

Kinetis Design Studio v1.1.1 Porting - User Guide

SOMNIUM® Document Number: SOMN-KDS-0003
Freescale Document Number: KDSPUG Rev 0, 07/14

Copyright © 2013-2014 SOMNIUM® Technologies Limited

Contents

1	Overview	1
2	Examples of warnings from an existing or imported CodeWarrior project	1
3	Converting a CodeWarrior project file	2
4	Building a migrated CodeWarrior project	2
5	Common build errors after migration	2
A	Notices	4

1 Overview

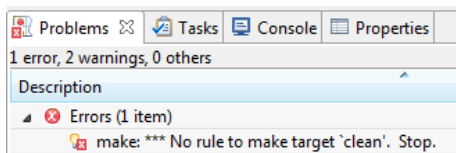
The *CodeWarrior to Kinetis Design Studio project migration assistant* assists in the migration of CodeWarrior projects into a form that is configurable by the Kinetis Design Studio IDE. It is necessary to manually alter certain project content to achieve a successful build in the Kinetis Design Studio IDE.

Existing CodeWarrior projects may be migrated into a form suitable for use with the Kinetis Design Studio IDE. To do this, first launch the Kinetis Design Studio IDE using a workspace which contains an existing CodeWarrior project or import an existing CodeWarrior project (File > Import > General > Existing Projects into Workspace) into a new workspace.

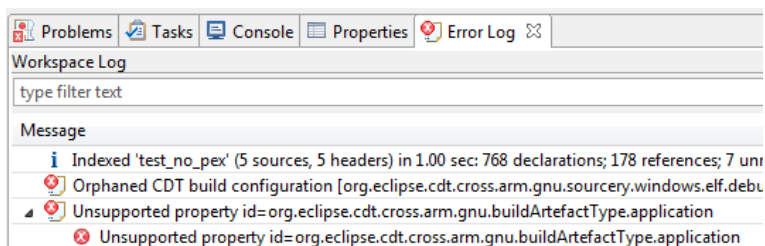
2 Examples of warnings from an existing or imported CodeWarrior project

An existing CodeWarrior project or one that has just been imported may exhibit the following warnings under the Kinetis Design Studio IDE:

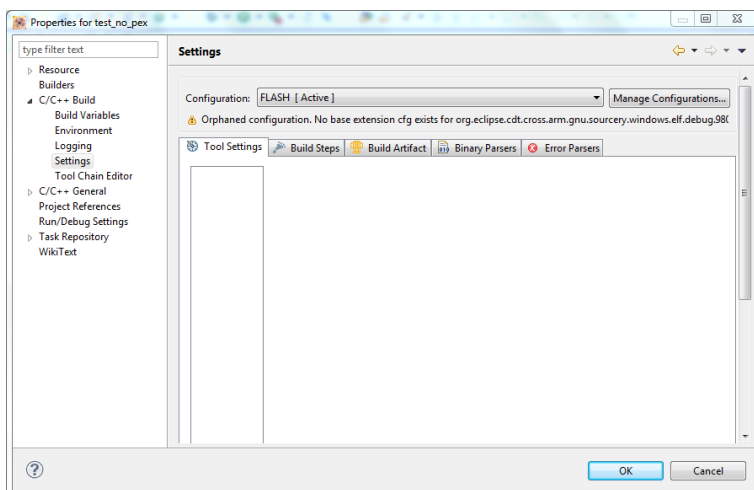
- *Problems view* (Windows > Show View > Problems)



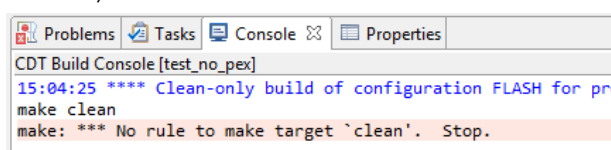
- *Error Log view* (Windows > Show View > Other > General > Error Log)



- The *Project Properties* (Project > Properties > C/C++ Build > Settings > Tool Settings) will appear unpopulated.

