

# CTIP for C

## CTIP Environment For C

200511349 장기웅

200711472 진교선

200511300 강정희

200511310 김진규

# Content

---

## Tools

- ✓ **IDE Selection**
- ✓ **Unit Testing Tools**
- ✓ **Selected Tools**
- ✓ **SCM / CI / Auto Build**
- ✓ **Overview**
- ✓ **Install Tools**

# Content

---

## Testing C Source

- ✓ Introduction
- ✓ Specification
- ✓ Pairwise Test case
- ✓ Test Result

## References

# Tools

Excluded

Selected

Install

IDE Selection

Unit Testing Tools

Code Coverage Tools

SCM/CI/AutoBuild

Overview

Install Tools

# IDE Selection

---

## Visual Studio 2008 vs Eclipse C Plugin

**Visual Studio 2008 Team Foundation Server를 활용하여 CI 환경을 구축 하고자 하였으나 C 환경은 지원하지 않았다. Java 환경에서와 같이 다양한 Plugin을 기대하고 Eclipse에서 C 개발 환경을 구축 하였다. 그러나 Code Management Tool을 지원하지 않는 문제가 있었다. 결국에는 Visual Studio Addin 들을 모아서 Subversion Commit 까지 성공하여 IDE 는 Visual Studio 2008 Professional 을 사용하게 되었다.**

# Excluded Tools

---

## Unit Testing Tools

xUnit 계열의 Tool을 선택하여 Hudson 환경에서 손쉽게 적용 가능 할 것이라는 예측에서 출발

- **CUnit**

- JUnit 과 가장 비슷한 형태로 사용 가능
- Visual Studio(VC Compiler)에 적용이 안됨

- **CppUnit**

- Visual Studio에 Addin 형태로 적용가능
- Native C code에 대한 테스트가 불가능

# Selection Tools

---

## Unit Testing Tool

- **Visual Assert**
  - JUnit 과 유사한 형태로 Test Case 작성이 가능 Assert 문을 통한 Test case Coding
  - Visual Studio와 완벽하게 연동(영문판)
  - Free ware
  - C 및 C++ Test 지원
  - Homepage에서 Test Framework에 대한 API 제공 (<http://www.visualassert.com/visual-studio-addin/doc/TestAPI.html>)
  - Test 이후 Coverage 측정 Tool과의 연계성이 부족

# Excluded Tools

---

## Code Coverage Tools

- **BullsEyeCoverage / Testwell CTC++**
  - Visual Studio와 연동 가능(BullsEye)
  - C/C++ Analysis 를 지원
  - 평가판을 요구 하였지만 답변이 없었음
- **TestCocoon**
  - Visual Studio와 연동(제한적)
  - C/C++ Code Coverage 지원
  - Open Source Project
  - Manual이 상세하지 않았음

# Selection Tools

---

## Code Coverage Tools

- **DevPartner Studio**
  - 유료 Tool이긴 하지만 Trial Version 제공
  - Error Detection & Code Coverage 지원
  - Visual Studio Addin 형태로 지원
  - 별도의 IDE 외부 설정이 불필요
  - Test Tool과의 연계성이 부족

# Selected Tools

---

## SCM Tools - Subversion

- **Visual SVN**
  - Repository Management Tool
  - Version 관리까지 되는 버전은 유료
- **AnkhSVN**
  - Visual Studio Addin
  - 2.1 이하 버전은 2008과 호환성이 떨어짐
  - Subclipse와 거의 유사한 형태로 사용이 가능
  - Visual SVN과의 연동이 잘됨

# Selected Tools

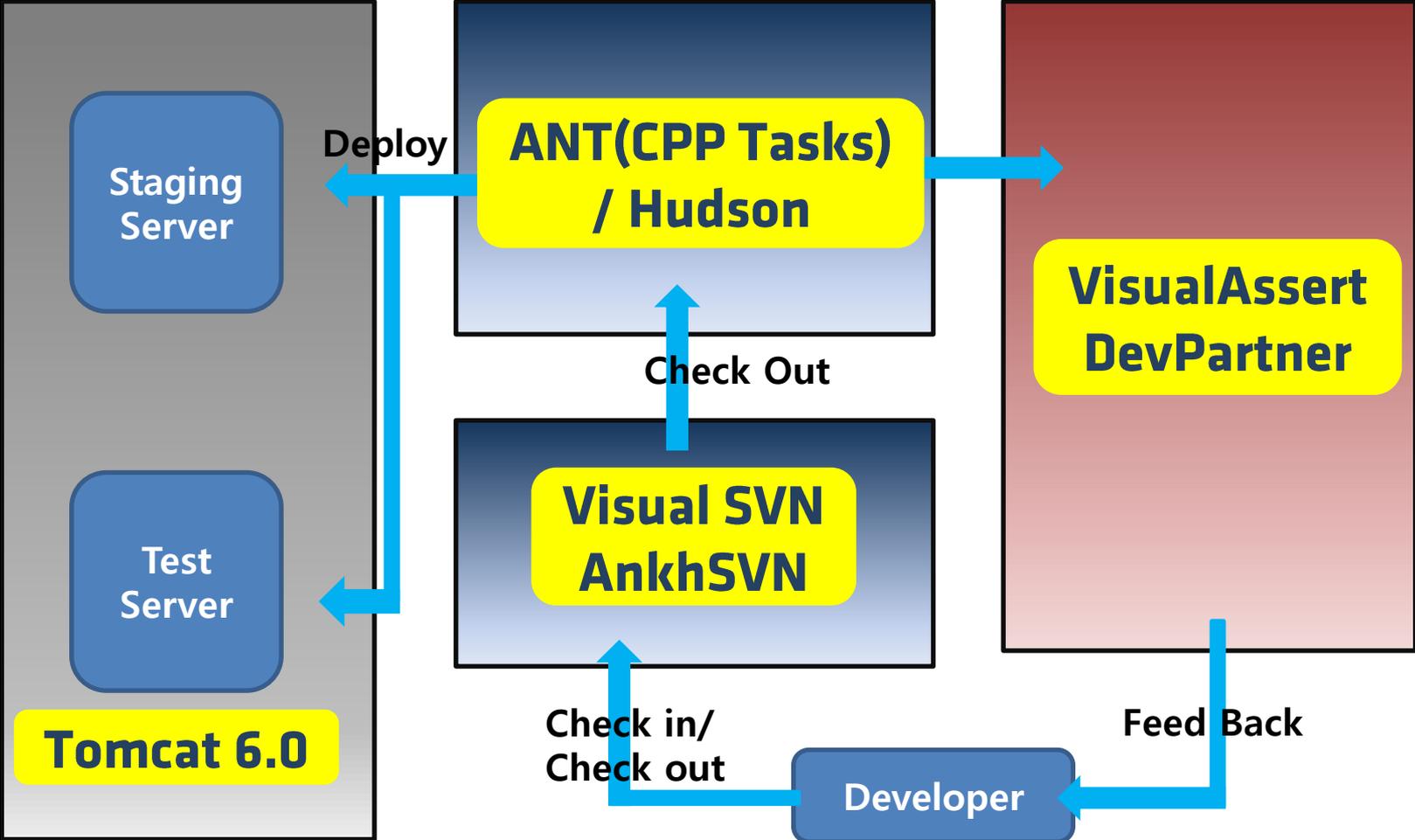
---

## Build Automation / CI Server

- **ANT(CPP Tasks)**
  - ANT C/C++ Build용 Plug-in
  - 이전 Java용 환경을 그대로 사용 할 수 있음
- **Hudson**
  - CI Server
  - 다른 SVN 이후 거치는 Tool들이 이전 Java 환경과 동일하여 그대로 사용

# CTIP Environment Overview

## Diagram For Our Environment



# Additional Tool Setting

## Cpptasks for Apache Ant

ant-contrib

SOURCEFORGE.NET®

Last Published: 2008-04-02 | ant-contrib > cpptasks

### Get cpptasks

[Download](#)  
[Changes](#)  
[License](#)

### About cpptasks

#### Usage

#### Elements

[cc](#)  
[compiler](#)  
[defineset](#)  
[distributer](#)  
[libraryset](#)  
[linker](#)  
[syslibraryset](#)  
[targetplatform](#)  
[versioninfo](#)

### Community

[Mailing Lists](#)  
[Issue Tracking](#)  
[Project Page](#)  
[News](#)

### Development

[Repository](#)  
[JavaDoc](#)  
[Continuous Integration](#)



## cpptasks for Apache Ant

The cc task can compile various source languages and produce executables, shared libraries (aka DLL's) and static libraries. Compiler adaptors are currently available for C/C++, FORTRAN, MIDL and Windows Resource compilers.

The task can be used with Apache Ant 1.5 and later. This software is not a product of the Apache Software Foundation (ASF) and no endorsement by the ASF is implied.

To use: \* Place cpptasks.jar into Ant's classpath by placing in Ant's lib directory, adding to CLASSPATH environment variable or using the -lib command line option.

- Add type and task definitions in build file using either taskdef or antlib.
- Add `cc` element to some target in your build file.
- Set path and environment variables to be able to run compiler from command line.
- Build project.

