

Building Kinetis example from command line

Rev1.0, updated September 29, 2016

Introduction

This tutorial describes how to build a project from command-line, using the GCC build tools included in the Kinetis Design Studio (KDS) tool from NXP. KDS uses open-source components including the Eclipse IDE and GCC build tools. The Eclipse IDE can be configured to automatically create make files for the project. After KDS creates these make files, they can be used to build the project from the command-line, without using the KDS IDE.

This tutorial uses an example for the Kinetis KE04Z microcontroller (MCU), using the FRDM-KE04Z development board. The following software is used in these tutorial steps:

- Microsoft Windows 7 64-bit
- Kinetis Design Studio (KDS) v3.2.0
- [FRDM-KEXX Driver Library Package](#) v1.3

Installing Software

Installing KDS

The latest KDS can be downloaded from <http://www.nxp.com/KDS>. This tutorial assumes KDS is installed to the default directory of C:\Freescale\KDS_v3.2.0. Run the KDS installer with default options to install KDS.

Installing KDS Example Project

This tutorial uses a project example from the [FRDM-KEXX Driver Library Package](#), although any KDS example project can be built from command-line like this. Download the package ZIP file, and extract it. This tutorial extracts the file to C:\Temp, placing the package path at C:\Temp\kexx_drv_lib.

Building Example Project in KDS

The project must first be built within the KDS IDE to generate the make files used for the command-line build. Also, this tutorial includes the optional steps to generate a HEX or BIN file after the project is built.

Import Example Project into KDS

1. Open KDS. If this is the first time using this KDS release, pick a workspace directory location. The workspace can be located anywhere.
2. Within KDS, import a project using the menu **File->Import**.

3. Expand General, and select “Existing Projects into Workspace”. Click the Next button

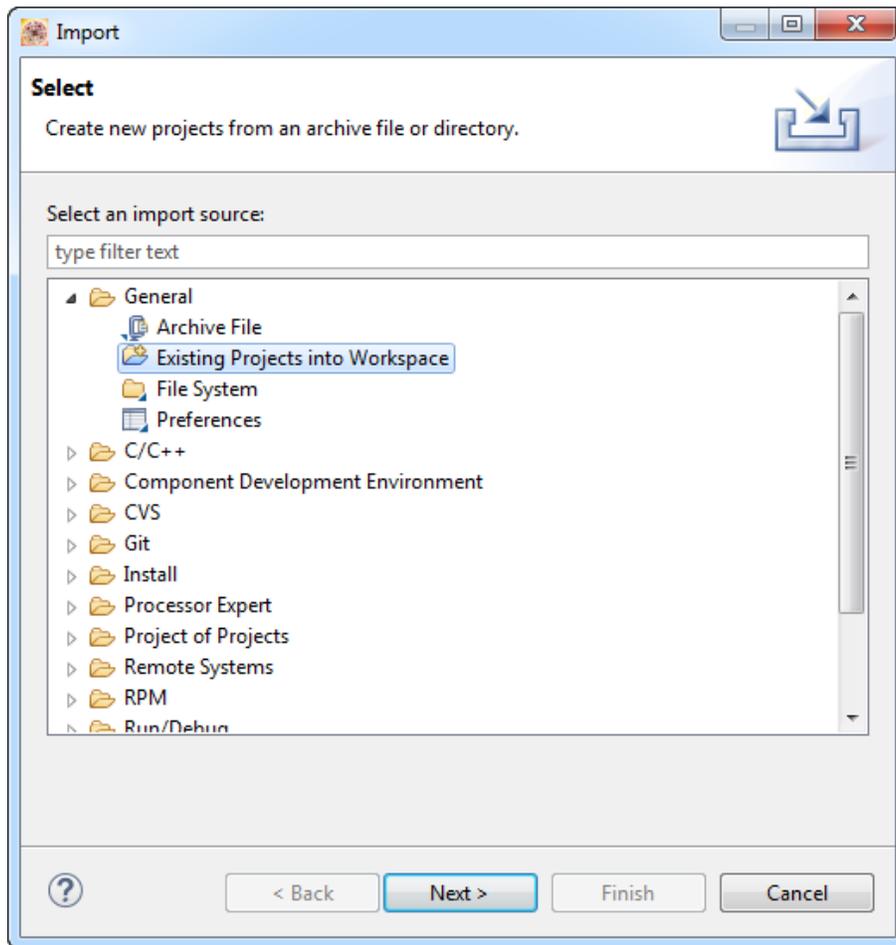


Figure 1: KDS Project Import

- Browse to the root directory of the example project. This tutorial uses the example at **C:\Temp\kexx_drv_lib\build\kds\ke04\RTC_demo**. Ensure the check box is unchecked for “Copy projects into workspace”. Click the Finish button.

