

CppUTest is a C/C++ based unit test framework for unit testing and test-driving your code. It is written in C++ but is used in C and C++ projects and frequently used in embedded systems but it works for any C/C++ project.

CppUTest's core design principles are:

- Simple in design and simple in use.
- Portable to old and new platforms.
- Build with Test-driven Development in mind.

# 1 MinGW

For CppUTest to work we need a C/C++ compiler. The compiler integrated in Keil is for ARM devices, but we want to compile and run the tests on the host computer.

The GNU Compiler Collection (available for Windows as part of MinGW) is used in this document.

## 1.1 Download MinGW

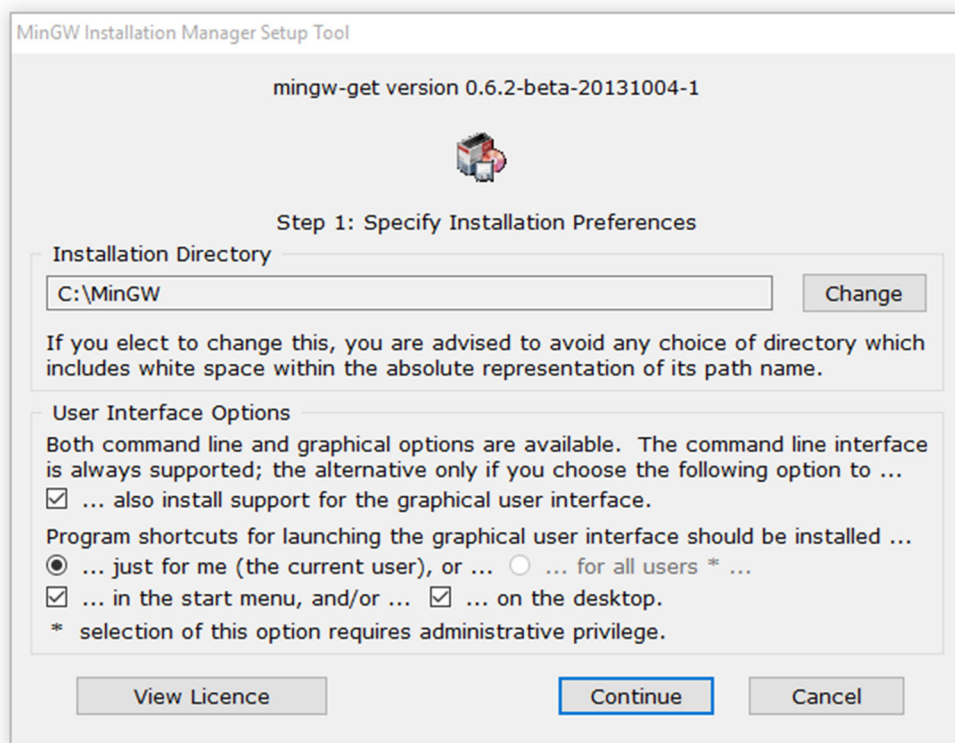
MinGW (Minimalist GNU for Windows) is a development environment for native Windows applications, it provides a complete open source programming tool set.

Download MinGW from here: <http://sourceforge.net/projects/mingw/files/>

## 1.2 Installation

Start the installation by executing the installer.

Install MinGW to the default directory "*C:\MinGW*". You need Admin Rights on this directory:

