CTIP for C

<T2>
200711431 박성훈
200711473 최가영
200813530 퐁이핑
201160049 티에리뒤랑
presenter : 박성훈

2011.5.20

Software Verification
Konkuk Univ.

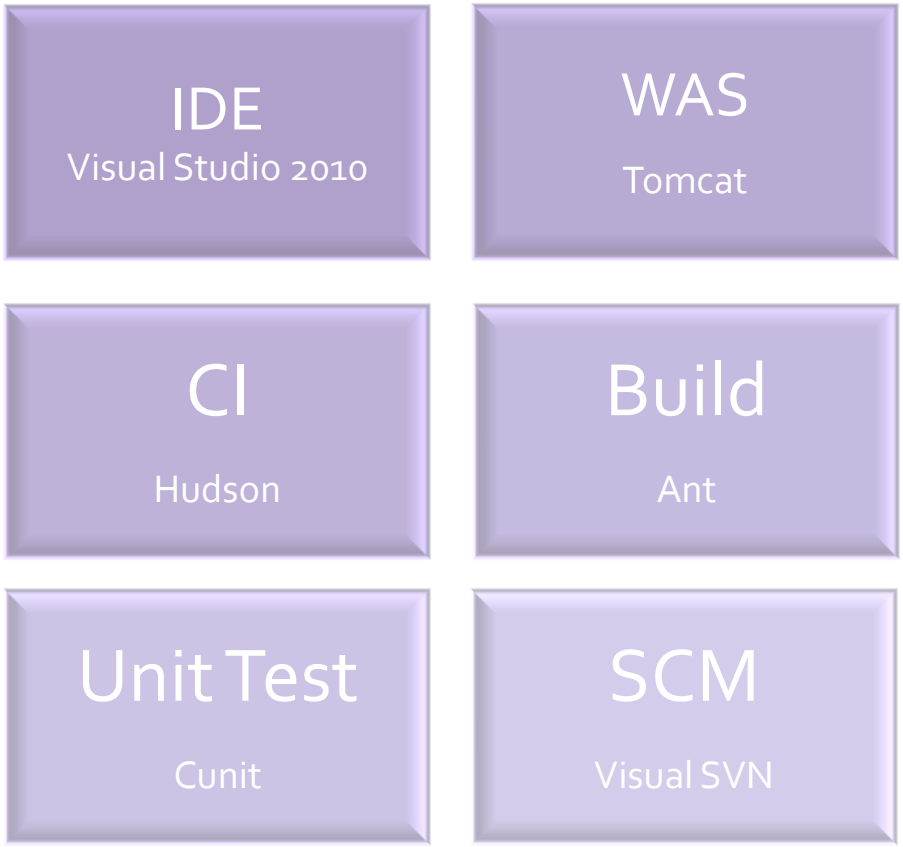
contents

- Chapter 1. Setting CTIP Environment for C
- Chapter 2. Testing

Chapter 1. Setting CTIP Environment for C

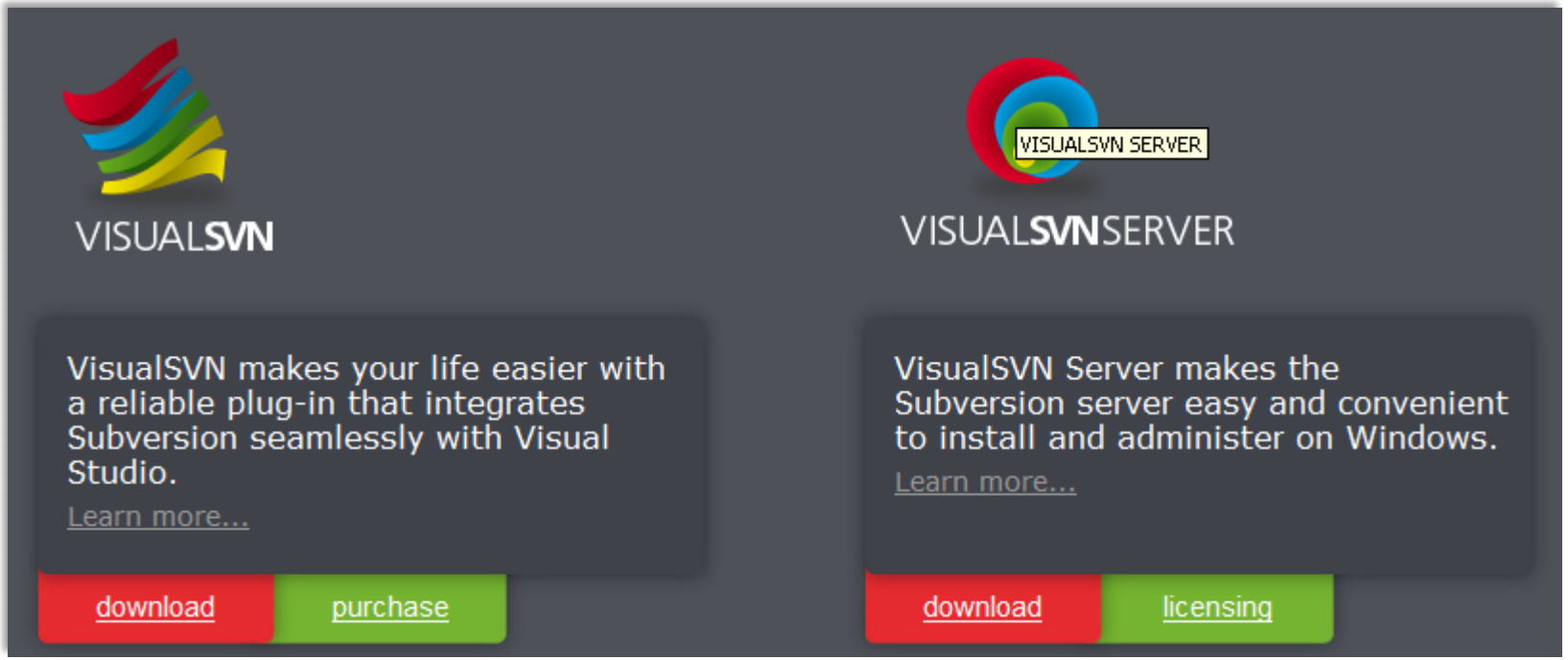
Set up CTIP Environment

- The tools of T2's project