Trivial Sample using taskdef (compatible with Ant 1.5 or later):

```
<project name="hello" default="compile">
  <taskdef resource="cpptasks.tasks"/>
  <target name="compile">
    <mkdir dir="target/main/obj"/>
    <cc outtype="executable" subsystem="console" outfile="target/hello" objdir="target/main/obj">
      <fileset dir="src/main/c" includes="*.c"/>
    </cc>
  </target>
</project>
```

Trivial Sample using antlib (compatible with Ant 1.6 or later):

```
<project name="hello" default="compile" xmlns:cpptasks="antlib:net.sf.antcontrib.cpptasks">
```

# Additional Tool Setting

## Hudson Project Configuration

**Advanced Project Options** Advanced...

---

**Source Code Management**

None  
 CVS  
 Subversion

Modules

Repository URL  ?

Local module directory (optional)  ?

Add more locations...

Use update

If checked, Hudson will use 'svn update' whenever possible, making the build faster. But this causes the artifacts from the previous build to remain when a new build starts.

Revert

If checked, Hudson will do 'svn revert' before doing 'svn update'. This slows it down, but will prevent files being modified from build to build.

Repository browser  ?

Advanced...

---

**Build Triggers**

Build after other projects are built ?

Poll SCM ?

Build periodically ?

---

**Build**

**Invoke Ant** ?

Ant Version  ?

Targets  ?

Advanced...

Delete

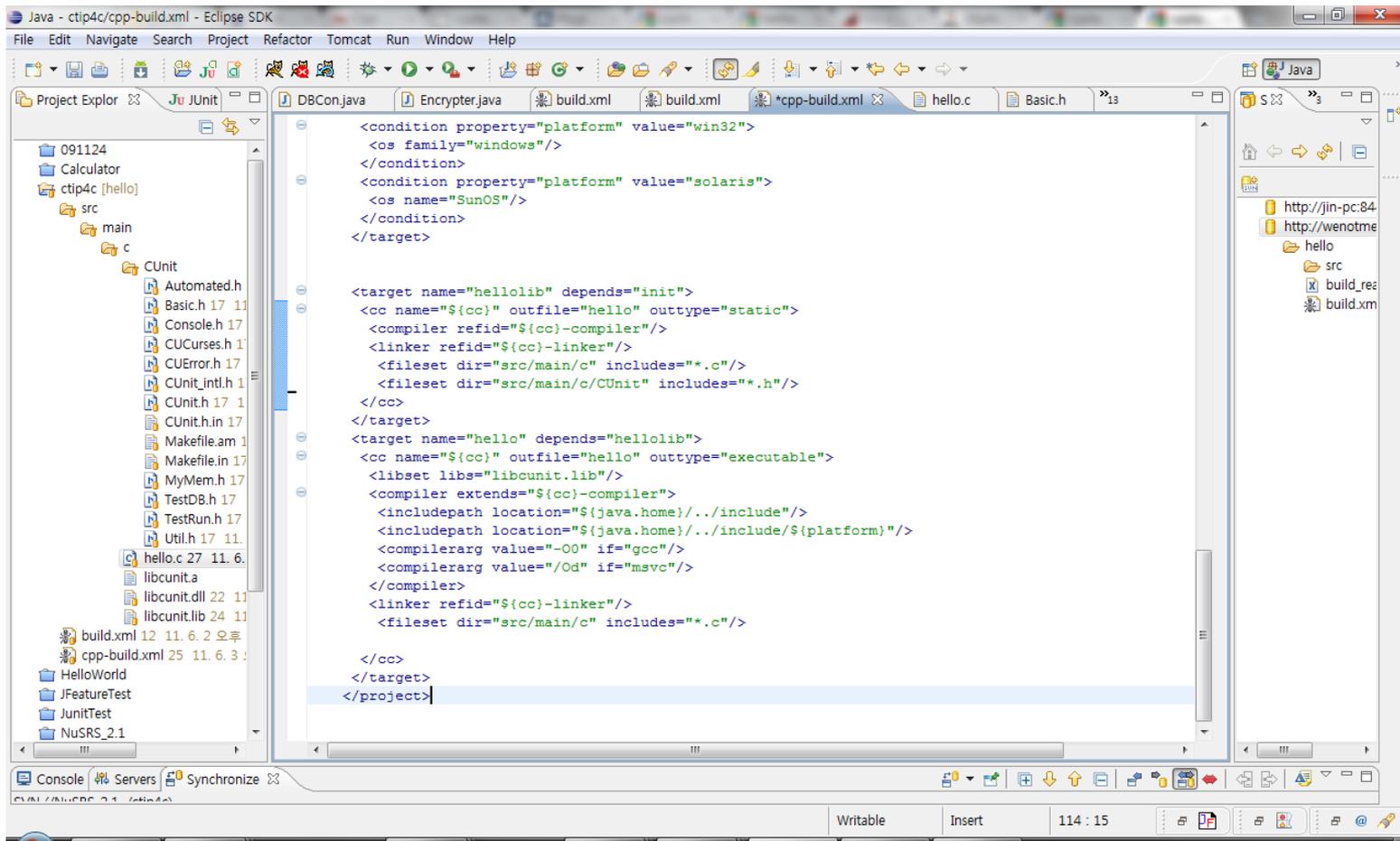
# Making Build.xml

## Setting Environmental values For MSVC

```
les#hello#build.xml - Eclipse SDK
efactor Tomcat Run Window Help
DBCon.java Encrypter.java build.xml *cpp-build.xml hello.c Basic.h »13
<project name="hello" default="build_wrapper" basedir=".">
  <target name="build_wrapper">
    <property environment="env"/>
    <exec executable="cmd" failonerror="true">
      <arg value="/C"/>
      <arg value="${env.VS90COMNTOOLS}/vsvars32.bat"/>
      <arg value="&cmd;&cmd;"/>
      <arg value="ant.bat"/>
      <arg value="-f"/>
      <arg value="cpp-build.xml" />
    </exec>
  </target>
</project>
```

