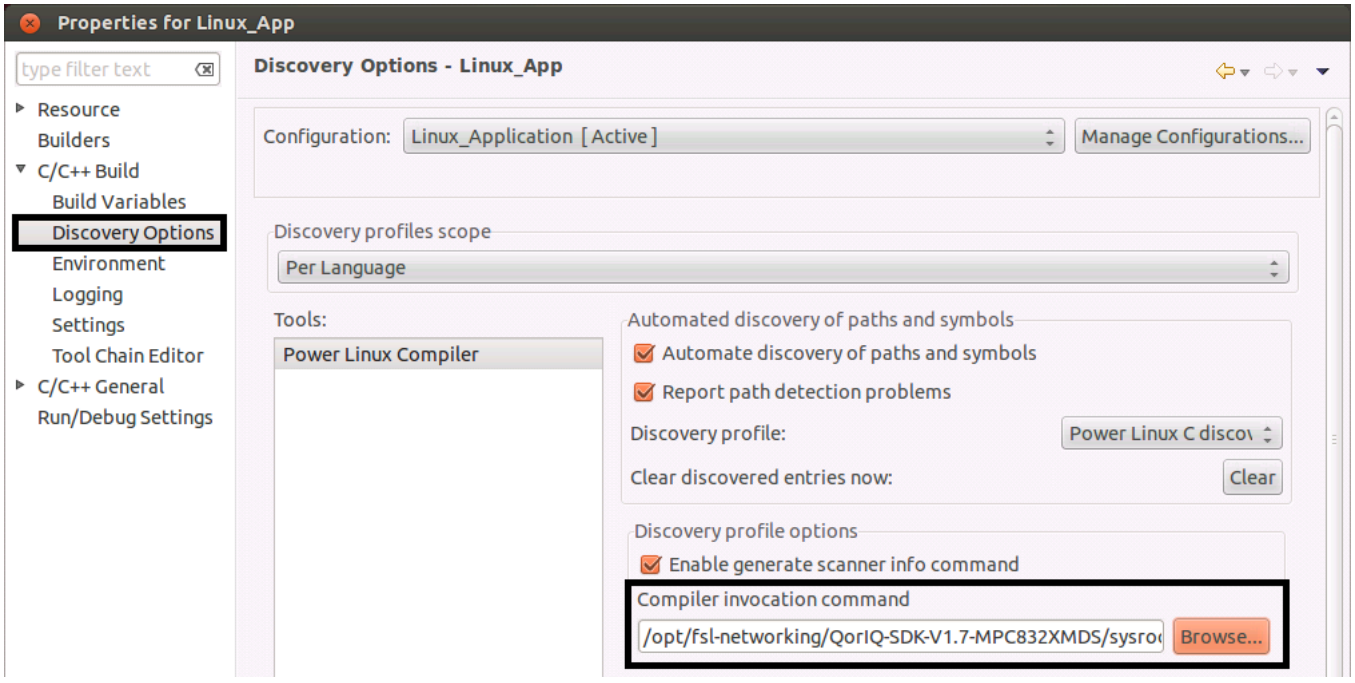
3. Specify the project name and location.
4. Select the processor and project output.
5. Configure the build settings.
6. Configure the connection details and click **Finish** to create the Linux application project.

NOTE	CodeWarrior doesn't provide Linux Project wizard for 8323 Processor. If a New Linux Project must be created using 8323, choose any Processor at step 4 and follow the steps from next section.