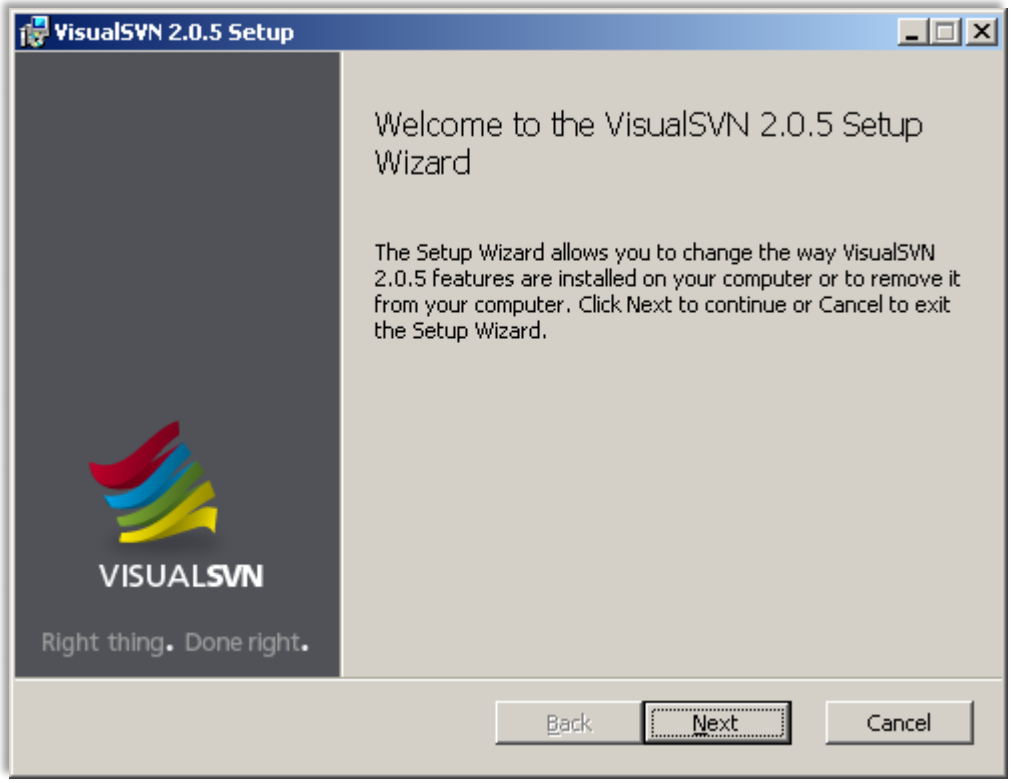


Install Visual SVN

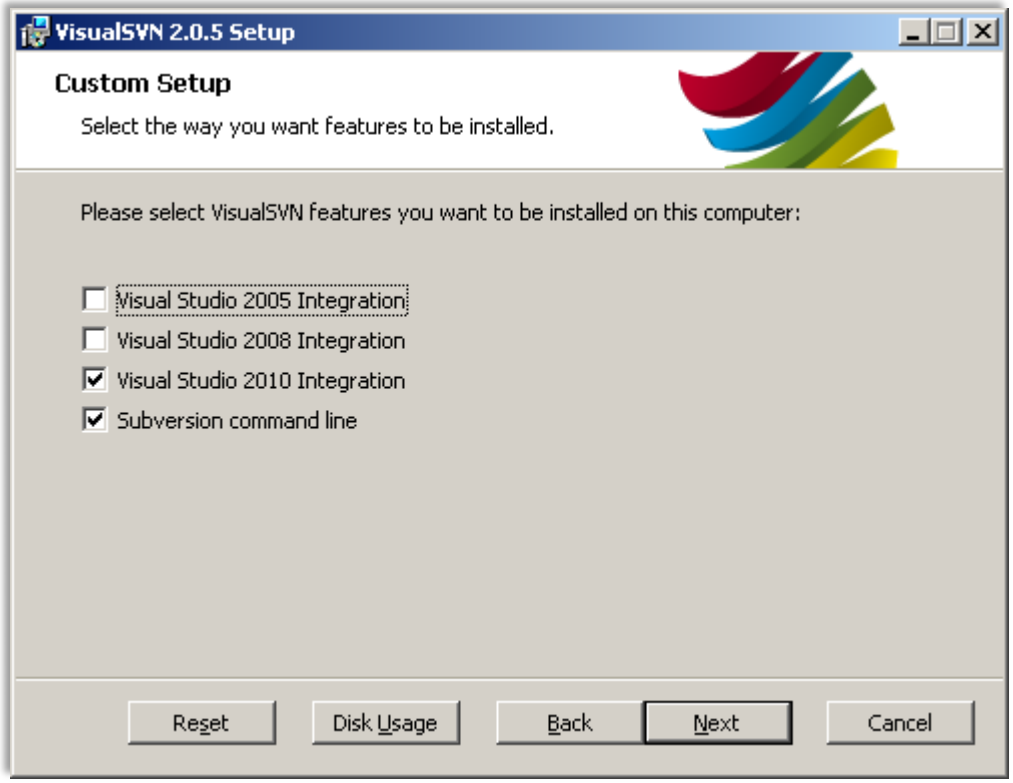
- http://www.visualsvn.com/



Install Visual SVN

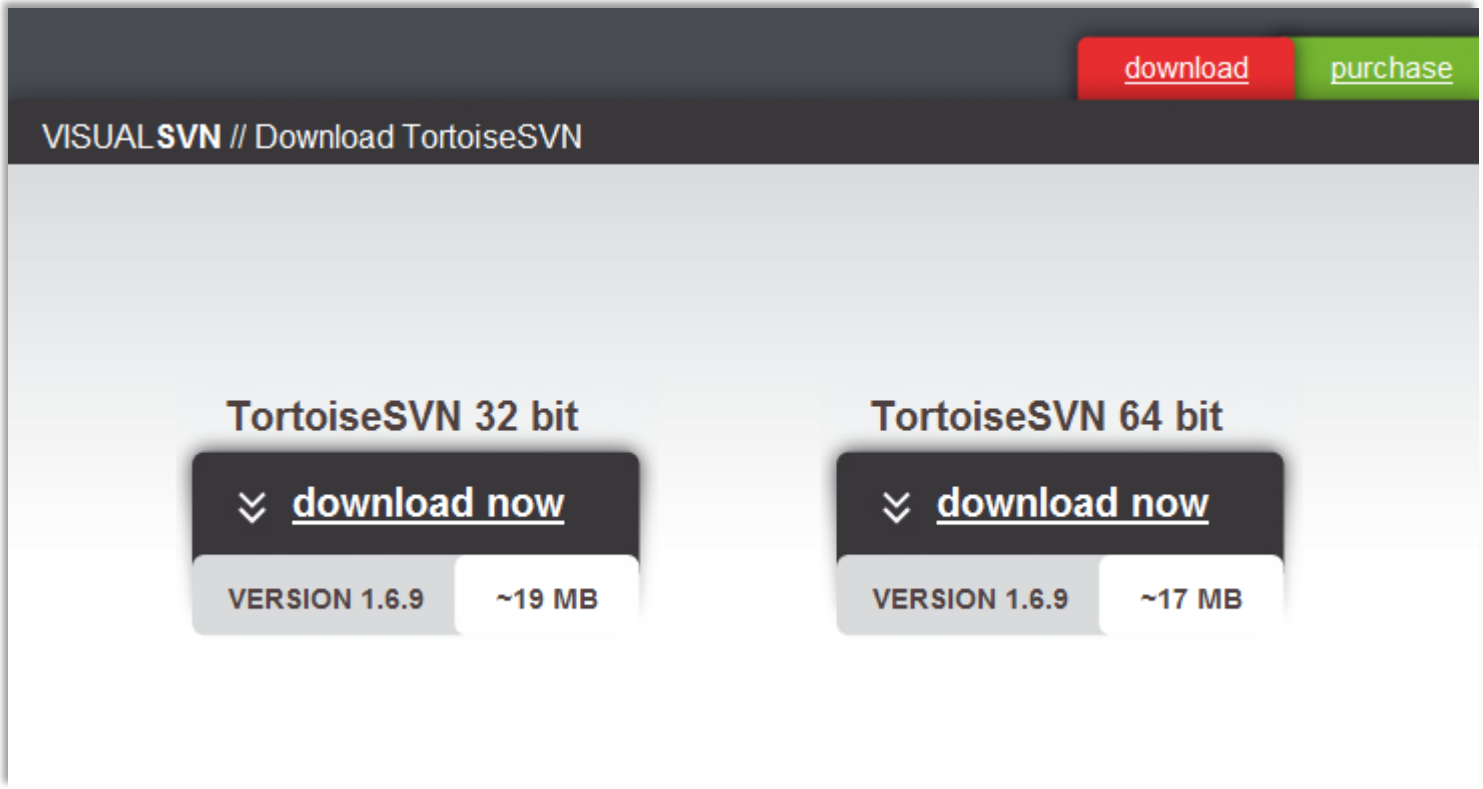


Install Visual SVN

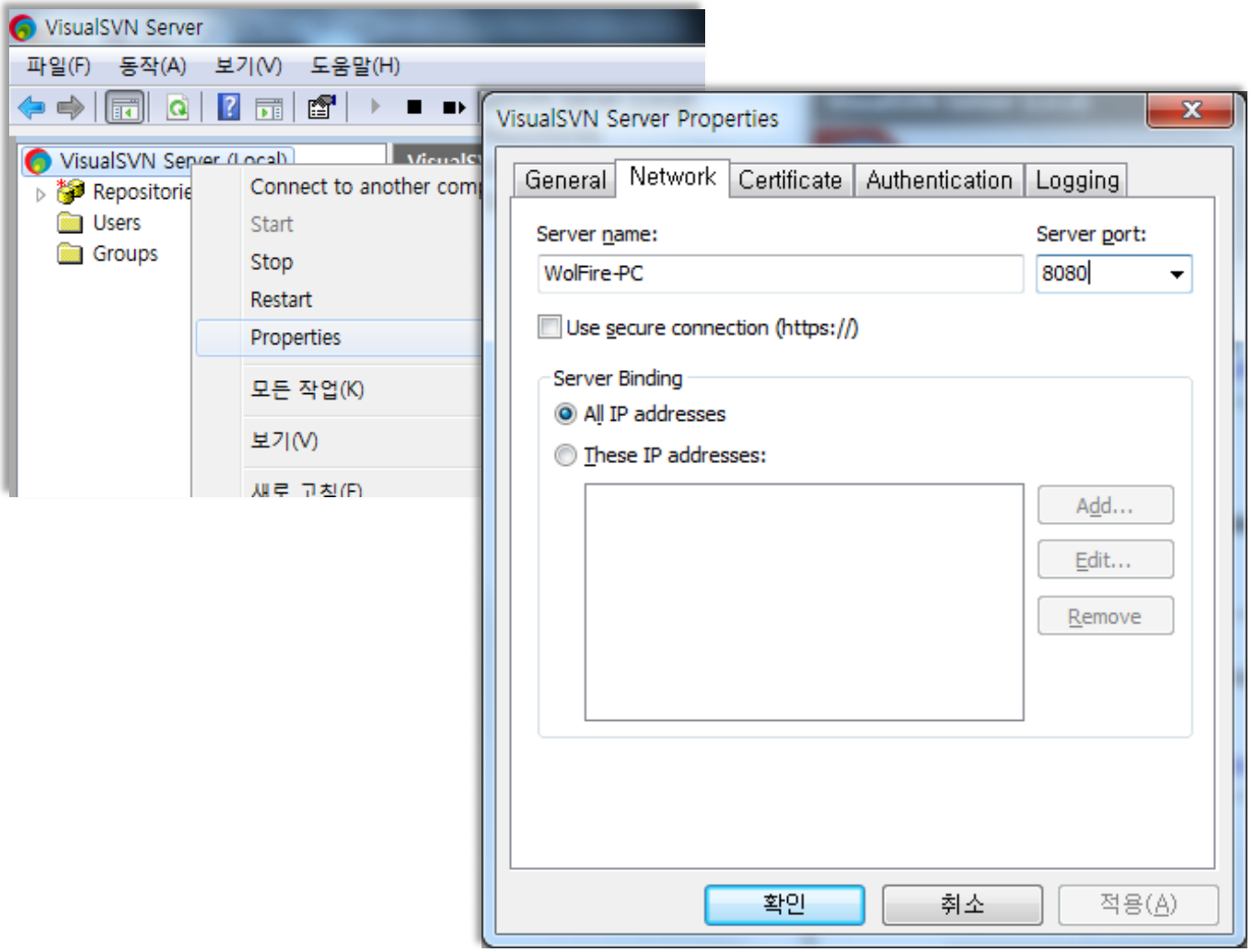


Install Visual SVN

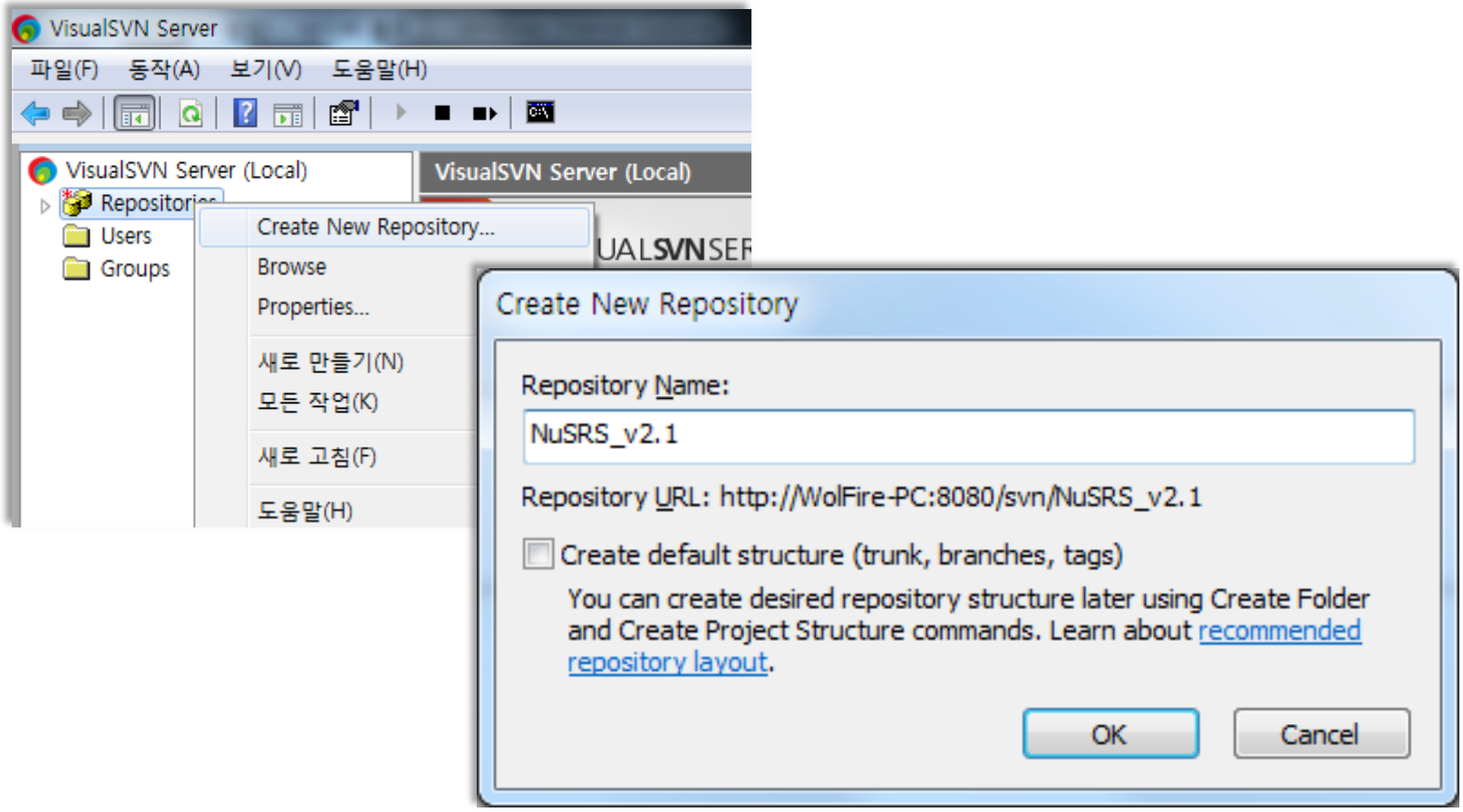
- http://www.visualsvn.com/visualsvn/download/tortoisesvn/



