

About Cross Platform Unit Testing

Adrian Kosmaczewski

2006-10-09

A lot has been said about unit testing à la JUnit in C++; the most interesting article about the subject is without any doubt Exploring the C++ Unit Testing Framework Jungle, By Noel Llopis, with a thorough comparison of the most important C++ unit testing frameworks out there:

CppUnit

Boost.Test

CppUnitLite

NanoCppUnit

Unit++

CxxTest

While I won't go into much detail about his article, I think that the only aspect that Noel forgot in his analysis is the cross-platform capabilities of these frameworks. And that is precisely the aspect that I was looking for while choosing a unit testing framework for a personal project: I needed it to work seamlessly in Ubuntu, Mac OS X and Windows.

I haven't looked at all of them, but I took a serious look to Boost.Test and CppUnit. The first one seems promising but much too complicated to install (you have to build the whole Boost libraries to get it - please correct me if I did wrong!), while CppUnit, that I used in my job, was a pleasure to work with. I have managed to build it directly from the sources doing a classic set of commands:

```
tar xvpz cppunit-1.12.0.tar.gz
cd cppunit-1.12.0
./configure
cd src/cppunit
make
sudo make install
```

This worked perfectly well in both Mac OS X 10.2 “Jaguar”, Mac OS X 10.4 “Tiger”, and Ubuntu Dapper. It installs the library in `/usr/local/lib` and after that, you can use it normally as any other lib.

On Windows, I found the instructions to make it work out-of-the-box in Visual Studio .NET 2003, and also using Eclipse CDT + MinGW.

Exactly what I wanted! Cross-platform, open-source, free, with good documentation, mature code base, easy to build, easy to include in projects (in my case, using the shared object library / `dylib` / `DLL`) and easy to link to.

PS: reading the comments in Noel’s article I found these references to two more C++ unit test frameworks:

`jUnitAsserter`

`TUT`