4.2. Change toolchain

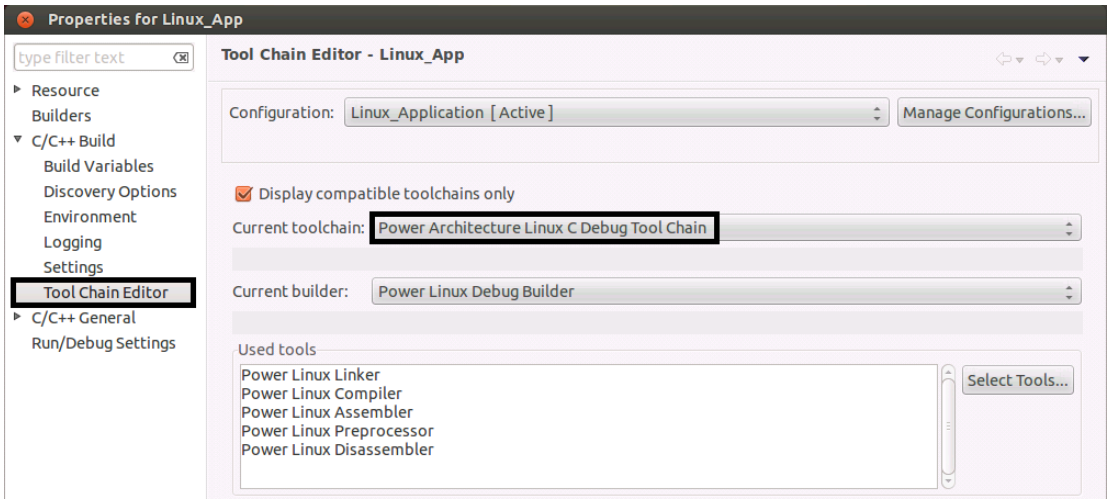
By default, the stationary project for Linux application includes the Freescale PPC or GCC binary toolchain. When toolchain is not provided by CodeWarrior, to add a default toolchain and an external build tool, follow these steps:

1. Choose **Project > Properties** from the CodeWarrior IDE menu bar. The **Properties** dialog appears.
2. Choose **C/C++ Build > Discovery Option** and select path for Freescale Linux SDK standalone toolchain at **Compiler invocation command.**



3. Click **Apply** to make the new settings available.
4. Choose **C/C++ Build > Tool Chain Editor** and select **Power Architecture Linux C Debug**

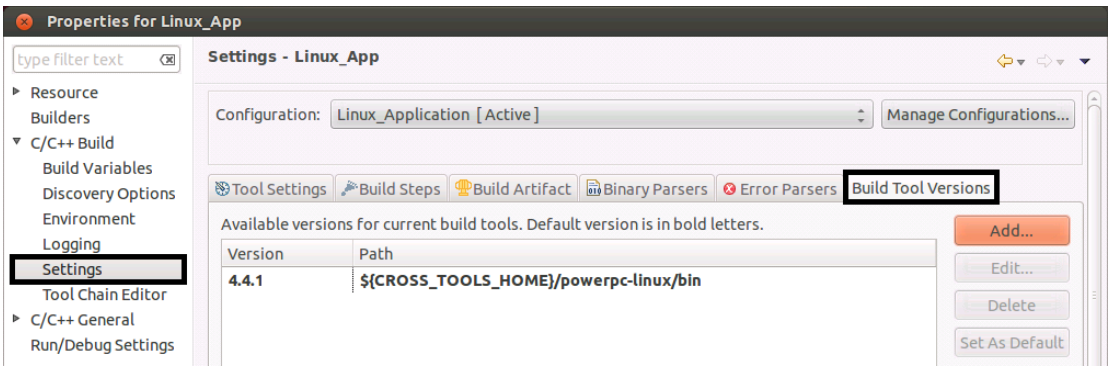
Tool Chain as **Current toolchain.** For Release configuration, select **Power Architecture Linux C Release Tool Chain** as **Current toolchain.**



NOTE	A dialog windows asks about Copy Tool Settings. It is indicated to copy compatible tool settings into the current toolchain.

5. Click **Apply** to make the new toolchain available.
6. Choose **C/C++ Build > Settings** in the left pane and click the **Build Tool Versions** tab in the right pane, as shown in the figure below.

