Cpputest

CppUTest unit testing and mocking framework for C/C++

CppUTest 

-
- Core Manual
-
- CppUMock Manual
-
- Plugin Manual
-
- Platforms stories
-
- View on GitHub

Download Release 3.8 as .zip Download Release 3.8 as .tar.gz

What is CppUTest.

CppUTest is a C /C++ based unit xUnit test framework for unit testing and for 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

Where to get CppUTest

Pre-packaged

Linux

There is an Debian and Ubuntu package available for CppUTest. This is by far the easiest way to install it, via:

```
$ apt-get install cpputest
```

MacOSX

For Mac, a Homebrew package is available too. You can install via:

```
$ brew install cpputest
```

From source

You can download the latest 'automatically released' version:

- Latest version passing the build This version is automatically packages after a build has passed.

Alternatively, you can clone the github repository, read-only:

```
$ git clone git://github.com/cpputest/cpputest.git
```

Or clone it via ssh (which requires a github account)

```
$ git clone git@github.com:cpputest/cpputest.git
```

After you cloned CppUTest, you can build it with your favorite build tool (CMake or autoconf).

Building with autoconf requires you to (this requires you to have installed GNU autotools, apt-get/brew install automake autoconf libtool):

```
$ cd cpputest_build
$ autoreconf .. -i
$ ../configure
$ make
```

How to create a coverage report

You can use autoconf to create a coverage report for CppUTests own tests. If you have lcov installed, a browsable html report will be generated as well. After the steps outlined in the previous paragraph, do the following:

```
$ make check_coverage
```

This will generate a file called gcov_report.txt with the coverage report in plain text format. It will also generate an HTML file called gcov_report.txt.html. If you have lcov installed, you will be able to browse the lcov report by opening ./cpputest_build/test_coverage/index.html . The lcov report is by far the easiest way to inspect CppUTest's own test coverage.

Alternatively, you can use CMake if that is the build tool you fancy (this requires you have install CMake, apt-get install cmake):

```
$ cd cpputest_build
$ cmake ..
$ make
```

For Windows users, the above work with cygwin. There are also several MS VC++ projects available.

Where to find more information

- If you have any questions, check out the Google Groups
- The source is at the main github page
- You can report bugs or features at the issues page
- You can follow CppUTest on twitter

Quick introduction (some code!)

To write your first test, all you need is a new cpp file with a TEST_GROUP and a TEST, like:

```
TEST_GROUP(FirstTestGroup)
{
};

TEST(FirstTestGroup, FirstTest)
{
  FAIL("Fail me!");
}
```

This test will fail.

You can add new tests to the test group by just writing more tests in the file, like this:

```
TEST(FirstTestGroup, SecondTest)
{
  STRCMP_EQUAL("hello", "world");
```

```
  LONGS_EQUAL(1, 2);
  CHECK(false);
}
```

You do need to create a main where you run all the unit tests. Such a main will look like this:

```
int main(int ac, char** av)
{
  return CommandLineTestRunner::RunAllTests(ac, av);
}
```

For more information, We'd recommend reading the manual or, even better, check some existing tests such as SimpleStringTest or (a bit more complicated) MemoryLeakDetectorTest or the mocking tests or just check out the Cheat Sheet

Related projects

For Eclipse users, also check:

- CppUTest Eclipse Test Runner This will allow you to run your tests JUnit style with red and green bars, and rerun arbitrary selections of tests. Prerequisites:
  - CppUTest off master
  - Eclipse Juno, Kepler, or later
  - Eclipse C/C++ Unit Plugin (if not already present in your version of Eclipse; install directly from Eclipse Help -> Install New Software…).
- CppUTest Eclipse Plugin Project

Authors and Contributors

CppUTest has had many contributions from its users. We can't remember all, but we appreciate it a lot! Much of the original code was written by Michael Feathers (based on CppUnit Lite). The current main maintainers are @jwgrenning and @basvodde