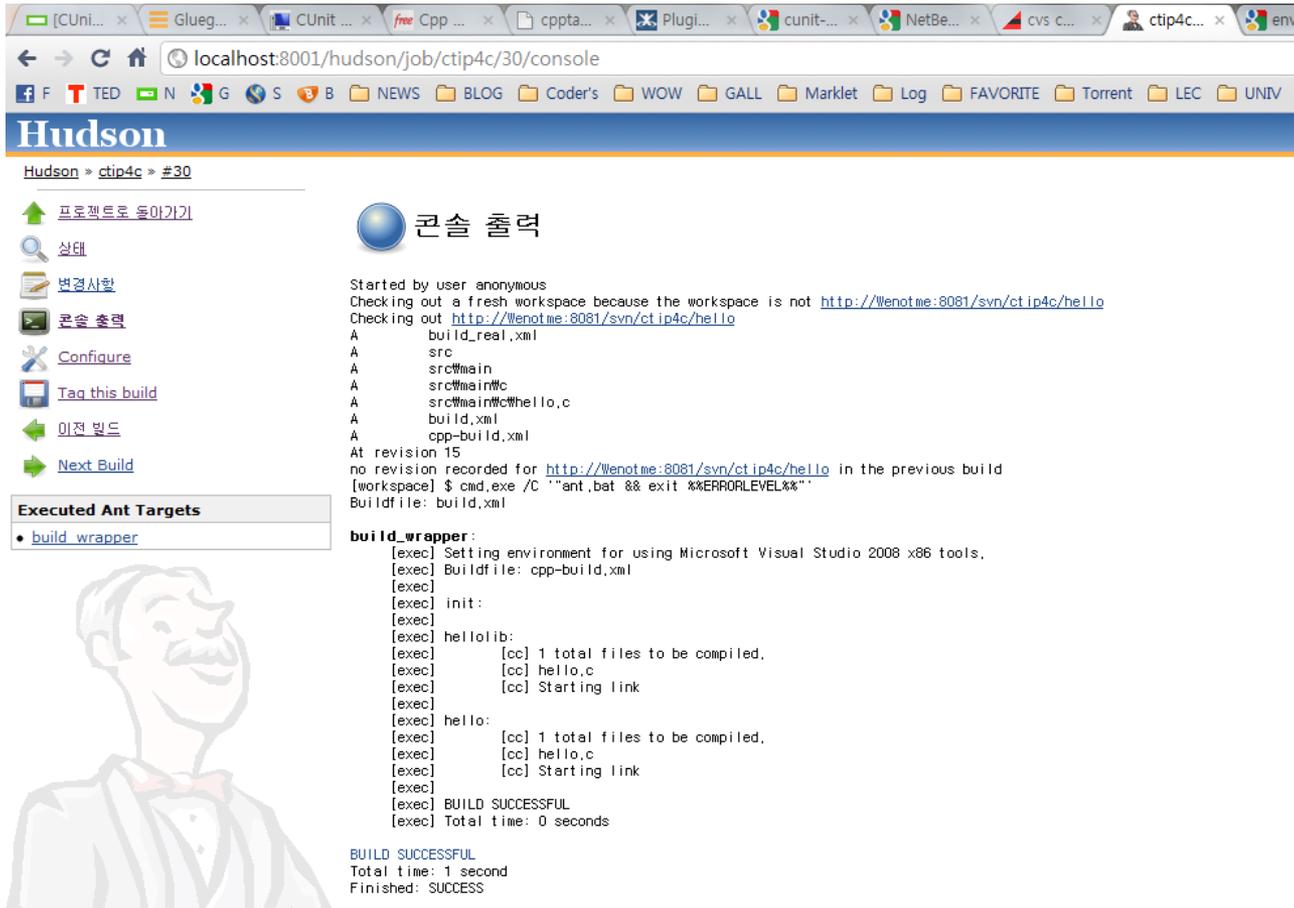
# Making Build.xml

## Build.xml for multi-platform Default C Compiler : MSVC



# Hudson Build Result Example

## Build Result



The screenshot shows a web browser window displaying the Hudson build result page for job 'ctip4c' at revision #30. The browser's address bar shows 'localhost:8001/hudson/job/ctip4c/30/console'. The page title is 'Hudson' and the breadcrumb is 'Hudson > ctip4c > #30'. On the left side, there are navigation links: '프로젝트로 돌아가기', '상태', '변경사항', '콘솔 출력', 'Configure', 'Tag this build', '이전 빌드', and 'Next Build'. Below these is a section titled 'Executed Ant Targets' with a sub-section 'build\_wrapper'. The main content area is titled '콘솔 출력' and shows the console output of the build process. The output indicates that the build was successful, with a total time of 1 second. A faint illustration of a man with a mustache and a bow tie is visible in the bottom-left corner of the screenshot.

localhost:8001/hudson/job/ctip4c/30/console

### Hudson

Hudson > ctip4c > #30

프로젝트로 돌아가기  
상태  
변경사항  
콘솔 출력  
Configure  
Tag this build  
이전 빌드  
Next Build

#### Executed Ant Targets

- build\_wrapper

#### 콘솔 출력

```
Started by user anonymous
Checking out a fresh workspace because the workspace is not http://Wenotme:8081/svn/ctip4c/hello
Checking out http://Wenotme:8081/svn/ctip4c/hello
A build_real.xml
A src
A src#main
A src#main#c
A src#main#c#hello.c
A build.xml
A cpp-build.xml
At revision 15
no revision recorded for http://Wenotme:8081/svn/ctip4c/hello in the previous build
[workspace] $ cmd.exe /C "ant.bat && exit %ERRORLEVEL%"
Buildfile: build.xml

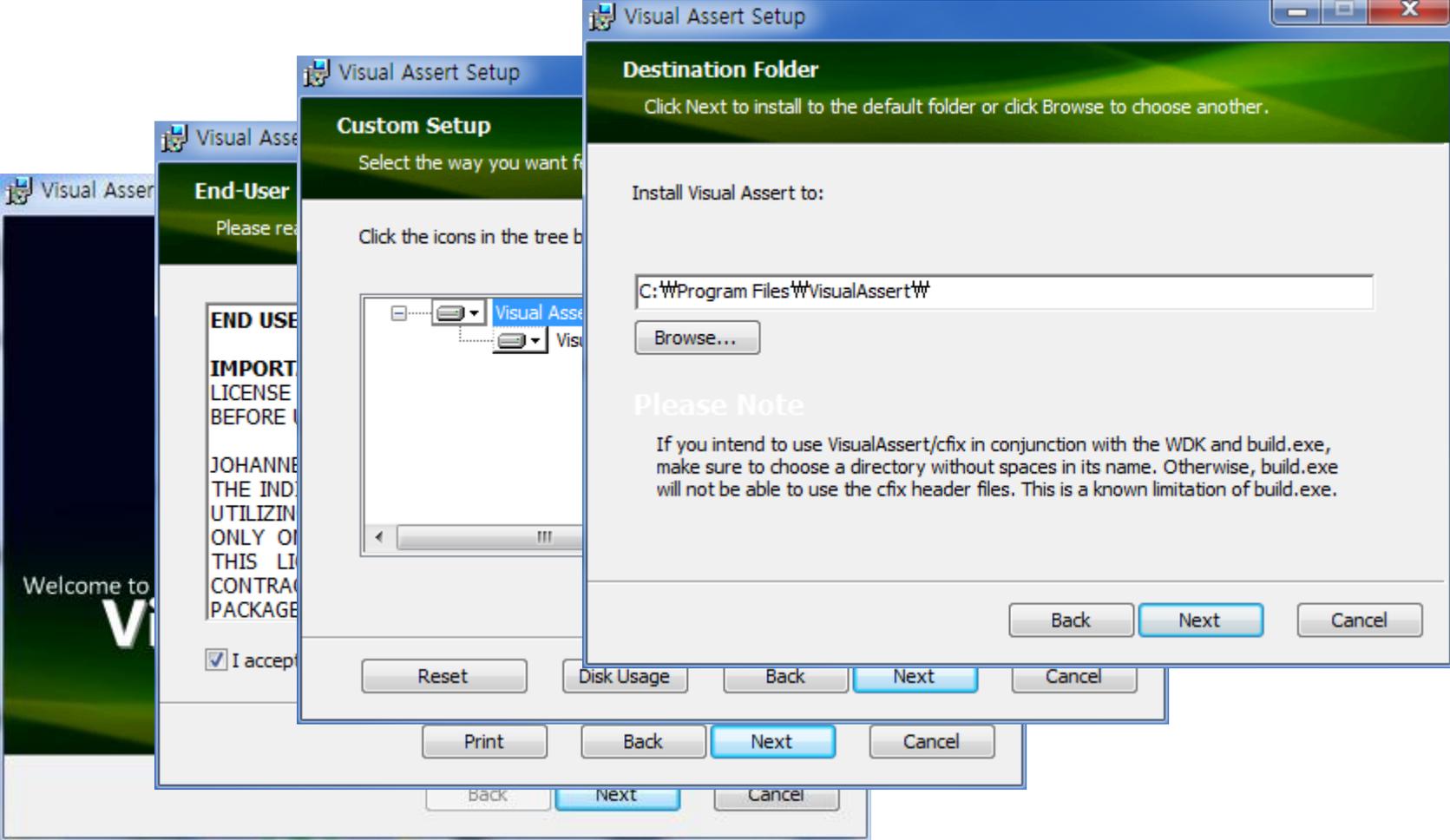
build_wrapper:
[exec] Setting environment for using Microsoft Visual Studio 2008 x86 tools.
[exec] Buildfile: cpp-build.xml
[exec]
[exec] init:
[exec]
[exec] hellolib:
[exec] [cc] 1 total files to be compiled.
[exec] [cc] hello.c
[exec] [cc] Starting link
[exec]
[exec] hello:
[exec] [cc] 1 total files to be compiled.
[exec] [cc] hello.c
[exec] [cc] Starting link
[exec]
[exec] BUILD SUCCESSFUL
[exec] Total time: 0 seconds

BUILD SUCCESSFUL
Total time: 1 second
Finished: SUCCESS
```



