

Readme - The C/C++ Source Code Unit Test Framework

Author: Rajinder Yadav
Date: July 21, 2007
Revision: Jan 27, 2008

Web: <http://devmentor.org>
Email: rajinder@devmentor.org

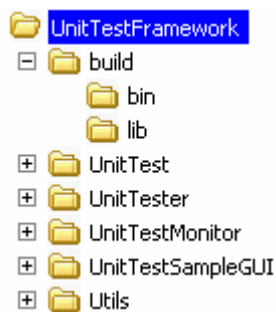
Getting Started

The UnitTest Framework was developed using Visual Studio 8, if you have an earlier version of Visual Studio you should still be able to build the projects but you will need to do some work to create the projects yourself by hands. The other option is to simply download the binaries.

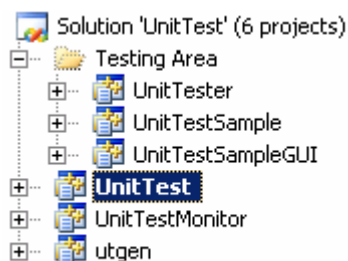
Download the **UnitTestFramework.zip** project files from <http://devmentor.org>

Extract all the files to your drive, once you have extracted the files you should see the following directory structure.

NOTE: the build folder will be generated after a project is compiled.



Once you extract the source files from **UnitTestFramework.zip** go to the **UnitTest** folder and double click on “UnitTest.sln” to open VisualStudio with all the projects loaded. What you will see are the 6 projects shown below:



Make sure you are set to build in **"Release"** mode inside Visual Studio, you must do this otherwise any assert macros in the debug build will halt unit testing when a dialog box is displayed.

In order to run "**UnitTestSampleGUI**" or "**UnitTestSample**" executables you will first need to build the **UnitTest** core lib project. For **UnitTestSampleGUI** you will also need to build **UnitTestMonitor** DLL project.

Note: the location where you extract the files will be referred as `%Source_Dir%`, keep this in mind when reading the **UnitTest Guide**.

UnitTest and **UnitTestMonitor** are the two projects you will need to build in the order mentioned before the samples can be built. I have not set the project dependencies because I like to do a clean build as I work on a projects and I don't particularly want all the other projects to rebuild.

- The **UnitTestMonitor** app is a GUI tool used to run and view the test case results.
- The **UnitTestSample** shows you how to develop a console and files logging unit test project.

Project **UnitTest** is the core DLL that contains the logic and bases classes to do all the unit testing you require, also packed in the DLL are two observers:

ConsoleLogger - outputs test results to the screen
StreamLogger - outputs test results to a log file

The Folder labeled "Testing Area" contains 3 projects:

UnitTester – build an exe to performs unit testing on the UnitTest classes

UnitTestSample and **UnitTestSampleGUI** are examples project to be browsed for reference.

NOTE: The UnitTest Monitor GUI will produce a log under the current folder. The default name of the log file is "**UnitTest.log**"

