Freescale Semiconductor
Application Note

Document Number: AN5277

Using an External GCC Toolchain with CodeWarrior for Power Architecture

1. Introduction

This document explains how to use an external GNU compiler collection (GCC) toolchain with CodeWarrior for Power Architecture. 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 Linux project to work with an SDK standalone toolchain
- Build the project using an external toolchain
- Import existing code as a makefile project

Contents

2. Preliminary background

CodeWarrior for Power Architecture may or may not include the Freescale PPC or GCC binary toolchain. If you are developing a Linux user space application with CodeWarrior, then you should 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 <sdk_install_path>
$ source ./fsl-setup-env -m <machine>
$ 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 Power Architecture 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 Power Architecture stationary project for Linux application, follow these steps:

1. Start CodeWarrior for Power Architecture.

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.

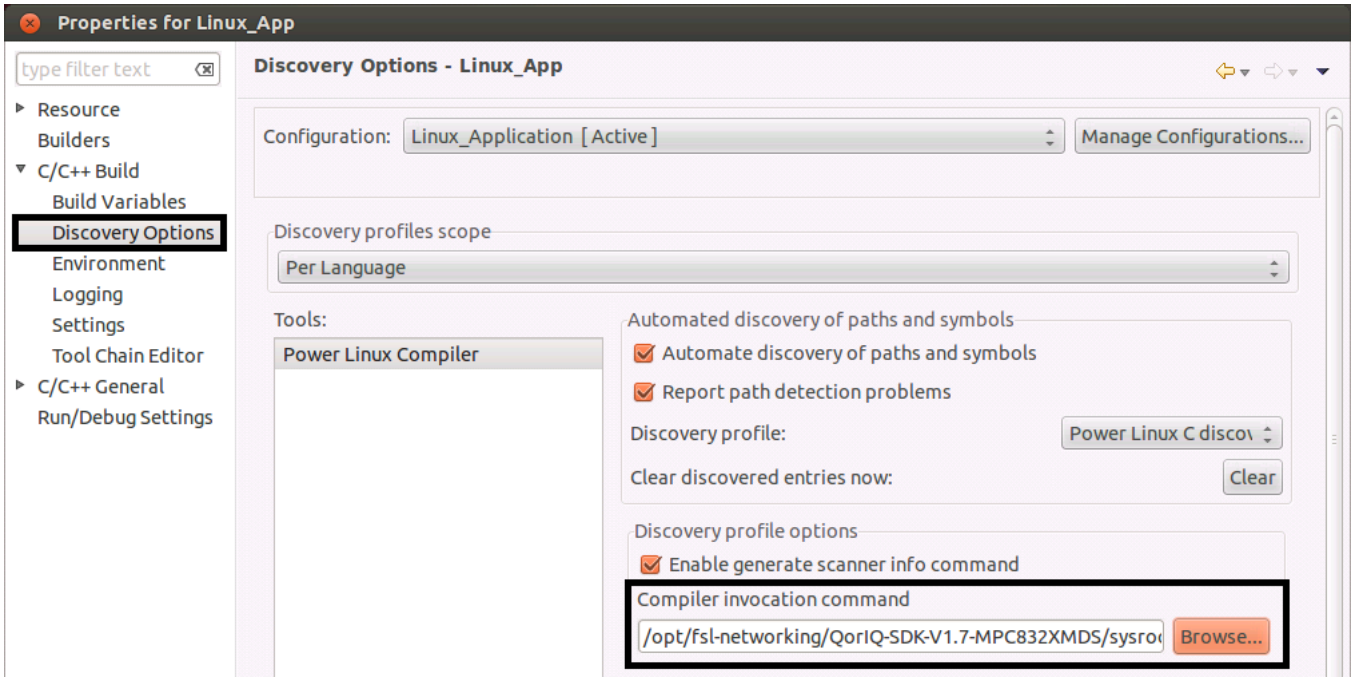
NOTE	To create Linux application projects, you need to install Linux build tool. CodeWarrior does not provide Linux project wizard for the MPC8323 processor. If you need to create a new Linux project using the 8323 processor, then choose any processor at step 4 and follow the steps from next section. Importing an example project can also be a starting point for creating a Linux application.