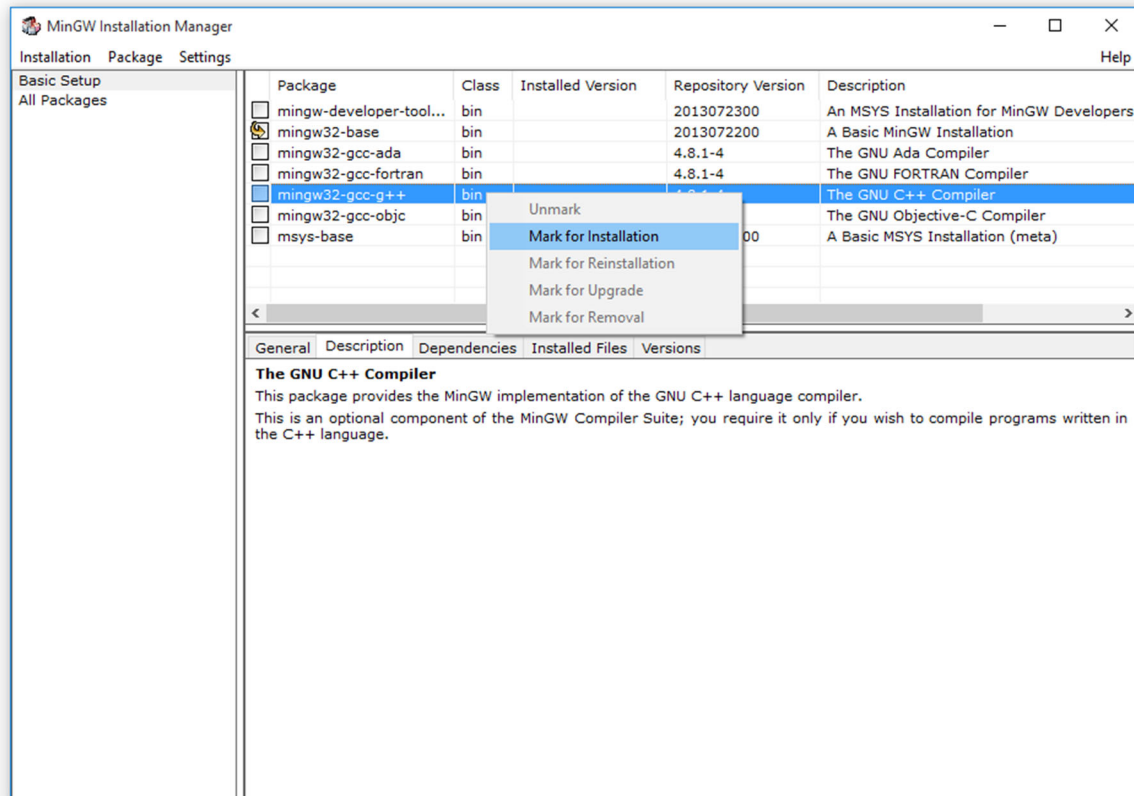


After the installation has finished you will see the MinGW Installation Manager. If it does not open automatically you can find it in the start menu.

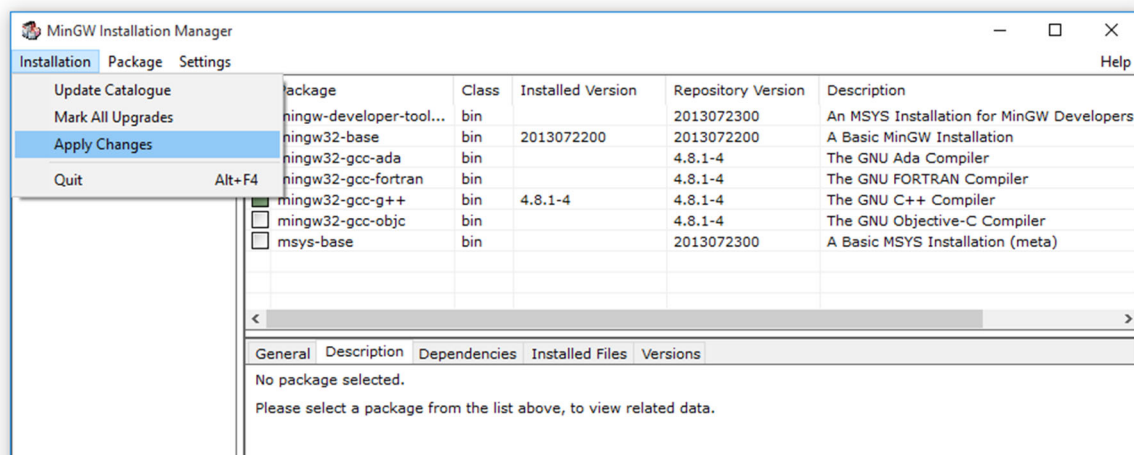
For the unit tests to work, you need the following packages:

- mingw32-base (*in Basic Setup*)
- mingw32-gcc-g++ (*in Basic Setup*)
- mingw32-pthreads-w32 (dev) (*in All Packages*)

To install a package right-click on the package name and select “Mark for Installation”:



Click on “Installation → Apply Changes” to definitely install the selected packages:



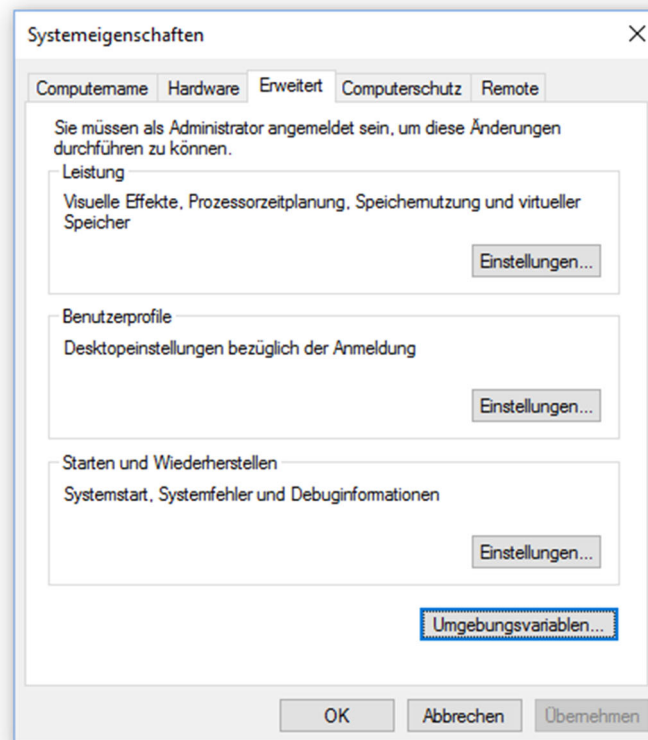
### 1.3 Environment Variables

To make sure Windows is aware of GCC we need to modify the systems PATH variable.

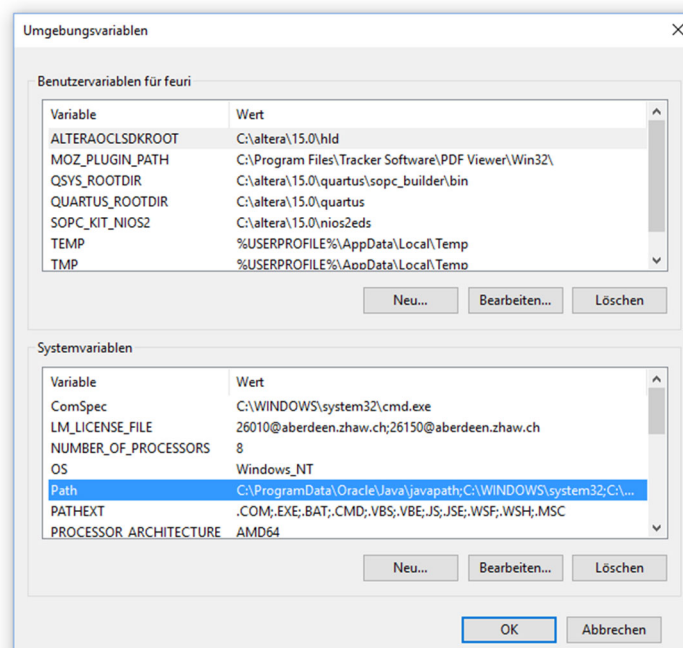
Go to the extended system settings in Windows and open the “System Variables” Menu:

Start → System → Advanced System Settings → Environment Variables

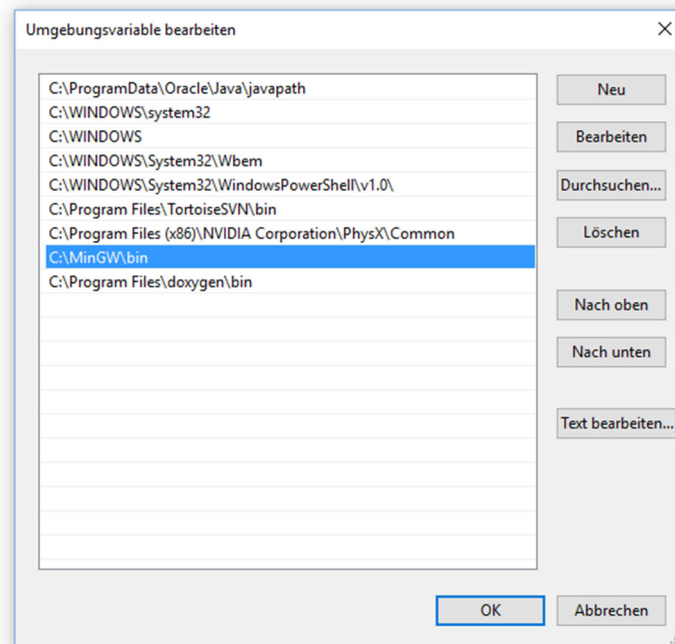
Start → System → Erweiterte Systemeinstellungen → Umgebungsvariablen...



Search for the “Path” variable and click edit:



Now add the path to the MinGW binaries "*C:\MinGW\bin*" to the variable:

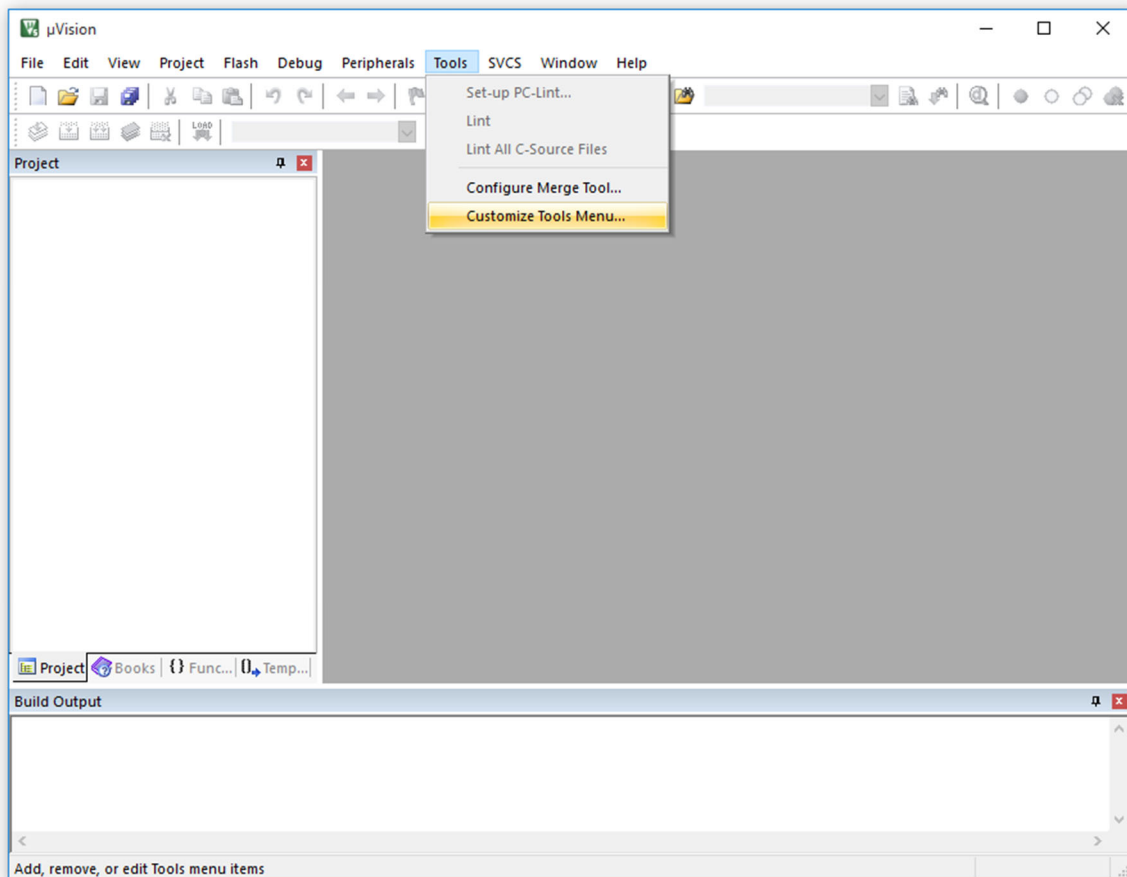


## 2 Keil $\mu$ Vision

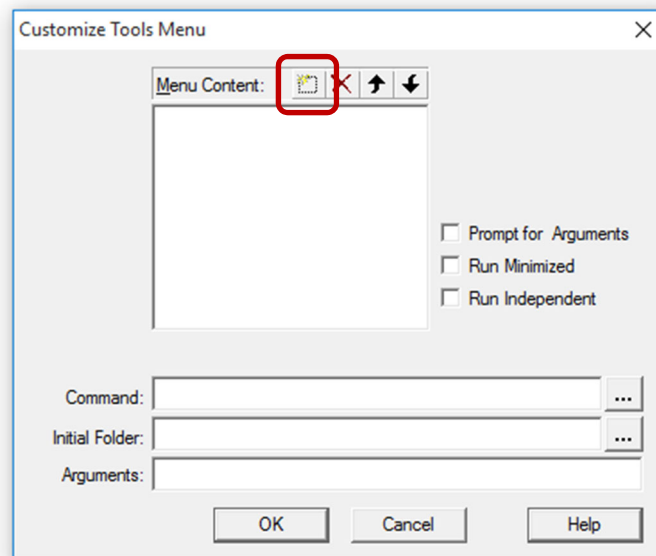
To integrate the unit tests in Keil, we need to add a command to the Keil Tools menu.

### 2.1 Customizing Keil $\mu$ Vision

To customize the tool menu in Keil open the *“Tools → Customize Tools Menu...”* Menu:

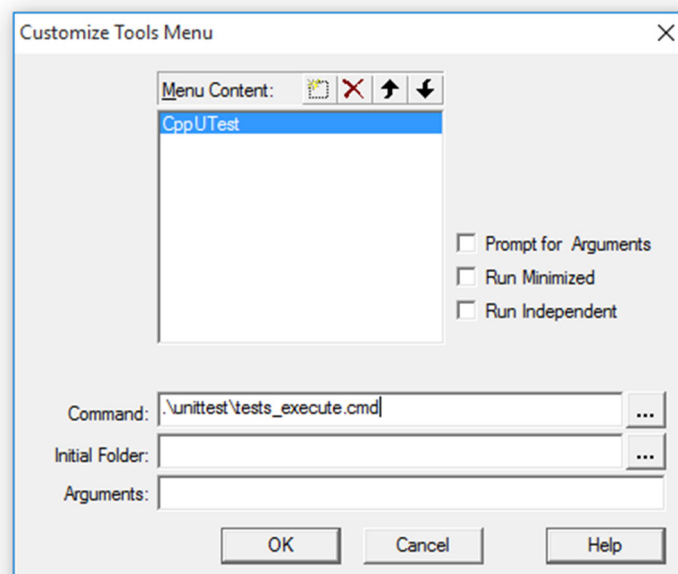


Create a new entry by clicking on the new entry button:



Fill in the following parameter:

- Titel : CppUTest
- Command: `.\unittest\tests_execute.cmd`
- Initial Folder: *[nothing]*
- Arguments: *[nothing]*



## 2.2 Execute a Unit Test

To execute a unit test simply click on “Tools → CppUTest” and, if the project is set up correctly, the project gets compiled and tested.