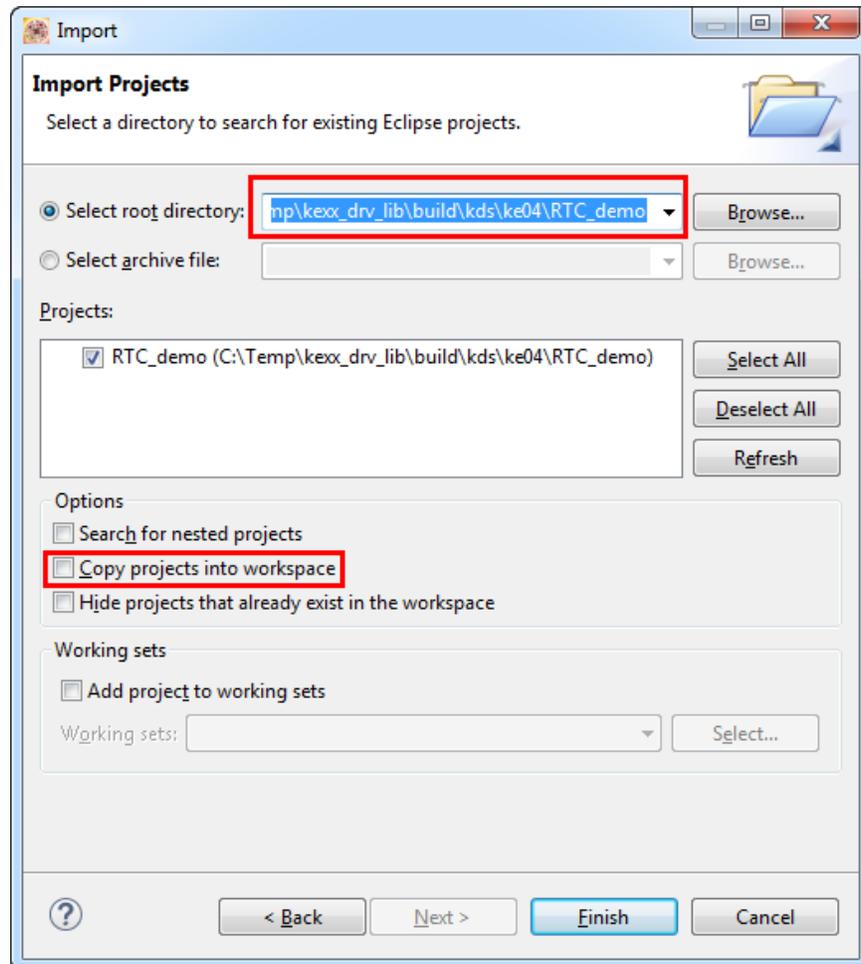


Figure 2: Import Project Directory

- The imported project is added to the Project Explorer view

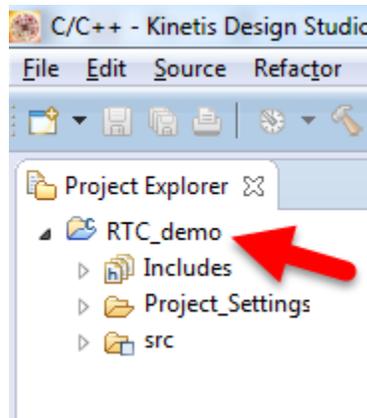


Figure 3: Imported project added to Project Explorer view

6. Open the project properties by **right-clicking the RTC_demo project** in the Project Explorer view, and select **Properties**.
7. Usually, Eclipse projects are already configured to generate makefiles by default. To confirm for this project, select the “C/C++ Build” option at the left. Under the “Builder Settings” tab, there is a check box for “Generate Makefiles automatically”