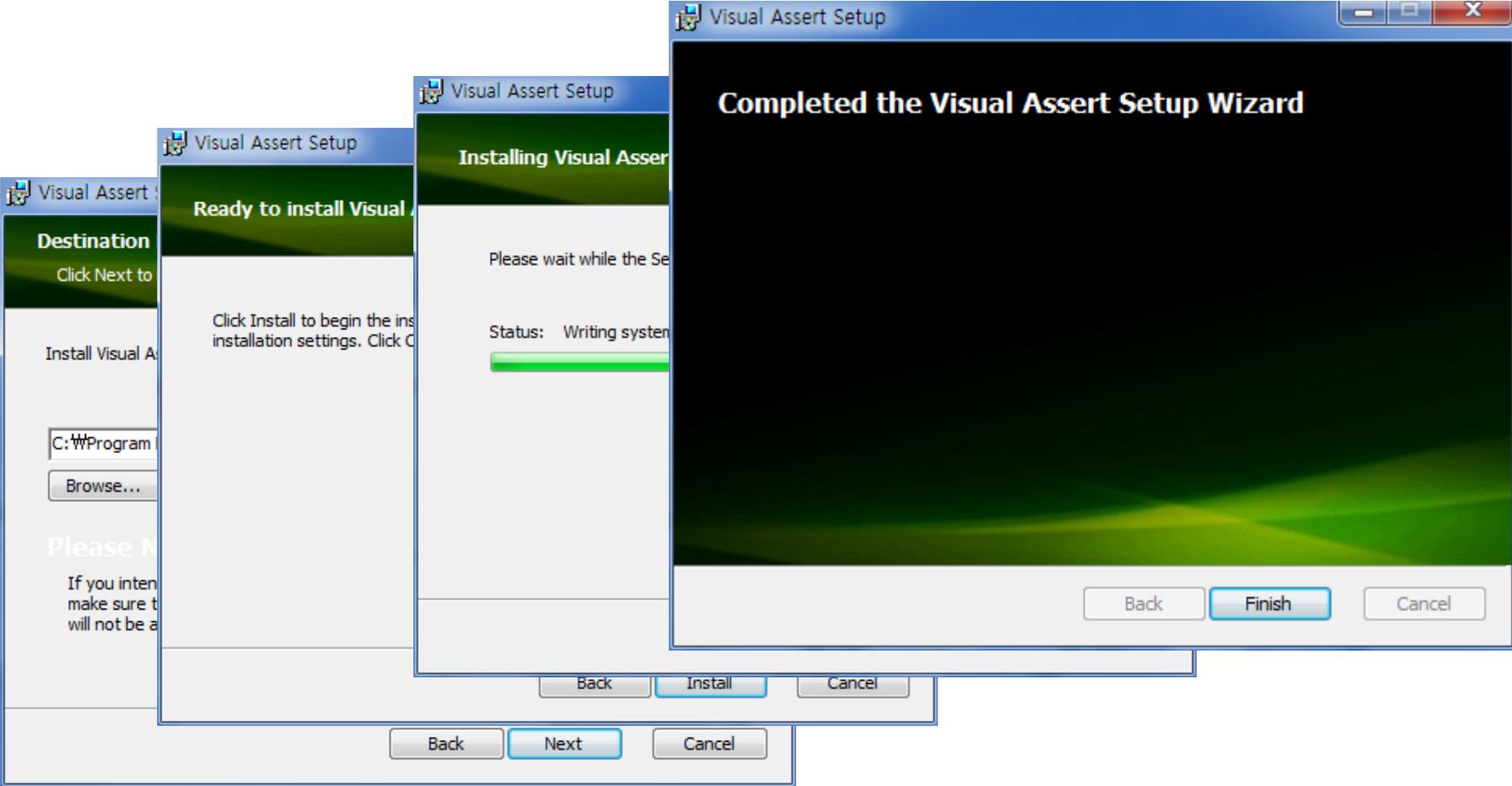
# Install Tools

## VisualAssert Install



# Install Tools

## VisualAssert Install



# Install Tools

## TestCocoon Download



The screenshot shows the TestCocoon website with the following content:

### TestCocoon

Open Source

Code Coverage Measurement for C/C++ and C#

- Overview
- Documentation
  - Code Coverage
  - CoverageBrowser
  - CoverageScanner
  - FAQ
  - Appendix
- Download
- User Forum
- License

**TestCocoon - Code Coverage Tool for C/C++ and C#**

TestCocoon is a complete code coverage tool chain for C/C++ and C# programs available under Apple® Mac OS X, Linux® or Microsoft® Windows. It analyzes the performance of a software validation and permits to measure the performance and optimizes the testing process of a C/C++ or C# applications:

- Finding untested code sections.
- Reducing the amount of tests by finding redundant tests.
- With TestCocoon it is possible to find which portion of the source code is covered only by one execution of a test, and to detect if a new test does not cover more source code line than existing tests.
- Finding dead code trough displaying the code parts which are never executed.
- Specially useful for manual testing:
  - TestCocoon is able to calculate the optimal execution order of tests which maximize the overall coverage after each run.
  - Also, TestCocoon is able to perform its analysis on a difference of two applications.

This permits to find which tests are impacted by a source code modification and permits to measure the test quality of a patch or a hot fix.

TestCocoon can be used for every testing steps and methodologies (e.g., unit tests, integration tests, etc...), and permits to collect and merge the execution reports to a single report.

It is composed of 3 tools:

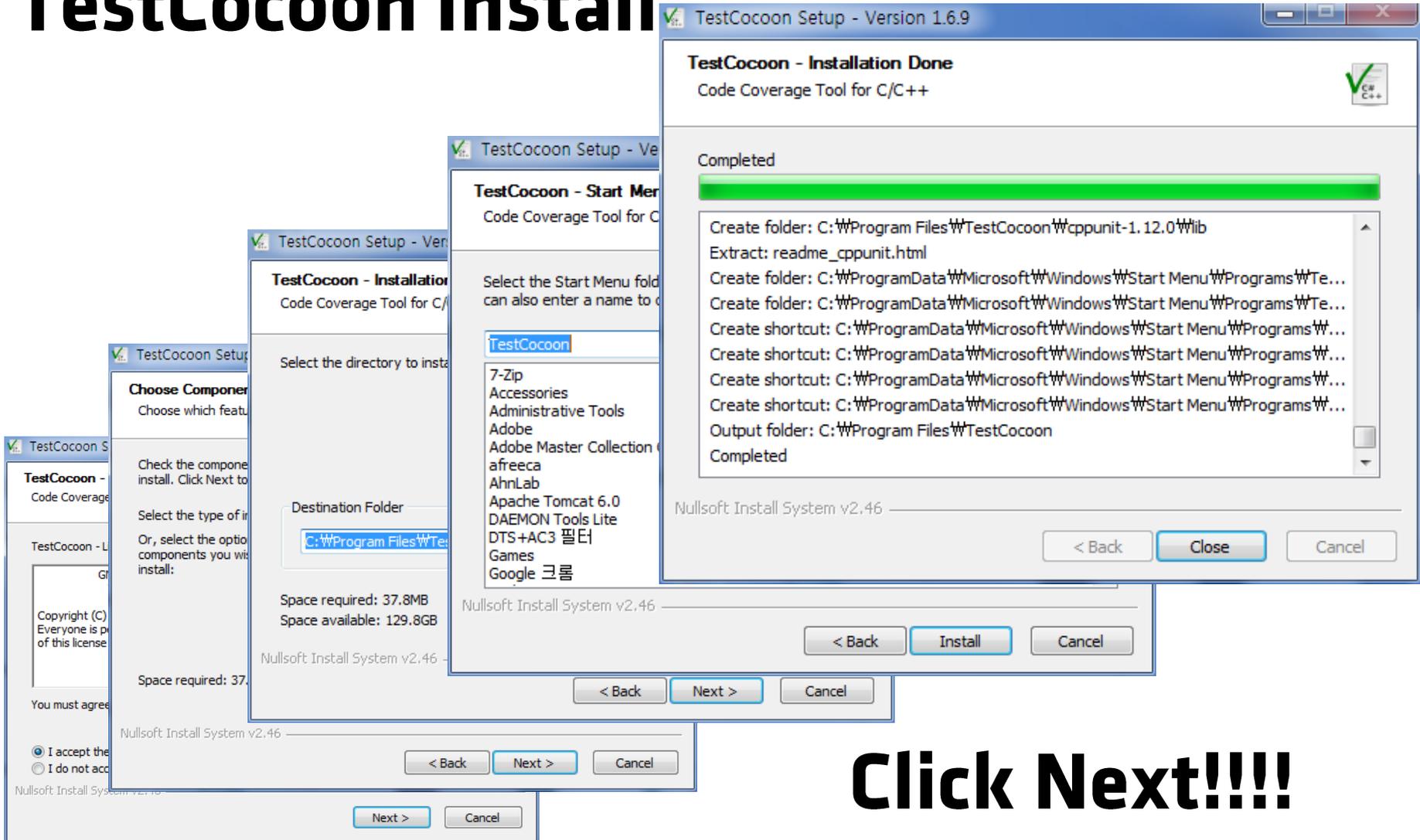
- CoverageScanner, which analyzes, instruments and generates the test results.
- CoverageBrowser, which displays and manages the results.