Project files Settings

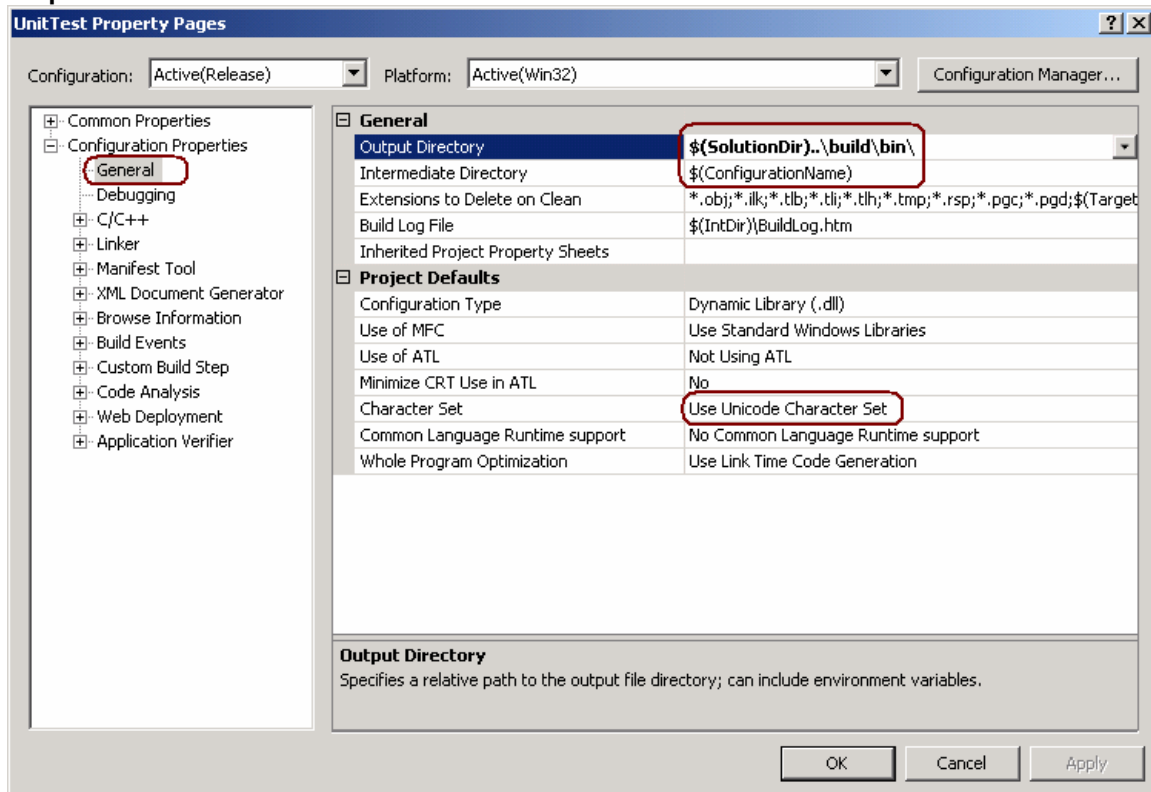
- 1) Make sure the base character set is Unicode
- 2) You will need to point to the following include directories:

```
%Source_Dir%\UnitTest\Include  
%Source_Dir%\UnitTestModitor  
%Source_Dir%\Utils\Include  
%Source_Dir%\UnitTest\UTUtils
```

- 3) Make sure the library path is set to the correct folder: `%Source_Dir%\build\lib`
- 4) You will need to link the **UnitTest.lib** files
- 5) If you will be using the UnitTestMonitor GUI DLL, you also need to link to **UnitTestMonitor.lib**

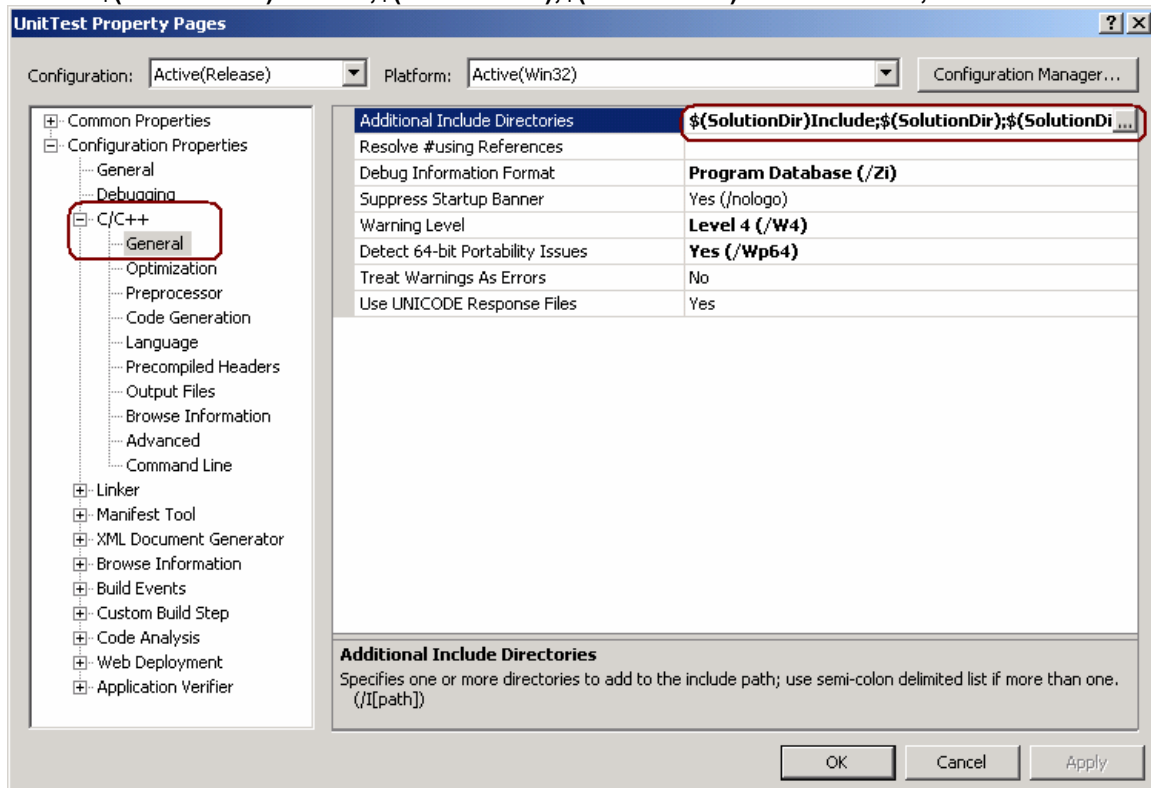
Following are the screen shots of the Project Properties showing you where to go to make the changes. Please refer to the **UnitTest_Guide** for more information.