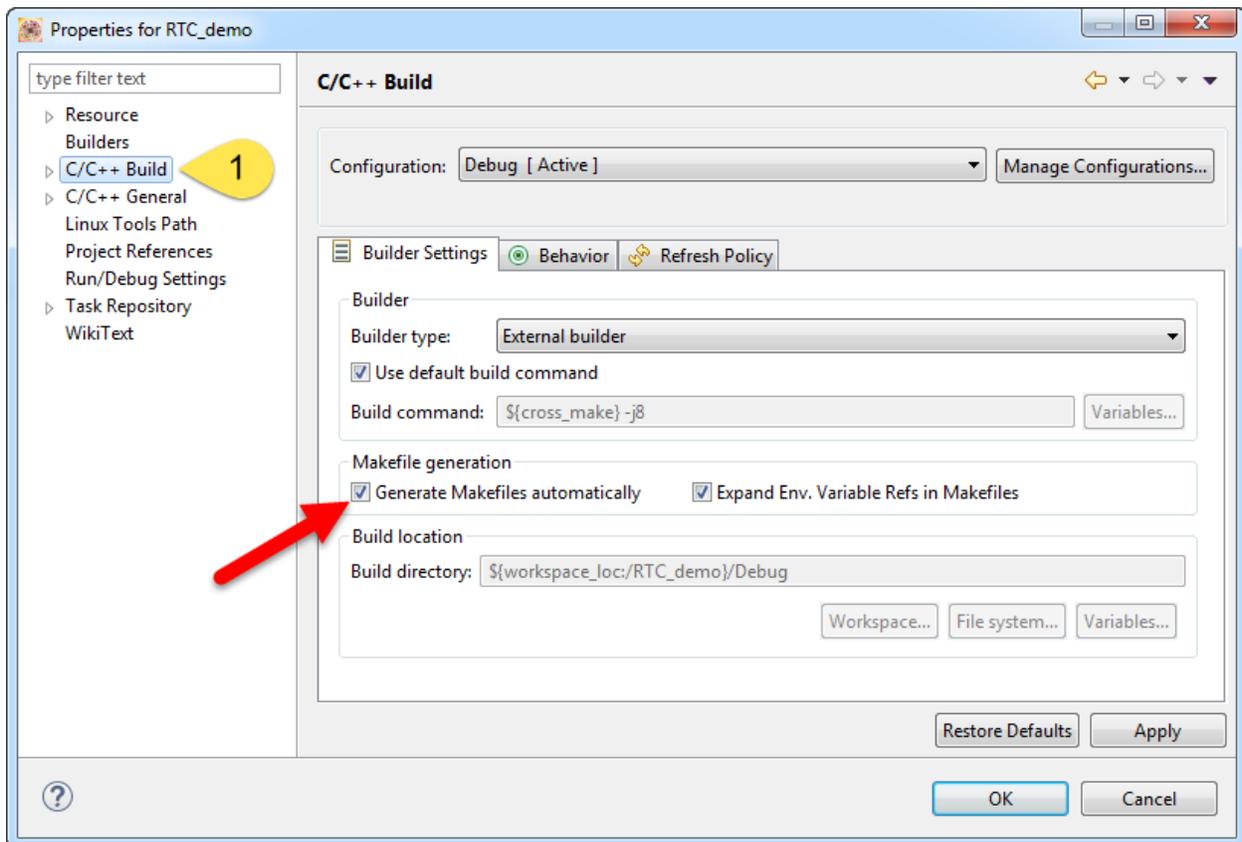


Figure 4: Generating Makefiles automatically in Eclipse

Configure Project to generate HEX/BIN file

This section includes the option steps to configure the KDS project to generate a HEX or BIN file when building the projects. These files can be used to program the MCU.

- With the project properties open, expand the “C/C++ Build” option on the left, and select Settings. Then click the Toolchains tab, and check the box for “Create flash image”. Click the Apply button.

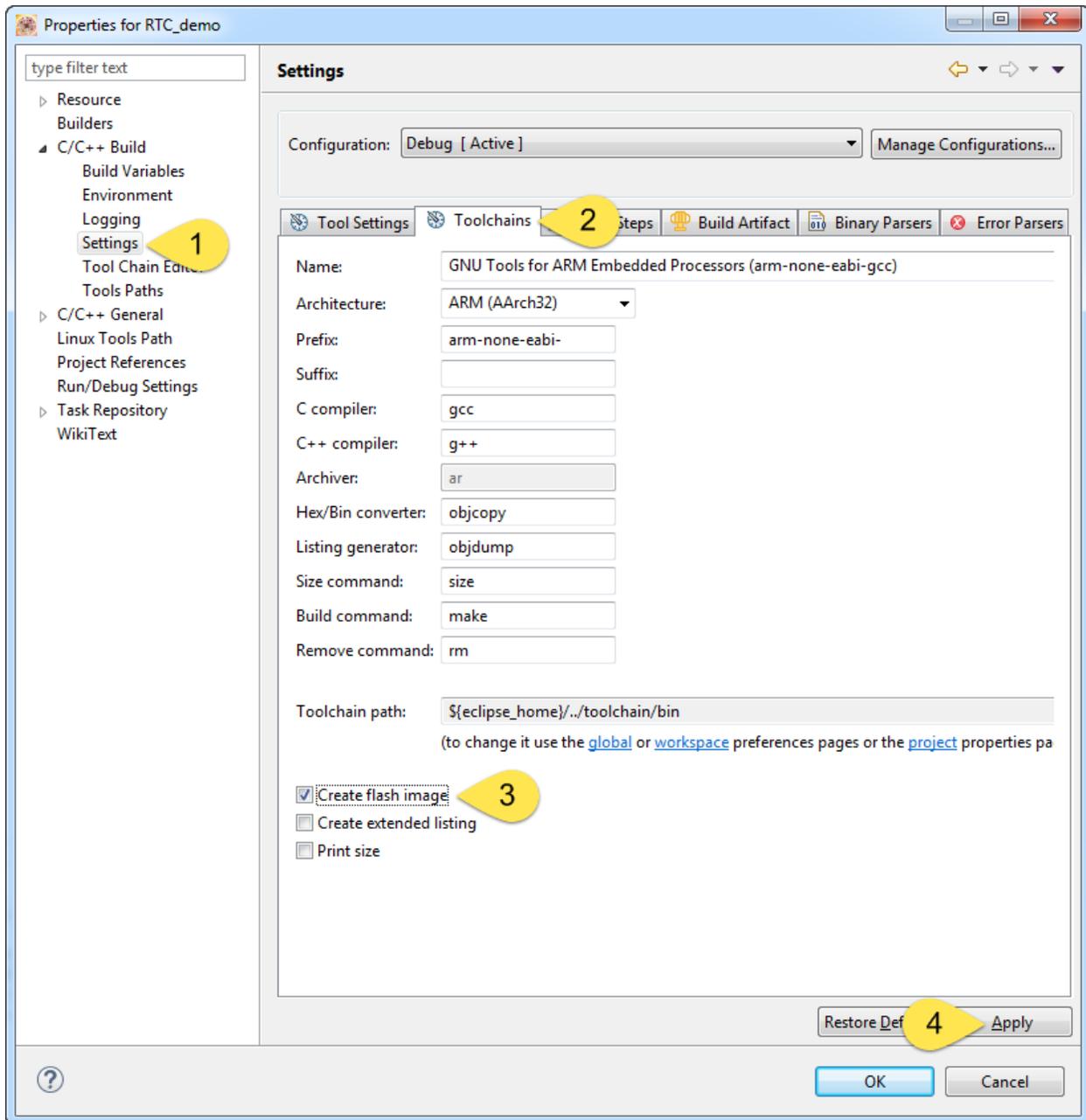


Figure 5: Configure to create flash image

- Click the “Tool Settings” tab, and select the General option under “Cross ARM GNU Create Flash Image”. For the output file format, select the desired option from Intel HEX, Motorola S-record, or a raw binary file. This tutorial selects the HEX file option. Then click the OK button to save these project properties.

