Freescale Semiconductor
Quick Start

Document Number: KDSPGQS
Rev. 1.0.0, 04/2015

Kinetis Design Studio Porting Guide

Contents

1 Introduction....1
2 Create a new project or import an existing project.... 1
3 Importing an existing project.... 2
4 Converting a legacy Kinetis project....5
5 Identifying the conversion errors and warnings....7
6 Manually updating and cleaning the converted project.... 10
7 Building the converted project....11

1 Introduction

The CodeWarrior for Kinetis Design Studio Porting migration assistant helps in migrating legacy Kinetis CodeWarrior GNU/gcc projects to projects configurable in Kinetis Design Studio. To succesfully build an imported projects, you need to manually make some changes to the project content and files. This document lists the steps to convert or import an existing Kinetis project to a Kinetis Design Studio project.

2 Create a new project or import an existing project

The process to import and convert projects is a semi-automated process. This is because the projects settings between CodeWarrior and Kinetis Design Studio are different, the import process outlined in the following sections tries to map the CodeWarrior settings to the Kinetis Design Studio settings.

However, as every project is different and can have different settings, the importer/converter might not be able to properly convert every project, especially if your CodeWarrior projects is highly customized.

In case you tryiong to import\convert a highly customized project, perform the following steps:

1. Create a new project, select File > New > Kinetis Project in Kinetis Design Studio.
2. Add your own sources to the new project.
3. Apply tool chain/build options, if appropriate.

3 Importing an existing project

To import an existing Kinetis project:

NOTE

CodeWarrior for Microcontrollers offers two compilers: legacy Freescale compiler, for Kinetis-K devices and GNU gcc for Kinetis-K and Kinetis-L devices. The legacy Freescale compiler projects settings are very different from the projects settings in Kinetis Design Studio. Therefore, Kinetis Design Studio can import\convert only the CodeWarrior GNU/gcc compiler projects.

1. Select **File > Import**, from the IDE menu.

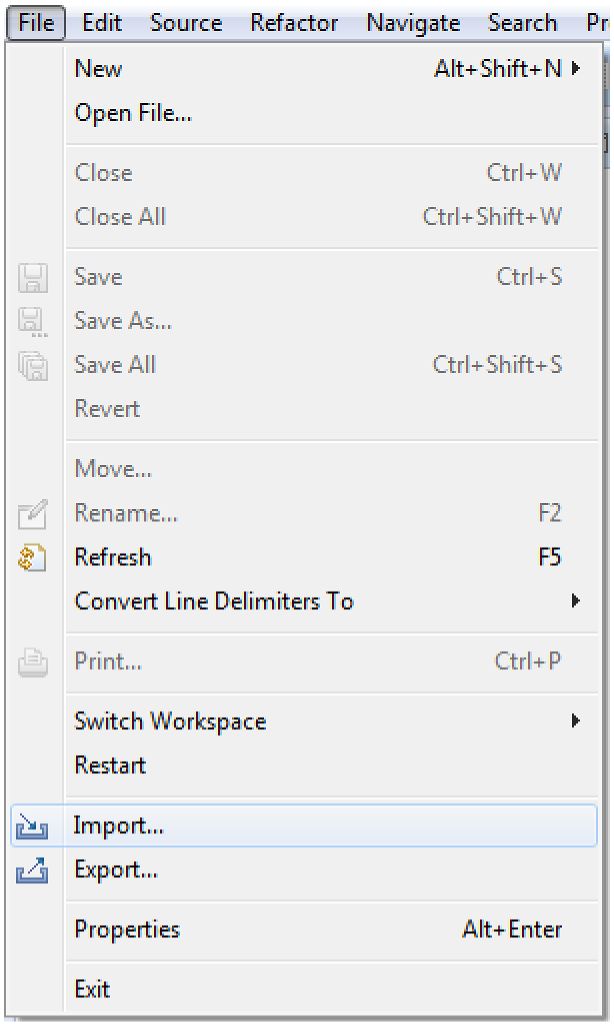


Figure 1. Select import option

The import dialog appears.

