NXP Semiconductors
Application Note

Document Number: AN5347
Rev. 0, 10/2016

Building U-Boot in CodeWarrior ARMv8

Contents

1 Introduction

This application note defines guidelines for configuring CodeWarrior for ARMv8 for U-Boot development.

This document explains:

- Installing SDK standalone toolchain
- Configuring CodeWarrior for ARMv8 for building U-Boot
- Building U-boot using CodeWarrior for ARMv8

2 Requirements

For building U-Boot using CodeWarrior for ARMv8, you need a host computer with Linux OS and CodeWarrior for ARMv8 Linux version installed.

3 Installing SDK standalone toolchain

Linux SDK provides a standalone toolchain that can be used for building different application outside Yocto. In this case, you can use the standalone toolchain for building U-Boot using CodeWarrior for ARMv8.

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-qoriq-glibc-<host-system>-<core>-toolchain-<release>.sh
```

4 Configuring CodeWarrior for ARMv8 for building U-Boot

To create a project for building U-Boot in CodeWarrior for ARMv8, perform these steps:

1. Select **File > New > C Project**.
2. Specify the project name and select **Empty Project** as **Project type**.
3. Uncheck the **Use default location** and Browse to the location of the U-Boot source.
4. Select **Cross GCC** as Toolchain.
5. Click **Next**.

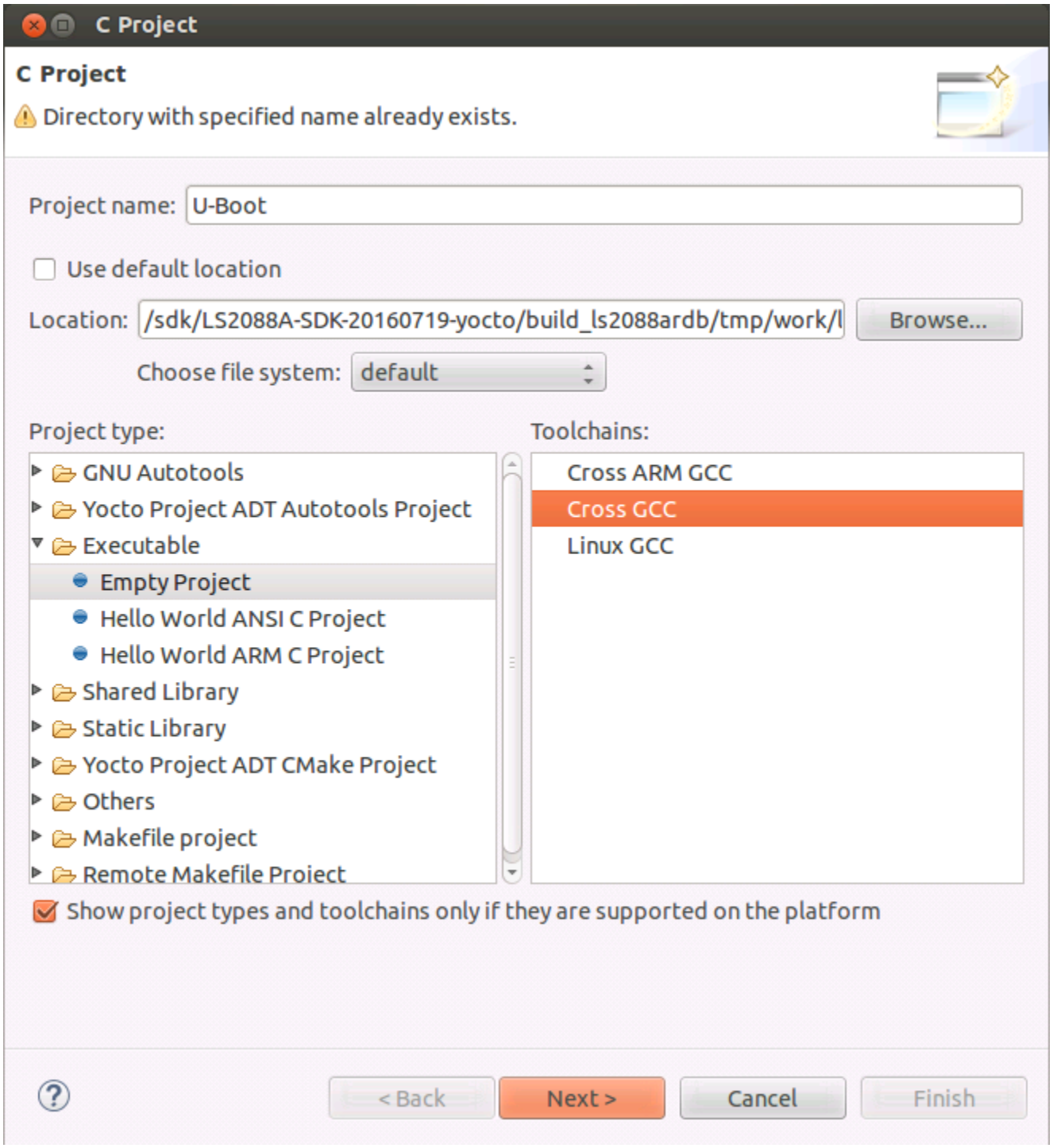


Figure 1. Specify project name