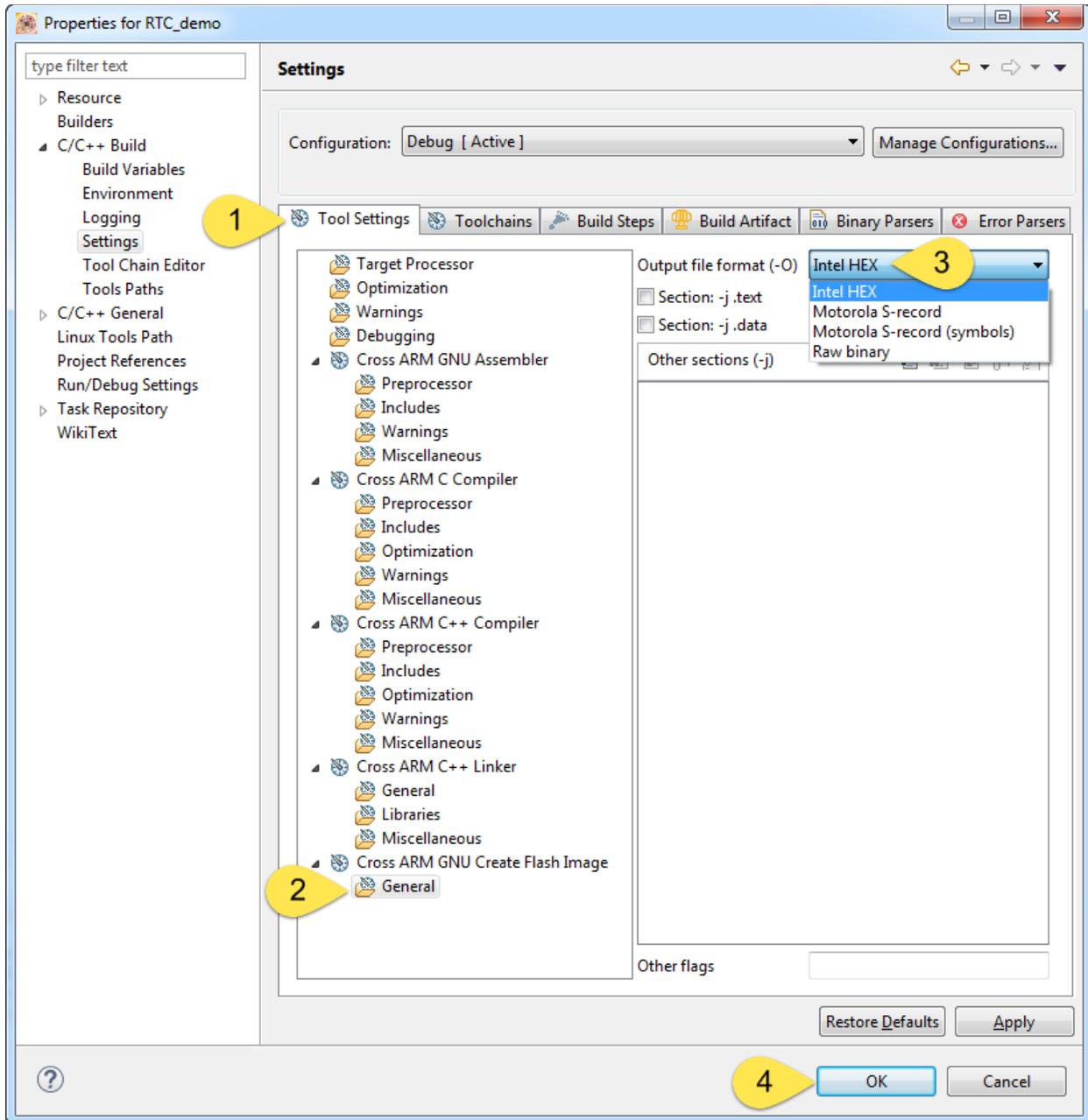


Figure 6: Select Flash Image file type

Build and Clean KDS Project

With the project configured, it can now be built by KDS to generate the makefiles. Once the build is completed, the project is cleaned to prepare it for the command-line build. Cleaning the project removes the object code and application image files generated by KDS, but leaves the makefiles.

10. Back in the KDS Project Explorer view, ensure the RTC_demo project is selected.

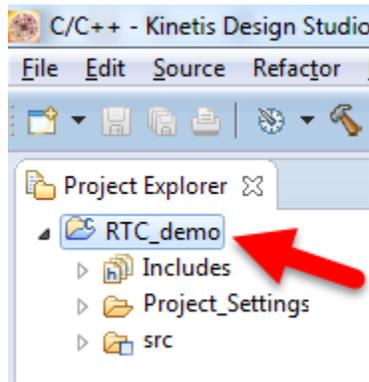


Figure 7: Select project in Project Explorer view

11. Use the hammer icon on the KDS toolbar to build the project. When building the project for the first time, you can use the pull-down arrow next to the hammer icon to select which build configuration to build. In this example, the Debug build configuration is selected and built.

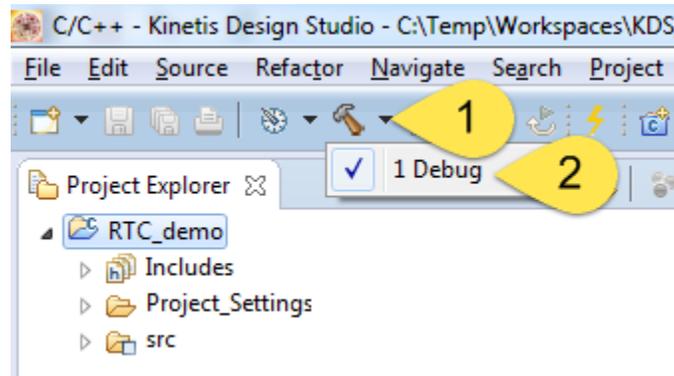


Figure 8: Build project

12. There should be no errors in the project build. The Problems view in KDS shows any errors or warnings when building the project. If no errors are listed, the project built successfully. The project is also built if any warnings are listed.