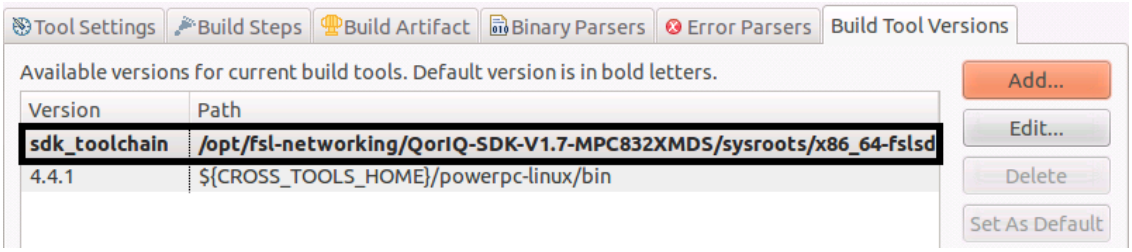
Figure 1. Project properties



7. Click **Add** and browse for the new toolchain location. The default installation path for Freescale Linux SDK standalone toolchain is: `/opt/fsl-networking/Layerscape-<release>/sysroot/<host-system>/usr/bin/powerpc-fsl-linux/`
8. Click **OK** to make the new toolchain available.

9. Select the new toolchain and click **Set As Default**, as shown in the figure below. This will make the new toolchain as the default toolchain for the project.