Step 1

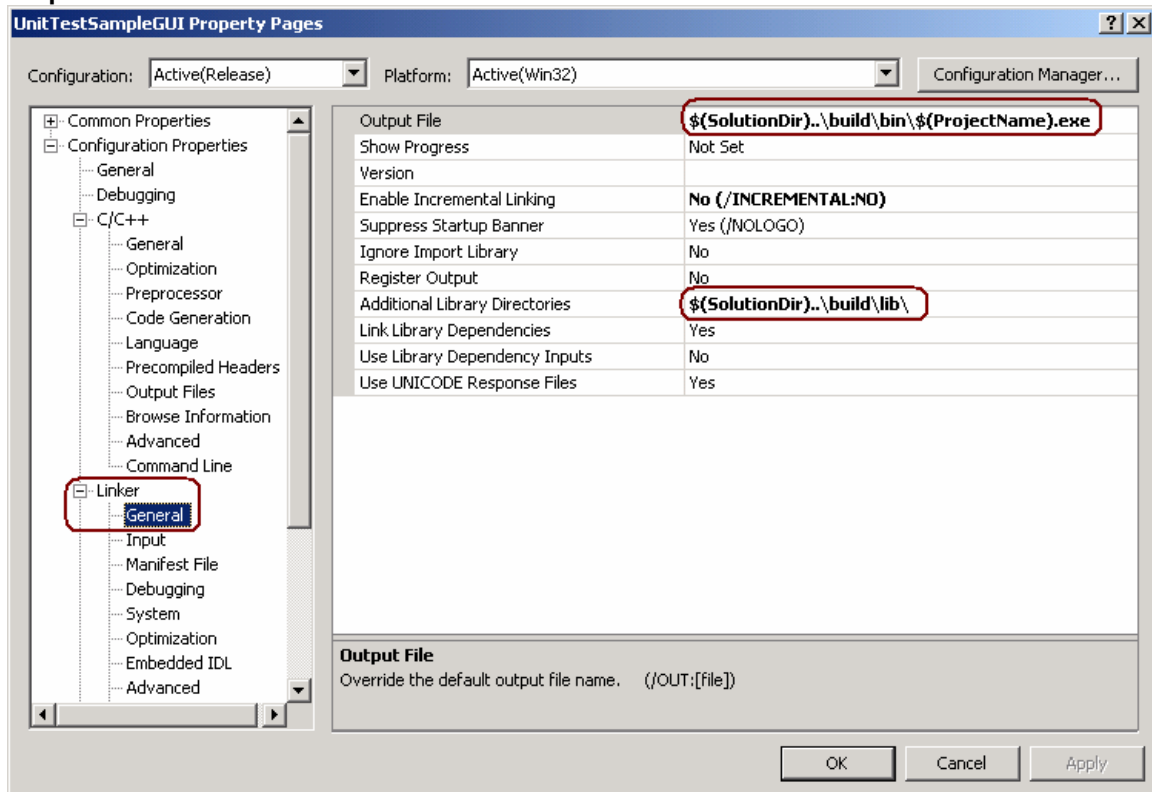


Step 2

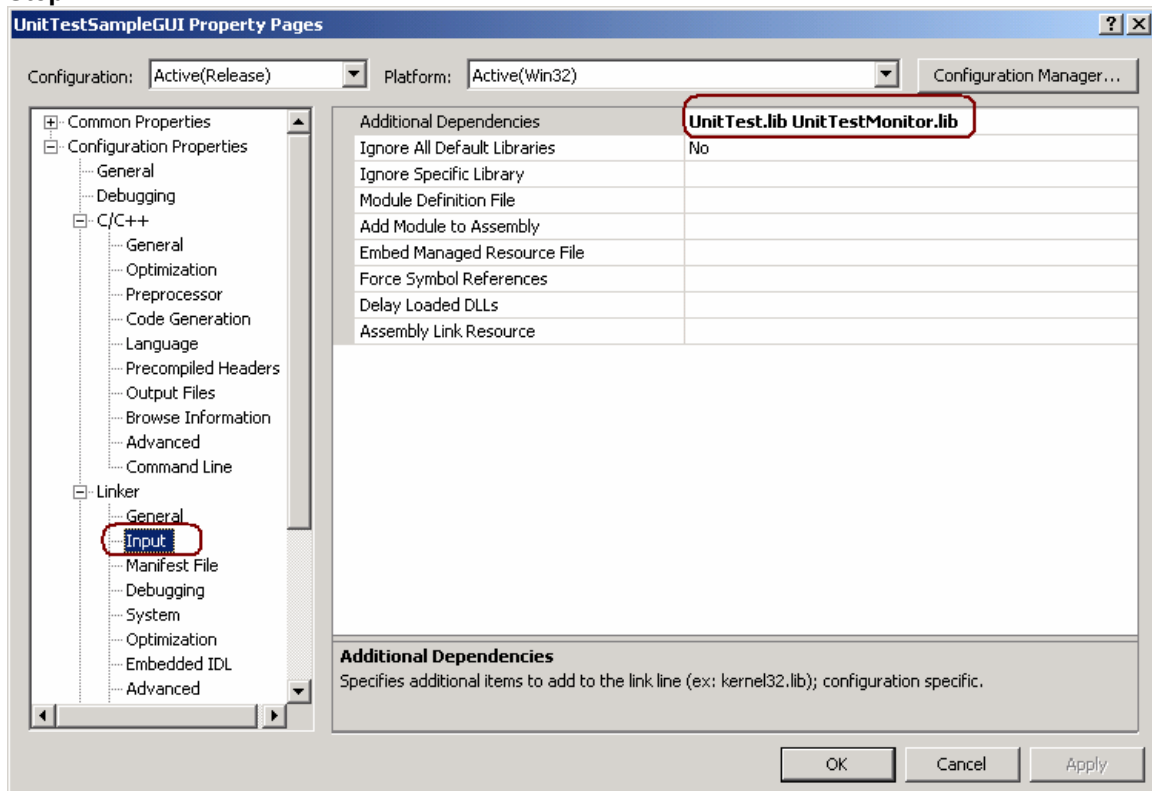
Inc Dir: `$(SolutionDir)Include;$(SolutionDir);$(SolutionDir)..\Utils\Include;`



Step 3



Step 4



UnitTest XML Input File UnitTest.xml

Finally an XML unit test input file now gets copied into the bin folder after the build of the UnitTest project. This XML file will need to be modified in order for all the unit test cases that come with UnitTestFramework.7z file to pass.

The basic format of the XML file is:

```
<?xml version="1.0" encoding="utf-8"?>
<UnitTest>
  <Input field="ReadMe.txt">G:\dev\src\UnitTester\TestInput\ReadMe.txt</Input>
  ...
  ....
</UnitTest>
```

You will need to modify all the paths for the TestInput folder to point to where the extracted folder is on your drive. In particular, search and replace "G:\dev\src\UnitTester\TestInput" and that should do the trick.