NXP Semiconductors
Application Note

Document Number: AN5224
Rev. 11.3.2, 08/2018

Using an external toolchain with CodeWarrior for ARMv8

Contents

1 Introduction

This document describes the steps to use an external toolchain, that is a non CodeWarrior default toochain, with CodeWarrior for ARMv8. This document explains:

- Building the toolchain available with the NXP Linux SDK
- Installing a new toolchain inside the CodeWarrior tool
- Customizing stationary projects (bare metal and Linux application) to work with the new toolchain
- Building a project using the external toolchain

2 Preliminary background

CodeWarrior for ARMv8 includes Linaro binary toolchain by default. If you want to develop a Linux user space application with CodeWarrior, you can use the toolchain supplied with the NXP Linux SDK or the external prebuild toolchain available in the SDK release.

To develop a bare metal application in the CodeWarrior software using an external bare metal toolchain, install the Service Pack that includes the toolchain and follow the steps described in this Application Note.

3 External toolchain

This section explains:

- Prebuild toolchain
- Standalone SDK toolchain

3.1 Prebuild toolchain

Ensure that you have extracted the NXP prebuild toolchain on your Linux machine. Current gcc version is indicated in the SDK release. Another way is to install the CodeWarrior ServicePack containing that toolchain.

See Change the toolchain for information about using a prebuild toolchain as the default build tool in the CodeWarrior software.

3.2 Standalone SDK toolchain

Another method of using an external toolchain is to compile the toochain available in SDK. To build and install the standalone toolchain using Yocto, refer the steps available in the QorIQ SDK Documentation. You can access the PDF or knowledge center from nxp.com:

https://www.nxp.com/support/developer-resources/run-time-software/linux-software-and-development-tools/linux-sdk-for-qoriq-processors:SDKLINUX?tab=Documentation_Tab

The default installation path for standalone toolchain is: `/opt/fsl-networking/`. The installation folder can be specified during the installation procedure. Additional information about building and installing the standalone toolchain with Yocto can be found in the SDK Documentation.

4 ARMv8 Linux application project

This section explains:

- Create stationary project for Linux application
- Change the toolchain

4.1 Create stationary project for Linux application

To create an ARMv8 stationary project for a Linux application:

1. In the CodeWarrior IDE, select **File > New > ARMv8 Stationary**.
2. Select **ARMv8 > Linux Application Debug > AArch64 > Hello World C Project**.
3. Type a Project Name.
4. Click **Finish**.

4.2 Change the toolchain

The stationary project for Linux application includes by default the Linaro GNU binary toolchain. To change the default toolchain for the created project, follow these steps:

1. Right-click on the project and select **Properties**.
2. In the **Properties** dialog, select **C\C++ Build > Tools Path**.