2. Expand the **General** tree and select **Existing Projects into Workspace**.

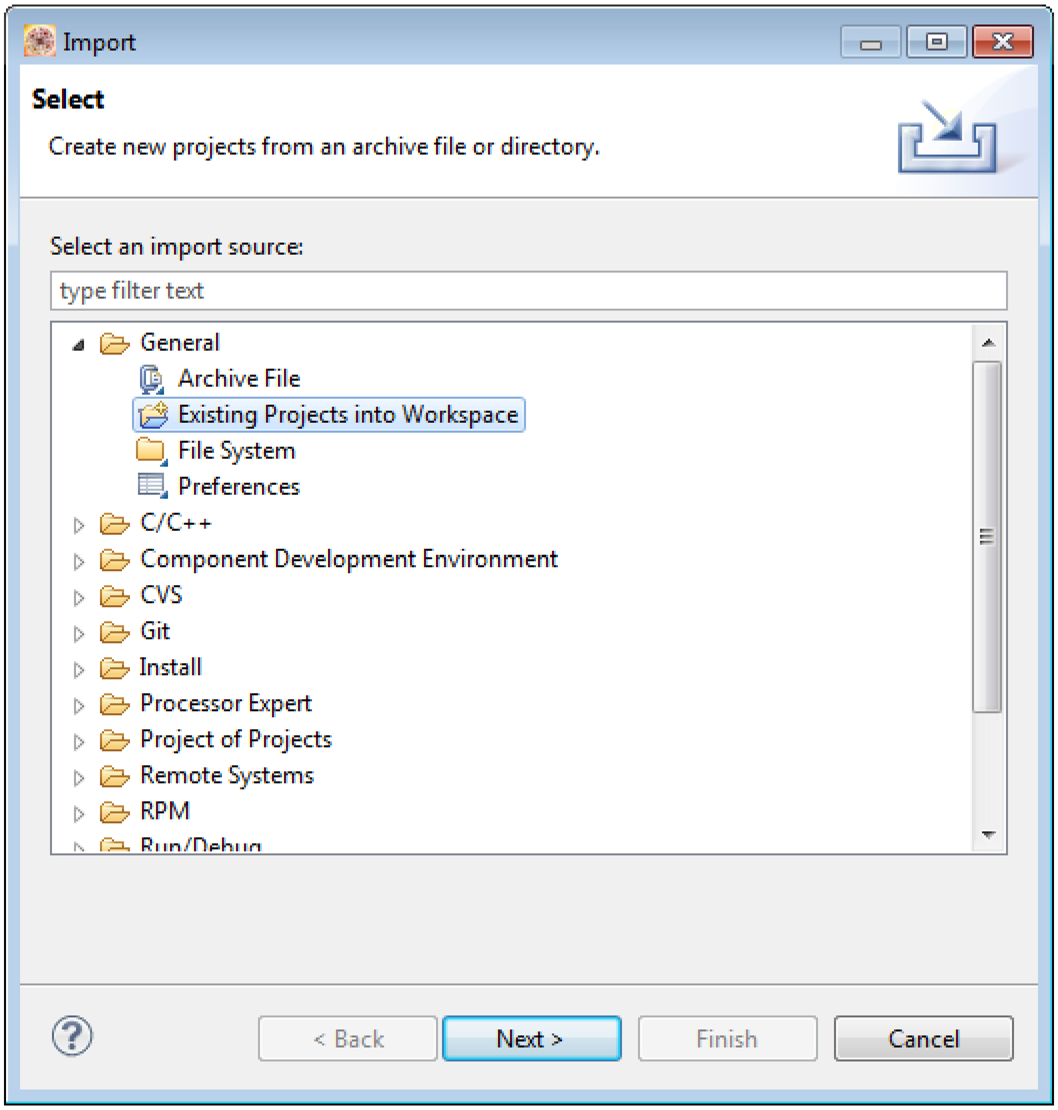
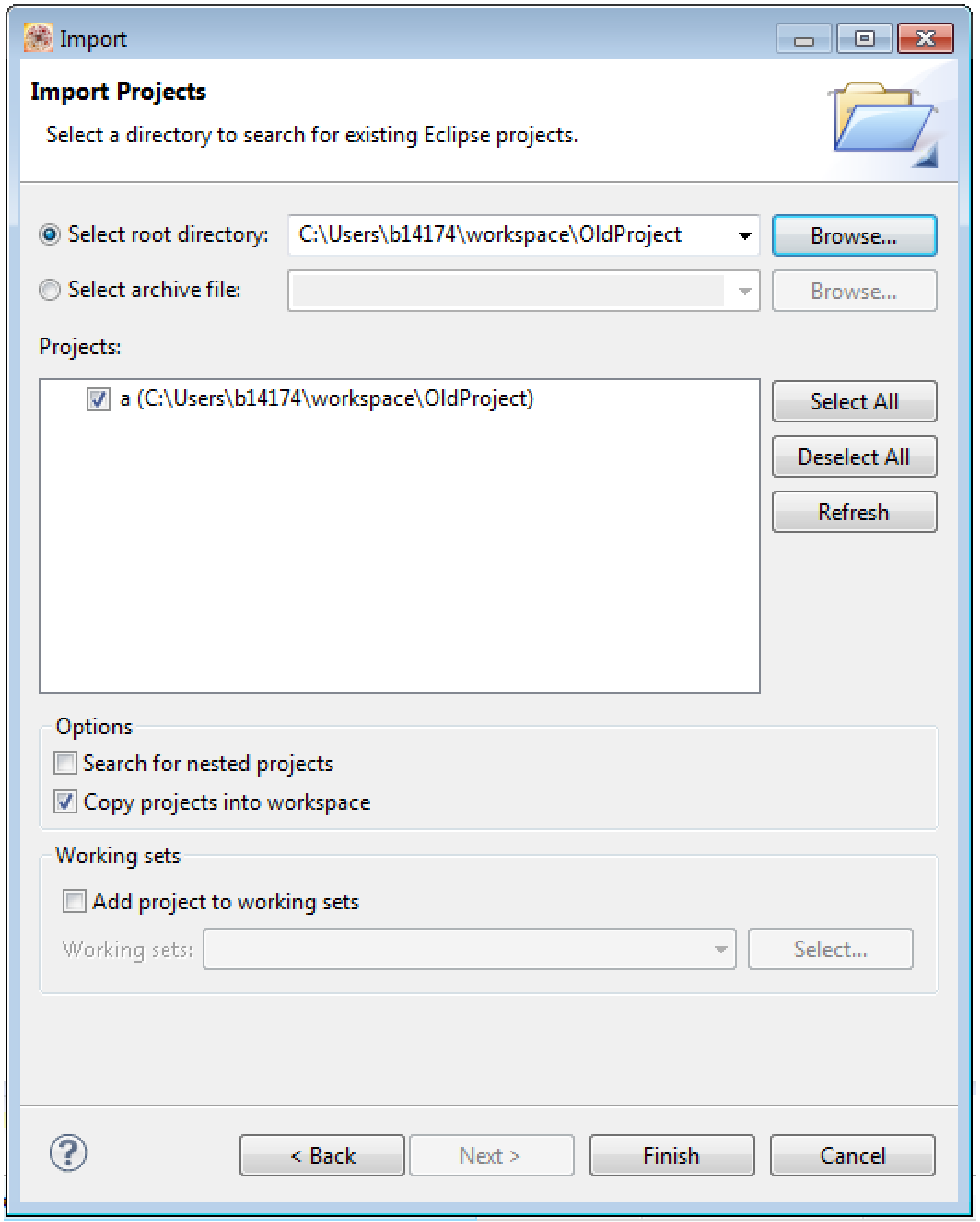