6. Select both **Debug** and **Release** configurations and click **Next**.

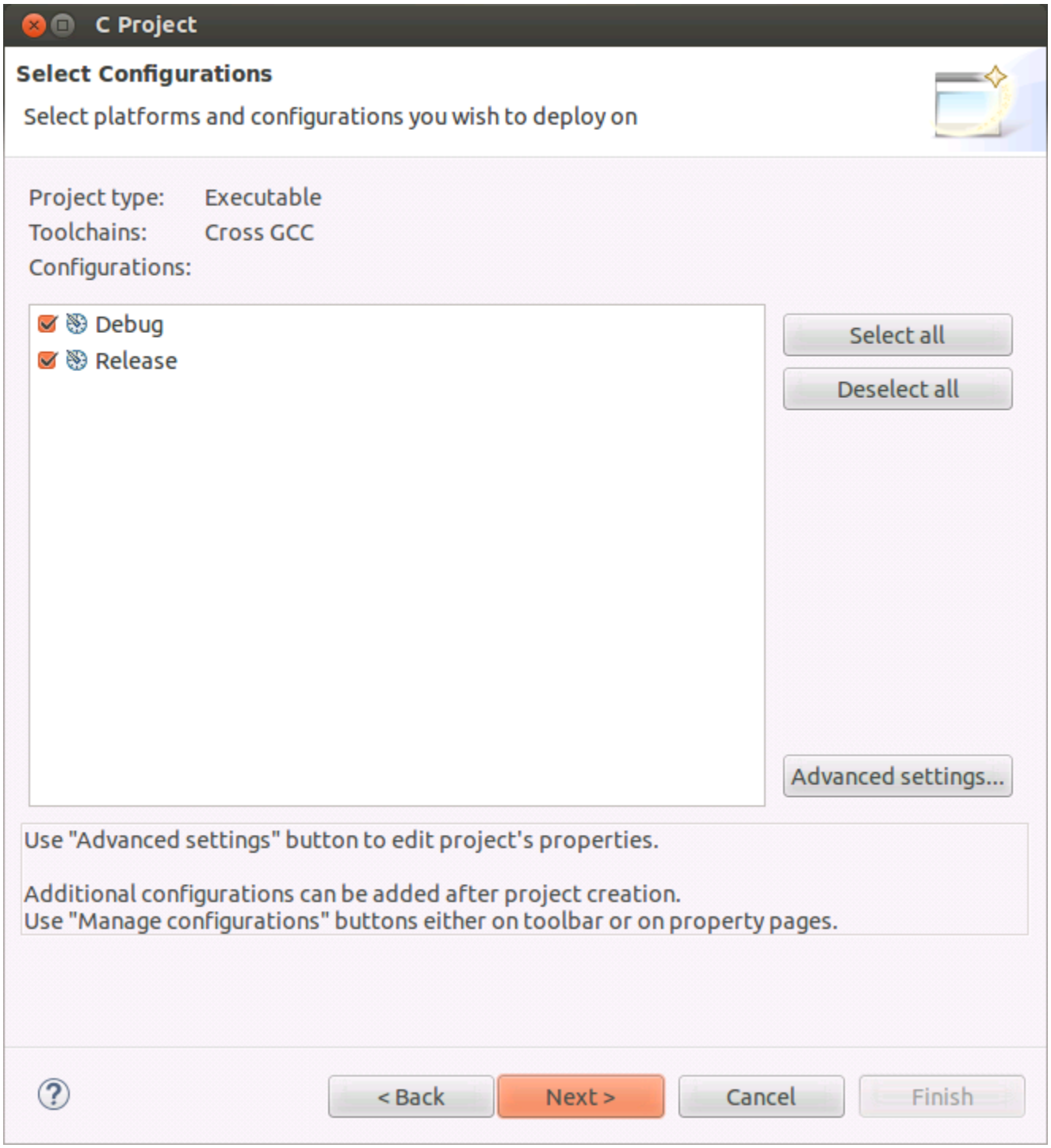


Figure 2. Select Debug and Release configurations

7. Specify the **Cross compiler prefix**, **Cross compiler path** and click **Finish**.

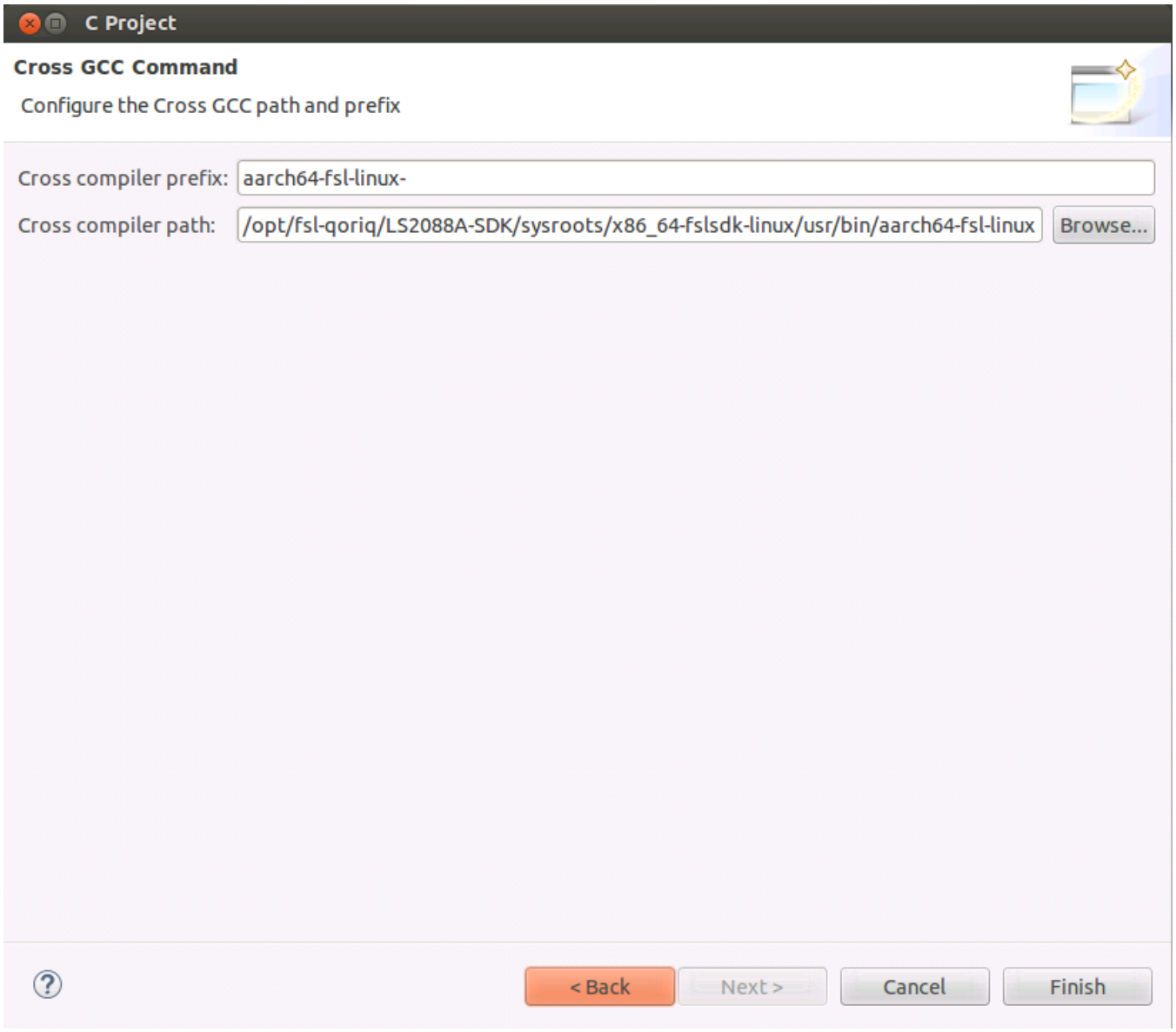