Figure 2. Change default toolchain

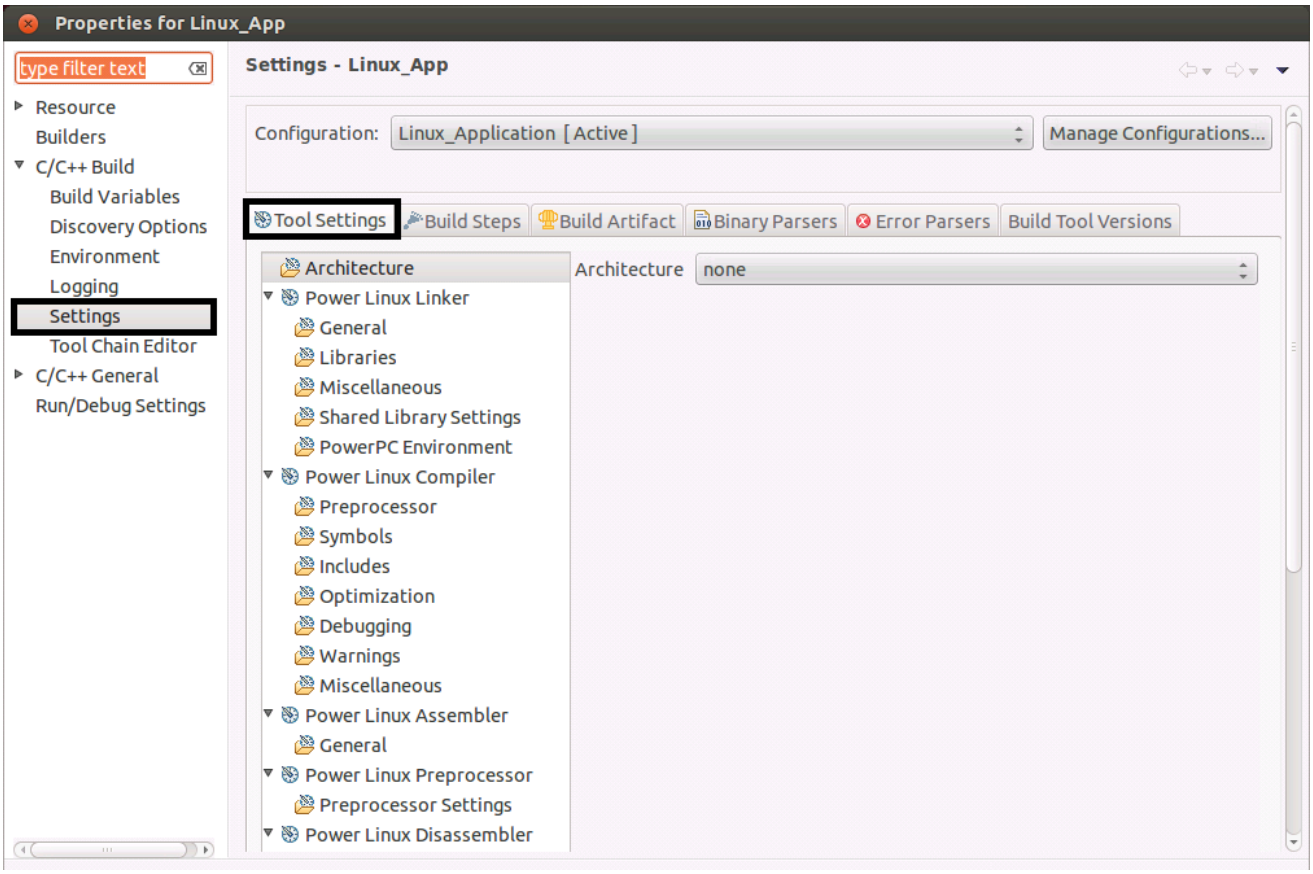


4.3. Verify build settings

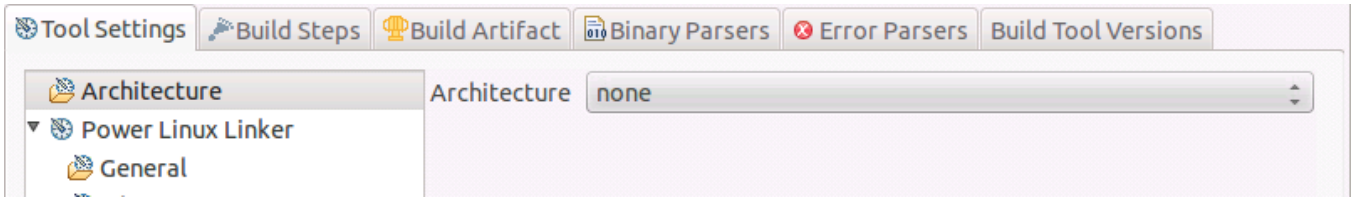
After setting the external toolchain as the default toolchain and before building your project, you should verify the build settings of the project. To verify build settings, follow these steps:

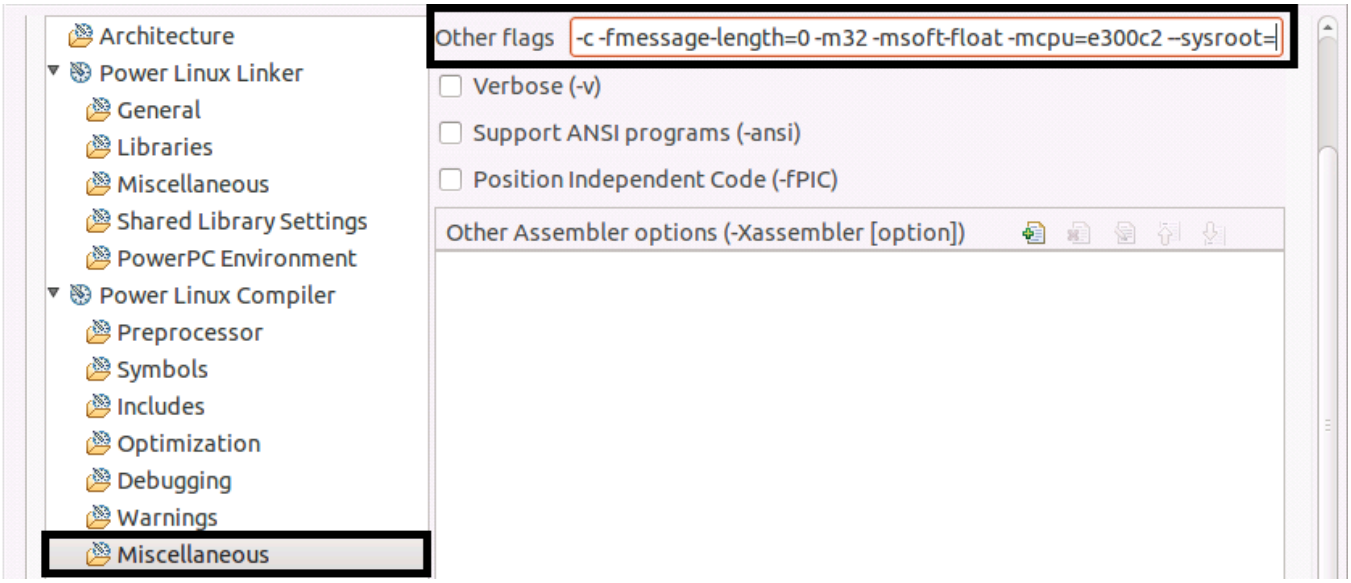
1. Choose **Project > Properties** from the CodeWarrior IDE menu bar. The **Properties** dialog appears.
2. Choose **C/C++ Build > Settings** in the left pane and click the **Tool Settings** tab in the right pane, as shown in the figure below.