Figure 2. Select an import source

3. Click **Next**.
4. Click **Browse** and select the root directory to search for an existing Eclipse project. You can also choose to import an archive file.
5. Check the **Copy projects into workspace** checkbox. This will import your existing project in the current Workspace. Changes made to the project in your Workspace will not impact your existing project or workspace makes a physical copy of your project. If your project contains links to other folders or locations, you will need to change them later accordingly.



Import

Import Projects

Select a directory to search for existing Eclipse projects.

● Select root directory: C:\Users\b14174\workspace\OldProject

○ Select archive file:

Projects:

- [✓] a (C:\Users\b14174\workspace\OldProject)

Options

- [ ] Search for nested projects
- [✓] Copy projects into workspace

Working sets

- [ ] Add project to working sets

Working sets:

Figure 3. Copy project in workspace

6. Click Finish.
   The imported project appears in the **Project Explorer** view.

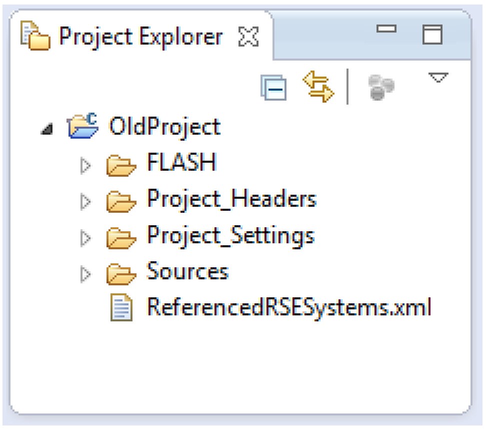


Figure 4. Project Explorer view

4 Converting a legacy Kinetis project

To use the imported legacy Kinetis project in Kinetis Design Studio, you need to convert it. The steps to convert a legacy Kinetis project are:

1. Select the recently imported legacy project you want to convert.

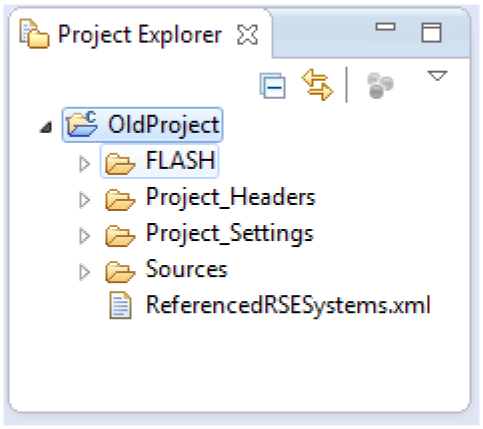


Figure 5. Select the project

2. Right-click and select **Convert CodeWarrior Project file**.

NOTE

The context menu **Convert CodeWarrior project file...** will only work with GNU/gcc CodeWarrior project and not for CodeWarrior projects using the legacy Freescale compiler.