<http://www.testcocoon.org>

접속 & 다운로드

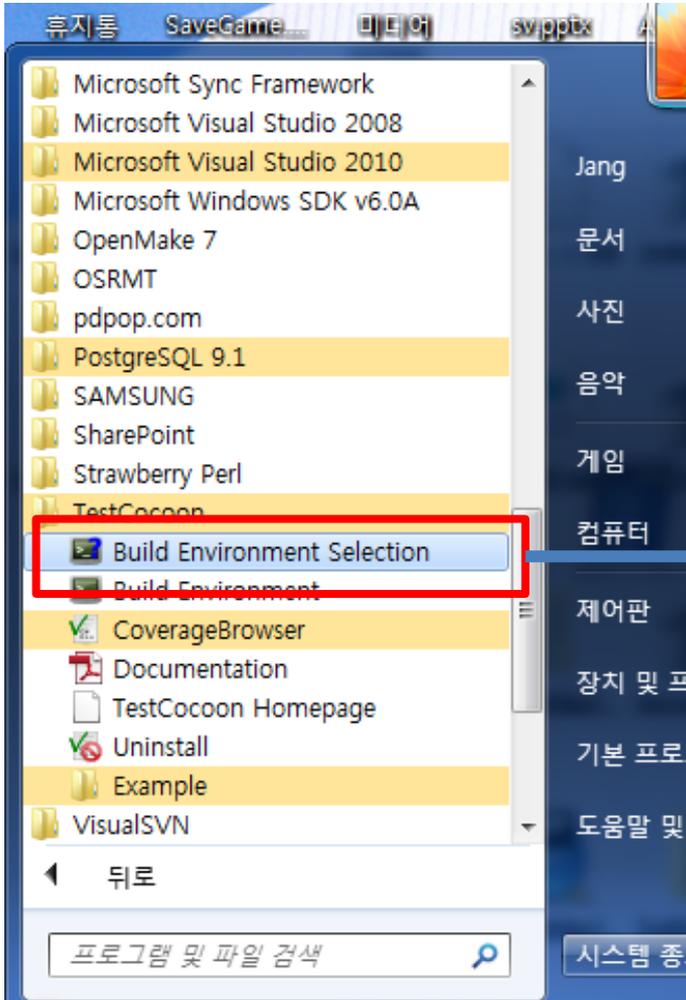
# Install Tools

## TestCocoon Install



# Install Tools

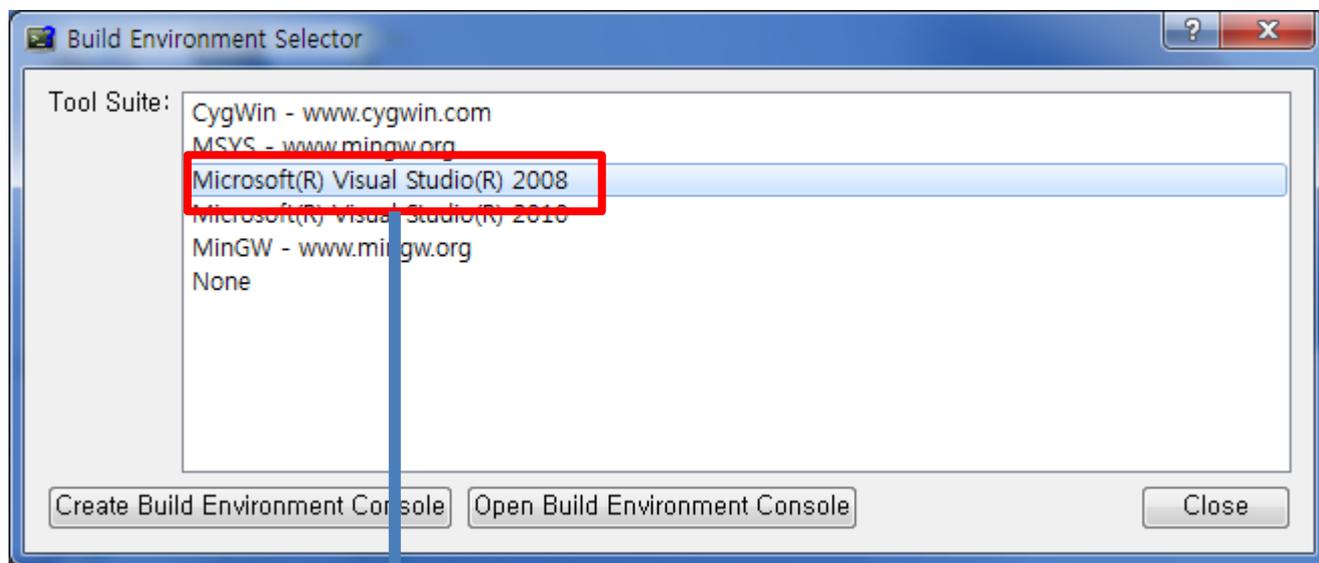
## TestCocoon Env. Configuration



**Build Environment Selection 클릭**

# Install Tools

## TestCocoon Env. Configuration



VS 2008 선택

# Install Tools

## TestCocoon Example

The screenshot displays the CoverageBrowser application interface. The main window shows a list of methods and their coverage percentages. The 'main' method is highlighted in green, indicating 100.00% coverage. The 'Sources' panel shows 'tutorial.cpp' with 100.00% coverage. The 'Executions' panel shows a table of execution results:

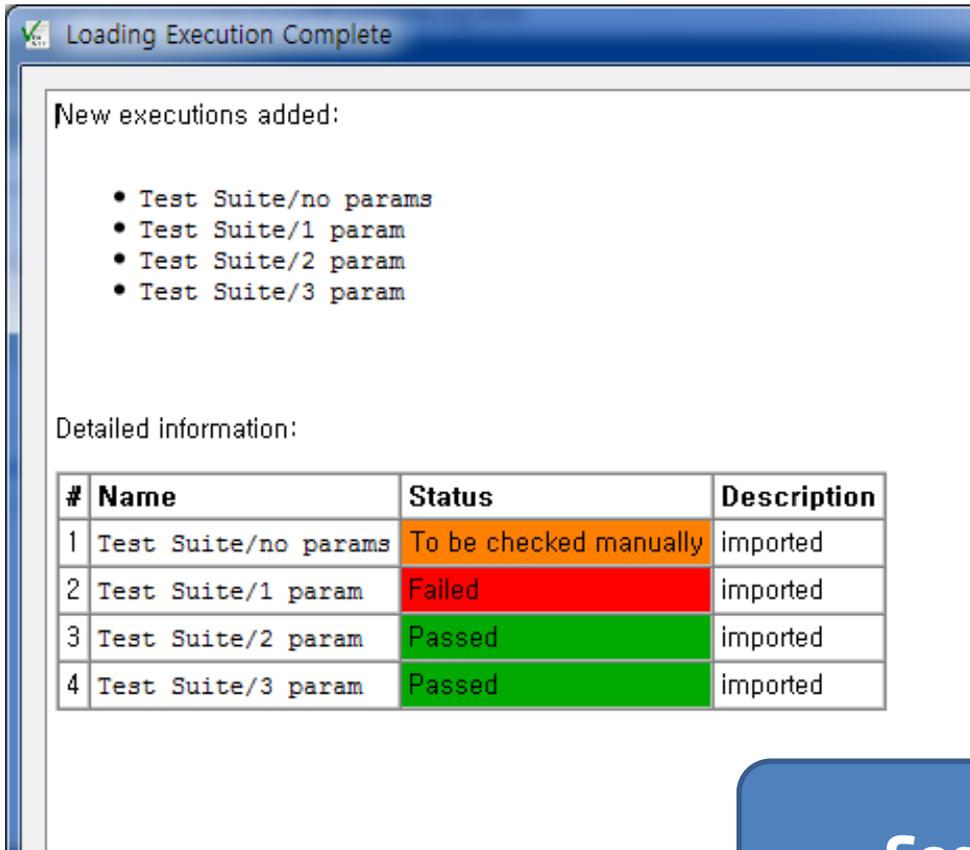
Executions	Coverage	State	Comment
no ...	25.000%	To be checke...	no
3 p ...	83.333%	Passed	no
...	...	Passed	no

The 'Statistic' panel shows a coverage of 100.000% (12/12). The 'Execution Comment' panel is empty.

주어진 Example로 Test Case에 대한  
Code Coverage 확인

# Install Tools

## TestCocoon Example



The screenshot shows a window titled "Loading Execution Complete". It contains a list of "New executions added:" and a table of "Detailed information:". The table has four columns: "#", "Name", "Status", and "Description". The rows show the results of four test suite executions, with one failed and three passed.

New executions added:

- Test Suite/no params
- Test Suite/1 param
- Test Suite/2 param
- Test Suite/3 param

Detailed information:

#	Name	Status	Description
1	Test Suite/no params	To be checked manually	imported
2	Test Suite/1 param	Failed	imported
3	Test Suite/2 param	Passed	imported
4	Test Suite/3 param	Passed	imported

**Code Coverage Report**

# Install Tools

## DevPartner Install

Setup - Micro Focus DevPartner Studio 10.5.786.0 Professional Edit...

**Installing**  
Please wait while Setup installs Micro Focus DevPartner Studio Professional Edition on your computer.

Extracting files...  
C:\Windows\WinSxS\Local\Temp\...

**DevPartner Studio**

- Install DevPartner Studio 10.5 Professional Edition**  
Installs static code analysis, error detection, performance, coverage and memory analysis and performance expert features for use with Visual Studio 2010, Visual Studio 2008, and Visual Studio 2005.  
[Read the release notes](#) | [Read the installation guide](#)
- Install DevPartner System Comparison**  
Installs a utility to compare the state of a computer to a past state, or to other computers.  
[Read the release notes](#)
- Install License Manager 4.4 (Optional)**  
Distributed License Management (DLM) software components validate that an appropriate license is available and facilitate license administration.  
[Read the installation guide](#)

**MICRO FOCUS**  
Leading the Evolution™

[Visit Micro Focus Support](#)  
[Download Adobe Reader](#)  
[Explore this Media](#)

**Select Install DevPartner Studio 10.5**

# Install Tools

## DevPartner Install

Micro Focus DevPartner Studio Professional Edition Installation

**Component Selection**

Select the program features you want installed.

Click on an icon in the list

- Program
- Manual
- Sample
- Visual S

Help

Micro Focus DevPartner Studio Professional Edition Installation

**Program Maintenance**

Modify, repair, or remove the program

- Modify**  
Change the program's settings. Custom Settings are installed.
- Repair**  
Repair the program if it is corrupt or missing files.
- Remove**  
Remove the program from the computer.

< Back

Micro Focus DevPartner Studio Professional Edition Installation

**Ready to Modify the Program**

The wizard is ready to begin installation.

Click Install to begin the installation.