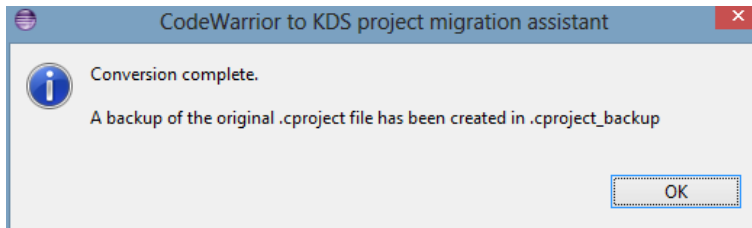


- Attempting to build the project results in failure with a message in the *CDT Build Console view* (Windows > Show View > Console):



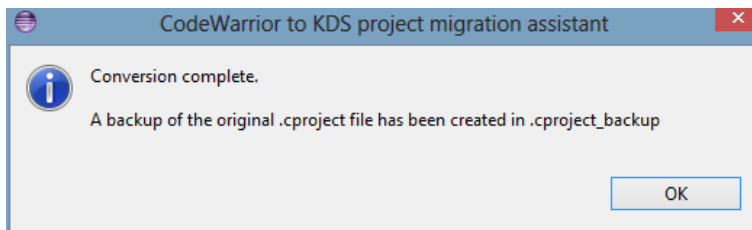
3 Converting a CodeWarrior project file

Next, open the *Project Explorer* view (Windows > Show View > Project Explorer) and open the project's context menu (right click) and select *Convert CodeWarrior project file....* The *CodeWarrior to KDS project migration assistant* dialog appears displaying the following question:



Click Yes to start the conversion.

Once completed, the *CodeWarrior to Kinetis Design Studio project migration assistant* dialog will then display the following information:



A log of the conversion process is stored in the file `KDSConverter.log` in the project's root.

Note Once a CodeWarrior project file has been converted, running the converter subsequent times has no effect.

4 Building a migrated CodeWarrior project

The project can be built and cleaned in the normal way (Project > Build Project, Clean...). Depending on the content of the CodeWarrior project it may be necessary to switch certain settings to use Kinetis Design Studio headers and libraries, rather than CodeWarrior headers and libraries.

5 Common build errors after migration

When attempting to build a migrated project the following problems maybe encountered. Solutions to these problems are offered as a temporary fix to demonstrate the build working. For complete migration it may be necessary to manually convert some of the project content to point to Kinetis Design Studio headers and libraries.

- `ansi_parms.h`: No such file or directory:

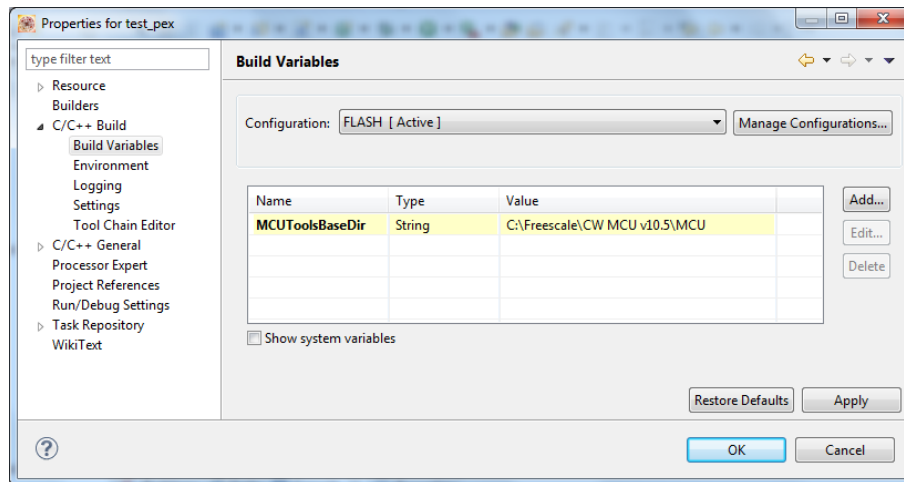
```

CDT Build Console [test_pex]
'Building file: ../Project_Settings/Startup_Code/__arm_end.c'
'Invoking: Cross ARM C Compiler'
arm-none-eabi-gcc -mcpu=cortex-m0plus -mthumb -O0 -fmessage-length=0 -fsigned-char -ffunction-sections -f...
../Project_Settings/Startup_Code/__arm_end.c:41:24: fatal error: ansi_parms.h: No such file or directory
}
^
compilation terminated.
make: *** [Project_Settings/Startup_Code/__arm_end.o] Error 1
    
```

