Last modified on 2013-04-19 18:50:55

ComTest for C++

ComTest for C++
- 1. Usage from command line
- 2. Usage from Eclipse without plugin
  - 2.1 Before use
- 2.2 Usage
3. Usage from other IDE's
- 3.1 Dev C++
- 3.2 Free-C
- 3.3 CodeLite
- 3.4 CodeBlocks
- 3.5 Visual Studio 2012
- 3.6 Other IDE's

- ComTest for C++ is in very alfa state.
- f.ex `"cat" === "cat"` works, so does C++ strings. Other pointer operations are not tested.
- no external test suite is used. Instead the source file is compiled to test file that is compiled and runned by g++
- at the moment destroys the old contents of the generated test file (Java-version deletes only lines done by ComTest)
- names C-test to name.c even it should be name.cpp because of the included C++ code.
- at the moment purpose is just to test single module units
- before usage: you must have g++ (f.ex MinGW in Windows) in the path (try writing g++ from command line)
- install: copy ComTestCpp.jar to some directory

1. Usage from command line

- usage from command line: example

```
e:\demo\>java -jar E:\kurssit\ohj2\comtest\comtestcpp.jar pali.cpp
pali.cpp => E:\kurssit\ohjelmointi2010\demot\vast\cpp1\paliTest.cpp ok
Test error: From: pali line: 24: expected [1] actual [0]
e:\demo\>
```

- example file (pali.cpp):

```
/**
 * \file pali.cpp
 * A function that checks if a C++- string is a palindrome
 *
 * @author Vesa Lappalainen
 * @version 27.4.2003
 */
#include <string>
#include <iostream>
using namespace std;

/**
 * Check if a word is a palindrome
 * @param word to study
 * @code
 * <pre name="test">
 *   isPalindrome("") === true;
 *   isPalindrome("a") === true;
 *   isPalindrome("aa") === true;
 *   isPalindrome("ab") === false;
 *   isPalindrome("aba") === true;
 *   isPalindrome("kissa") === false;
 *   isPalindrome("saippuakauppias") === true;
 *   isPalindrome("abbat") === false;
 *   isPalindrome("apua") === false;
 * </pre>
 * @endcode
 */
bool isPalindrome(const string &word)
{
    // TODO: implement correctly
    return true;
}
```

2. Usage from Eclipse without plugin

2.1 Before use

- First remove automatic comment formating from Eclipse
  - `Window/Preferences/Code style/Formater/New`
  - Give name for formater and copy from some existing one
  - Edit (if needed)
  - Comments and remove Enable comment formating
- Create new run command to Eclipse:
  - `Run/External Tools/Extrenal Tools Configuration...`
  - Choose Program
  - Press new icon
  - Name as ComTestCppRun
  - Location is you JDK path, f.ex:

    ```
    C:\Program Files\Java\jre7\bin\java.exe
    ```

    in Linux use / in dircetories.
  - to Arguments put your .jar:in address, f.ex:

    ```
    -jar E:\kurssit\ohj2\comtest\comtestcpp.jar ${resource_loc}
    ```

- IF your g++ is not in the path, you may add it to path in system or add it in this command:
  - goto commands Environment
  - add new variable:

    ```
    Name:  path
    Value: c:\MinGW\bin
    ```

- Now you can run ComTest for C++ by pressing in toolbar :the small green arrow that has small suitcase icon in the right bottom corner.
- Remember to wait the run until finished (red square disappears from Console window)
- Do yourself a Template so that it is fast to add ComTest-kommentti:
  - `Window/Preferences/C++/Editor/Templates/New`
  - Name: comtestcpp
  - Description: Adds a ComTest C++ comment
  - Context: Comment
  - Pattern:

    ```
    @code
     * <pre name="test">
     * ${cursor}
     * </pre>
     * @endcode
    ```

  - OK
  - OK
  - Now when you need C++ test, write comtest and Ctrl-Space in C++ comments.
- if you want tests to different folder, do `ComTest.ini` where you tell where to put the test files:

```
#DIRECTORY=dir
#PACKAGE=packagename
```

lines can be put also to source code comments

2.2 Usage

- Add ComTest macro language comments to your source file, remember that comtest Ctrl-Space will help your writing.
- Run ComTestCpp from green arrow
- follow the Console window and look if you get any errors