Figure 3. Specify cross compiler prefix and path

8. You can see the newly created project in the **Project Explorer** view.

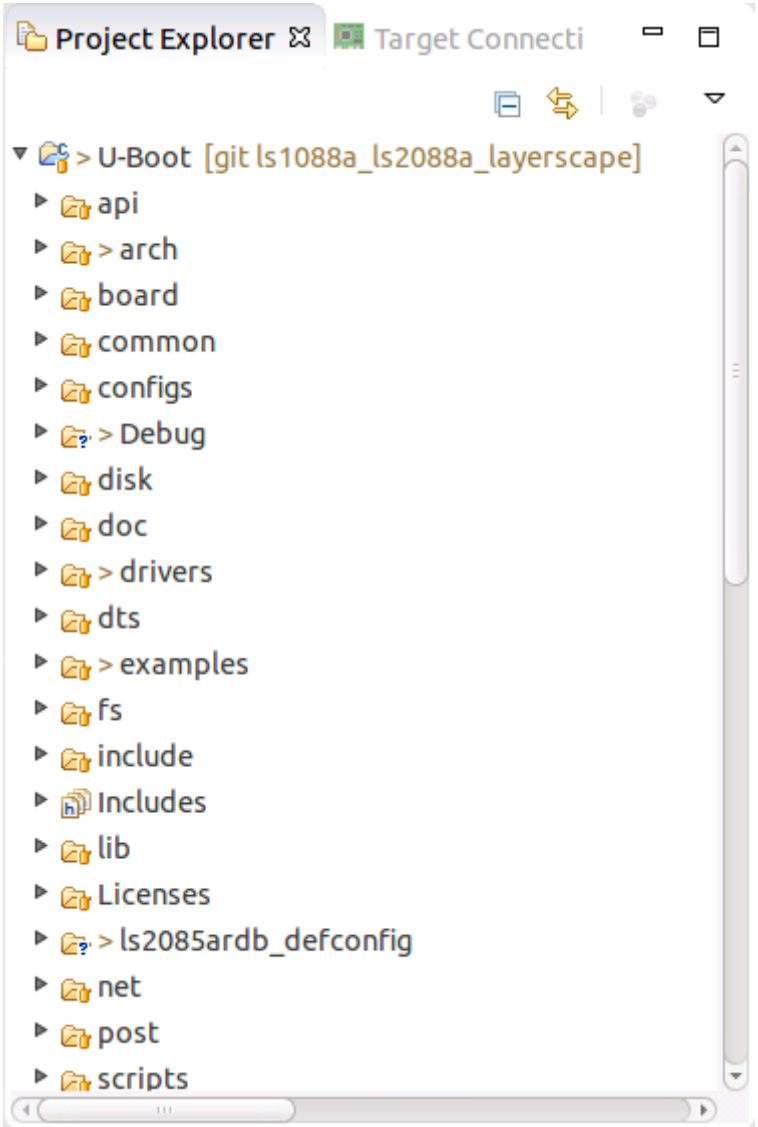


Figure 4. Project Explorer view

9. Go to **Project > Properties > C/C++** build, click the **Builder settings** tab and uncheck **Generate Makefiles automatically**.