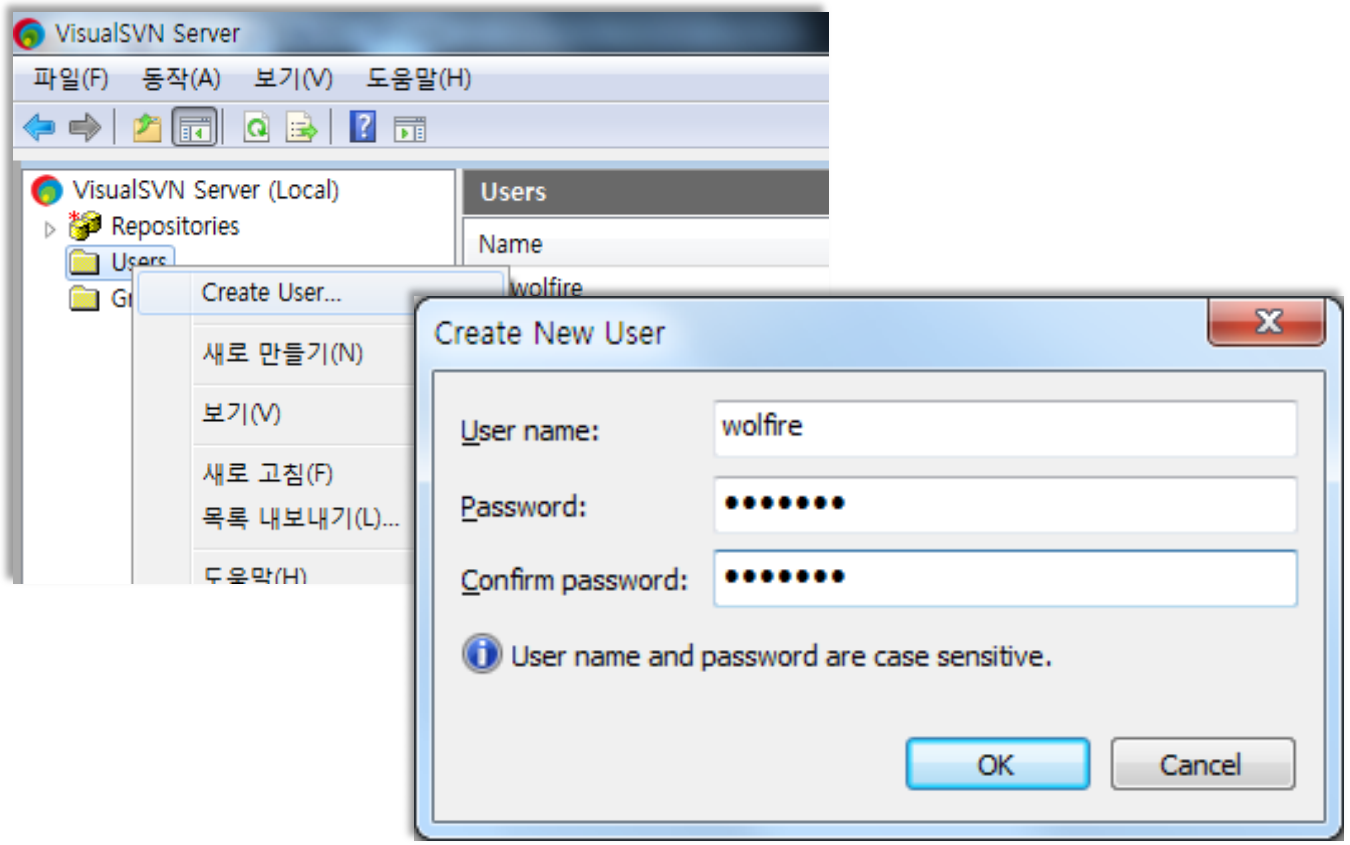
Setting Visual SVN



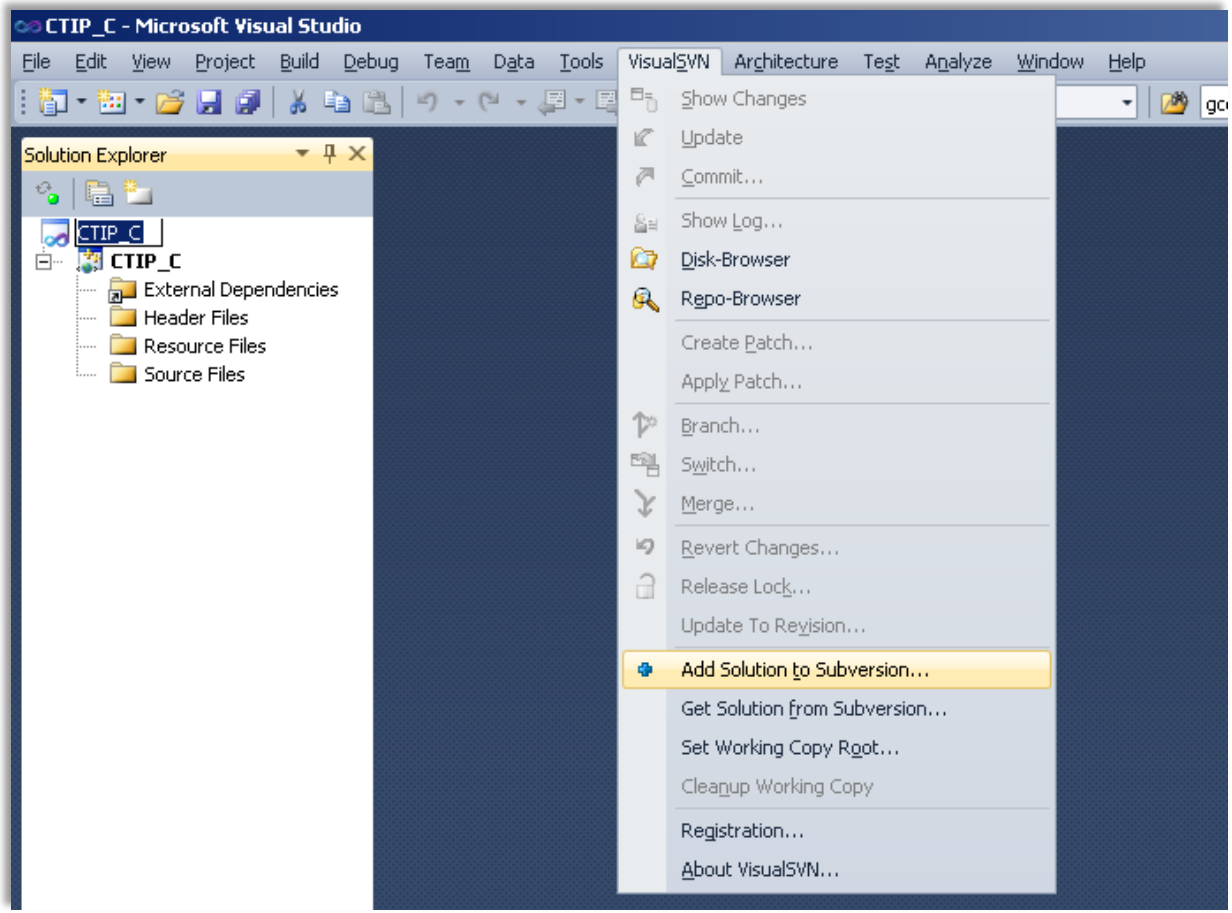
Setting Visual SVN



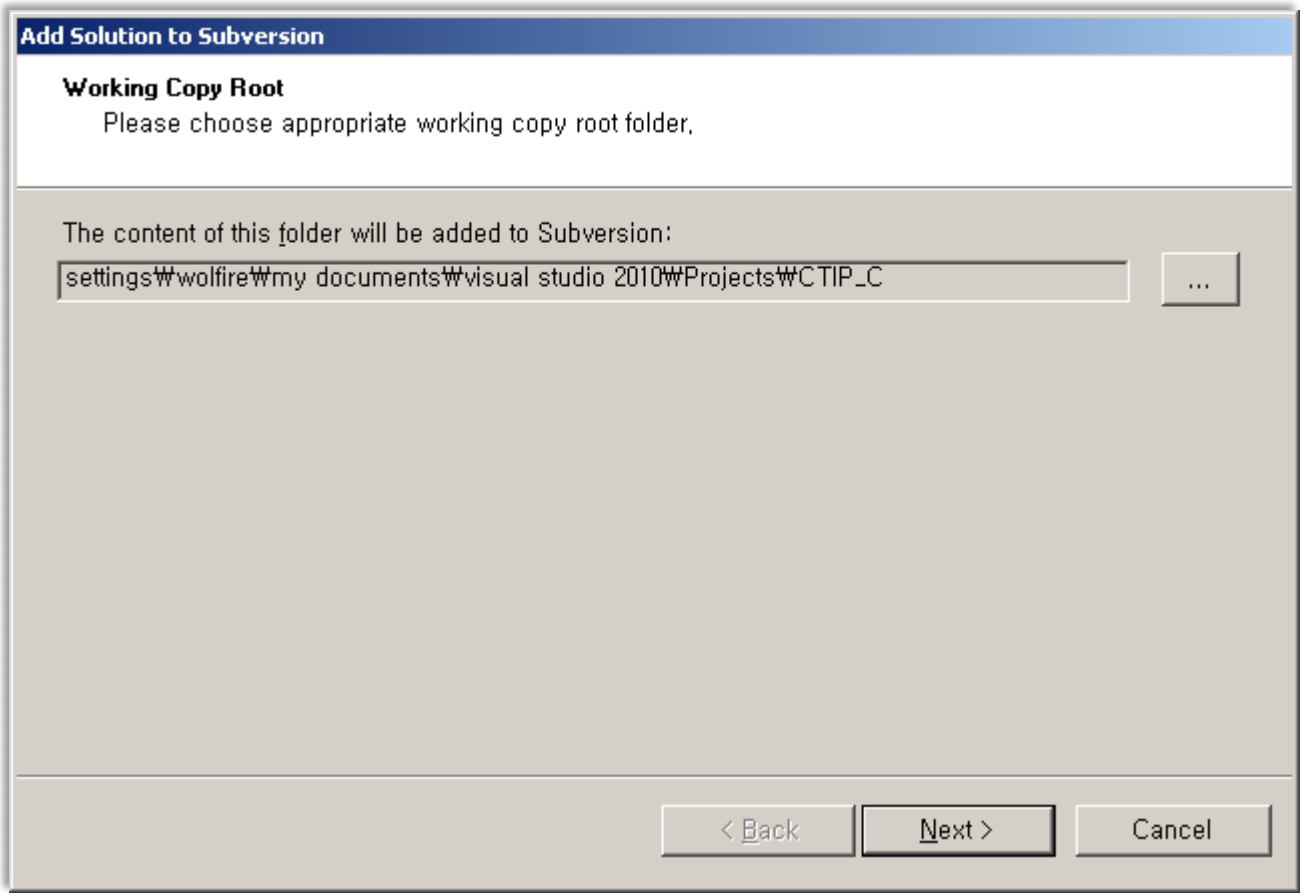
Setting Visual SVN



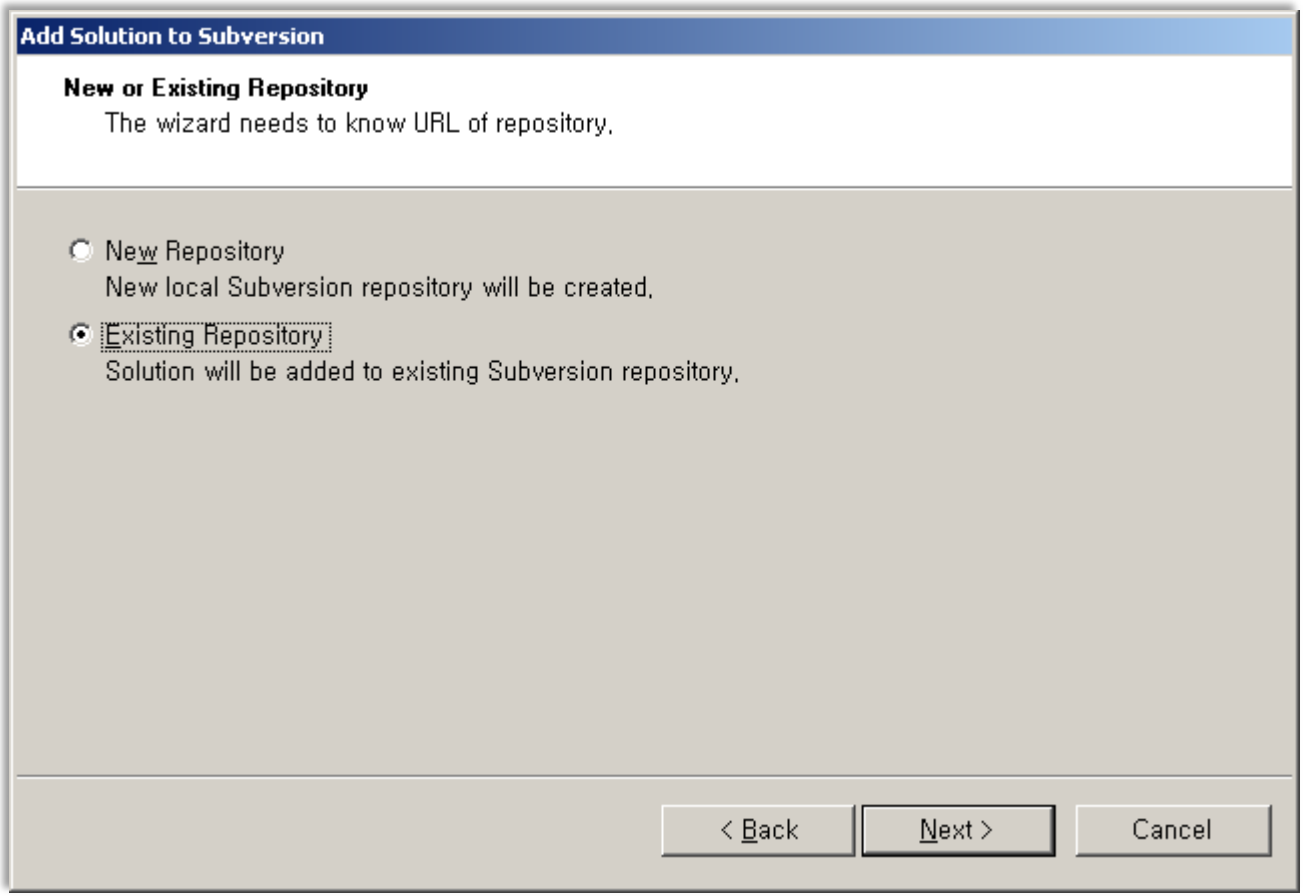
Setting Visual SVN



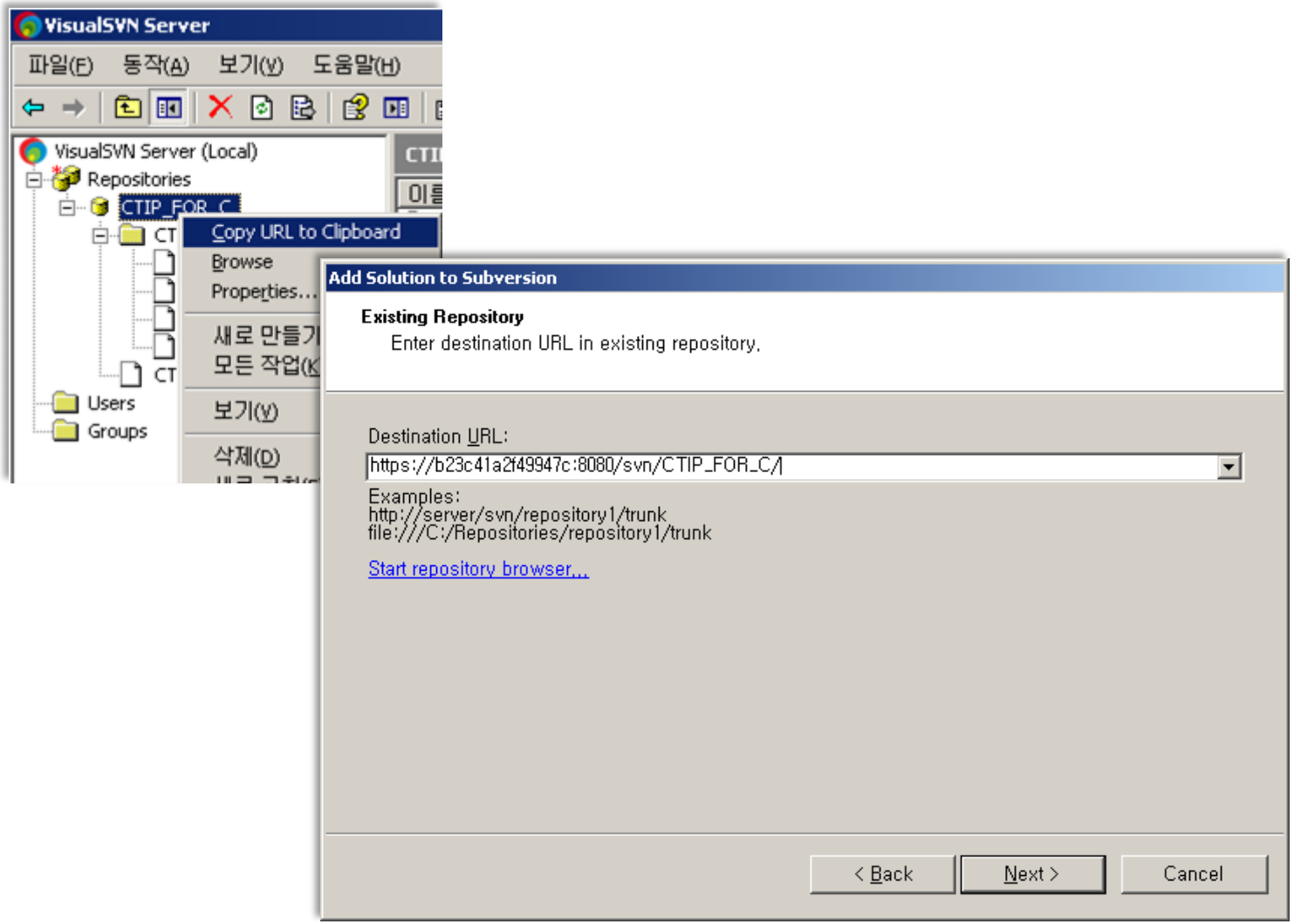
Setting Visual SVN



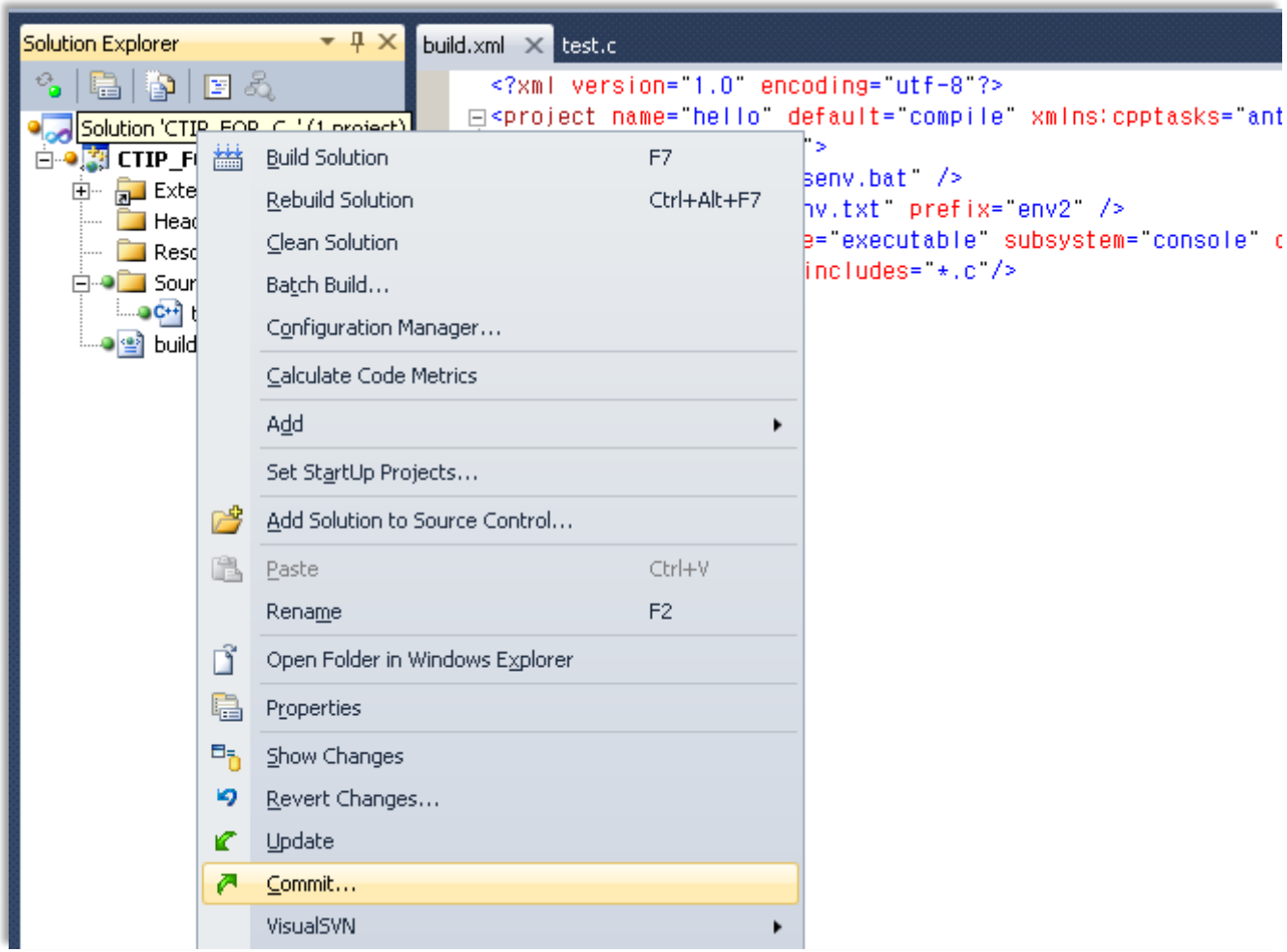