4.2. Change toolchain

By default, the stationary project for Linux application includes the Freescale PPC or GCC binary toolchain. In case, CodeWarrior does not provide the required toolchain, then you can add a default toolchain and an external build tool, by following these steps:

1. Choose **Project > Properties** from the CodeWarrior IDE menu bar. The **Properties** dialog appears.
2. Select **C/C++ Build > Discovery Options** in the left pane and specify Freescale Linux SDK standalone toolchain path in the **Compiler invocation command** field in the right pane.

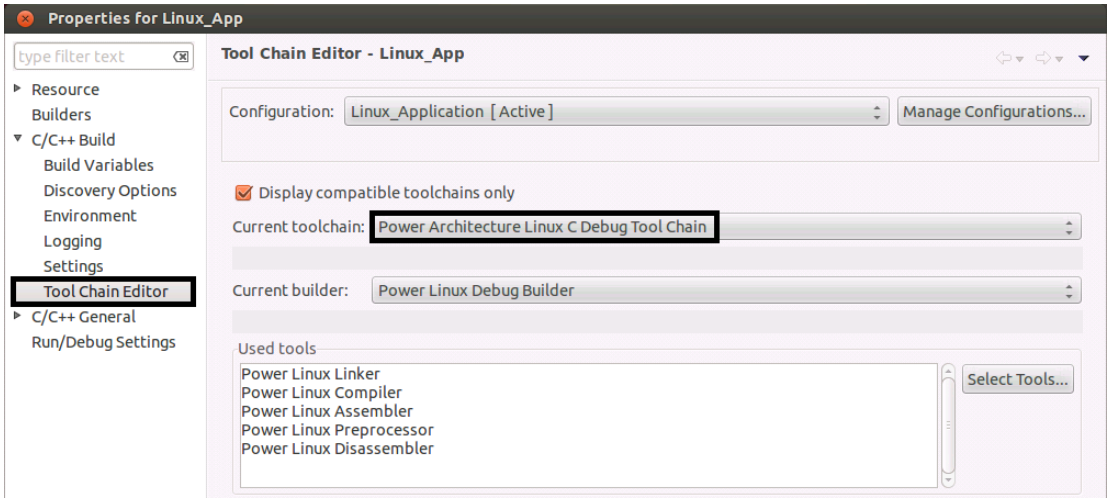
Figure 1. Specify Freescale Linux SDK standalone toolchain path



NOTE	Compiler invocation command should be something like this: `/opt/fsl-networking/QorIQ-SDK-<release>/sysroot/<target_architecture>-fsl-linux`.

3. Click **Apply** to apply the new settings.
4. Select **C/C++ Build > Tool Chain Editor** in the left pane and:
   - For an active configuration, choose **Power Architecture Linux C Debug Tool Chain** from the **Current toolchain** menu in the right pane
   - For a release configuration, choose **Power Architecture Linux C Release Tool Chain** from the **Current toolchain** menu in the right pane

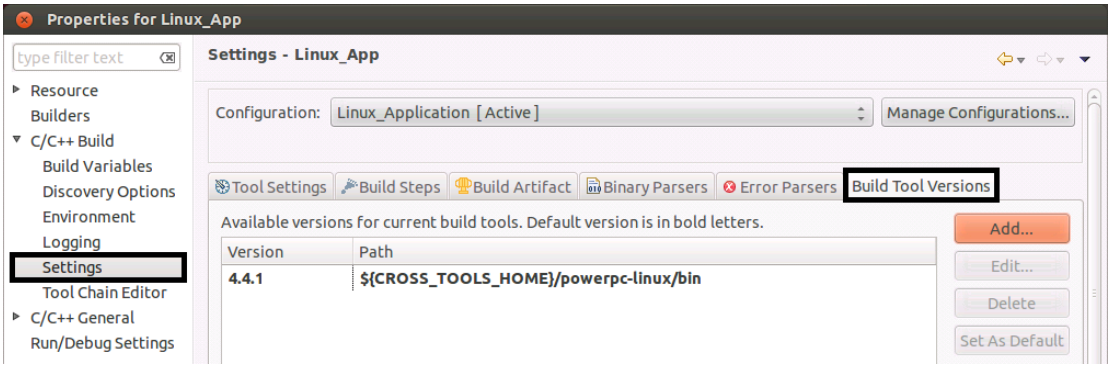
Figure 2. Specify toolchain type



NOTE A dialog appears, asking you whether you want to copy tool settings or not. You are recommended to copy compatible tool settings into the current toolchain.