Figure 3. Tool settings



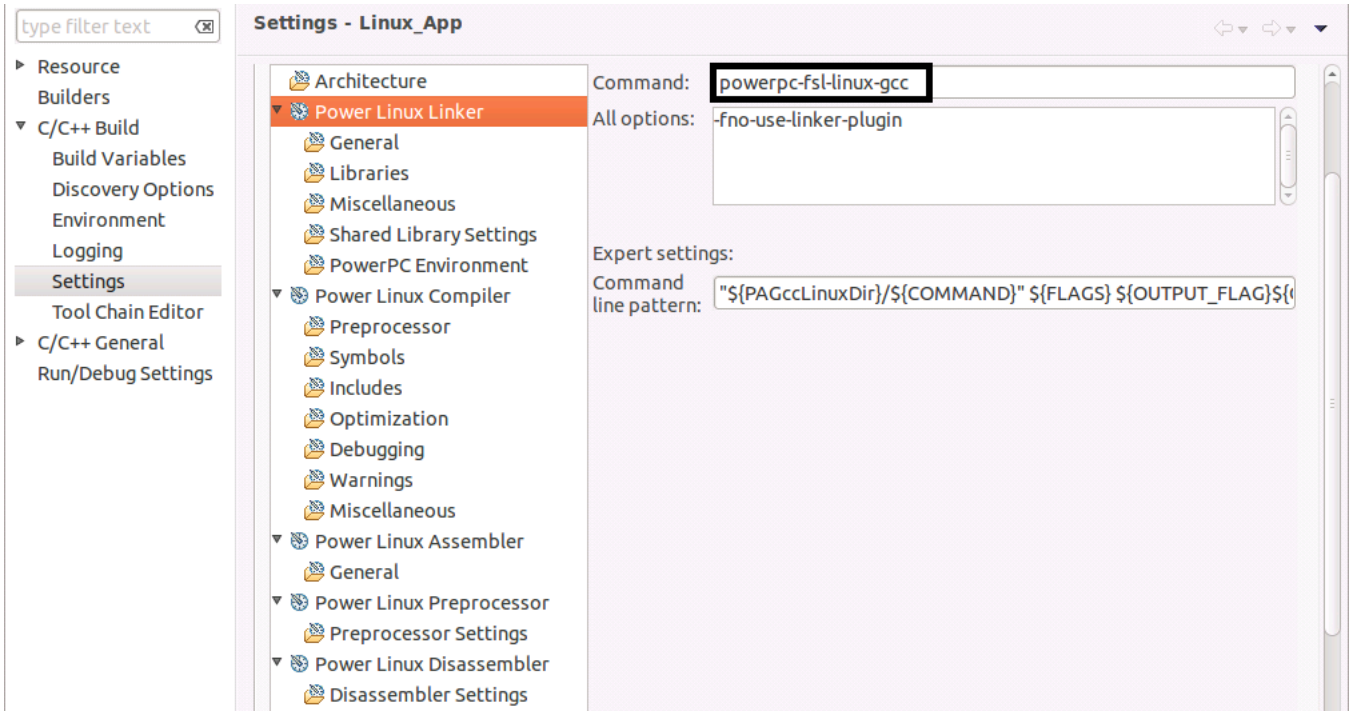
3. Choose correct Architecture type. If the desired Architecture is not available for selection, choose none and enter the Architecture type as compiler options.





4. For Power Linux Assembler, Compiler, Linker, Preprocessor, Disassembler, verify if the command is same as in the external toolchain (see the figure below).

Figure 4. Project settings



5. Click **OK** to save the project settings.

NOTE	Some build options may not be valid for a specific toolchain. For e300c2 toolchian – *fdebug-unwind-tables* is not a valid option and must be removed from Project > Settings > Power Linux Compiler > Miscellaneous > Other flags.