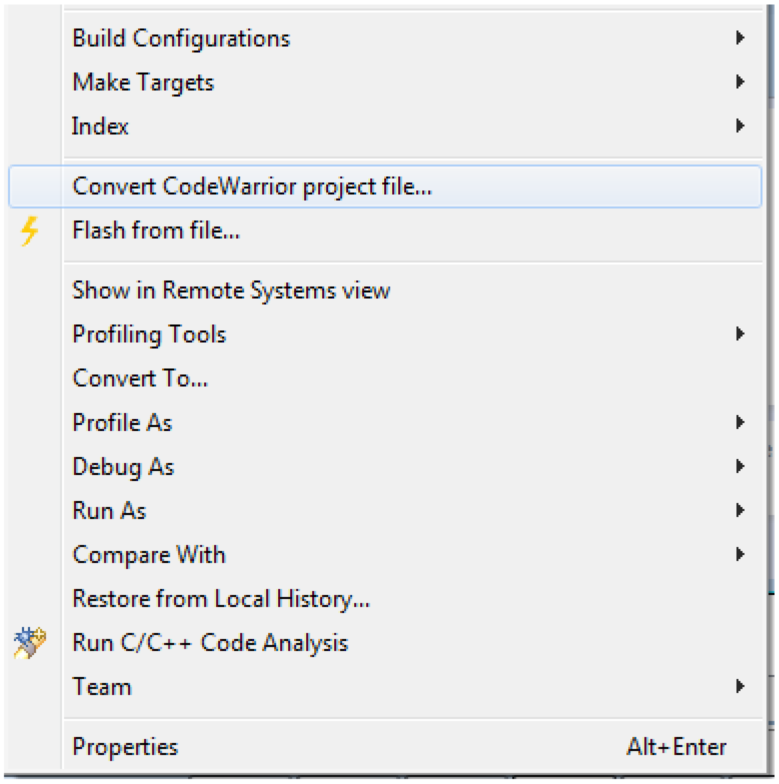
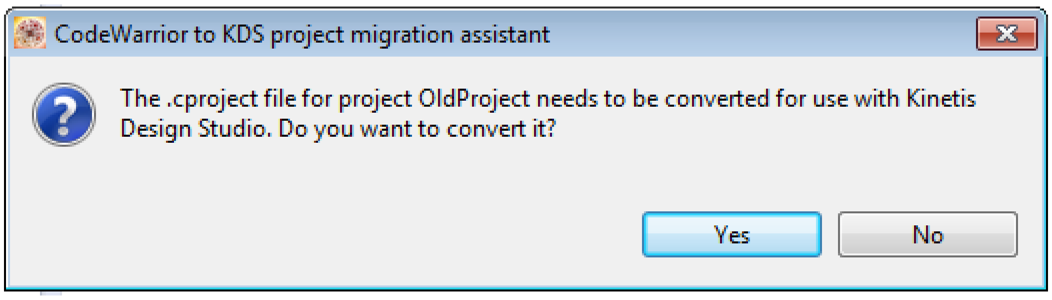


Figure 6. Select Convert CodeWarrior Project file

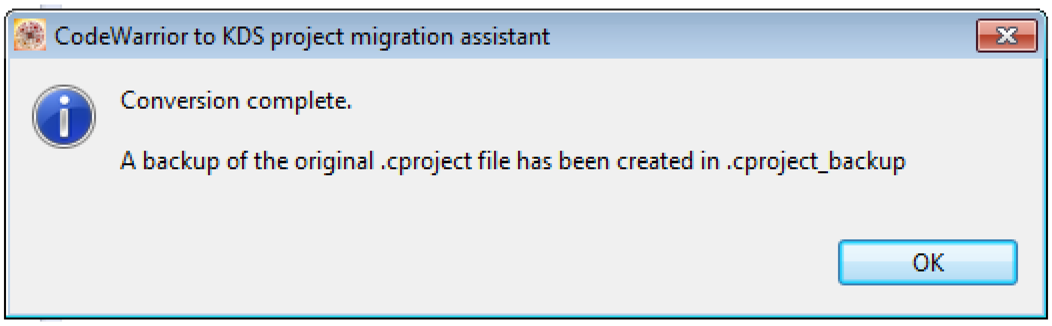
The CodeWarrior to KDS project migration assistant dialog appears and prompts whether you want to convert the project.