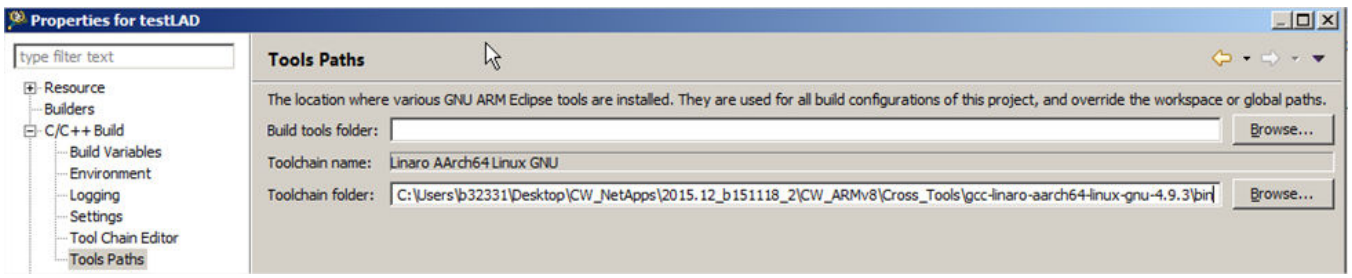

Figure 1. Set toolchain

3. Browse to the desired toolchain in the **Toolchain folder** field.
   a. For the Service Pack, the path is: `<CW_Layout>\CW_ARMv8\Cross_Tools\gcc-linaro-aarch64-linux-gnu-<ver>\bin`
   b. For the SDK Linux, go to the path where aarch64 cross-compiler was installed during Toolchain standalone installation.
      1. SDK toolchain is a sysrooted toolchain. This means that GCC will start to look for target files and libraries starting from the path specified by the sysroot option. To have a working build configuration, follow these steps:
         a. Select **Project Properties > C/C++ Build > Settings > Tool Settings > Cross Arm C Compiler > Miscellaneous**.
         b. Set the option **Other compiler flags** to `--sysroot="path_to_SDK_sysroot`
         c. Select **Project Properties > C/C++ Build > Settings > Tool Settings > Cross Arm C Linker > Miscellaneous**.
         d. Set the option **Other linker flags** to `--sysroot="path_to_SDK_sysroot`
   c. For the LSDK, go to the path where aarch64 cross-compiler was installed during LSDK prebuild action. For example, `/usr/aarch64-linux-gnu`
4. To check if the new external toochain is configured, select **Project Properties > C/C++ Build > Settings > Toolchains**.
5. Click the **Toolchains** tab, the **Toolchain path** should be configured as shown in the following figure:

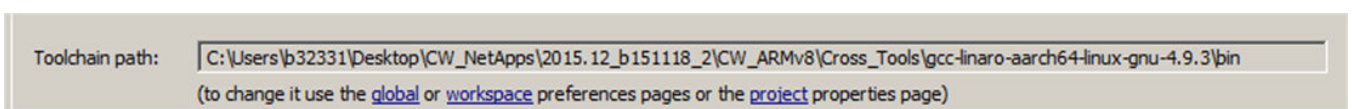

Figure 2. Toolchain path

5 ARMv8 Bareboard project

This topic explains:

5.1 Create stationary project for bareboard

To create an ARMv8 stationary project for Linux application:

1. Select **File > New > ARMv8 Stationary**.
2. Select **ARMv8 > Bare board > Hello World C Project**.
3. Type a Project Name.
4. Click **Finish**.

5.2 Change the toolchain

The stationary project for Bare board application includes by default the Linaro ELF binary toolchain. To change the default toolchain for the created project. follow these steps:

1. Right-click on the project and select **Properties**.
2. In the **Properties** dialog, select **C\C++ Build > Tools Path**.
3. Browse to the desired toolchain in the **Toolchain folder** field.
    a. For the Service Pack, the path is: `<CW_Layout>\CW_ARMv8\Cross_Tools\gcc-linaro-aarch64-none-elf-<ver>\bin`

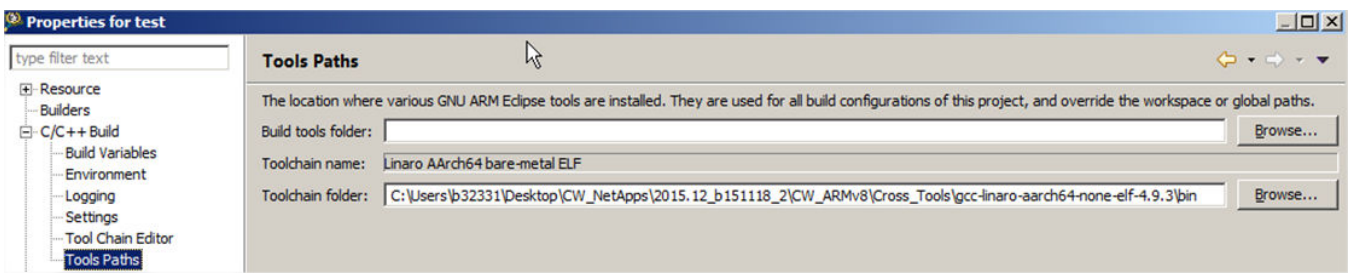

Figure 3. Set toochain

6 Troubleshooting

This topic lists the troubleshooting steps for the issue that you can encounter while integrating the new toolchain.