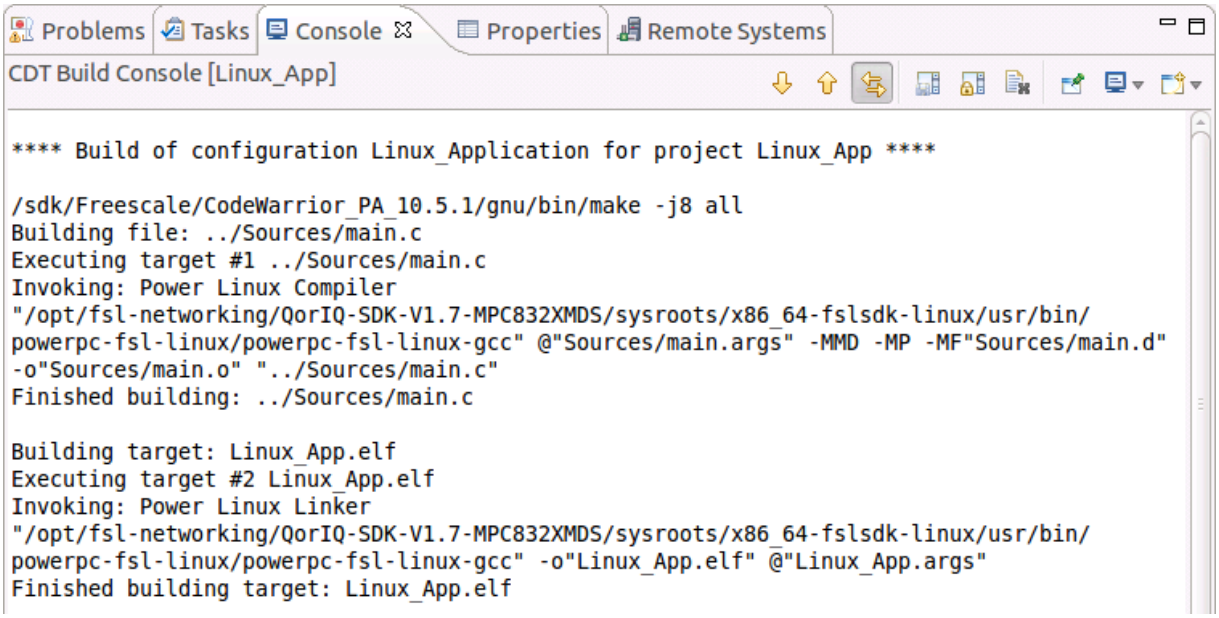
SDK toolchain is a sysrooted toolchain. This means that toolchain will start to look for target fragments and libraries starting from the path specified by the sysroot option. To have a working build configuration, follow these steps:

1. For Power Linux Compiler, go to **Miscellaneous > Other flags** and add *--sysroot="<path_to_target_sysroot>"* as an option.
2. For Power Linux Linker, go to **Miscellaneous > Linker flags** and add *--sysroot="<path_to_target_sysroot>"* as an option.

4.4. Build project using an external toolchain

To build the project, choose **Project > Build Project** from the CodeWarrior IDE menu bar. The project should be built with no errors, as shown in the figure below.