Setting Visual SVN



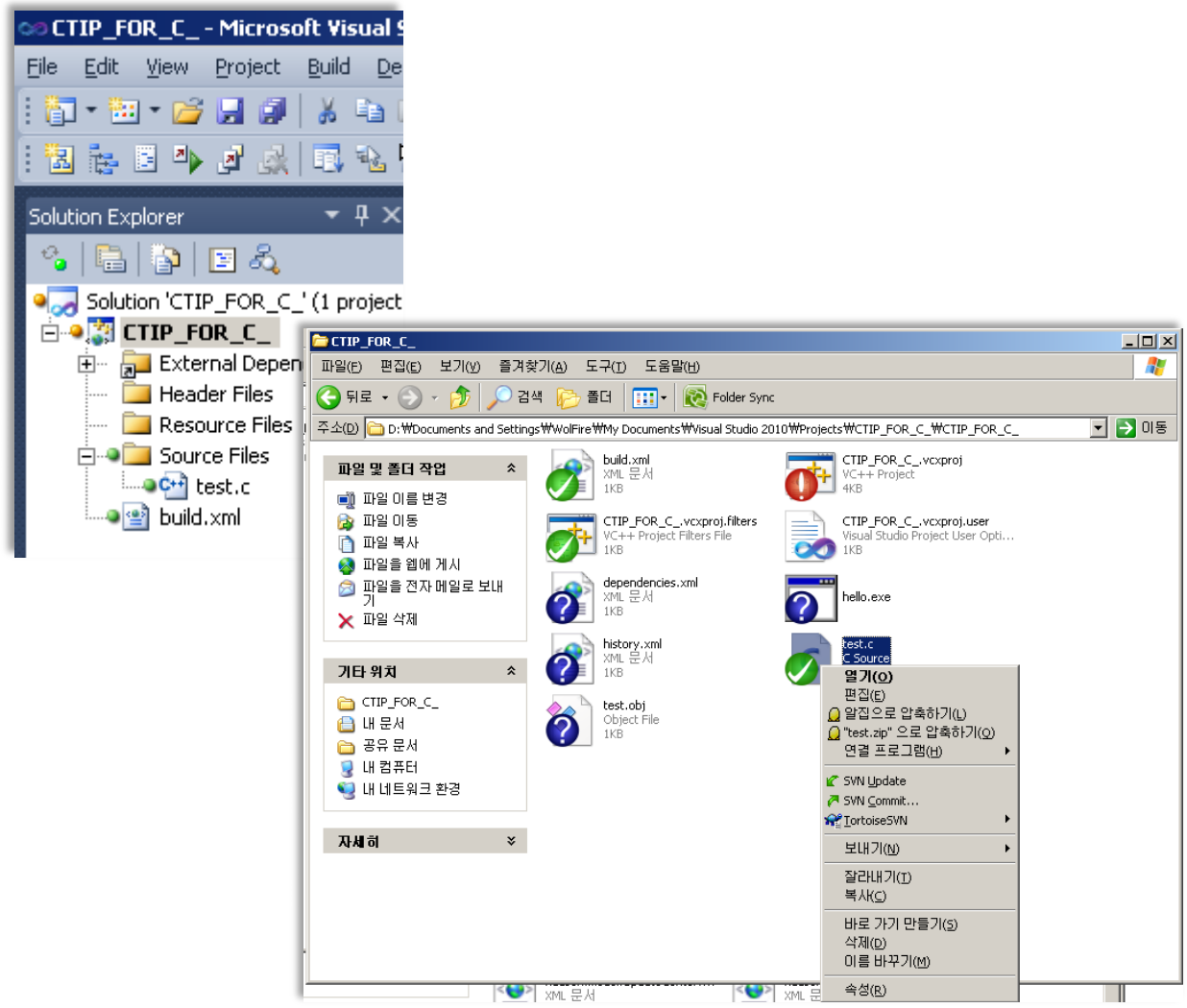
Setting Visual SVN



Setting Visual SVN

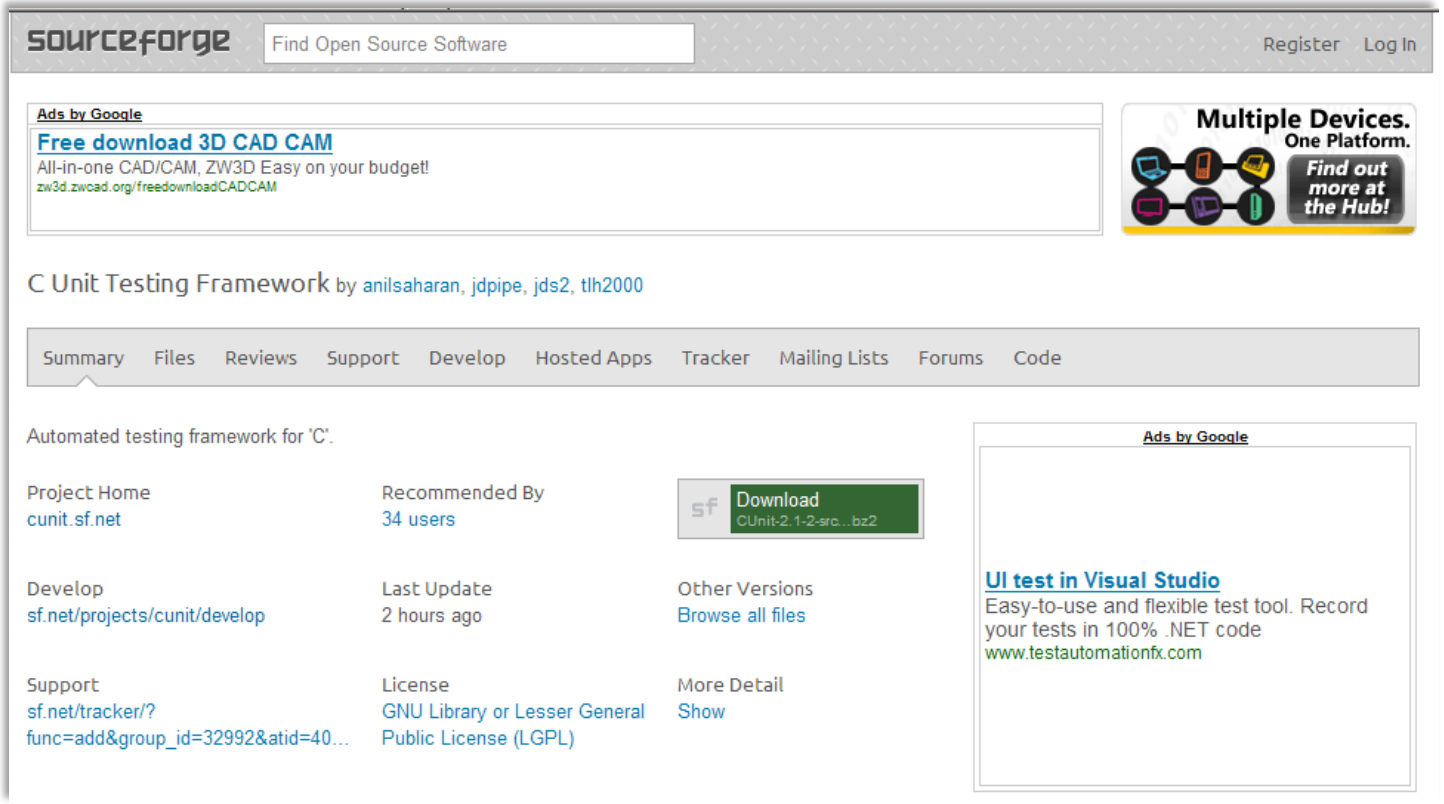


Setting Visual SVN



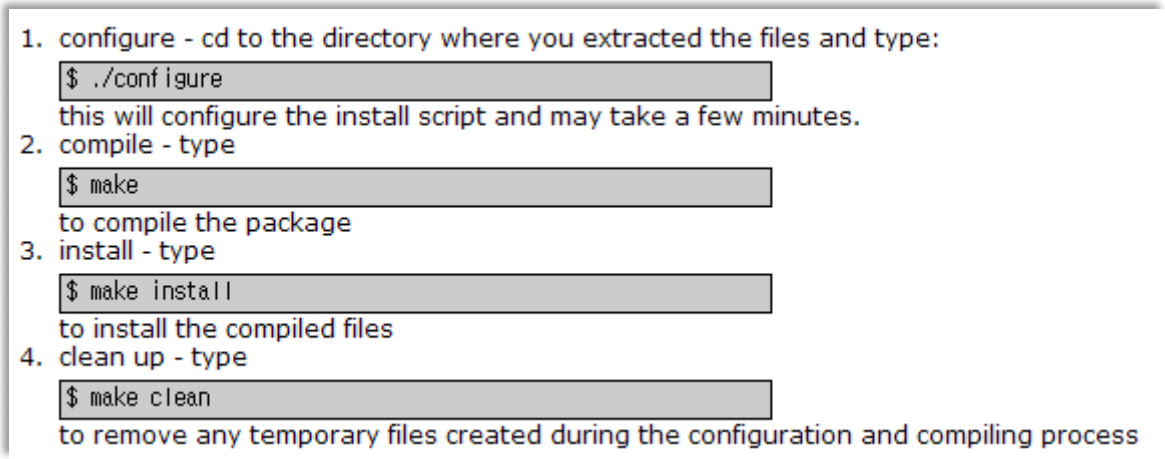
Install CUnit

- http://sourceforge.net/projects/cunit/



Install CUnit

- http://w3.scs.ryerson.ca/~schi/cps707/install-linux.html