The .cproject file for project OldProject needs to be converted for use with Kinetis Design Studio. Do you want to convert it?

Figure 7. CodeWarrior to KDS project migration assistant dialog

3. Click Yes.
   The CodeWarrior to KDS project migration assistant will prompt that the conversion has completed and a backup of the original *cproject file has been created in .cproject_backup.



Conversion complete.

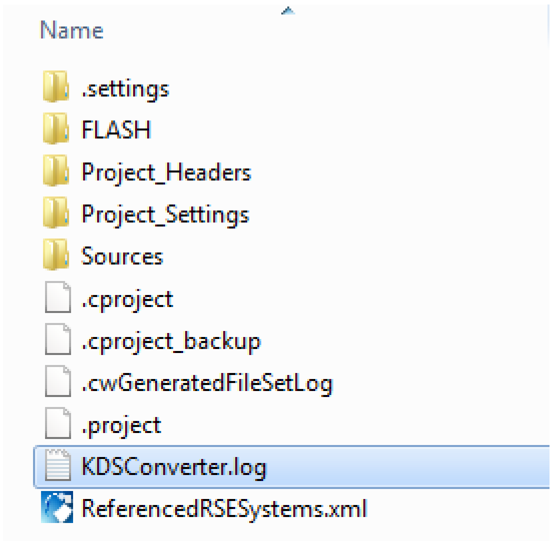
A backup of the original .cproject file has been created in .cproject_backup

Figure 8. Conversion completed

4. Click **OK** to close the CodeWarrior to KDS project migration assistant dialog.

NOTE

>To see the conversion process log, navigate to the project's root directory and open the KDSConverter.log file.



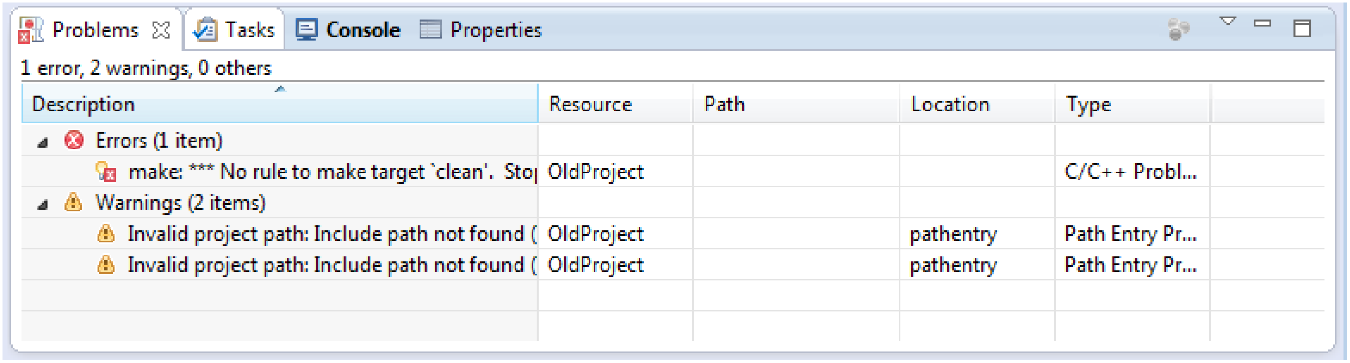
Name
.settings
FLASH
Project_Headers
Project_Settings
Sources
.cproject
.cproject_backup
.cwGeneratedFileSetLog
.project
KDSConverter.log
ReferencedRSESystems.xml

Figure 9. Conversion log

5 Identifying the conversion errors and warnings

To ensure that you can successfully build and configure the converted project, you need to first identify the errors and warnings encountered during the conversion.

The errors and warnings encountered during conversion are listed in the **Problems** view.



1 error, 2 warnings, 0 others

Description	Resource	Path	Location	Type
Errors (1 item)				
make: *** No rule to make target `clean'. Sto	OldProject			C/C++ Probl...
Warnings (2 items)				
Invalid project path: Include path not found (	OldProject		pathentry	Path Entry Pr...
Invalid project path: Include path not found (	OldProject		pathentry	Path Entry Pr...

Figure 10. Problems view

NOTE

The Problems view appears in the IDE by default. However, you can also access and open the **Problems** view by selecting **Window > Show View > Problems** from the IDE menu bar.

