Creating and Configuring C++test Projects for RVDS

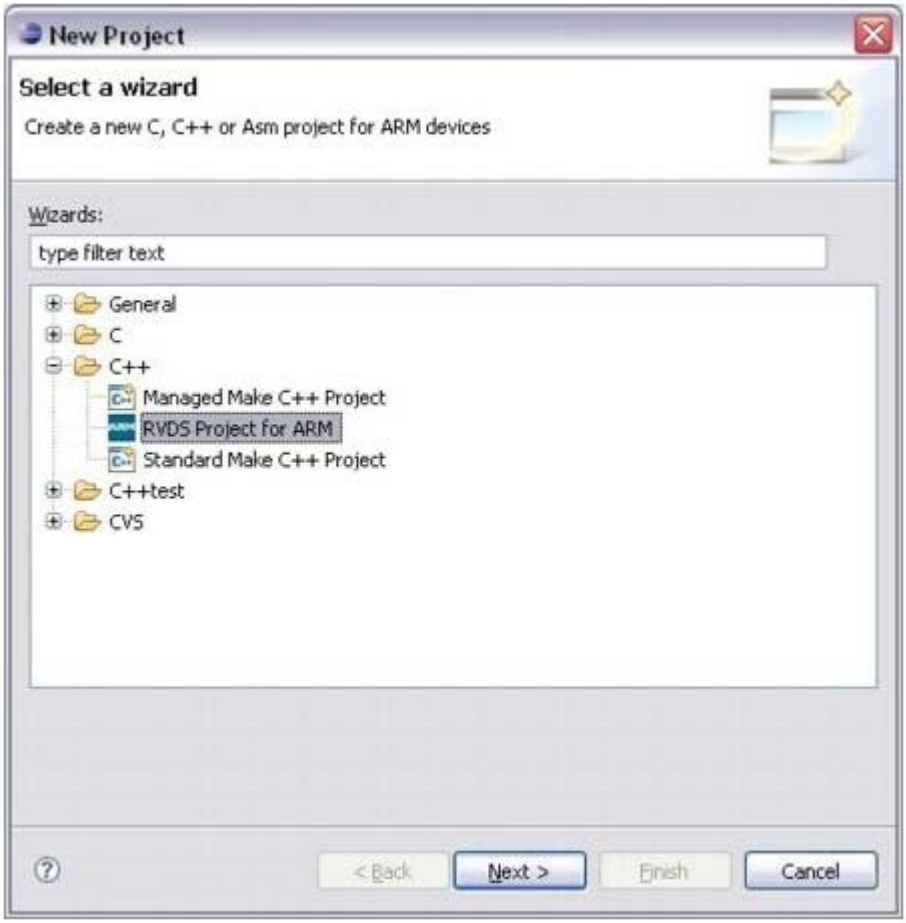
This topic explains how to create and configure C++test projects for code that is designed to be compiled/built using ARM compilers and/or written with the aid of the RVDS IDE.

C++test integrates with ARM Development Studio 5 (DS-5) projects. Both managed and static makefile build types are supported

- CDT-managed make projects should not require additional setup activities.
- Static make projects may require the **C++test> Build Settings> Option Source** project properties to be reconfigured. For more information please refer to Creating a Project Using an Existing Build System.

> **Note**
>
> When testing a project that uses the ARM C/C++ Compiler to build an ARM Linux application or library (e.g.using the `armcc --arm_linux_configure --arm_linux_config_file=path` command), you must manually change the linker executable from `armlink` to `armcc` in the project compiler settings. Choose **Project Properties> Parasoft> C++test > Build Settings** to edit the compiler settings. Changing the linker executable allows the behavior of ARM C/C++ toolchain to be mimicked in ARM Linux mode.