1. configure - cd to the directory where you extracted the files and type:

```
$ ./configure
```

this will configure the install script and may take a few minutes.

2. compile - type

```
$ make
```

to compile the package

3. install - type

```
$ make install
```

to install the compiled files

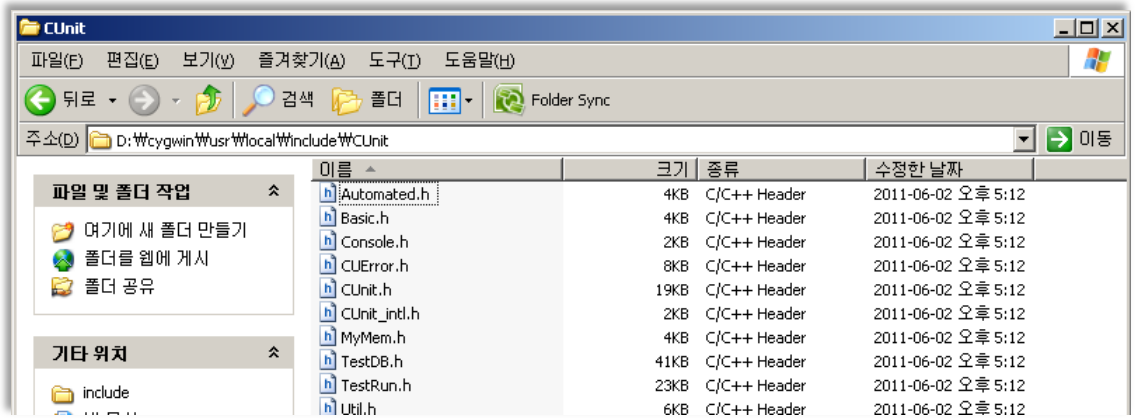
4. clean up - type

```
$ make clean
```