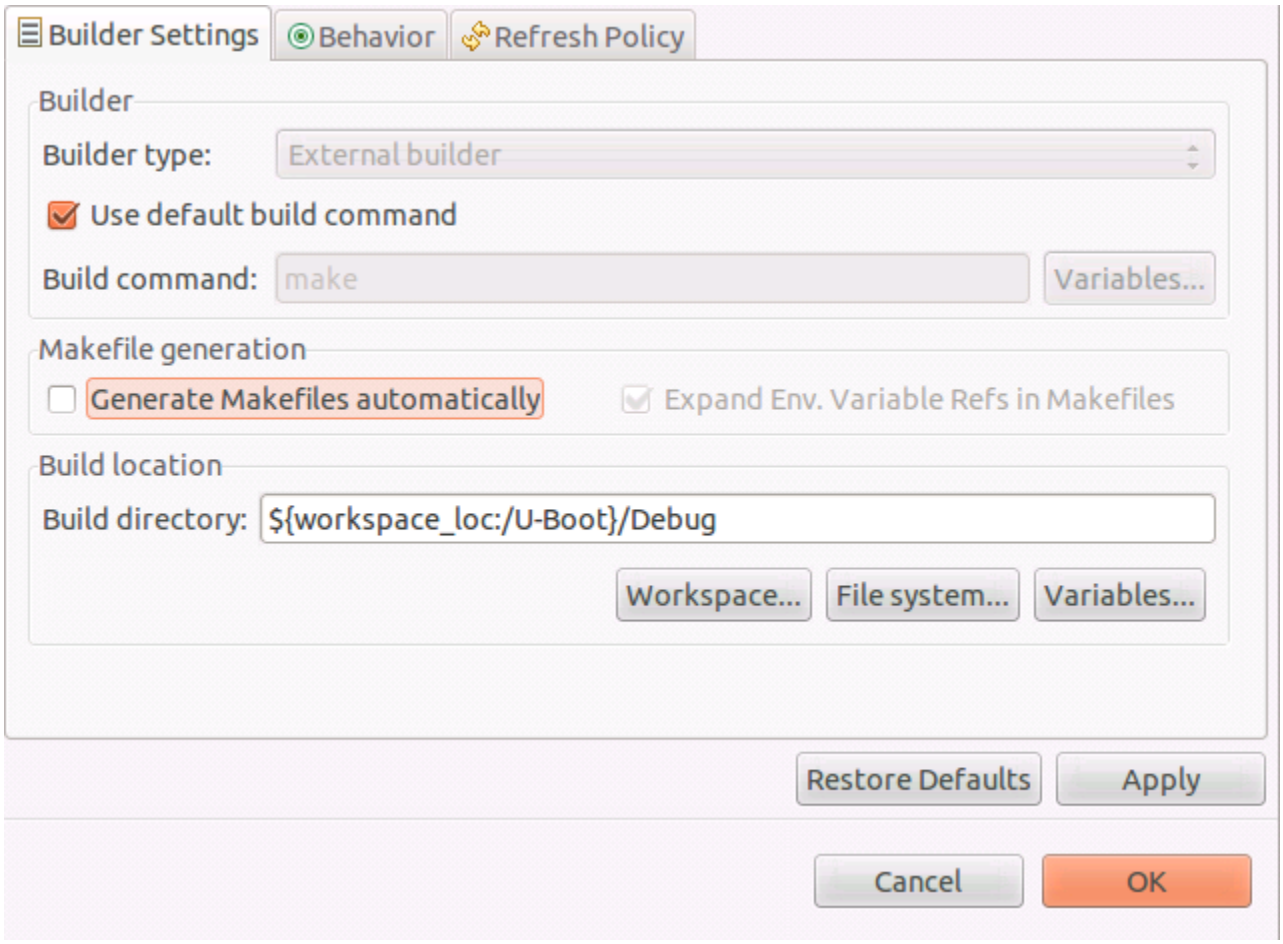


Figure 5. Configure Builder Settings

10. Update the **Build directory** with U-Boot source code path.
11. Click the **Behavior** tab, remove text from the **Build (incremental build)** field and enter **distclean** in the **Clean** field.

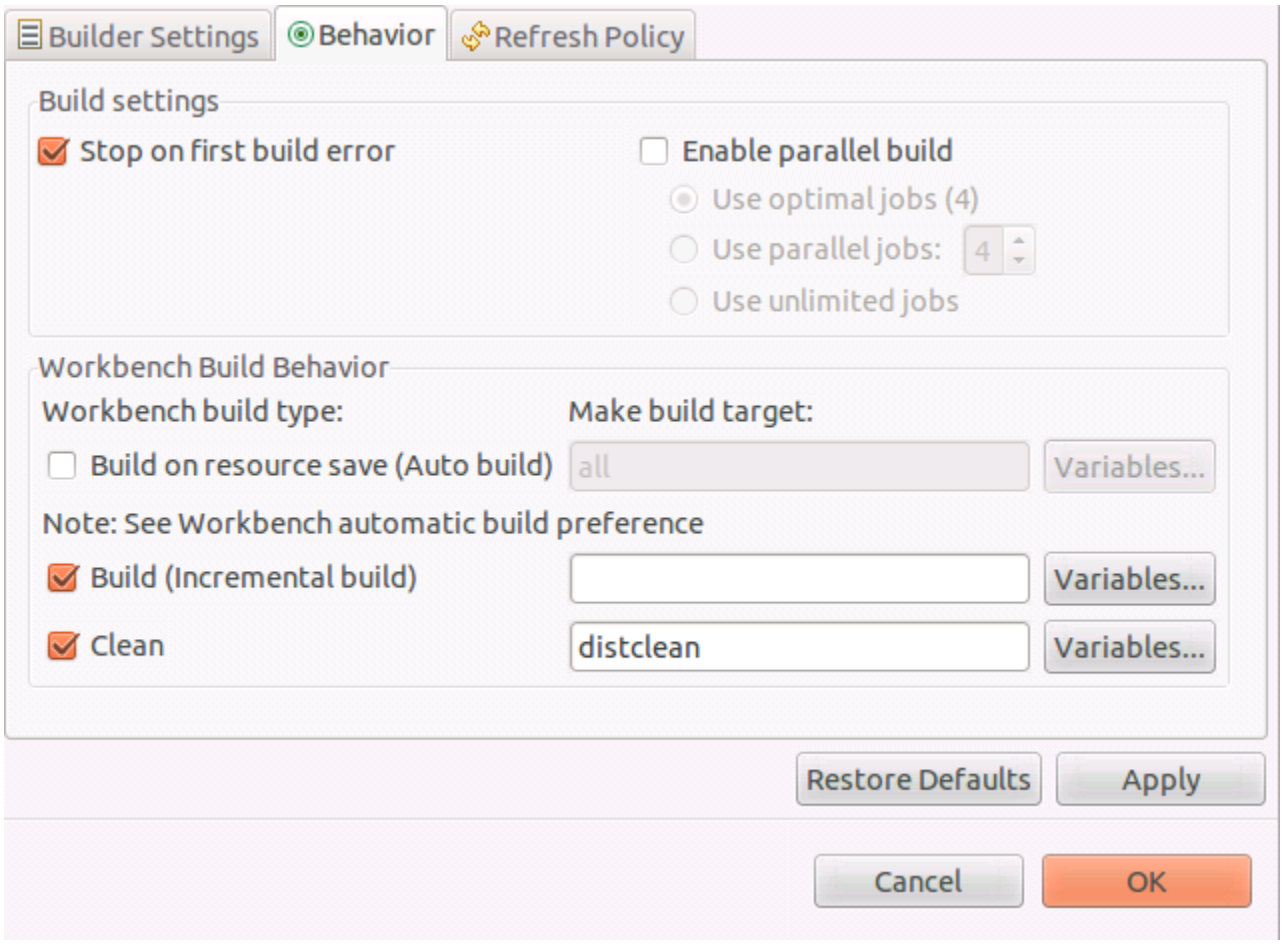


Figure 6. Configure Behavior

12. Go to **Project > Properties > C/C++ build > Environment** and add environmental variables for:

    Name: **CROSS_COMPILE**

    Value: **aarch64-fsl-linux-**

    Click **Add to all configuration**

    Name: **ARCH**

    Value: **arm64**

    Click **Add to all configuration**

    Name: **SDKTARGETSYSROOT**

    Value: **/opt/fsl-qoriq/LS2088A-SDK/sysroots/aarch64-fsl-linux**