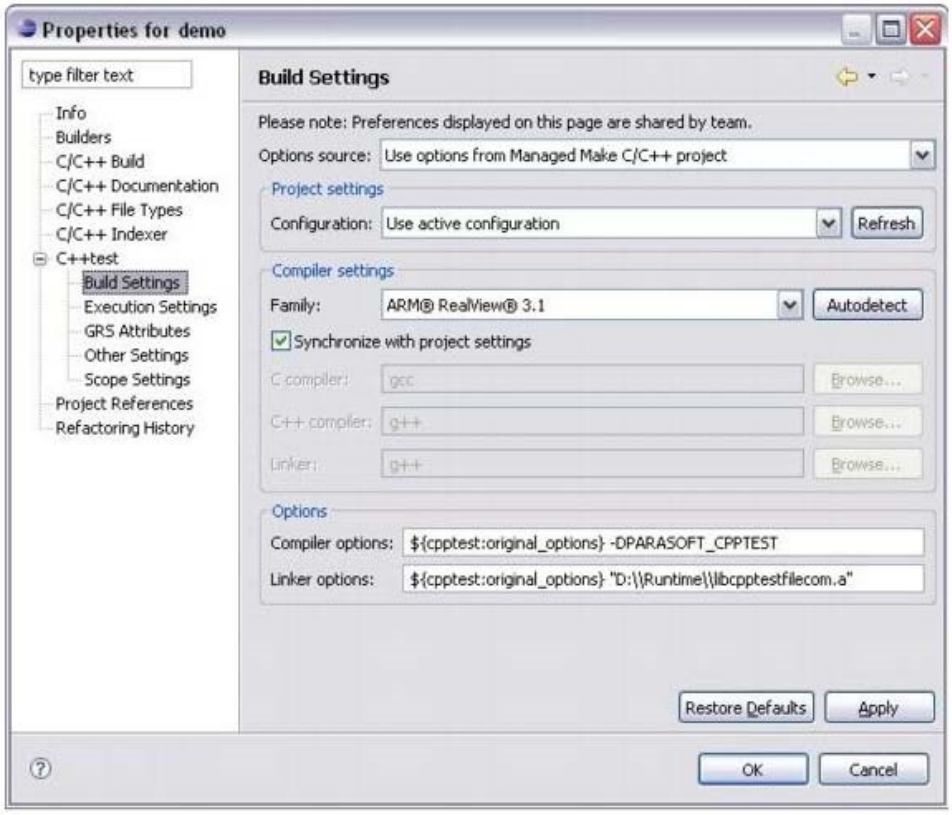
RVDS provides two basic CDT project configurations (Manage Make Project as well as Standard Make Project) plus an additional project configuration (RVDS Project for ARM). RVDS Project for ARM is the most commonly-used and convenient, since it already provides all of the defines and switches that would otherwise have to be set manually. However, C++test can work with any of these project types.



If you intend to unit test your project, you will need a correct build of the C++test runtime. The default Test configuration "Run RVDS 3.x, 4.x Tests" contains a step for automatically building the C++test runtime library. In some cases, you may need to prepare a custom build of the C++test runtime library; if so, see the chapter Working with the C++test Runtime Library.

When you are ready to configure your project for C++test, review and modify settings as follows:

1. Right-click the C/C++ Projects tree (a.k.a. "the project tree") node for the project whose settings you want to review and modify, then choose **Properties** from the shortcut menu. The Properties dialog will open.
2. Select **Parasoft> C++test> Build Settings** in the left pane.



3. Verify that the proper compiler family is specified.
    - In most cases, you can use the **Autodetect** button to automatically point C++test to the compiler's executables. If you use another compiler's binaries, you may need to clear the **Synchronize with project settings** check box, and then manually specify the appropriate location(s).
4. If you want to use an alternative project configuration (e.g., Debug, Release, etc.) choose it in the **Configurations** box.
5. If you will be performing unit testing with a special build of customized runtime library, specify an additional linker library. The default value is ${cpptest:original_options}. Append this with the path to the C++test runtime library (see Working with the C++test Runtime Library to learn how to build the runtime library). In the graphic above, the runtime library is under the Runtime directory on drive D:
    - Note that if you are using the default Test Configuration with the default build of the runtime library, you don't need to add anything to the linker flags. The "Run RVDS 3.x 4.x Tests" Test Configuration will prepare a default build of the runtime library and pass it to the linker.

For more information about C++test project properties, see Reviewing and Modifying Settings.