Figure 5. Console view



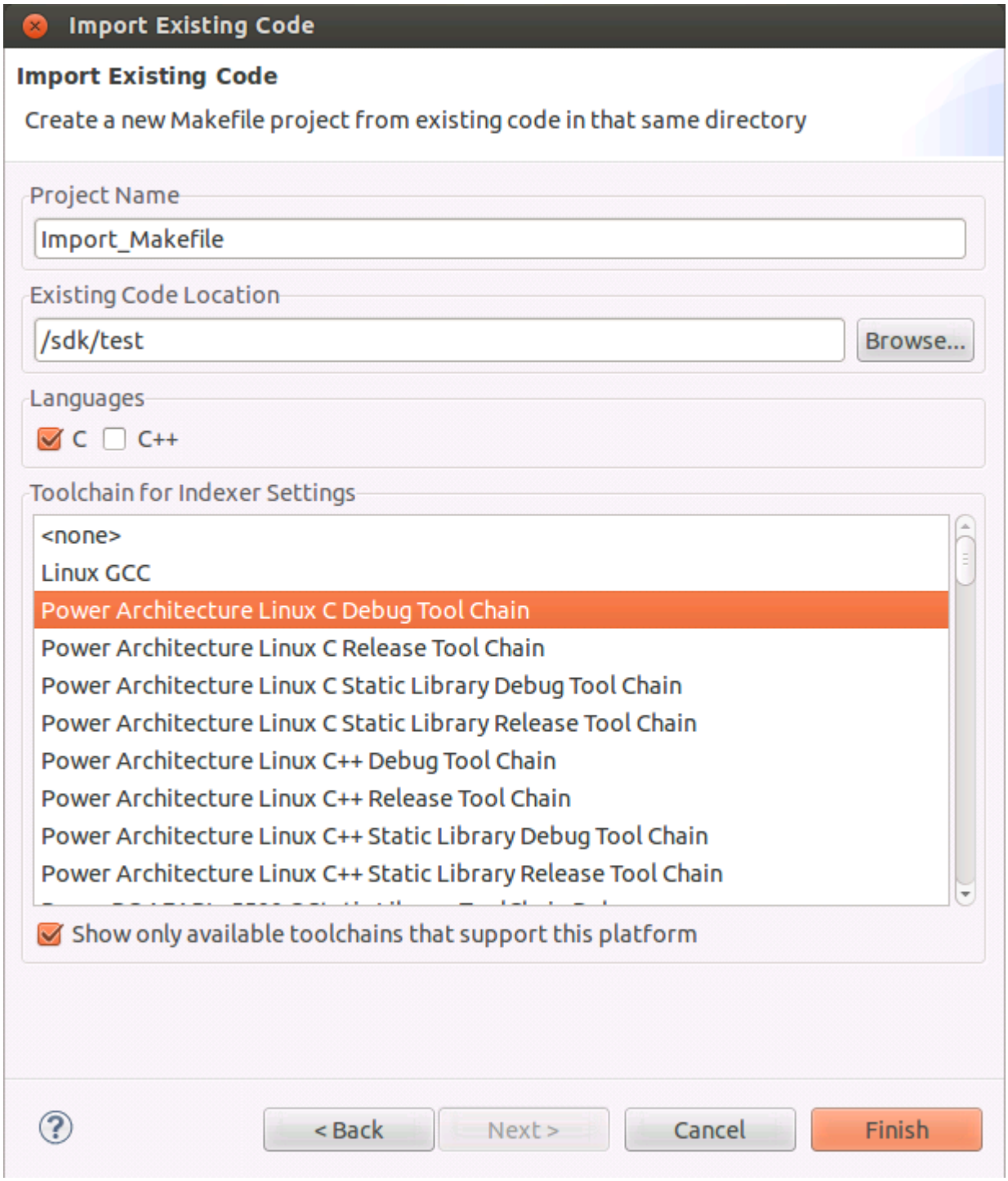
5. Importing Existing Code as Makefile Project