    Click **Add to all configuration**

    NOTE

    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.

    Name: **PATH**

    Value: **/opt/fsl-qoriq/LS2088A-SDK/sysroots/x86_64-fslsdk-linux/usr/bin:/opt/fsl-qoriq/LS2088A-SDK/sysroots/x86_64-fslsdk-linux/usr/bin/aarch64-fsl-linux:/usr/sbin:/bin**

Click **Add to all configuration**

Name: **KCFLAGS**

Value: **"--sysroot=${ SDKTARGETSYSROOT }"**

Click **Add to all configuration**

NOTE

When SDK standalone toolchain is built in other location than default, it is possible that other environmental variables must be set. Check the error from **Console** view and add the necessary variables.

13. Go to **Project > Properties > C/C++ build > Settings** and uncheck **Elf Parser** and check **GNU Elf Parser**.

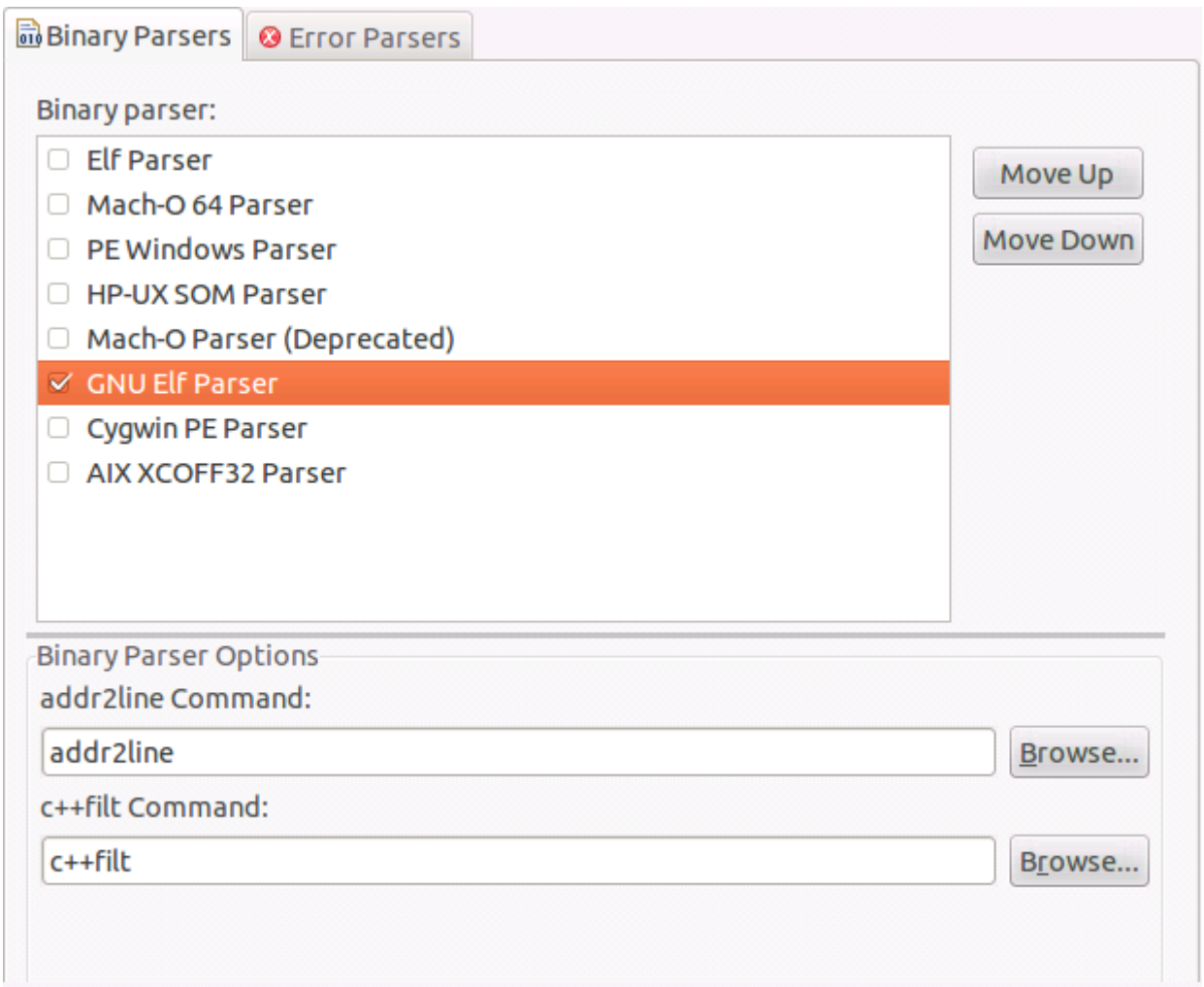
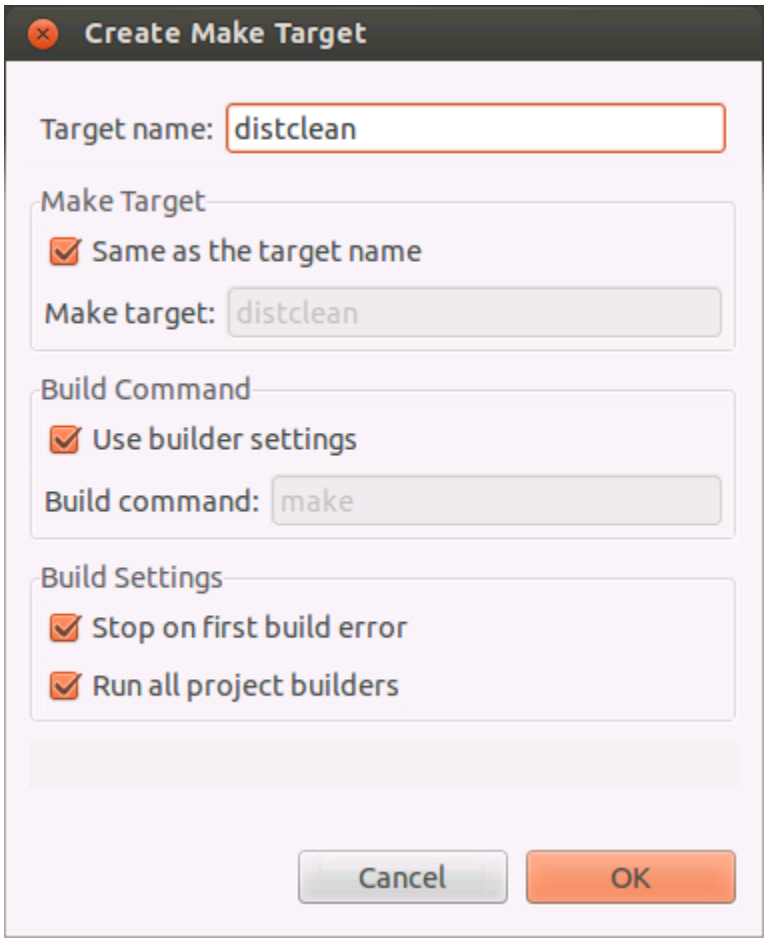


Figure 7. Configure Binary Parsers

5 Building U-boot using CodeWarrior for ARMv8

To build U-Boot using CodeWarrior for ARMv8, two build activities must be created under **Project > Make Target > Build** from the menu bar.


Create Make Target
Target name: distclean
Make Target
Same as the target name
Make target: distclean
Build Command
Use builder settings
Build command: make
Build Settings
Stop on first build error
Run all project builders
Cancel
OK

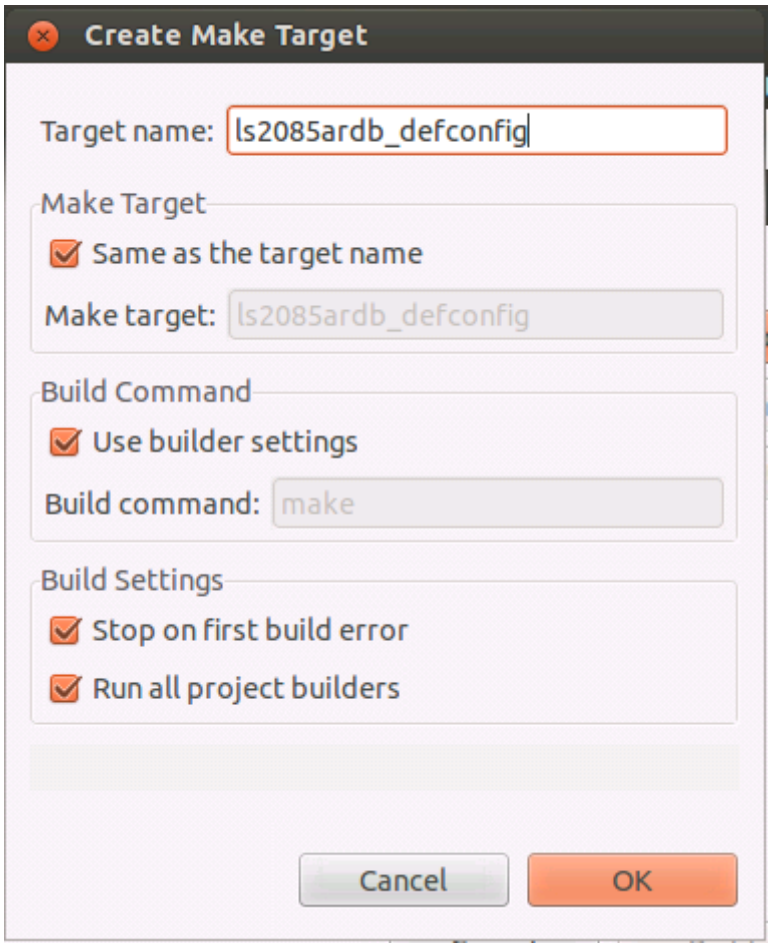


Figure 8. Create Make Target

Once configured we have two build targets.