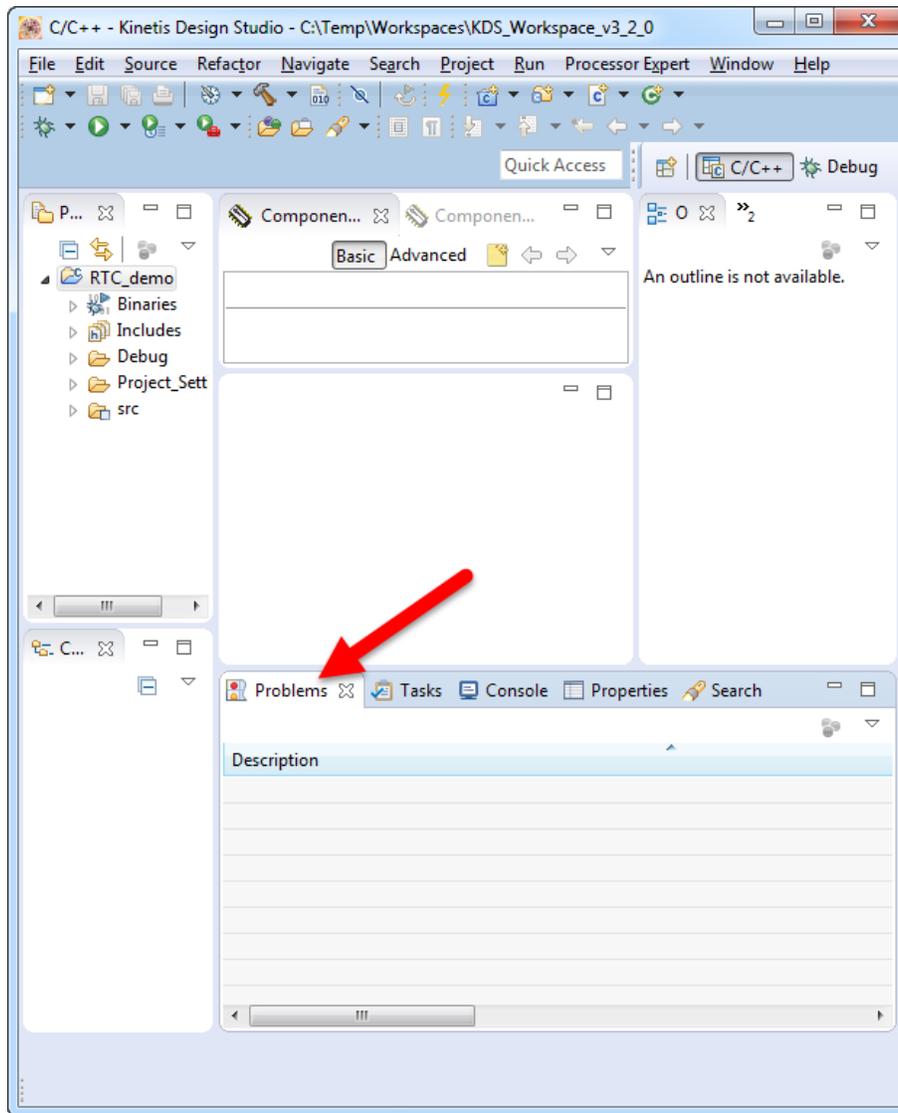


Figure 9: KDS Problems view

13. The Console view gives more details on the build process and results. In this screenshot, the end of the Console text is shown, which shows KDS built the application image file RTC_demo.elf, and generated the flash image file RTC_demo.hex.