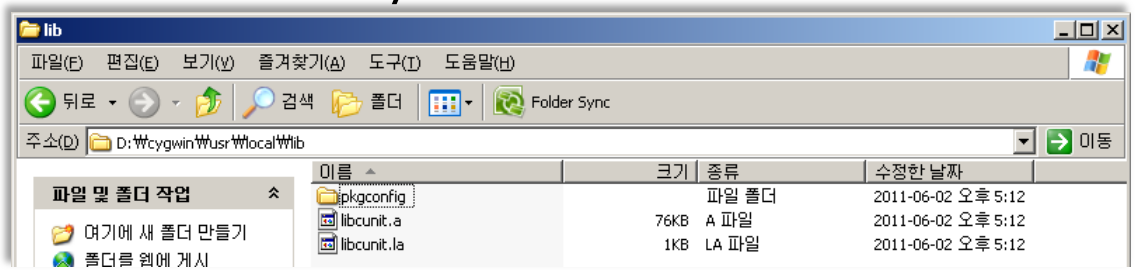
to remove any temporary files created during the configuration and compiling process

Install CUnit

- /usr/local/include/Cunit - Header File

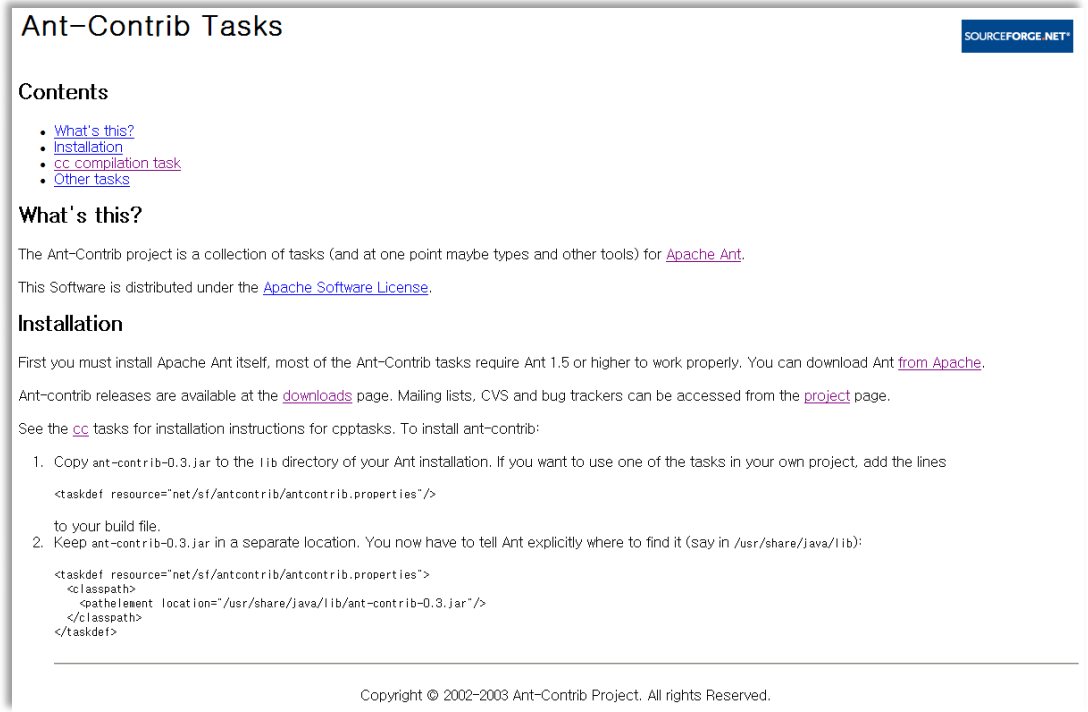


- /usr/local/lib - Library File



Install Ant CPPTASKS

- http://ant-contrib.sourceforge.net/
  Download : ant-contrib & cpptasks



Ant–Contrib Tasks

SOURCEFORGE.NET

Contents

- What's this?
- Installation
- cc compilation task
- Other tasks

What's this?

The Ant–Contrib project is a collection of tasks (and at one point maybe types and other tools) for Apache Ant.

This Software is distributed under the Apache Software License.

Installation

First you must install Apache Ant itself, most of the Ant–Contrib tasks require Ant 1.5 or higher to work properly. You can download Ant from Apache.

Ant–contrib releases are available at the downloads page. Mailing lists, CVS and bug trackers can be accessed from the project page.

See the cc tasks for installation instructions for cpptasks. To install ant–contrib:

1. Copy `ant-contrib-0.3.jar` to the `lib` directory of your Ant installation. If you want to use one of the tasks in your own project, add the lines

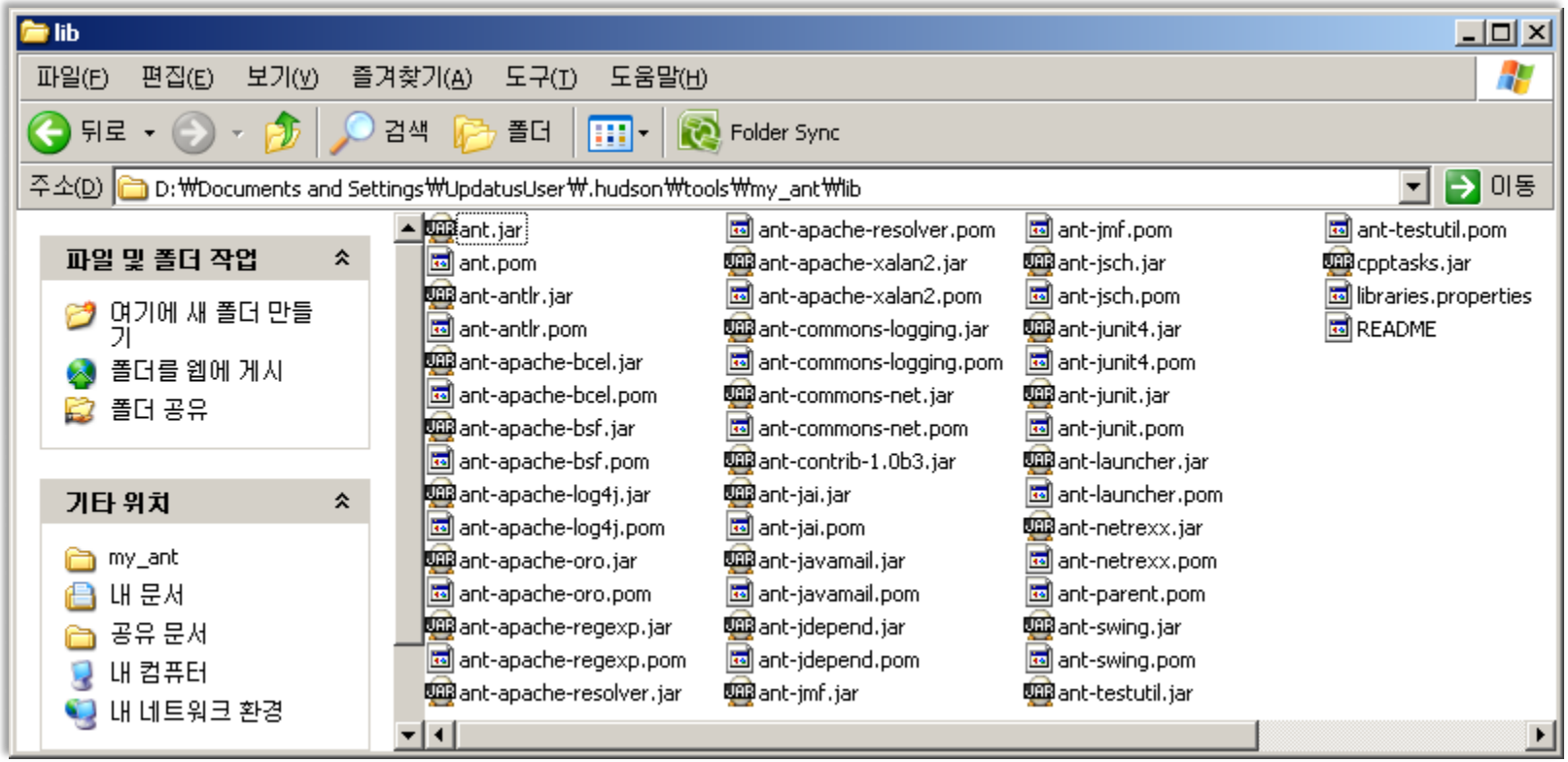
   ```
   <taskdef resource="net/sf/antcontrib/antcontrib.properties"/>
   ```

   to your build file.
2. Keep `ant-contrib-0.3.jar` in a separate location. You now have to tell Ant explicitly where to find it (say in `/usr/share/java/lib`):

   ```
   <taskdef resource="net/sf/antcontrib/antcontrib.properties">
     <classpath>
       <pathelement location="/usr/share/java/lib/ant-contrib-0.3.jar"/>
     </classpath>
   </taskdef>
   ```