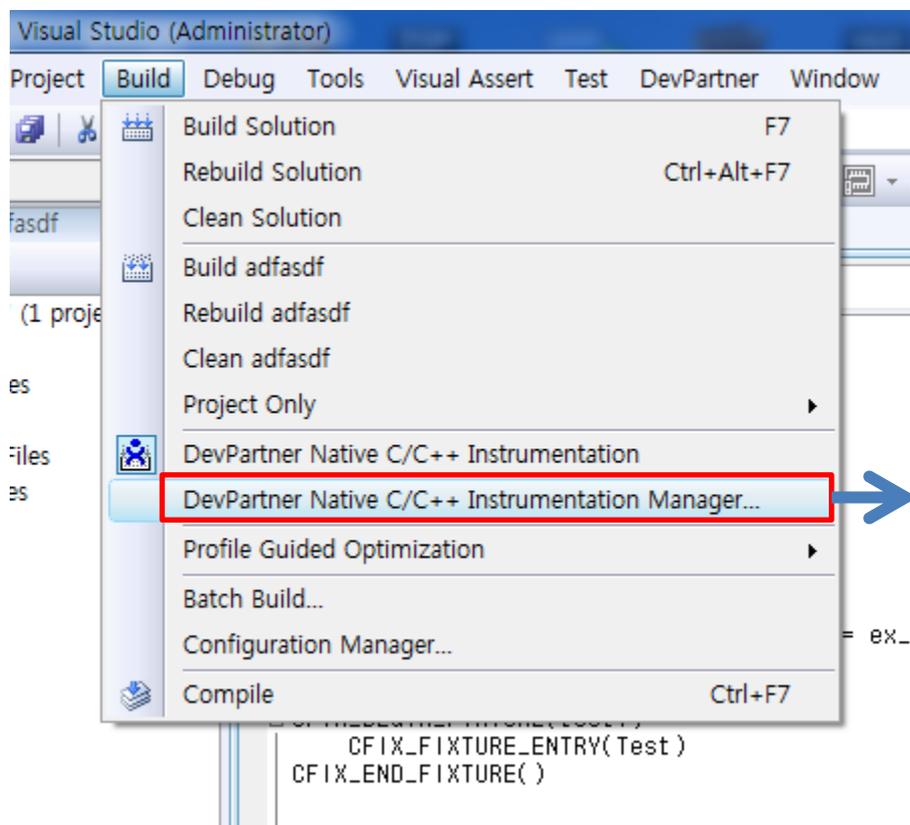
If you want to review or change any of your installation settings, click Back. Click Cancel to exit the wizard.

< Back   **Install**   Cancel

Micro Focus  
Leading the way

# Install Tools

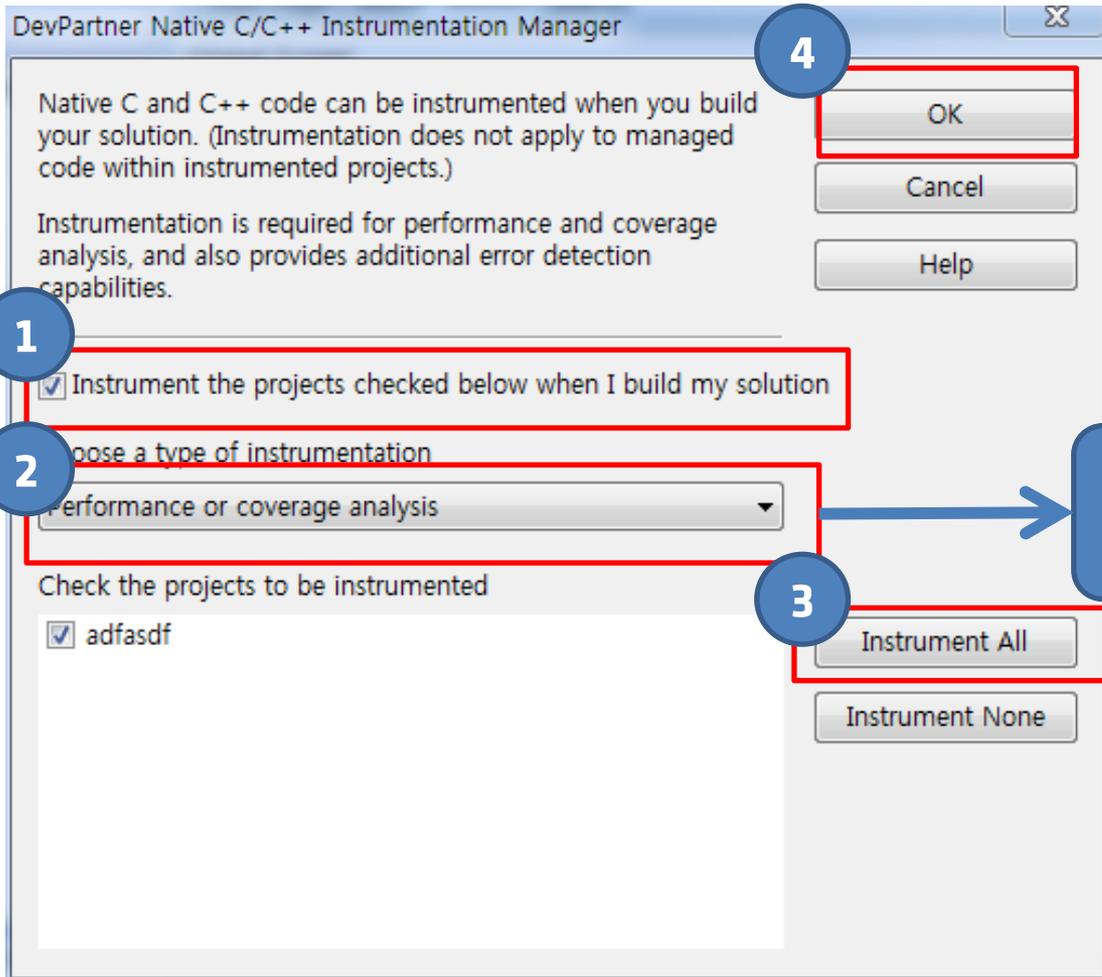
## DevPartner Menu



**Instrumentation Manger에서  
Project Coverage Report File  
Setting**

# Install Tools

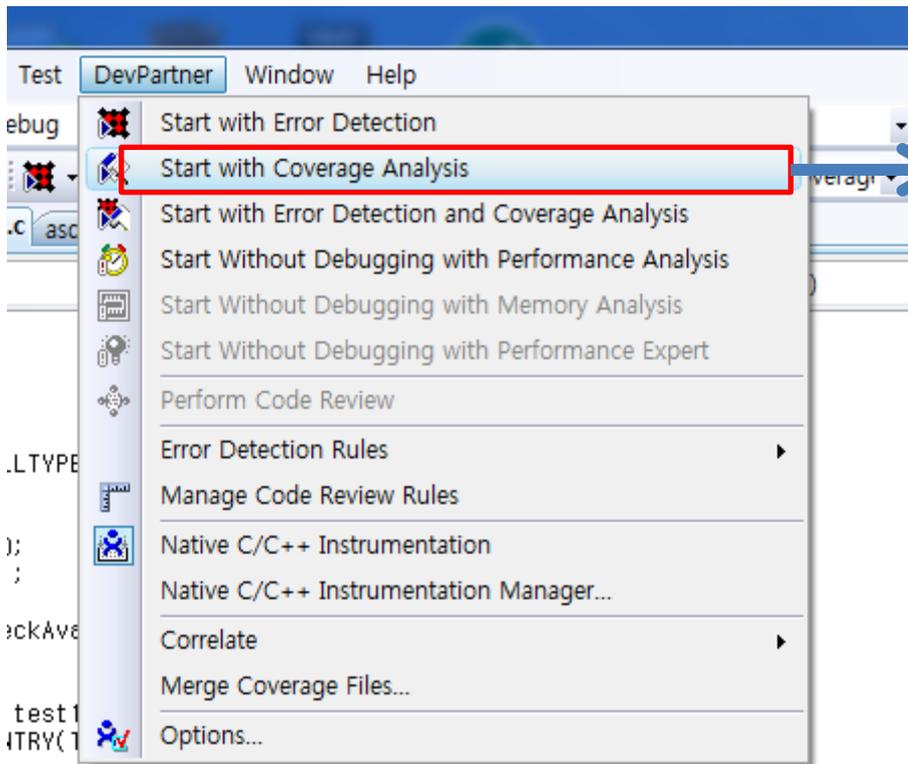
## DevPartner Menu



Performance or coverage  
Analysis 선택

# Install Tools

## DevPartner Menu



**Code Coverage 측정**

# Install Tools

## DevPartner Coverage Report

The screenshot shows the Microsoft Visual Studio interface with the DevPartner Coverage Report open. The report is titled 'adfasdf1.dpcov\*' and shows the following summary:

- All ( 82.1% of 134 lines )
- 110 of 134 lines executed (82,1%)
- 6 of 7 methods called (85,7%)

The 'Method List' section is highlighted with a red box and contains the following table:

Method Name	% Covered	Called	# Lines Not Execut...
checkOrder	0.0	0	4
checkAvailable	64.3	17	5
selectionToPrice	80.0	17	1
main	81.1	1	14
makeChange	100.0	1	0
printAvailable	100.0	17	0
printMenu	100.0	17	0

The 'Output' window at the bottom shows the following text:

```
'adfasdf.exe': Loaded 'C:\Windows\System32\imm32.dll'  
'adfasdf.exe': Loaded 'C:\Windows\System32\misctf.dll'  
'adfasdf.exe': Loaded 'C:\Program Files\Microsoft Focus\DevPartner Studio\Analysis\TTCore.dll'  
Runtime in coverage mode  
'adfasdf.exe': Loaded 'C:\Windows\System32\shell32.dll'  
'adfasdf.exe': Loaded 'C:\Windows\System32\shlwapi.dll'  
'adfasdf.exe': Unloaded 'C:\Windows\System32\shell32.dll'  
'adfasdf.exe': Unloaded 'C:\Windows\System32\shlwapi.dll'  
The thread 'Win32 Thread' (0x12e8) has exited with code 0 (0x0).  
The program '[564] adfasdf.exe: Native' has exited with code 0 (0x0).
```

# Install Tools

## AnkhSVN Download

ankhsvn: Subversion Support for Visual Studio - Windows Internet Explorer  
http://ankhsvn.open.collab.net/

COLLABNET®  
collabxchange | Partners | Customers | News & Events | Global Sites | Store | Contact

WHY COLLABNET? | PRODUCTS | CLOUD SERVICES | DOWNLOADS | SUPPORT | TRAINING | SERVICES

Login | Join Now  
COLLABNET COMMUNITY

**About AnkhSVN**  
News  
Features  
Screenshots  
Documentation  
FAQ  
History  
License

**Getting AnkhSVN**  
Download  
Source Code  
AnkhSVN Extensions

**Community**  
Community Support  
Mailing Lists  
Issue Tracker  
Getting Involved

**Related Projects**  
CollabNet Desktop - Visual Studio Edition

**AnkhSVN - Subversion Support for Visual Studio**  
The 100% OpenSource Source Code Management Solution for Microsoft Visual Studio.

**AnkhSVN** is a Subversion Source Control Provider for Microsoft Visual Studio 2005, 2008 and 2010.

**AnkhSVN** provides Apache Subversion source code management support to all project types supported by Visual Studio and allows you to perform the most common version control operations directly from inside the Microsoft Visual Studio IDE.

The **Pending Changes dashboard** gives you a unique insight in your development process and provides easy access to the source code and issue management features. The deep source code control (SCC) integration allows you to focus on developing, while AnkhSVN keeps track of all your changes and provides you the tools to effectively handle your specific needs.

» AnkhSVN Features  
» Screenshots  
**Download AnkhSVN 2.1 Now!**

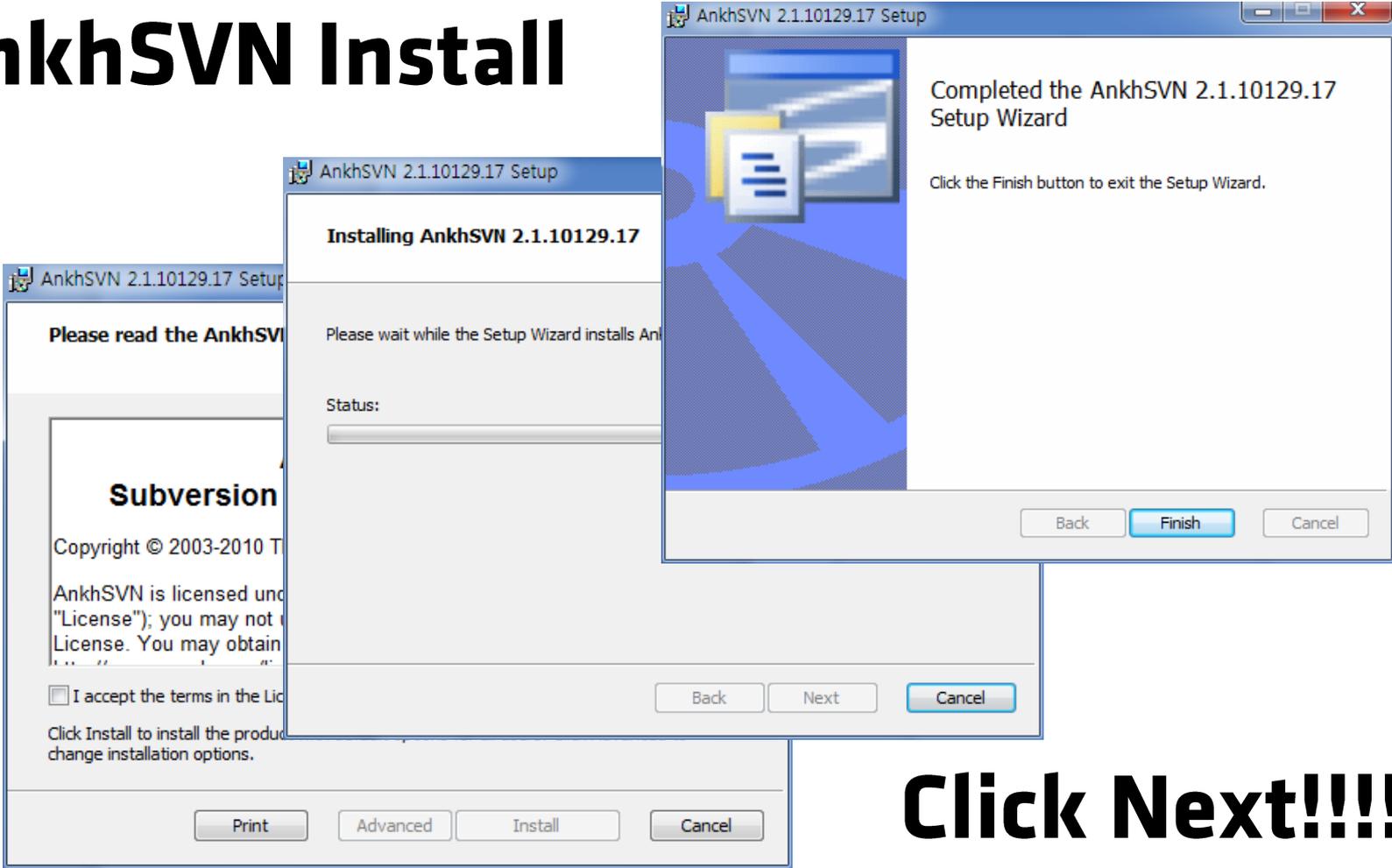
**News**  
**AnkhSVN 2.1.10129 released** (2011년 3월 11일 금요일)