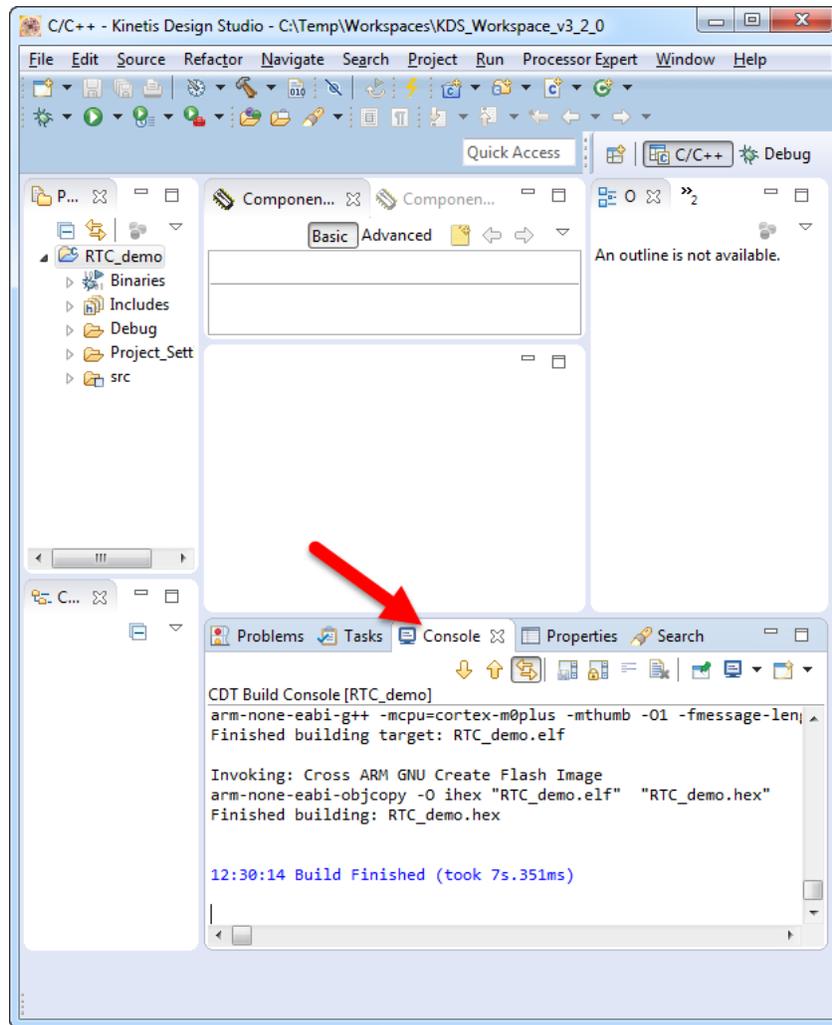


Figure 10: KDS Console view after building

14. Back in the Project Explorer view, you will find the generated makefiles and application image files, like the .ELF and .HEX files. Under the RTC_demo project, expand the directory for the Debug build configuration that was just built. The .ELF and .HEX files are located here, along with some makefiles.

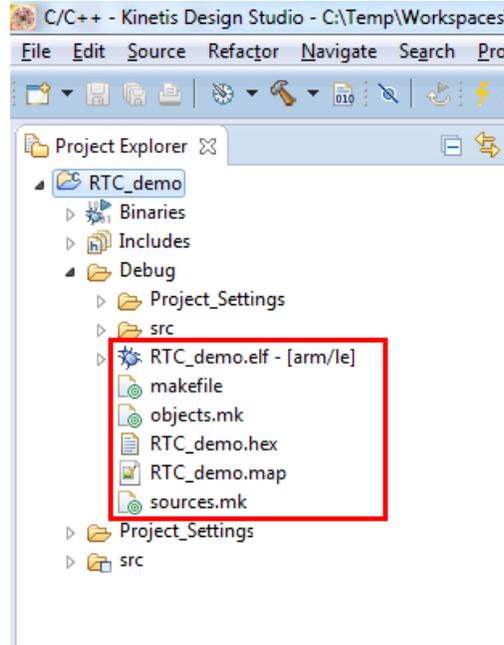


Figure 11: Makefiles and application image files

15. This step cleans the project, which will force the command-line build in the later section to rebuild the entire project. Right-click the RTC_Demo project, and select “**Clean Project**”.
16. Cleaning the project removes the application image .ELF and .HEX files, but leaves the makefiles. The project can now be built from the command-line

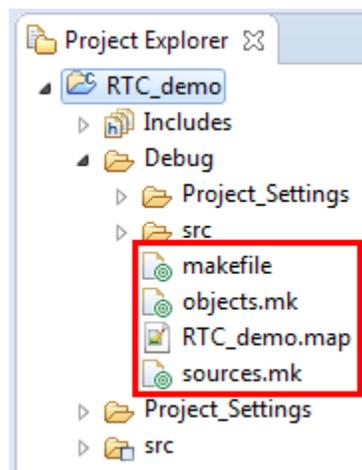


Figure 12: Cleaned project

Updating Windows PATH Environment Variable

Before doing a command-line build, the Windows PATH Environment variable needs to be updated. This will tell Windows what path directories to search for the GCC executables used during the command-line build.

17. In Windows, click the Start menu, and right-click on **Computer**. Select **Properties**.
18. Click “Advanced System Settings”
19. Click the Advanced tab, and click the “**Environment Variables**” button.

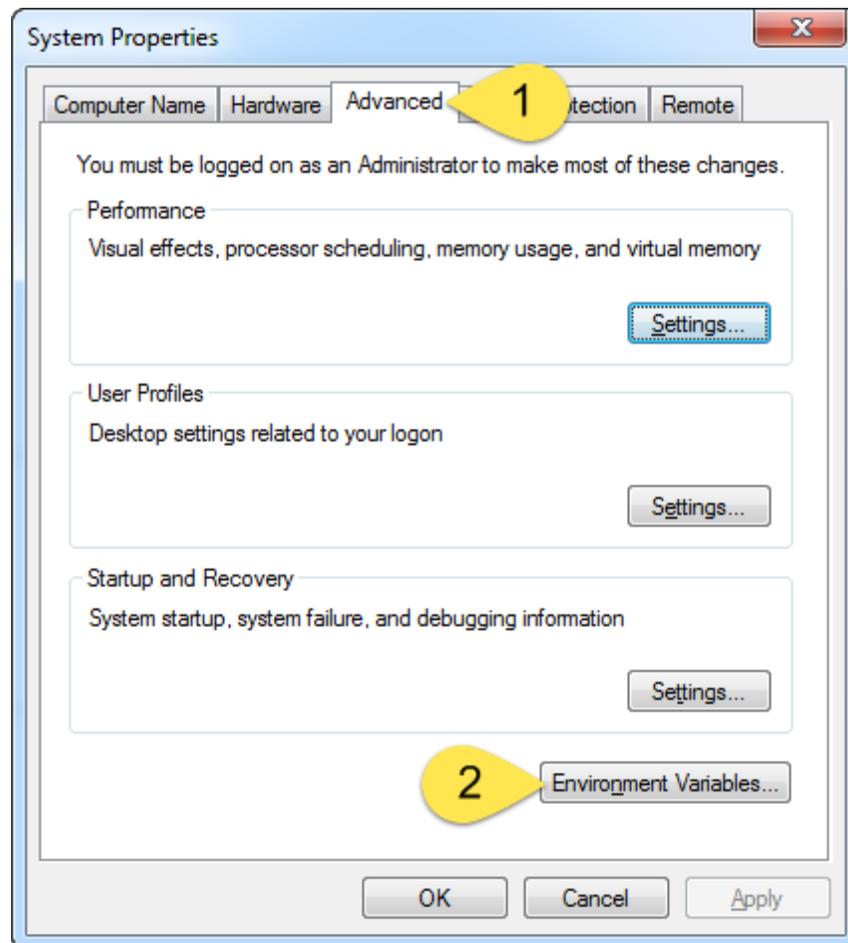


Figure 13: Windows System Properties

20. Scroll down in the User variables section and find the **Path variable**. Select the Path variable, and click the **Edit button**.