Install Ant CPPTASKS

- Hudson - Ant

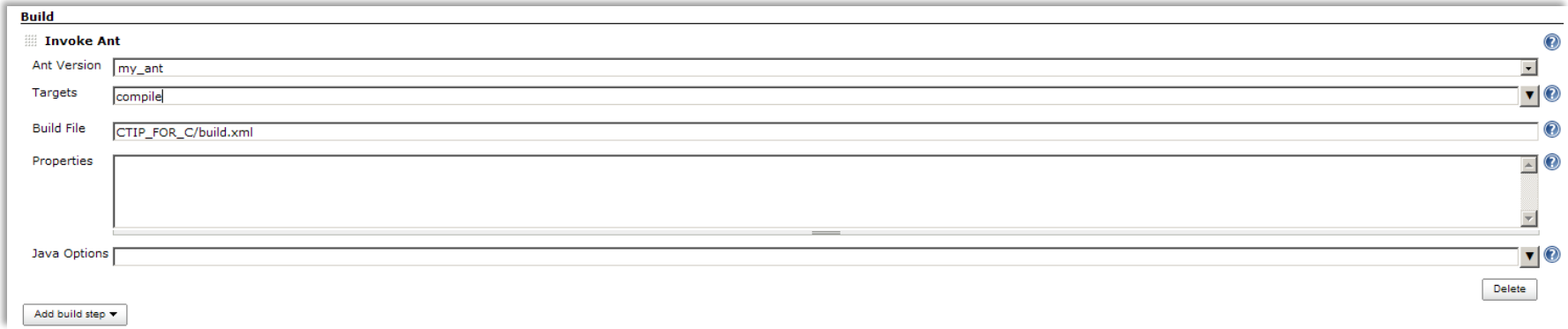


Hudson Setting

- Subversion

Install Ant CPPTASKS

- Ant



Ant Build Script

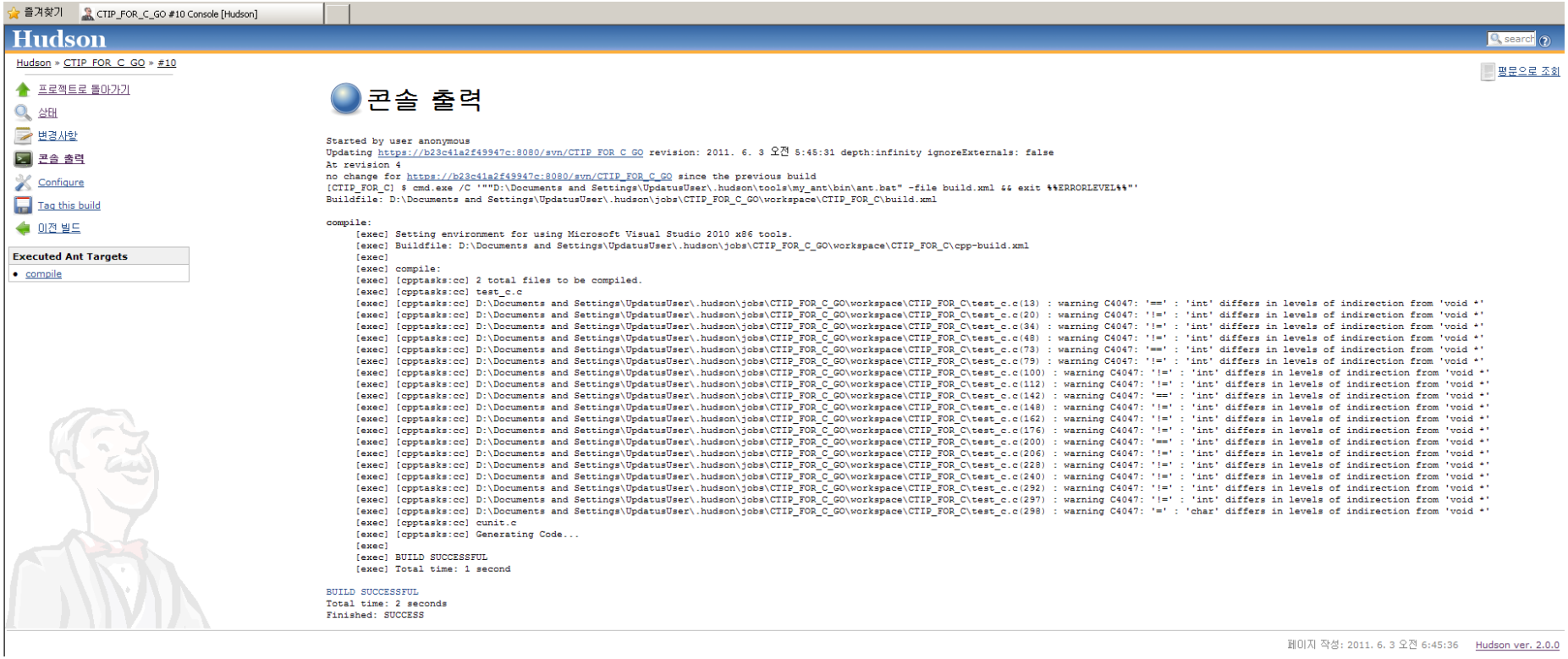
```
<?xml version="1.0" encoding="utf-8"?>

<project name="hello" default="compile">

  <property environment="env"/>

  <target name="compile">
    <exec executable="cmd" failonerror="true">
      <arg value="/C"/>
      <arg value="vsvars32.bat"/>
      <arg value="&amp;&amp;"/>
      <arg value="ant.bat"/>
      <arg value="-f"/>
      <arg value="cpp-build.xml"/>
    </exec>
  </target>

</project>
```

- Compiler - MSVC

Ant Build Script

```
<?xml version="1.0" encoding="utf-8"?>

<project name="hello" default="compile" xmlns:cpptasks="antlib:net.sf.antcontrib.cpptasks">
  <property name="link1" value="lib/libcunit.a"/>
  <property name="link2" value="liblibcunit.la"/>

  <target name="compile">

    <cpptasks:cc subsystem="console" name="msvc" link="static" multithreaded="true" exceptions="true">
      <fileset dir="." includes="*.c"/>
      <linker name="msvc">

        <libset libs="${link1}"/>
        <libset libs="${link2}"/>
      </linker>
    </cpptasks:cc>
  </target>

</project>
```

Hudson Build



Chapter 2. Testing

Target Project

- Digital Encoding
  - NRZL (Non-return-to-zero)
  - NRZI (Non-return-to-zero, inverted)
  - Manchester
  - DifMan

Target Project

- NRZL (Non-return-to-zero)
  - Input : 10110101

- NRZI (Non-return-to-zero, inverted)
  - Input : 10110101

Target Project

- Manchester
  - Input : 10110101

- DifMan
  - Input :10110101

Test Case

- NRZL & NRZI

Test Case - Input
00
01
10
11

- Manchester & DifMan

Test Case - Input
000
001
010
011
100
101
110
111

Test C File

```
//NRGL Encoding
int NRZL(char a[100], char b[100]){ ... }

//NRZI Encoding
int NRZI(char a[100], char b[100]){ ... }

// Manchester Encoding
int Manchester(char a[100], char b[100]){ ... }

//Differential Manchester
int DifMan(char a[100], char b[100]){ ... }
```

```
#include "CUnit/Console.h"
#include "CUnit/CUnit.h"

void test_NRZL_001(){ ... }
void test_NRZL_002(){ ... }
void test_NRZL_003(){ ... }
void test_NRZL_004(){ ... }

void test_NRZI_001(){ ... }
void test_NRZI_002(){ ... }
void test_NRZI_003(){ ... }
void test_NRZI_004(){ ... }

void test_MANCHESTER_001(){ ... }
void test_MANCHESTER_002(){ ... }
void test_MANCHESTER_003(){ ... }
void test_MANCHESTER_004(){ ... }
void test_MANCHESTER_005(){ ... }
void test_MANCHESTER_006(){ ... }
void test_MANCHESTER_007(){ ... }
void test_MANCHESTER_008(){ ... }

void test_DIFMAN_001(){ ... }
void test_DIFMAN_002(){ ... }
void test_DIFMAN_003(){ ... }
void test_DIFMAN_004(){ ... }
void test_DIFMAN_005(){ ... }
void test_DIFMAN_006(){ ... }
void test_DIFMAN_007(){ ... }
void test_DIFMAN_008(){ ... }
```

Test C File

```c
int main(void)
{
    CU_pSuite NRZL_suite, NRZI_suite, MANCHESTER_suite, DIFMAN_suite;

    CU_initialize_registry();

    NRZL_suite = CU_add_suite("NRZL_TEST", NULL, NULL);
    CU_add_test(NRZL_suite, "NRZL_TEST_01", test_NRZL_001);
    CU_add_test(NRZL_suite, "NRZL_TEST_02", test_NRZL_002);
    CU_add_test(NRZL_suite, "NRZL_TEST_03", test_NRZL_003);
    CU_add_test(NRZL_suite, "NRZL_TEST_04", test_NRZL_004);

    NRZI_suite = CU_add_suite("NRZI_TEST", NULL, NULL);
    CU_add_test(NRZI_suite, "NRZI_TEST_01", test_NRZI_001);
    CU_add_test(NRZI_suite, "NRZI_TEST_02", test_NRZI_002);
    CU_add_test(NRZI_suite, "NRZI_TEST_03", test_NRZI_003);
    CU_add_test(NRZI_suite, "NRZI_TEST_04", test_NRZI_004);

    MANCHESTER_suite = CU_add_suite("MANCHESTER_TEST", NULL, NULL);
    CU_add_test(MANCHESTER_suite, "MANCHESTER_TEST_01", test_MANCHESTER_001);
    CU_add_test(MANCHESTER_suite, "MANCHESTER_TEST_02", test_MANCHESTER_002);
    CU_add_test(MANCHESTER_suite, "MANCHESTER_TEST_03", test_MANCHESTER_003);
    CU_add_test(MANCHESTER_suite, "MANCHESTER_TEST_04", test_MANCHESTER_004);
    CU_add_test(MANCHESTER_suite, "MANCHESTER_TEST_05", test_MANCHESTER_005);
    CU_add_test(MANCHESTER_suite, "MANCHESTER_TEST_06", test_MANCHESTER_006);
    CU_add_test(MANCHESTER_suite, "MANCHESTER_TEST_07", test_MANCHESTER_007);
    CU_add_test(MANCHESTER_suite, "MANCHESTER_TEST_08", test_MANCHESTER_008);

    DIFMAN_suite = CU_add_suite("DIFMAN_TEST", NULL, NULL);
    CU_add_test(DIFMAN_suite, "DIFMAN_TEST_01", test_DIFMAN_001);
    CU_add_test(DIFMAN_suite, "DIFMAN_TEST_02", test_DIFMAN_002);
    CU_add_test(DIFMAN_suite, "DIFMAN_TEST_03", test_DIFMAN_003);
    CU_add_test(DIFMAN_suite, "DIFMAN_TEST_04", test_DIFMAN_004);
    CU_add_test(DIFMAN_suite, "DIFMAN_TEST_05", test_DIFMAN_005);
    CU_add_test(DIFMAN_suite, "DIFMAN_TEST_06", test_DIFMAN_006);
    CU_add_test(DIFMAN_suite, "DIFMAN_TEST_07", test_DIFMAN_007);
    CU_add_test(DIFMAN_suite, "DIFMAN_TEST_08", test_DIFMAN_008);

    CU_console_run_tests();
    CU_cleanup_registry();

 return 0;
}
```

- CU_initialize_registry()
- CU_add_suite()
- CU_add_test()
- CU_console_run_tests()
- CU_cleanup_registry()

Console Testing

```
WolFire@b23c41a2f49947c /home/CUnit-2.1-2/Examples
$ gcc -I/usr/local/include/CUnit -L/usr/local/lib test.c -lcunit -o test
```

```
WolFire@b23c41a2f49947c /home/CUnit-2.1-2/Examples
$ ./test


     CUnit - A Unit testing framework for C - Version 2.1-2
             http://cunit.sourceforge.net/


****************** CUNIT CONSOLE - MAIN MENU ********************************
(R)un  (S)elect  (L)ist  (A)ctivate  (F)ailures  (O)ptions  (H)elp  (Q)uit
Enter command: _
```

Console Testing

```
Running Suite : NRZL_TEST
     Running Test : NRZL_TEST_01
     Running Test : NRZL_TEST_02
     Running Test : NRZL_TEST_03
     Running Test : NRZL_TEST_04
Running Suite : NRZI_TEST
     Running Test : NRZI_TEST_01
     Running Test : NRZI_TEST_02
     Running Test : NRZI_TEST_03
     Running Test : NRZI_TEST_04
Running Suite : MANCHESTER_TEST
     Running Test : MANCHESTER_TEST_01
     Running Test : MANCHESTER_TEST_02
     Running Test : MANCHESTER_TEST_03
     Running Test : MANCHESTER_TEST_04
     Running Test : MANCHESTER_TEST_05
     Running Test : MANCHESTER_TEST_06
     Running Test : MANCHESTER_TEST_07
     Running Test : MANCHESTER_TEST_08
Running Suite : DIFMAN_TEST
     Running Test : DIFMAN_TEST_01
     Running Test : DIFMAN_TEST_02
     Running Test : DIFMAN_TEST_03
     Running Test : DIFMAN_TEST_04
     Running Test : DIFMAN_TEST_05
     Running Test : DIFMAN_TEST_06
     Running Test : DIFMAN_TEST_07
     Running Test : DIFMAN_TEST_08

Run Summary:    Type  Total    Ran Passed Failed Inactive
              suites      4      4    n/a      0        0
               tests     24     24     24      0        0
             asserts     64     64     64      0      n/a

Elapsed time =    0.000 seconds


****************** CUNIT CONSOLE - MAIN MENU ****************************
(R)un  (S)elect  (L)ist  (A)ctivate  (F)ailures  (O)ptions  (H)elp  (Q)uit
Enter command: _
```