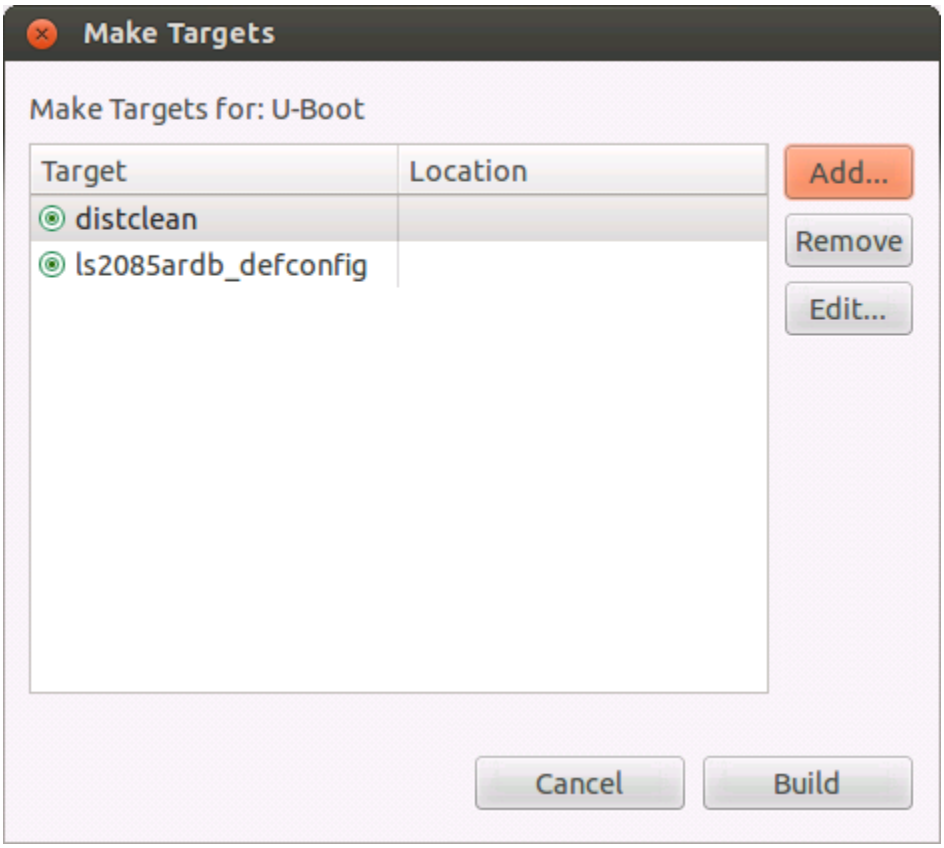


Figure 9. Build targets

Go to **Project > Make Target > Build**, select **distclean** and click **Build**. A “make distclean” command will run removing all the object and temporary files. Below message will be displayed when build is complete in the **Console** view.

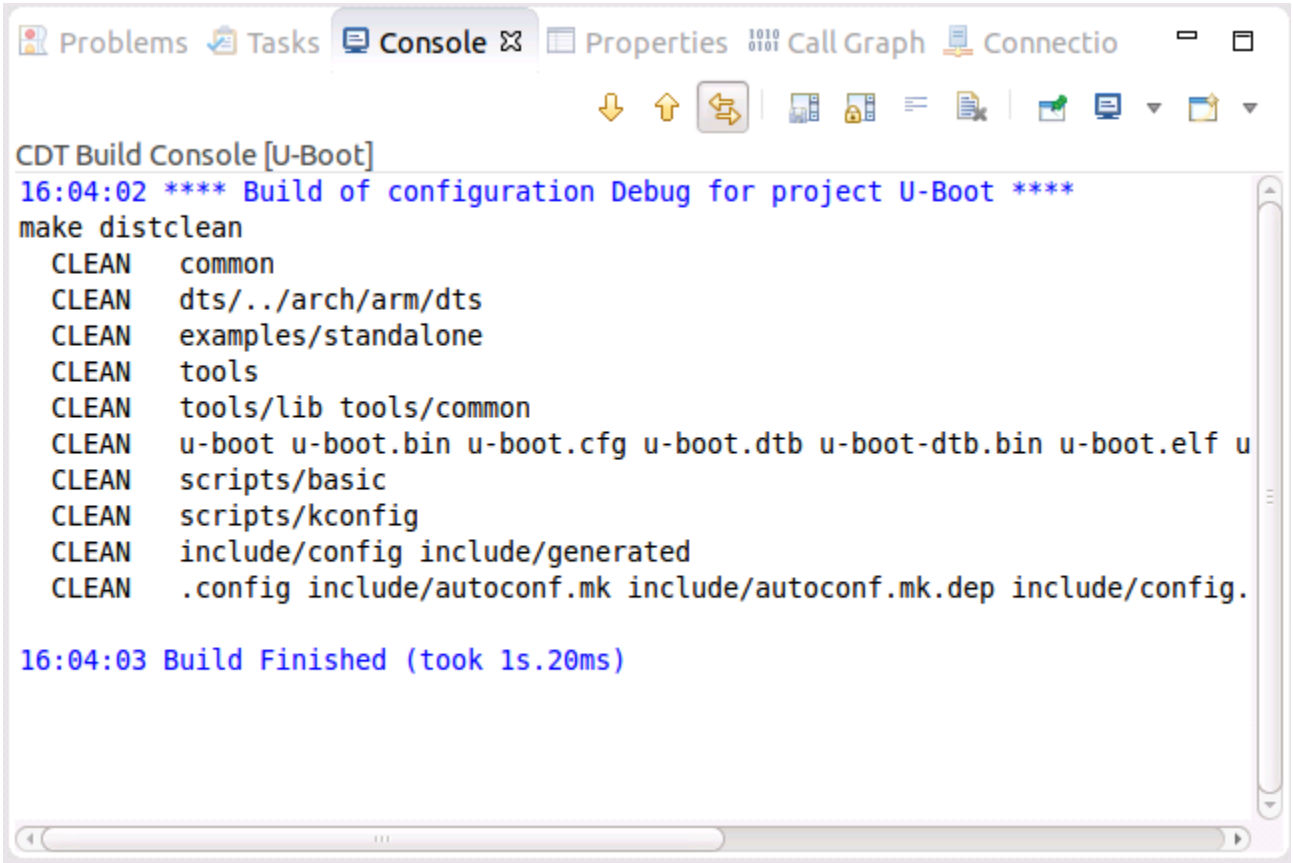


Figure 10. Console view

Go again to **Project > Make Target > Build**, select **ls2085ardb_defconfig** and click **Build**. A ”make ls2085ardb_defconfig” command will run and configure the U-Boot to be built for the LS2088ARDB board in this case.

Figure 11. Console view

To build U-Boot, go to **Project > Build Project** from the menu bar. Below message will be displayed when build is complete in the **Console** view.

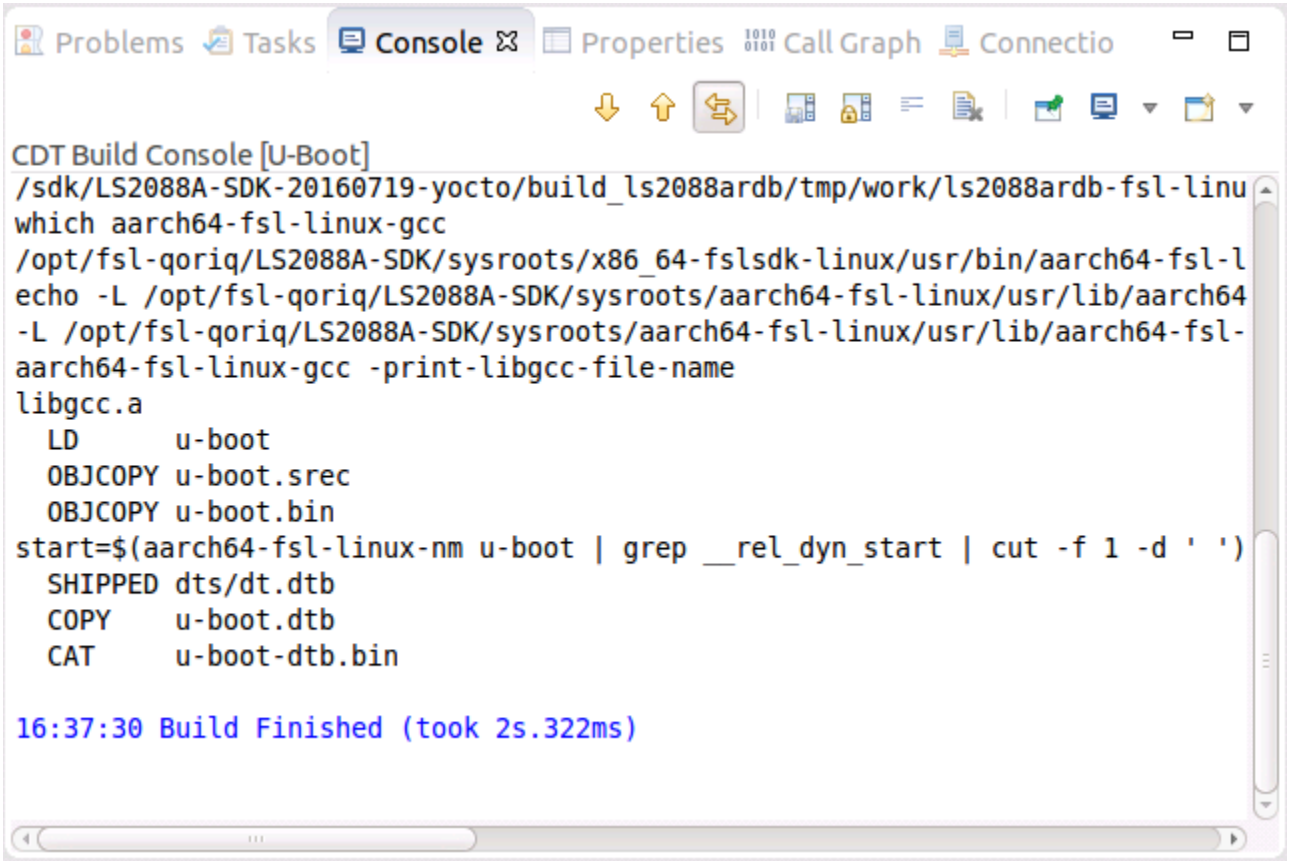


Figure 12. Console view

How to Reach Us:

Home Page:
nxp.com

Web Support:
nxp.com/support

Document Number AN5347
Revision 0, 10/2016