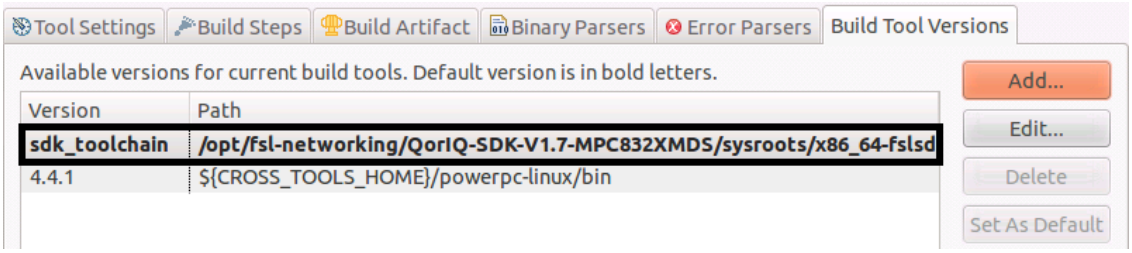
5. Click **Apply** to apply the new toolchain.
6. Select **C/C++ Build > Settings** in the left pane and click the **Build Tool Versions** tab in the right pane, as shown in the figure below.

Figure 3. Add new toolchain



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

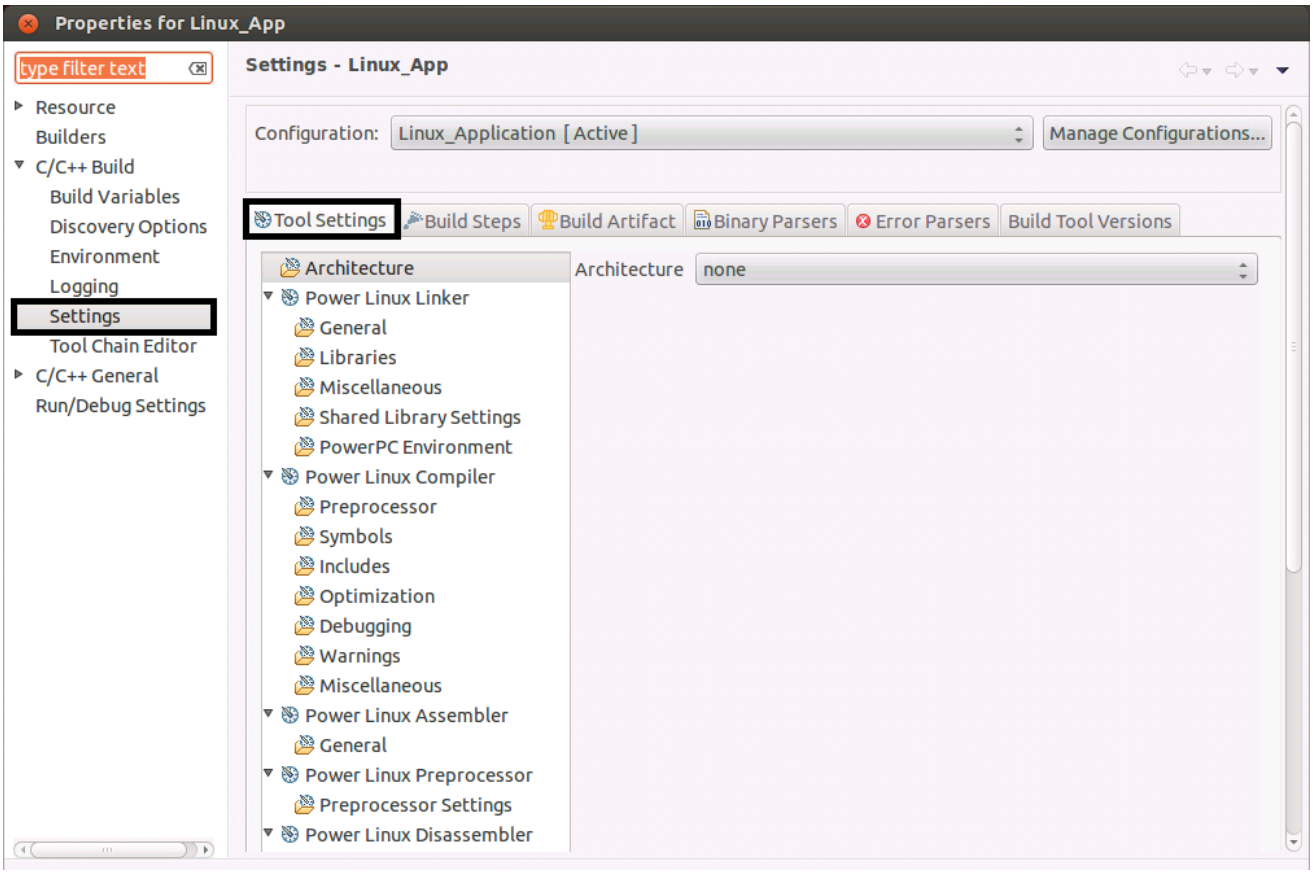


4.3. Verify build settings

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. Select **C/C++ Build > Settings** in the left pane of the **Properties** dialog and click the **Tool Settings** tab in the right pane, as shown in the figure below.

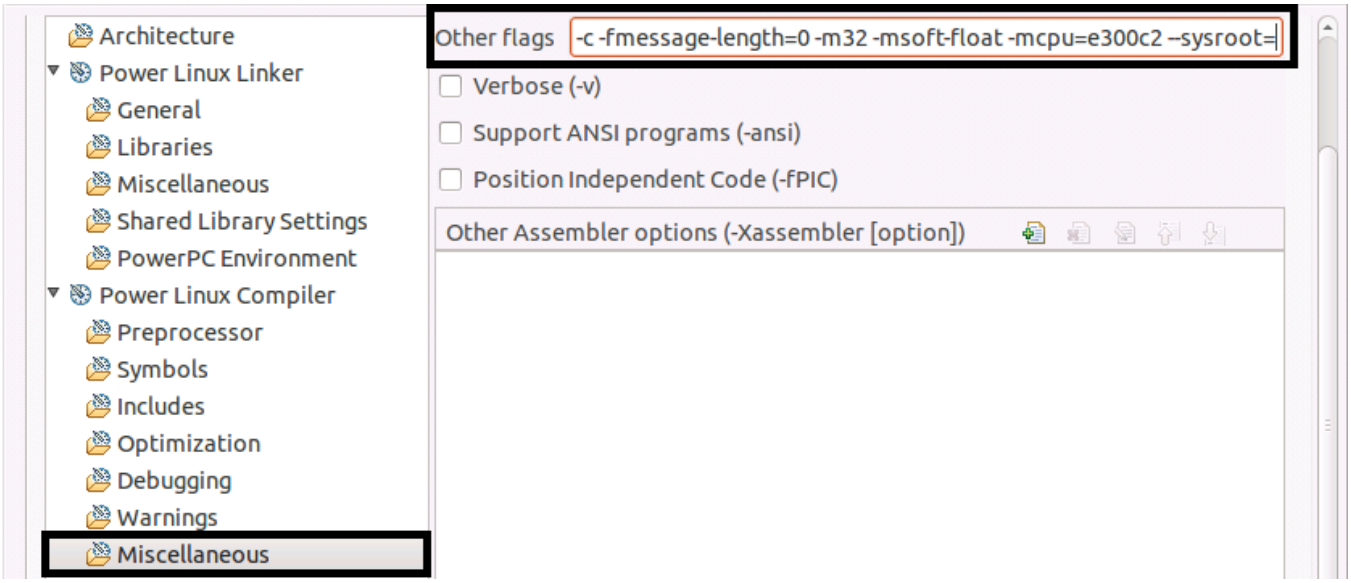
Figure 5. Verify build settings



2. Choose the correct architecture type from the Architecture menu on the **Architecture** page of the **Tool Settings** page. If the desired architecture type is not available for selection, then choose

none and specify the architecture type as compiler options on the **Miscellaneous** page, as shown in the figure below.