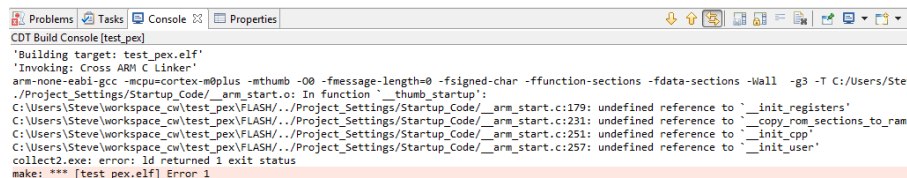
This problem may be caused by the use of the environment variable `MCUToolsBaseDir` in the compiler include paths (Project > Properties > C/C++ Build > Settings > Cross ARM C Compiler > Includes).

To fix this problem either define the variable so it is accessible from within the Eclipse environment or replace the include path directives to use absolute paths.

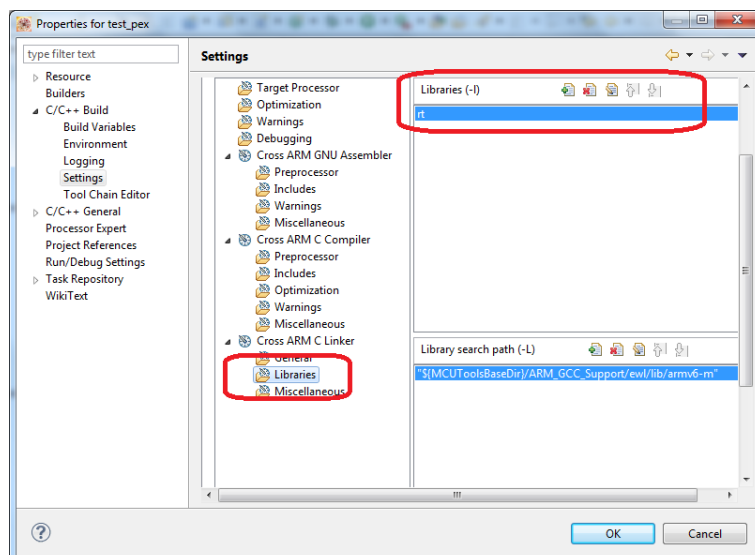
A convenient method to define an environment variable for use in the build is to define it in Project > Properties > C/C++ Build > Build Variables. For example press Add... and in *Variable name* add `MCUToolsBaseDir` and in *Value* add the directory to the CodeWarrior support libraries, for example `C:\Freescale\CW MCU v10.5\MCU`.



- collect2.exe: error: ld returned 1 exit status



This problem may be caused by the missing library `librt`. To fix this problem add the `librt` library to the project by selecting Project > Properties > C/C++ Build > Settings > Cross ARM C Linker > Libraries > Libraries (-l) and add the library "rt" (without quotes).



A Notices

Copyright © 2013-2014 SOMNIUM® Technologies Limited

Kinetis Design Studio is produced for Freescale by SOMNIUM® Technologies <http://www.somniumtech.com>. All rights reserved.

Information in this document is provided solely to enable system and software implementers to use Freescale products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document.

SOMNIUM and the SOMNIUM logo are registered trademarks of SOMNIUM® Technologies Limited.

Freescale, the Freescale logo, CodeWarrior, and Kinetis are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. All other product or service names are the property of their respective owners. ARM and Cortex are registered trademarks of ARM Limited (or its subsidiaries) in the EU and/or elsewhere. mbed is a trademark of ARM Limited (or its subsidiaries) in the EU and/or elsewhere.

Freescale reserves the right to make changes without further notice to any products herein. Freescale makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters that may be provided in Freescale data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including “typicals,” must be validated for each customer application by customer’s technical experts. Freescale does not convey any license under its patent rights nor the rights of others. Freescale sells products pursuant to standard terms and conditions of sale, which can be found at the following address: <http://www.freescale.com/SalesTermsandConditions>.