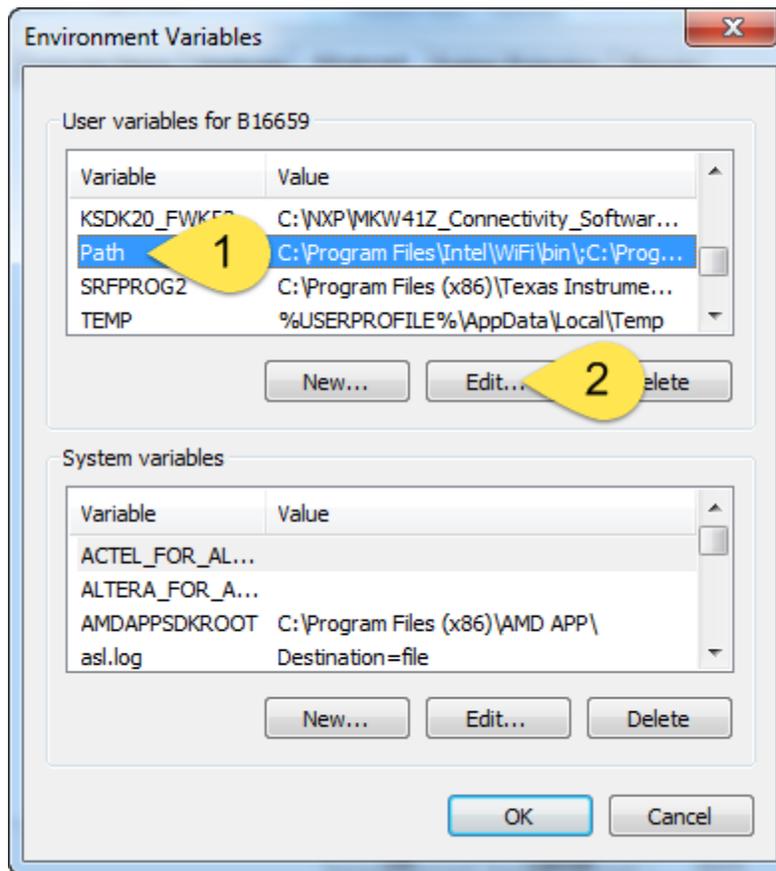


Figure 14: Edit Path variable

21. Add the path to the KDS toolchain executables to the end of the path variable. Place a semicolon ';' at the end of the current paths if not already there, and add the new path. In this tutorial, the path to add is **C:\Freescale\KDS_v3.2.0\Toolchain\bin**

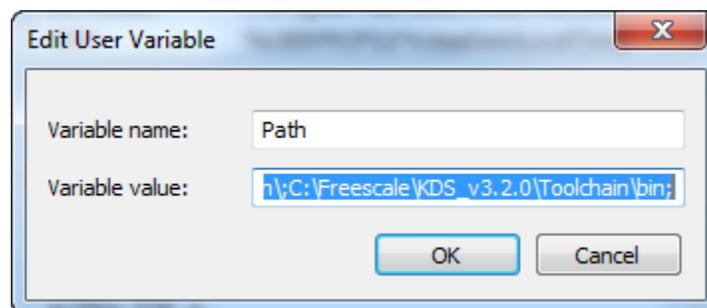


Figure 15: Added path to Path variable

22. Click the three OK buttons to save this change to the Windows System settings.

Building Project from command-line

Now the project can be built using the Windows command-line

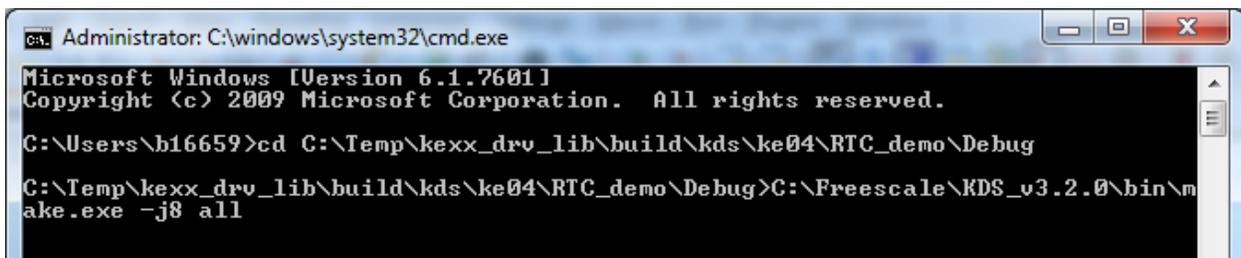
23. Open the Windows Command Line window. Click the Start menu. In the “Search programs and files” field, type `cmd`

24. In the Command Line Window, change the directory to the build configuration directory under the project directory. In this tutorial, this is the Debug directory under the RTC_demo project directory. Type in this command below:

```
cd C:\Temp\kexx_drv_lib\build\kds\ke04\RTC_demo\Debug
```