1. For the CodeWarrior releases based on Eclipse version, before updating the toolchain path, ensure that the default toolchain is configured for the Linux application or the bare metal project. To specify the default toolchain:
    a. Select **Window > Preferences > C/C++ > Build > Global Tools Paths**.
    b. Select the toolchain as per your requirement, for example, select **Linaro AArch64 bare-metal ELF** for the bare metal toolchain update or select **Linaro AArch64Linux GNU** for the linux application toolchain update.

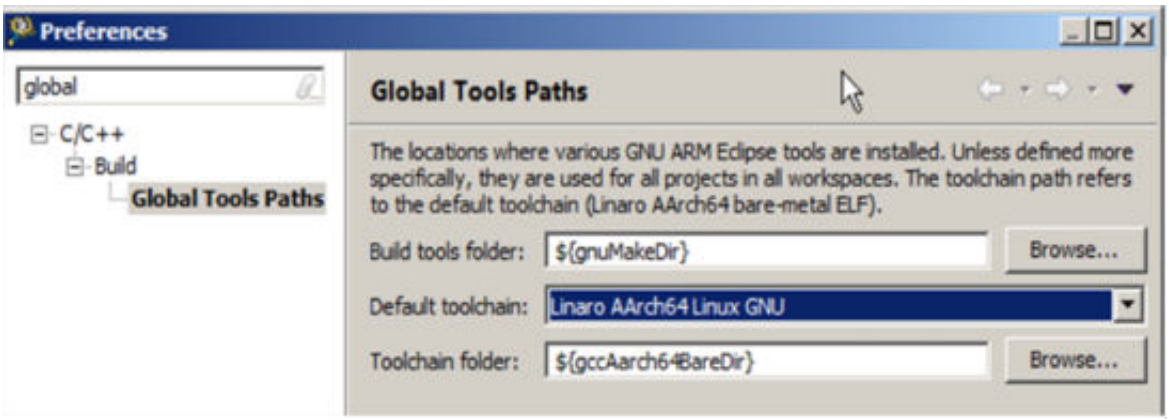

Figure 4. Global Tool Paths

c. Click **OK** to close the **Preferences** dialog before updating the Tools path in the **Project Properties** dialog.

2. Toolchain prefix is different for different toolchains, that is the binaries have different names. An example is when migrating from bare-metal toolchain 4.8.3 to 4.9.3, the prefix should updated to `aarch64-elf-` instead of `aarch64-none-elf-` . This can be done from **C/C++ Build > Settings > Toolchains tab > Prefix** field.

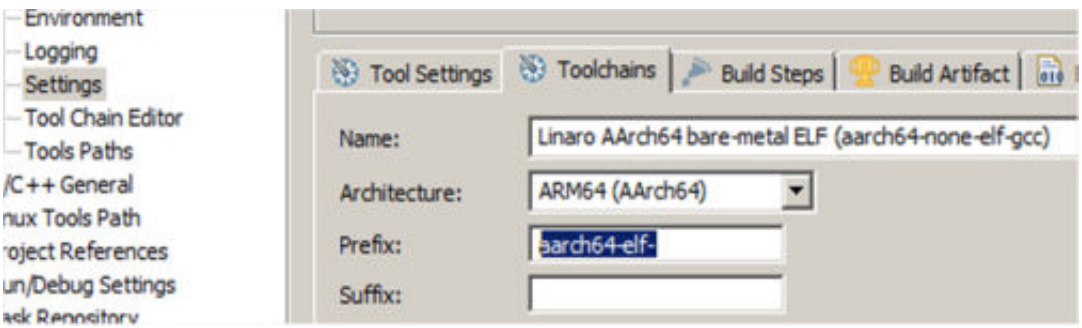

Figure 5. Toolchains

3. Tools settings need to be updated with different parameters and compiler options. This can be done from the **C/C++ Build > Settings > Tool Settings** tab.

How to Reach Us:

Home Page:
nxp.com

Web Support:
nxp.com/support

Document Number AN5224
Revision 11.3.2, 08/2018