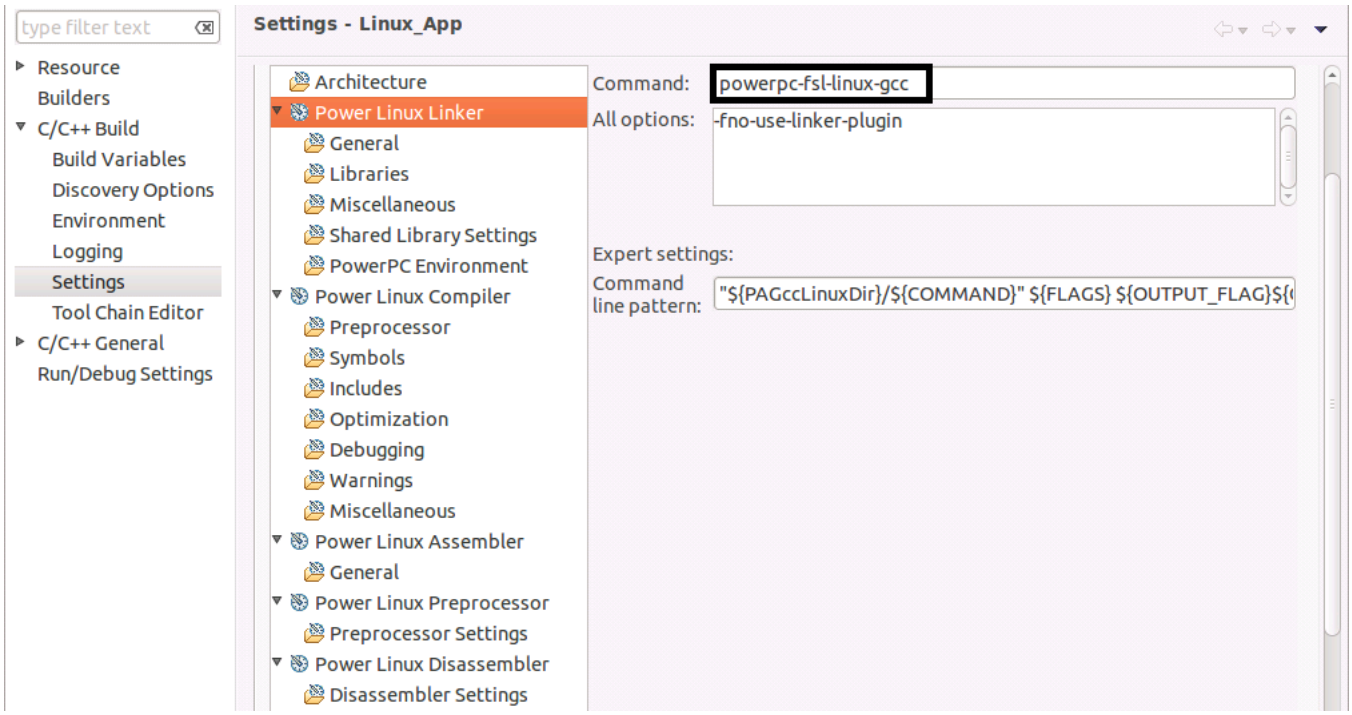
Figure 6. Specify architecture type as compiler options



NOTE	The example shown in the figure above is for the MPC8323 processor.

3. For PowerPC Linux assembler, compiler, linker, preprocessor, and disassembler, verify if the command is same as in the external toolchain (see the figure below).

Figure 7. Verify linker command



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

NOTE	Some build options may not be valid for a specific toolchain. For example, `-fdebug-unwind-tables` is not a valid option for the e300c2 toolchain. You can remove this option using the **Other flags** field on the **C/C++ Build > Settings > Tool Settings > Power Linux Compiler > Miscellaneous** page of the **Properties** dialog.