25. Build the project using the typed command below

```
C:\Freescale\KDS_v3.2.0\bin\make.exe -j8 all
```

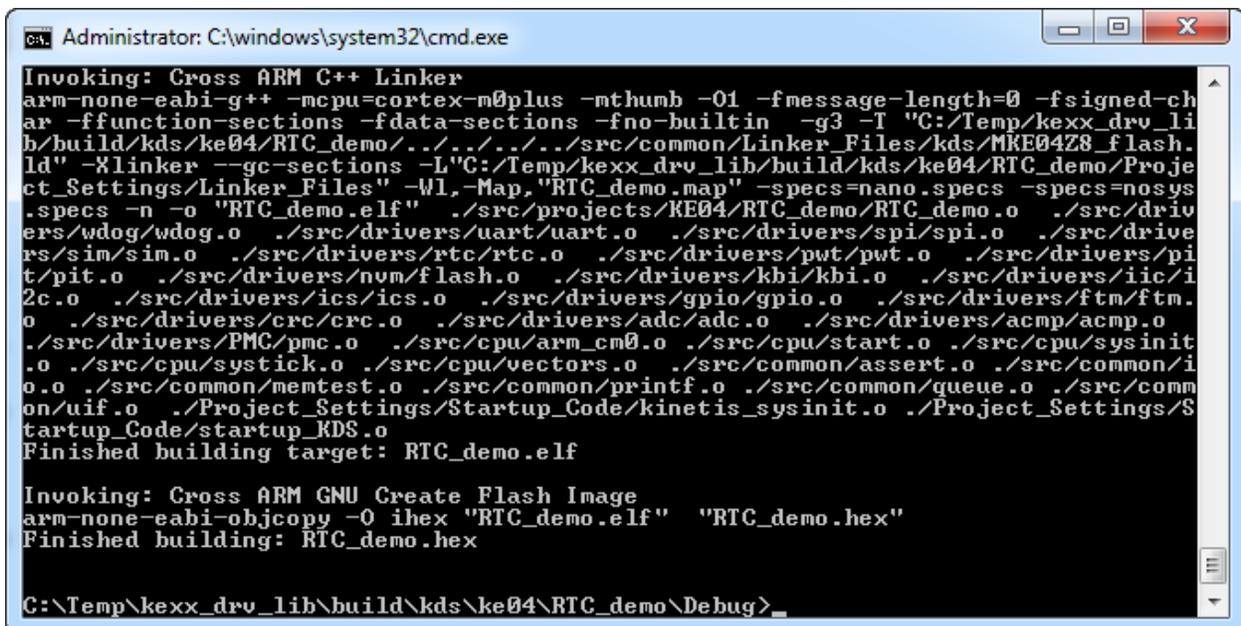


```
Administrator: C:\windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\b16659>cd C:\Temp\kexx_drv_lib\build\kds\ke04\RTC_demo\Debug
C:\Temp\kexx_drv_lib\build\kds\ke04\RTC_demo\Debug>C:\Freescale\KDS_v3.2.0\bin\make.exe -j8 all
```

Figure 16: Windows command-line commands

26. The project builds, and one again the RTC_demo.elf and RTC_demo.hex files are generated.



```
Administrator: C:\windows\system32\cmd.exe
Invoking: Cross ARM C++ Linker
arm-none-eabi-g++ -mcpu=cortex-m0plus -mthumb -O1 -fmessage-length=0 -fsigned-char -ffunction-sections -fdata-sections -fno-builtin -g3 -I "C:\Temp\kexx_drv_lib\build\kds\ke04\RTC_demo\..\..\..\src\common\Linker_Files\kds\MKE04Z8_flash.ld" -xlinker --gc-sections -L"C:\Temp\kexx_drv_lib\build\kds\ke04\RTC_demo\Project_Settings\Linker_Files" -Wl,-Map,"RTC_demo.map" -specs=nano.specs -specs=nosys.specs -n -o "RTC_demo.elf" ./src/projects/KE04/RTC_demo/RTC_demo.o ./src/drivers/wdog/wdog.o ./src/drivers/uart/uart.o ./src/drivers/spi/spi.o ./src/drivers/sim/sim.o ./src/drivers/rtc/rtc.o ./src/drivers/pwt/pwt.o ./src/drivers/pit/pit.o ./src/drivers/nvm/flash.o ./src/drivers/kbi/kbi.o ./src/drivers/iic/i2c.o ./src/drivers/ics/ics.o ./src/drivers/gpio/gpio.o ./src/drivers/ftm/ftm.o ./src/drivers/crc/crc.o ./src/drivers/adc/adc.o ./src/drivers/acmp/acmp.o ./src/drivers/PMC/pmc.o ./src/cpu/arm_cm0.o ./src/cpu/start.o ./src/cpu/sysinit.o ./src/cpu/systick.o ./src/cpu/vectors.o ./src/common/assert.o ./src/common/i.o ./src/common/mentest.o ./src/common/printf.o ./src/common/queue.o ./src/common/uif.o ./Project_Settings/Startup_Code/kinetis_sysinit.o ./Project_Settings/Startup_Code/startup_KDS.o
Finished building target: RTC_demo.elf

Invoking: Cross ARM GNU Create Flash Image
arm-none-eabi-objcopy -O ihex "RTC_demo.elf" "RTC_demo.hex"
Finished building: RTC_demo.hex

C:\Temp\kexx_drv_lib\build\kds\ke04\RTC_demo\Debug>
```

Figure 17: Project build complete from command-line