Using an External GCC Toolchain with CodeWarrior for ARMv7

1. Introduction

This document explains how to use an external GNU compiler collection (GCC) toolchain with CodeWarrior for QorIQ LS series – ARM V7 ISA. 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

1. Introduction.................................................................1
2. Preliminary background................................................2
3. Using SDK standalone toolchain .................................2
4. Switching between different CodeWarrior toolchains...............2
5. Working with an ARMv7 Linux application project................3
6. Changing toolchain path..............................................7
2. Preliminary background

CodeWarrior for QorIQ LS series - ARM V7 ISA includes the GCC Linaro 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. Using 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. Switching between different CodeWarrior toolchains

CodeWarrior toolchain versions are provided as service packs. This helps you to choose the desired toolchain version.

To switch from Linaro GCC toolchain 4.9.3 to 4.8.3, follow these steps:

2. Click Installation Details. The CodeWarrior Development Studio Installation Details dialog appears.
4. Restart CodeWarrior as indicated by the installer.
5. Choose Help > Install New Software from the CodeWarrior IDE menu bar and select the service pack corresponding to Linaro GCC toolchain 4.8.3.
6. Install the new service pack.
7. Restart CodeWarrior as indicated by the installer.

To switch back to the original Linaro GCC toolchain 4.9.3, follow these steps:

2. Click Installation Details. The CodeWarrior Development Studio Installation Details dialog appears.
4. Restart CodeWarrior as indicated by the installer.
5. Go to Help > Install New Software and reinstall the service pack corresponding to Linaro GCC toolchain 4.9.3.

| NOTE | Due to the differences between toolchains 4.8.3 and 4.9.3, when a project created using toolchain 4.8.3 service pack is built using toolchain 4.9.3 service pack, the correct architecture needs to be chosen from the Architecture list on the Project Properties > C/C++ Build > Settings > Tool Settings > Target Processor page. In addition, when a project created using toolchain 4.9.3 service pack is built using toolchain 4.8.3 service pack, the rdimon library needs to be added to the Libraries pane on the Project Properties > C/C++ Build > Settings > Tool Settings > ARM Sourcery GCC C Linker > Libraries page and the --specs=rdimon.specs flag needs to be added to the Other flags field on the Project Properties > C/C++ Build > Settings > Tool Settings > ARM Sourcery GCC C Linker > Miscellaneous page. Changes in other settings may be required depending on the processor type or compiler options. |

5. Working with an ARMv7 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

5.1. Create a stationary project for Linux application

To create an ARMv7 stationary project for Linux application, follow these steps:
Working with an ARMv7 Linux application project

1. Start CodeWarrior for QorIQ LS series - ARM V7 ISA.
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.

### 5.2. Change toolchain

By default, the stationary project for Linux application includes the GCC Linaro binary toolchain. To change the default toolchain used by CodeWarrior, 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 **Build Tool Versions** tab in the right pane, as shown in the figure below.

![Figure 1. Project properties](image)

3. 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/arm-fsl-linux-gnueabihf/`
4. Click **OK** to make the new toolchain available.
5. 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

![Tool Settings](image)

NOTE In the CodeWarrior for QorIQ LS series - ARM V7 ISA 10.0.3 release, the Add option is not available and you need to implement a workaround to use an external toolchain as the default toolchain. See Changing toolchain path for details.

5.3. 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:

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.

![Tool Settings](image)

3. For ARM Sourcery GCC Assembler, C Compiler, C Linker, C Preprocessor, C++ Preprocessor,
Working with an ARMv7 Linux application project

verify if the command is same as in the external toolchain (see the figure below).

**NOTE**
In the toolchain that is shipped with CodeWarrior, the command is: `arm-linux-gnueabihf-`, and in the Freescale SDK Linux toolchain, the command is: `arm-fsl-linux-gnueabi-`.

Figure 4. Project settings

Figure 4. Project settings

SDK toolchain is a sysrooted toolchain. This means that GCC 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 ARM Sourcery GCC C Compiler, go to Miscellaneous > Other flags and add `--sysroot='<path_to_target_sysroot>'` as an option.

2. For ARM Sourcery GCC C Linker, go to Miscellaneous > Other flags and add `--sysroot='<path_to_target_sysroot>'` as an option.

5.4. **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.
6. Changing toolchain path

To specify the external toolchain path as the path for the `ARMv7GCCToolsDir` and `ARMv7GCCLinToolsDir` variables, follow these steps:

1. Choose **Project > Properties** from the CodeWarrior IDE menu bar. The **Properties** dialog appears.
2. Choose **C/C++ Build > Build Variables** in the left pane and click **Add** in the right pane. The **Define a New Build Variable** dialog appears.
3. Specify external toolchain path as the value (path) for each of the variables, `ARMv7GCCToolsDir` and `ARMv7GCCLinToolsDir`, as shown in the figure below.
Changing toolchain path

**Figure 6. Change toolchain path**

4. Click **OK**.

5. Click **OK**.