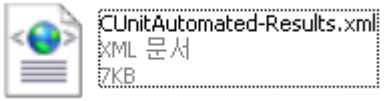
Test C File

```c
#include "CUnit/Automated.h"
#include "CUnit/CUnit.h"

int main(void)
{
    CU_pSuite NRZL_suite, NRZI_suite, MANCHESTER_suite, DIFMAN_suite;

    CU_initialize_registry();

    NRZL_suite = CU_add_suite("NRZL_TEST", NULL, NULL);
    CU_add_test(NRZL_suite, "NRZL_TEST_01", test_NRZL_001);
    CU_add_test(NRZL_suite, "NRZL_TEST_02", test_NRZL_002);
    CU_add_test(NRZL_suite, "NRZL_TEST_03", test_NRZL_003);
    CU_add_test(NRZL_suite, "NRZL_TEST_04", test_NRZL_004);

    NRZI_suite = CU_add_suite("NRZI_TEST", NULL, NULL);
    CU_add_test(NRZI_suite, "NRZI_TEST_01", test_NRZI_001);
    CU_add_test(NRZI_suite, "NRZI_TEST_02", test_NRZI_002);
    CU_add_test(NRZI_suite, "NRZI_TEST_03", test_NRZI_003);
    CU_add_test(NRZI_suite, "NRZI_TEST_04", test_NRZI_004);

    MANCHESTER_suite = CU_add_suite("MANCHESTER_TEST", NULL, NULL);
    CU_add_test(MANCHESTER_suite, "MANCHESTER_TEST_01", test_MANCHESTER_001);
    CU_add_test(MANCHESTER_suite, "MANCHESTER_TEST_02", test_MANCHESTER_002);
    CU_add_test(MANCHESTER_suite, "MANCHESTER_TEST_03", test_MANCHESTER_003);
    CU_add_test(MANCHESTER_suite, "MANCHESTER_TEST_04", test_MANCHESTER_004);
    CU_add_test(MANCHESTER_suite, "MANCHESTER_TEST_05", test_MANCHESTER_005);
    CU_add_test(MANCHESTER_suite, "MANCHESTER_TEST_06", test_MANCHESTER_006);
    CU_add_test(MANCHESTER_suite, "MANCHESTER_TEST_07", test_MANCHESTER_007);
    CU_add_test(MANCHESTER_suite, "MANCHESTER_TEST_08", test_MANCHESTER_008);

    DIFMAN_suite = CU_add_suite("DIFMAN_TEST", NULL, NULL);
    CU_add_test(DIFMAN_suite, "DIFMAN_TEST_01", test_DIFMAN_001);
    CU_add_test(DIFMAN_suite, "DIFMAN_TEST_02", test_DIFMAN_002);
    CU_add_test(DIFMAN_suite, "DIFMAN_TEST_03", test_DIFMAN_003);
    CU_add_test(DIFMAN_suite, "DIFMAN_TEST_04", test_DIFMAN_004);
    CU_add_test(DIFMAN_suite, "DIFMAN_TEST_05", test_DIFMAN_005);
    CU_add_test(DIFMAN_suite, "DIFMAN_TEST_06", test_DIFMAN_006);
    CU_add_test(DIFMAN_suite, "DIFMAN_TEST_07", test_DIFMAN_007);
    CU_add_test(DIFMAN_suite, "DIFMAN_TEST_08", test_DIFMAN_008);

    CU_automated_run_tests();
    CU_list_tests_to_file();
    CU_set_output_filename("Result.xml");
    CU_cleanup_registry();

 return 0;
}
```

- CU_initialize_registry()
- CU_add_suite()
- CU_add_test()
- CU_automated_run_tests()
- CU_list_tests_to_file()
- CU_set_output_filename()
- CU_cleanup_registry()

CUnit XML

- CUnit Result Xml




```
<?xml version="1.0" ?>
<?xml-stylesheet type="text/xsl" href="CUnit-Run.xsl" ?>
<!DOCTYPE CUNIT_TEST_RUN_REPORT SYSTEM "CUnit-Run.dtd">
<CUNIT_TEST_RUN_REPORT>
  <CUNIT_HEADER/>
  <CUNIT_RESULT_LISTING>
    <CUNIT_RUN_SUITE>
      <CUNIT_RUN_SUITE_SUCCESS>
        <SUITE_NAME> NRZL_TEST </SUITE_NAME>
        <CUNIT_RUN_TEST_RECORD>
          <CUNIT_RUN_TEST_SUCCESS>
            <TEST_NAME> NRZL_TEST_01 </TEST_NAME>
          </CUNIT_RUN_TEST_SUCCESS>
        </CUNIT_RUN_TEST_RECORD>
        <CUNIT_RUN_TEST_RECORD>
          <CUNIT_RUN_TEST_SUCCESS>
            <TEST_NAME> NRZL_TEST_02 </TEST_NAME>
          </CUNIT_RUN_TEST_SUCCESS>
        </CUNIT_RUN_TEST_RECORD>
        <CUNIT_RUN_TEST_RECORD>
          <CUNIT_RUN_TEST_SUCCESS>
            <TEST_NAME> NRZL_TEST_03 </TEST_NAME>
          </CUNIT_RUN_TEST_SUCCESS>
        </CUNIT_RUN_TEST_RECORD>
        <CUNIT_RUN_TEST_RECORD>
          <CUNIT_RUN_TEST_SUCCESS>
            <TEST_NAME> NRZL_TEST_04 </TEST_NAME>
          </CUNIT_RUN_TEST_SUCCESS>
        </CUNIT_RUN_TEST_RECORD>
      </CUNIT_RUN_SUITE_SUCCESS>
    </CUNIT_RUN_SUITE>
    <CUNIT_RUN_SUITE>
      <CUNIT_RUN_SUITE_SUCCESS>
        <SUITE_NAME> NRZI_TEST </SUITE_NAME>
        <CUNIT_RUN_TEST_RECORD>
          <CUNIT_RUN_TEST_SUCCESS>
            <TEST_NAME> NRZI_TEST_01 </TEST_NAME>
          </CUNIT_RUN_TEST_SUCCESS>
        </CUNIT_RUN_TEST_RECORD>
        <CUNIT_RUN_TEST_RECORD>
          <CUNIT_RUN_TEST_SUCCESS>
            <TEST_NAME> NRZI_TEST_02 </TEST_NAME>
```

CUnit To JUnit


```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:output method="xml" indent="yes" />

  <xsl:param name="suitename" />

  <xsl:variable name="cunitCount" select="count(//CUNIT_RUN_TEST_RECORD)"/>
  <xsl:variable name="cunitFailureCount" select="count(//CUNIT_RUN_TEST_FAILURE)"/>

  <xsl:template match="/">
    <testsuites>
      <xsl:attribute name="errors">0</xsl:attribute>
      <xsl:attribute name="failures">
        <xsl:value-of select="$cunitFailureCount"/>
      </xsl:attribute>
      <xsl:attribute name="tests">
        <xsl:value-of select="$cunitCount"/>
      </xsl:attribute>
      <xsl:attribute name="name">
        <xsl:value-of select="$suitename" />
      </xsl:attribute>
      <xsl:apply-templates />
    </testsuites>
  </xsl:template>

  <xsl:template match="/CUNIT_TEST_RUN_REPORT/CUNIT_RESULT_LISTING">
    <xsl:for-each select="CUNIT_RUN_SUITE/CUNIT_RUN_SUITE_SUCCESS">
      <xsl:variable name="localCunitFailureCount" select="count(CUNIT_RUN_TEST_RECORD/CUNIT_RUN_TEST_FAILURE)"/>
      <xsl:variable name="localCunitCount" select="count(CUNIT_RUN_TEST_RECORD)"/>
      <xsl:variable name="sn" select="normalize-space(SUITE_NAME/text())"/>
      <testsuite>
        <xsl:attribute name="errors">0</xsl:attribute>
        <xsl:attribute name="failures">
          <xsl:value-of select="$localCunitFailureCount"/>
        </xsl:attribute>
        <xsl:attribute name="tests">
          <xsl:value-of select="$localCunitCount"/>
        </xsl:attribute>
        <xsl:attribute name="name">
          <xsl:value-of select="$sn"/>
        </xsl:attribute>
        <xsl:apply-templates select="CUNIT_RUN_TEST_RECORD" />
```

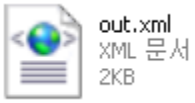
XSLTPROC

- http://xmlsoft.org/XSLT/index.html
  - Used Cygwin Plug-in

```
WolFire@b23c41a2f49947c ~
$ xsltproc --stringparam suitename testall -o out.xml cunit-to-junit.xsl CUnitA
utomated-Results.xml
```

JUnit Result Report

- Out.xml




```
<?xml version="1.0"?>
<testsuites errors="0" failures="0" tests="24" name="testall">
  <testsuite errors="0" failures="0" tests="4" name="NRZL_TEST">
    <testcase classname="" name="NRZL_TEST_01" time="0"/>
    <testcase classname="" name="NRZL_TEST_02" time="0"/>
    <testcase classname="" name="NRZL_TEST_03" time="0"/>
    <testcase classname="" name="NRZL_TEST_04" time="0"/>
  </testsuite>
  <testsuite errors="0" failures="0" tests="4" name="NRZI_TEST">
    <testcase classname="" name="NRZI_TEST_01" time="0"/>
    <testcase classname="" name="NRZI_TEST_02" time="0"/>
    <testcase classname="" name="NRZI_TEST_03" time="0"/>
    <testcase classname="" name="NRZI_TEST_04" time="0"/>
  </testsuite>
  <testsuite errors="0" failures="0" tests="8" name="MANCHESTER_TEST">
    <testcase classname="" name="MANCHESTER_TEST_01" time="0"/>
    <testcase classname="" name="MANCHESTER_TEST_02" time="0"/>
    <testcase classname="" name="MANCHESTER_TEST_03" time="0"/>
    <testcase classname="" name="MANCHESTER_TEST_04" time="0"/>
    <testcase classname="" name="MANCHESTER_TEST_05" time="0"/>
    <testcase classname="" name="MANCHESTER_TEST_06" time="0"/>
    <testcase classname="" name="MANCHESTER_TEST_07" time="0"/>
    <testcase classname="" name="MANCHESTER_TEST_08" time="0"/>
  </testsuite>
  <testsuite errors="0" failures="0" tests="8" name="DIFMAN_TEST">
    <testcase classname="" name="DIFMAN_TEST_01" time="0"/>
    <testcase classname="" name="DIFMAN_TEST_02" time="0"/>
    <testcase classname="" name="DIFMAN_TEST_03" time="0"/>
    <testcase classname="" name="DIFMAN_TEST_04" time="0"/>
    <testcase classname="" name="DIFMAN_TEST_05" time="0"/>
    <testcase classname="" name="DIFMAN_TEST_06" time="0"/>
    <testcase classname="" name="DIFMAN_TEST_07" time="0"/>
    <testcase classname="" name="DIFMAN_TEST_08" time="0"/>
  </testsuite>
</testsuites>
```

JUnit Result Report

```
<?xml version="1.0" encoding="utf-8"?>

<project name="hello" default="compile" xmlns:cpptasks="antlib:net.sf.antcontrib.cpptasks">
  <property name="link1" value="lib/libcunit.a"/>
  <property name="link2" value="liblibcunit.la"/>

  <target name="compile">

    <cpptasks:cc subsystem="console" name="msvc" multithreaded="true" exceptions="true" outtype="executable" outfile="test.exe">
      <fileset dir="." includes="*.c"/>

    </cpptasks:cc>
  </target>

</project>
```

- Output … test.exe.lib
  - I don't know Why…