3. Usage from other IDE's

3.1 Dev C++

- new tool can be made by same procedure than in Eclipse. The only problem is that it opens a console that shuts down after run. So make first a comtestcpp.bat:

  ```
  java -jar E:\kurssit\ohj2\comtest\comtestcpp.jar %*
  pause
  ```

- make a new tool:
  - `Tools/Configre Tools`
  - Add
  - Title: ComTestCpp
  - Program: C:\bat\comtestcpp.bat
  - Working directory: <SOURCEPATH>
  - Parameters: <SOURCENAME>
  - OK
- run: `Tools/ComTestCpp`
- remember to save the file before run

3.2 Free-C

- new tool can be made by same procedure than in Eclipse. The only problem is that it opens a console that shuts down after run. So make first a comtestcpp.bat (f.ex in directory c:\bat):

  ```
  java -jar E:\kurssit\ohj2\comtest\comtestcpp.jar %*
  pause
  ```

- make a new tool:
  - `Tools/Configre Tools`
  - Add
  - Tool: ComTestCpp
  - ShortCut: Ctrl+Q
  - Program: C:\bat\comtestcpp.bat
  - Parameters: $(FileName)
  - Working directory: .
  - OK
- run: `Tools/ComTestCpp or press Qtrl+Q`
- be sure that you have g++ in path before starting C-Free

  ```
  write
    path=%path%;c:\Program Files (x86)\C-Free 5\mingw\bin
  and start C-Free from command line by
    start "" "c:\Program Files (x86)\C-Free 5\CppIDE.exe"
  or make a cf.bat that has that command inside and start by cf
  ```

3.3 CodeLite

- make a new tool:
  - `Plugins/External Tools/Configure Extrenal Tools.../New...`
  - Id: external_tool_0
  - Name: ComTestCpp
  - Tool Path: `C:\Program Files\Java\jre7\bin\java.exe` (tai missä Java onkin)
  - Arguments: `-jar E:\kurssit\ohj2\comtest\comtestcpp.jar $(CurrentFileFullPath)`
  - General: X Save... ja X Captuter...
  - OK
- make a keyboard shortcut:
  - `Settings/Keyboard Shortcuts...`
  - find External Tool 0
  - Edit and put f.ex Ctrl-Q
  - OK OK
- run: `Plugins/External Tools/!ComTestCpp` or `Ctrl-Q`
- be sure that you have \MinGW\bin in your path before starting CodeLite
- to add a ComTest code block, make a snippet:
  - `Plugins/SnipWiz/Settings.../Snippet`
  - Meny Entry: !comt
  - Keyboard shortcut: Ctrl-Shift-Q
  - koodiksi:

    ```
    \@code
     * <pre name="test">
     *   @
     * </pre>
     * \@endcode
    ```

  - Add
- to add a comment for function save the code (`Ctrl-S`) select the function line and press first `Ctrl-Shift-D` (shortcut for: right button/Code generation/Insert Doxygen comment)
- then fill teh comment and go to end of comment (before line */) and press `Ctrl-Shift-Q`

3.4 CodeBlocks

- make a new tool:
  - `Tools/Configure Tools.../Add`
  - Name: ComTestCpp
  - Executable: `C:\Program Files\Java\jre7\bin\java.exe` (tai missä Java onkin)
  - Arguments: `-jar E:\kurssit\ohj2\comtest\comtestcpp.jar ${ACTIVE_EDITOR_FILENAME}`
  - X Launch tool hidden with standard output redirected
  - OK
- run: `Plugins/External Tools/ComTestCpp`
- be sure that you have g++ in path before starting CodeBlocks

3.5 Visual Studio 2012

- make a new tool:
  - `Tools/External Tools.../Add`
  - Title: ComTestCpp
  - Command: `C:\Program Files\Java\jre7\bin\java.exe` (or where your Java is)

  - Arguments: `-jar E:\kurssit\ohj2\comtest\comtestcpp.jar $(ItemFileName)$(ItemExt)`
  - Initial directory: `$(ItemDir)`
  - X Use Output windows
  - OK
- run: `Tools/ComTestCpp`

3.6 Other IDE's

- copy idea's from previous examples