5.1. Import Makefile

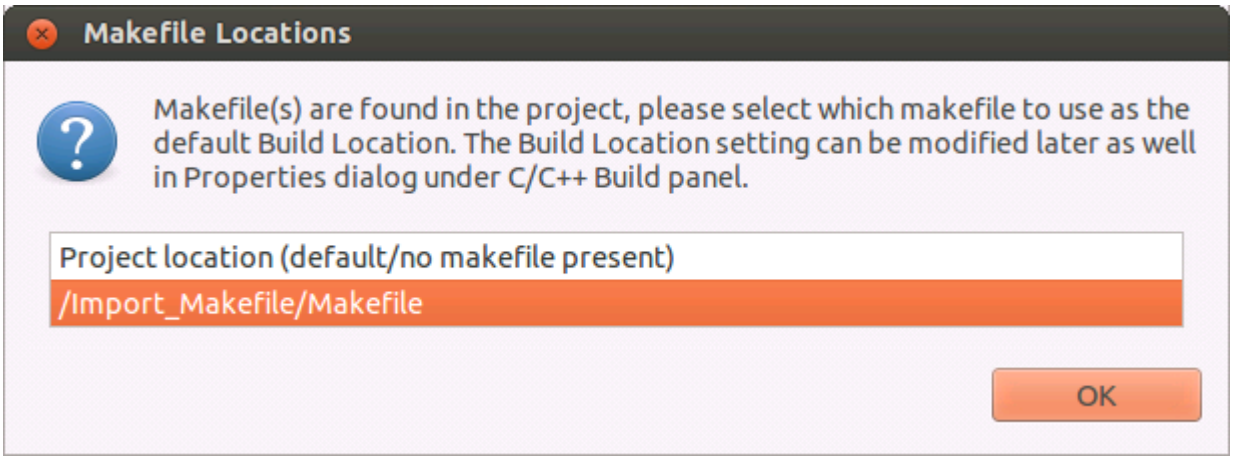
To import in CodeWarrior a Makefile Project, follow these steps:

1. Choose **File > Import** from the CodeWarrior IDE menu bar. The **Import** dialog appears.
2. Choose **C/C++ > Existing Code as Makefile Project** and click Next.
3. Choose Project Name, Existing Code Location, Language and Toolchain.

Figure 6. Change toolchain path



4. Click **Finish**. A dialog windows appears asking for Makefile.

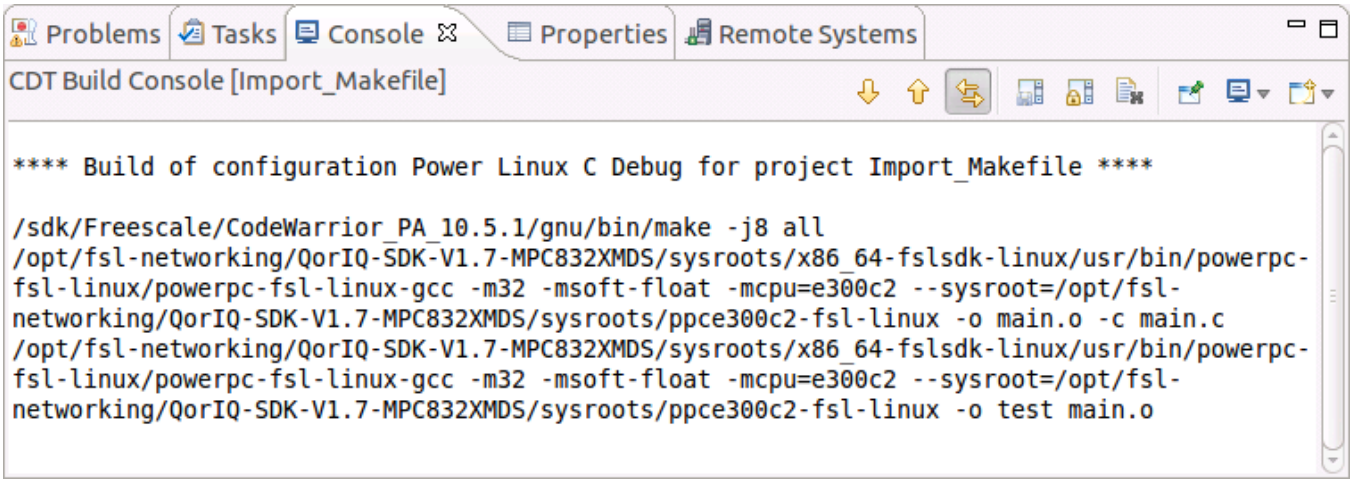


5. Click **OK**. The project is created.
6. Follow the steps from section **4.2**, to change the Compiler invocation command and add external build tool.

5.2. Build imported makefile project using an external toolchain

To build the project, choose **Project > Build Project** from the CodeWarrior IDE menu bar. The project should be built with no errors, as shown in the figure below.

Figure 7.Console view



How to Reach Us:

Home Page:
freescale.com

E-mail:
freescale.com/support

Document Number: AN5048

22 March 2016