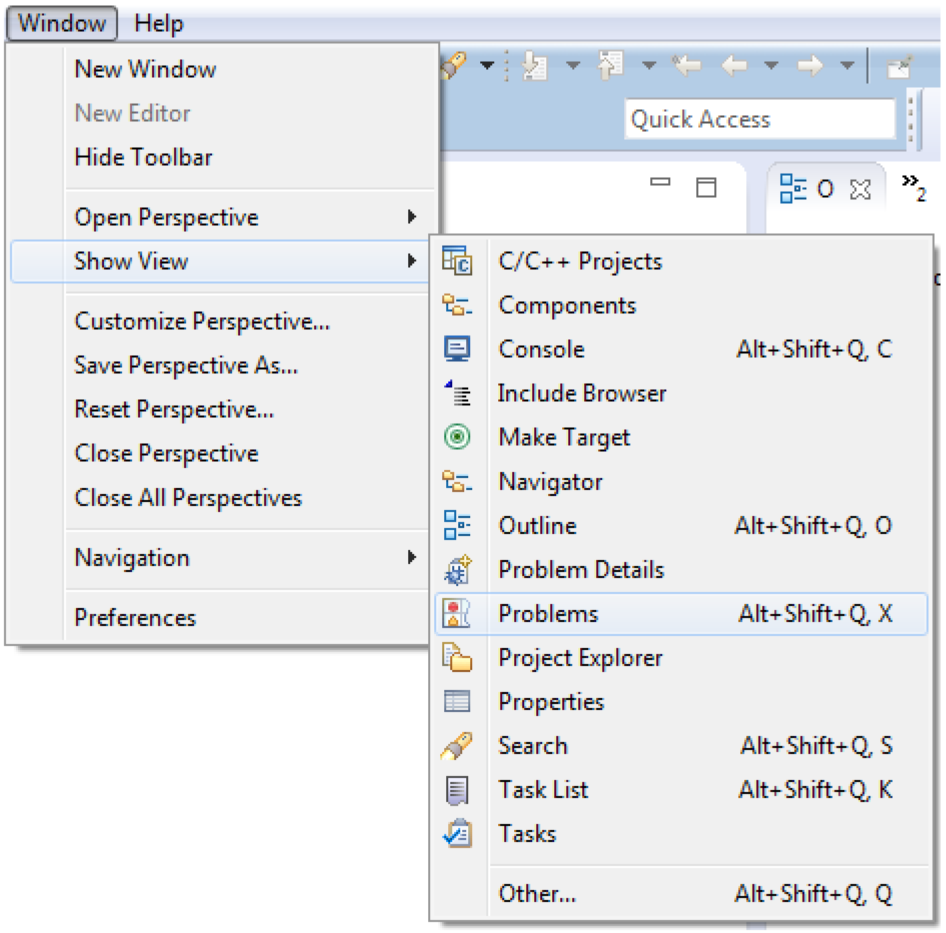


Figure 11. Accessing the Problems view

The complete list of errors and warnings encountered in the Workspace is available in the **Error log** view.

To open the **Error Log** view:

1. Select **Windows > Show View > Other > Error Log**.

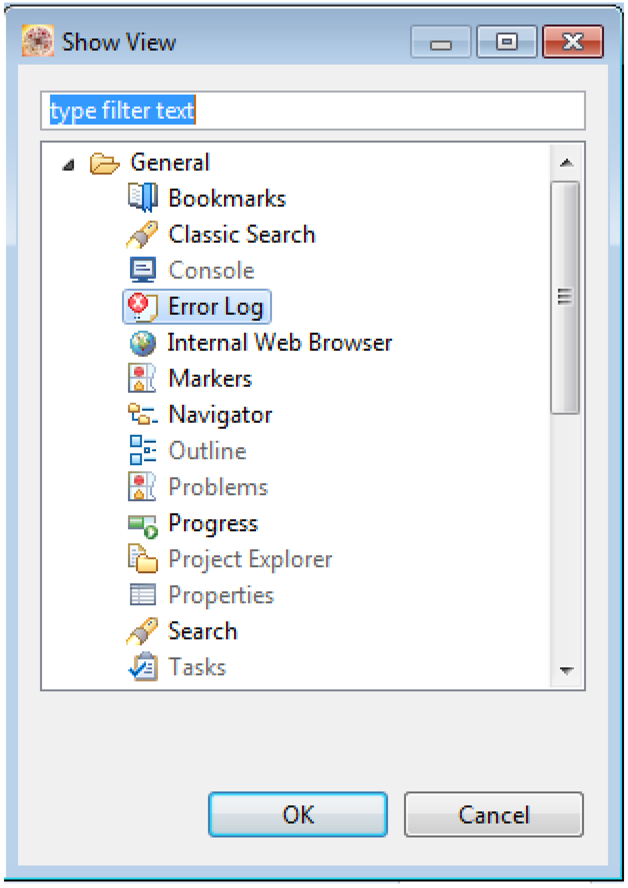
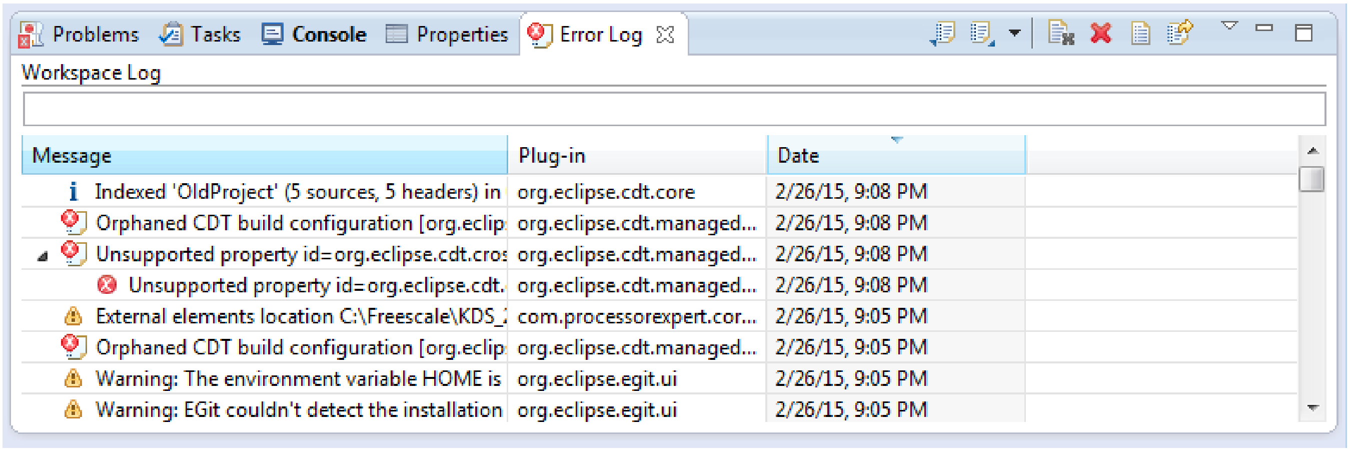


Figure 12. Accessing the Error log view

2. Click **OK**. The **Error Log** view appears stacked in the IDE.



Message	Plug-in	Date
Indexed 'OldProject' (5 sources, 5 headers) in	org.eclipse.cdt.core	2/26/15, 9:08 PM
Orphaned CDT build configuration [org.eclip	org.eclipse.cdt.managed...	2/26/15, 9:08 PM
Unsupported property id=org.eclipse.cdt.cros	org.eclipse.cdt.managed...	2/26/15, 9:08 PM
Unsupported property id=org.eclipse.cdt.	org.eclipse.cdt.managed...	2/26/15, 9:08 PM
External elements location C:\Freescale\KDS_	com.processorexpert.cor...	2/26/15, 9:05 PM
Orphaned CDT build configuration [org.eclip	org.eclipse.cdt.managed...	2/26/15, 9:05 PM
Warning: The environment variable HOME is	org.eclipse.egit.ui	2/26/15, 9:05 PM
Warning: EGit couldn't detect the installation	org.eclipse.egit.ui	2/26/15, 9:05 PM