The SDK toolchain is a sysrooted toolchain. It 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:

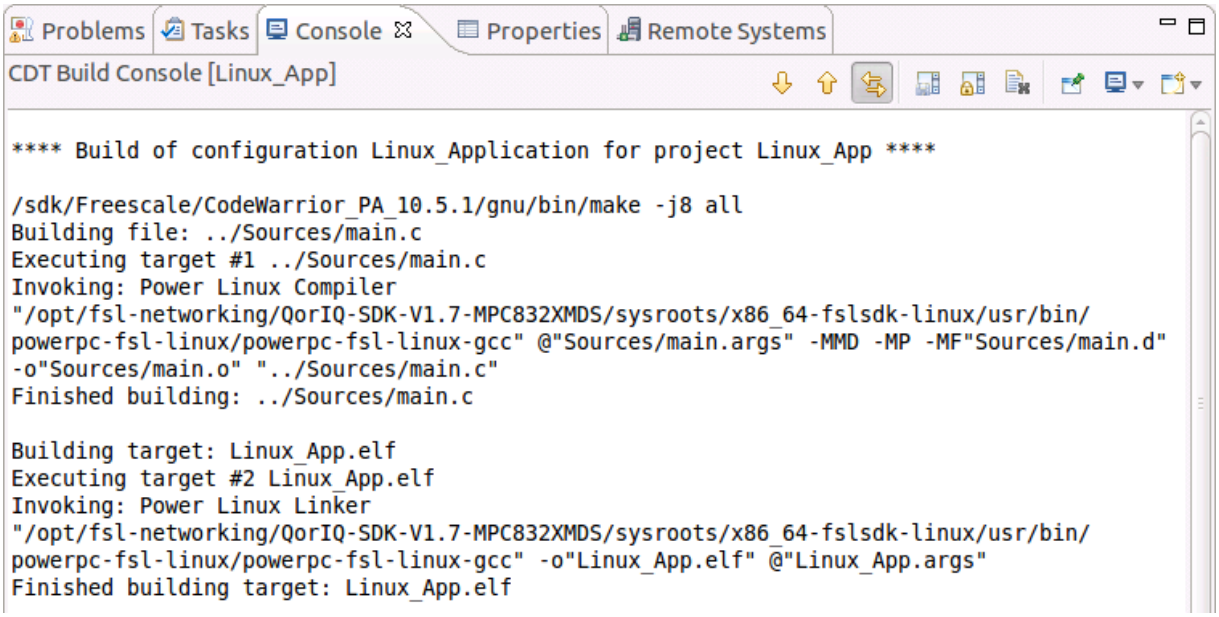
1. Go to the **C/C++ Build > Settings > Tool Settings > Power Linux Compiler > Miscellaneous** page of the **Properties** dialog and specify `--sysroot="<path_to_target_sysroot>"` in the **Other flags** field.
2. Go to the **C/C++ Build > Settings > Tool Settings > Power Linux Linker > Miscellaneous** page and specify `--sysroot="<path_to_target_sysroot>"` in the **Linker flags** field.

4.4. Build project using an external toolchain

To build the project, choose **Project > Build Project** from the CodeWarrior IDE menu bar. The project starts building and the **Console** view displays the progress of building the project, as shown in the figure

below.

Figure 8. Build project



5. Importing existing code as a Makefile project

This section contains the following subsections:

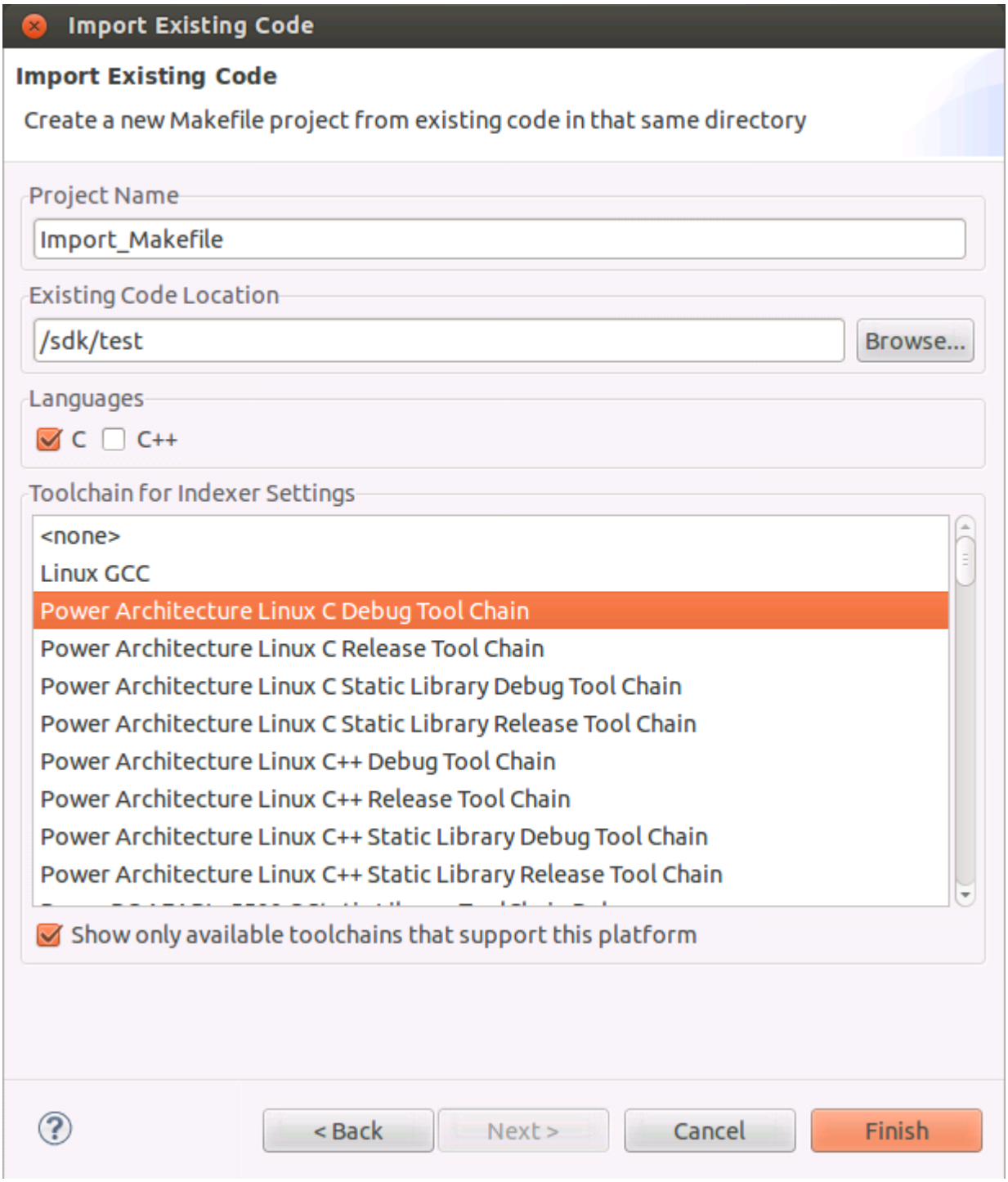
- Import Makefile project
- Build imported Makefile project using an external toolchain

5.1. Import Makefile project

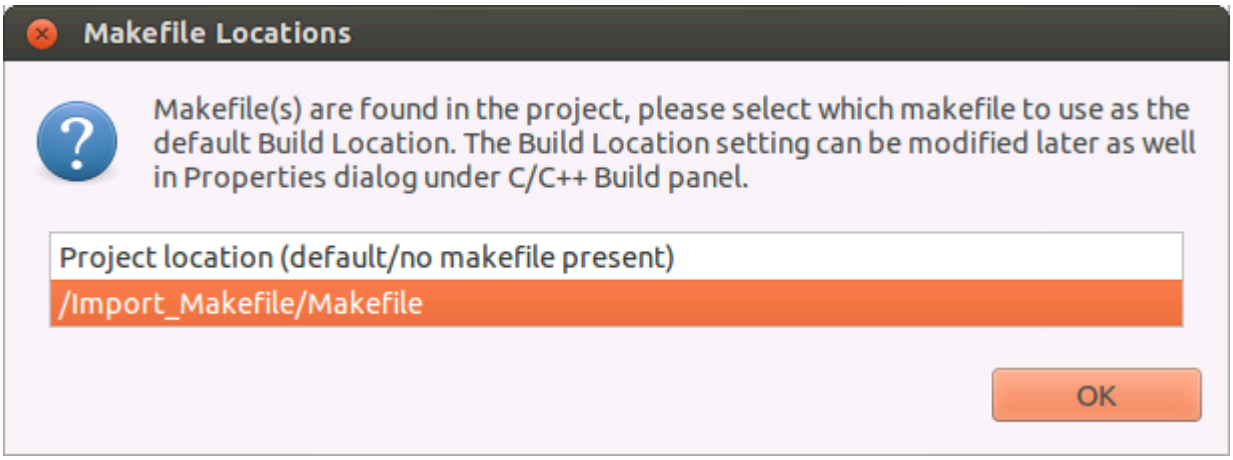
To import a Makefile Project into CodeWarrior, follow these steps:

1. Choose **File > Import** from the CodeWarrior IDE menu bar. The **Import** wizard starts, displaying the **Select** page.
2. Choose **C/C++ > Existing Code as Makefile Project** and click **Next**. The wizard name and page name changes to **Import Existing Code**.
3. Specify project name, existing code location, language, and toolchain, as shown in the figure below.

Figure 9. Specify toolchain



4. Click **Finish**. The **Makefile Locations** dialog appears asking for Makefile.

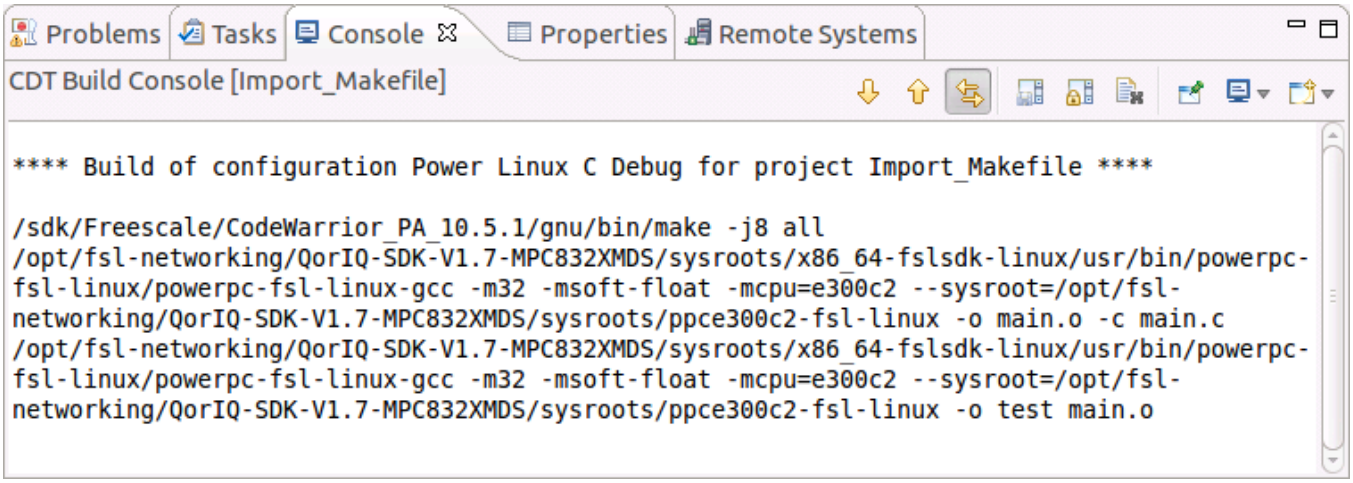


5. Click **OK**. The project is created.
6. Follow the steps from Change toolchain, to change the compiler invocation command and add the external build tool.

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 starts building and the **Console** view displays the progress of building the project, as shown in the figure below.

Figure 10. Build project



How to Reach Us:

Home Page:
nxp.com

E-mail:
nxp.com/support

Document Number: AN5277

10 May 2016