Figure 13. Error Log view

NOTE

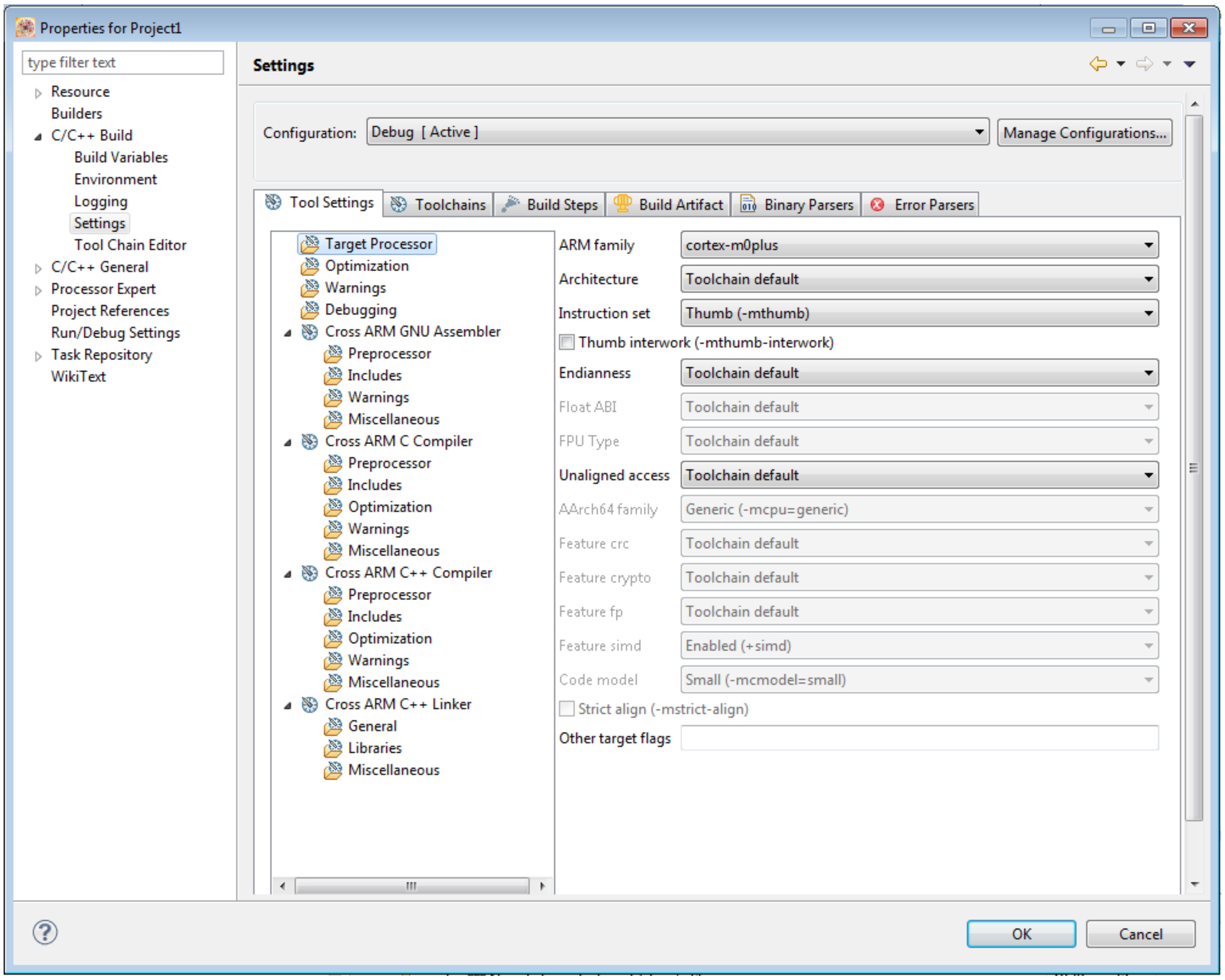
These errors occurs if manual updates to the project are not done. However, once the update is done these errors will not go away (because they are stored in the log file) but should stop occurring each time the project is opened after the project has been updated.

6 Manually updating and cleaning the converted project

Once you have identified the issues

To clean a project:

1. Select the convert project from the Project Explorer view.
2. Specify the path for compiler include paths.
   a. Right-click on the selected project and select **Properties**
   b. Expand the **C/C++ Build** tree and select **Settings > Tool Settings**.
   c. Navigate to **Cross ARM C Compiler** and select **Includes**.



Configuration: Debug [Active]

ARM family: cortex-m0plus
Architecture: Toolchain default
Instruction set: Thumb (-mthumb)
Thumb interwork (-mthumb-interwork)
Endianness: Toolchain default
Float ABI: Toolchain default
FPU Type: Toolchain default
Unaligned access: Toolchain default
AArch64 family: Generic (-mcpu=generic)
Feature crc: Toolchain default
Feature crypto: Toolchain default
Feature fp: Toolchain default
Feature simd: Enabled (+simd)
Code model: Small (-mcmodel=small)
Strict align (-mstrict-align)
Other target flags:

Figure 14. Build settings

d. Click the button and specify the absolute path of the libraries you want to include.

Alternatively, you can navigate to Project > Properties > C/C++ Build > Build Variables and define an environment variable ${MCUToolsBaseDir}. Defining the variable makes it accessible within the Eclipse environment. To define the variable, click **Add** and specify the variable name and Value. Check the **Apply to all configurations** checkbox if you want to access the variable from other configurations. Click **OK** to close the **Define a New Build Variable** dialog.

3. Navigate to Project > Properties > C/C++ Build > Settings > Cross ARM C Linker > Libraries > Libraries (-l).
4. Add the library `rt`.
5. Click **OK** to close the **Properties** dialog.
6. Right-click on the project and select **Clean Project**.

7 Building the converted project

To build the project converted project:

1. Select the project in the Project Explorer view.
2. Select Project > Build Project.

Index

Document Number KDSPGQS
Revision 1.0.0, 04/2015