feedback

<http://ankhsvn.open.collab.net>  
접속 & 다운로드

# Install Tools

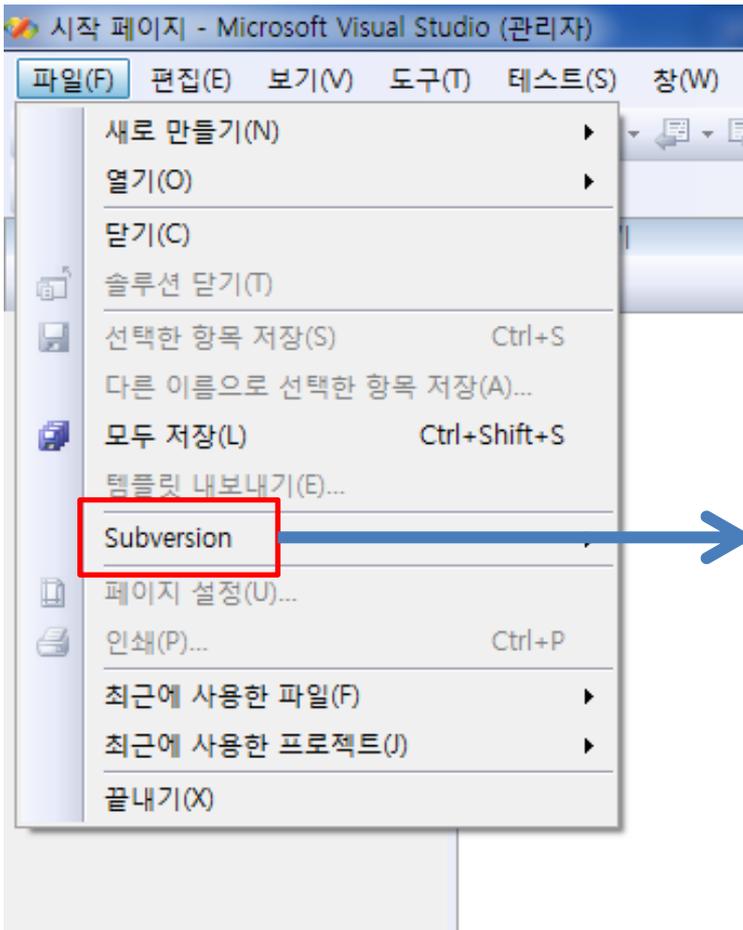
## AnkhSVN Install



**Click Next!!!!**

# Install Tools

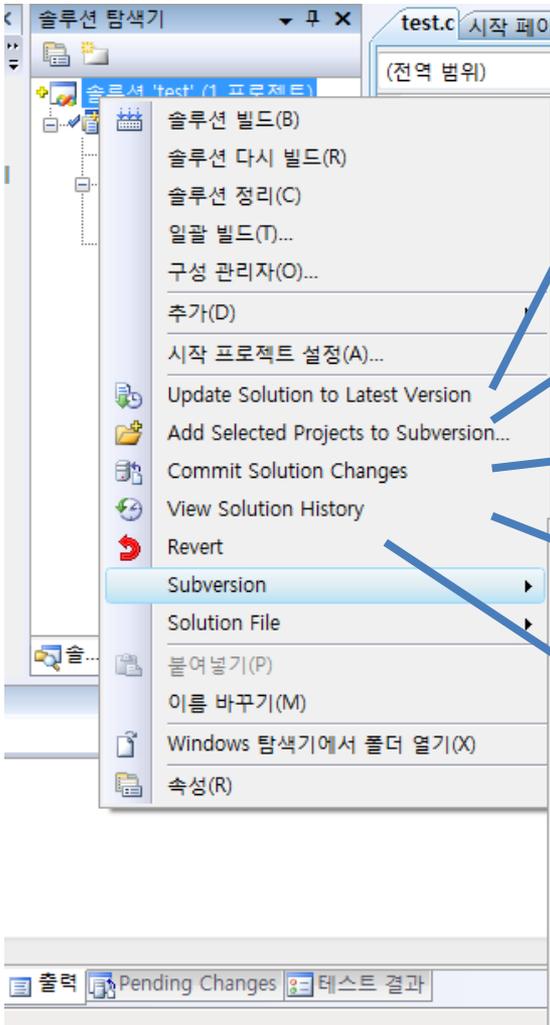
## AnkhSVN Install



**File -> Subversion 메뉴 추가 확인**

# Install Tools

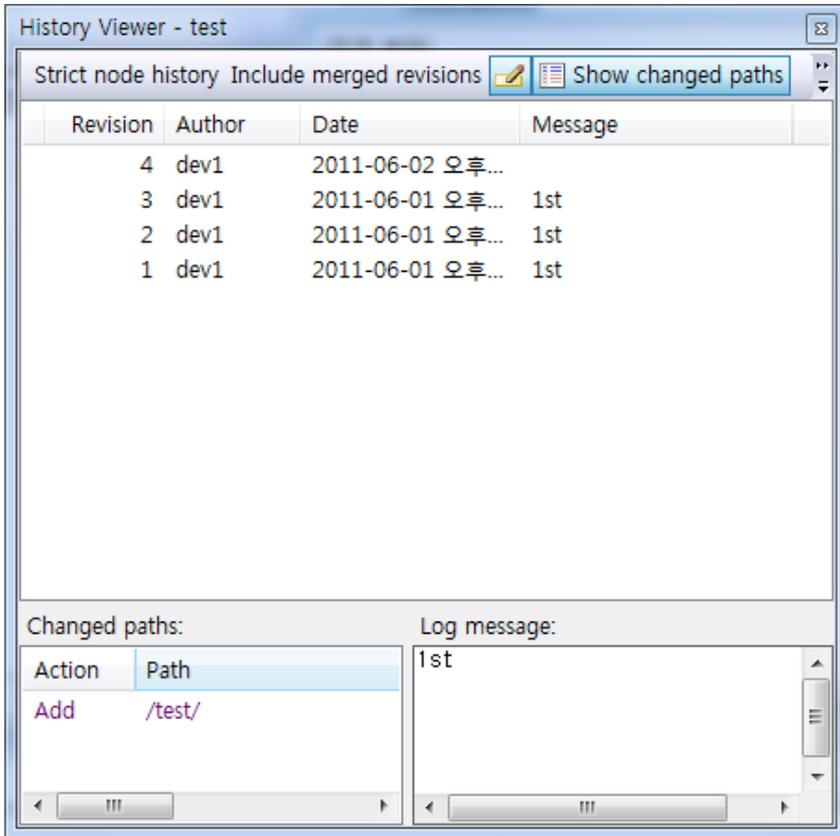
## AnkhSVN Example



- SVNRepository 의 Latest Update 가져오기
- SVNRepository에 Project 등록
- 바뀐 부분만 Commit
- History V
- 이전 버전으로 되돌리기

# Install Tools

## AnkhSVN History View / Repository



History Viewer - test

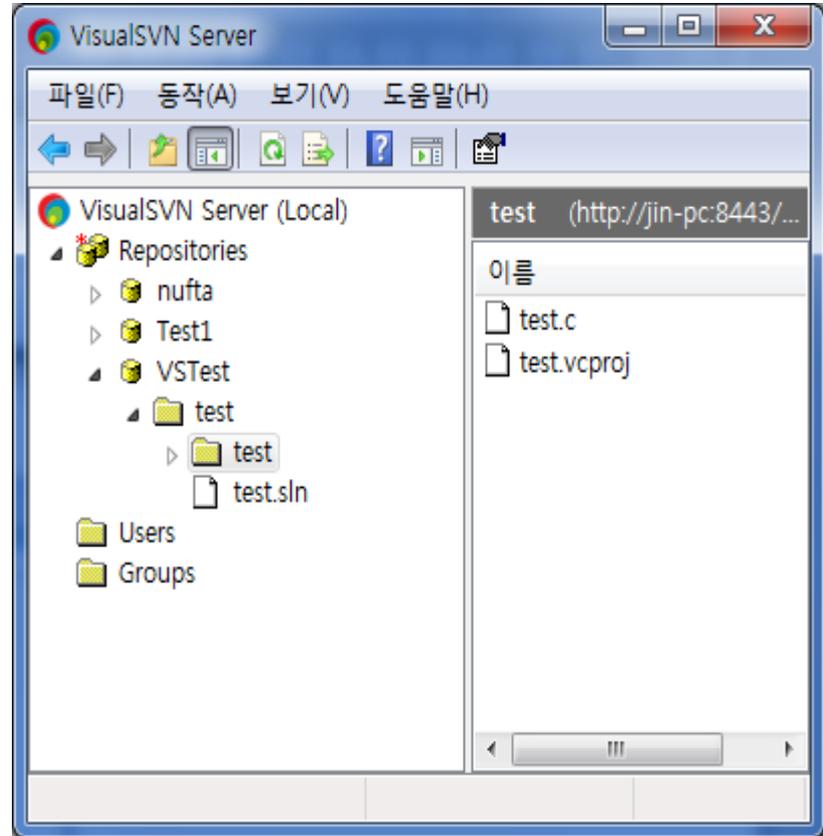
Strict node history Include merged revisions Show changed paths

Revision	Author	Date	Message
4	dev1	2011-06-02 오후...	
3	dev1	2011-06-01 오후...	1st
2	dev1	2011-06-01 오후...	1st
1	dev1	2011-06-01 오후...	1st

Changed paths:

Action	Path
Add	/test/

Log message:  
1st



VisualSVN Server

파일(F) 동작(A) 보기(V) 도움말(H)

VisualSVN Server (Local)

- Repositories
  - nufta
  - Test1
  - VSTest
    - test
      - test
      - test.sln
- Users
- Groups

test (http://jin-pc:8443/...)

이름

- test.c
- test.vcproj

# Testing C Source

Bending Machine

System Testing

Introduction

Specification

Pairwise Testing

Test Result

# Introduction

---

## Bending machine Source

- 일정 액수를 입력하면 자판기에서 음료수를 뽑을 수 있는 프로그램
- 입력 값은 초기 금액 투입과 음료수 선택
- 결과 값은 각 음료당 구입 개수와 거스름 돈, 잔고 보유량
- 음료 주문 전에 주문 가능한 음료를 표시

# Specification

---

- 초기 금액을 입력 받는다.
- 콜라:500원/포카리:600원/커피:700원
- 환타:800원/사이다:900원/미닛메이드:1000원
- 거스름돈은 5000원 1000원 500원 100원 단위로 지급한다.

# Pairwise Testing

---

## Representative Values

- 초기 입력 금액에 대한 설정
  - 500원 미만/500원/600원/700원/800원/900원/1000원/1000원 이상
- 음료 선택에 대한 값
  - 1~6까지의 값/ 그 이외의 값
- 거스름 돈
  - 5000원/ 4000원/ 3500원/400원 (동전 출력 확인용)

# Pairwise Testing

## AllPairs Testcase Generation

case	초기입력	음료선택	거스름돈	pairings
1	500원미만	유효한 값	5000원	3
2	500원미만	유효하지않은값	4000원	3
3	500원	유효하지않은값	5000원	3
4	500원	유효한 값	4000원	3
5	600원	유효한 값	3500원	3
6	600원	유효하지않은값	400원	3
7	700원	유효하지않은값	3500원	3
8	700원	유효한 값	400원	3
9	800원	유효한 값	5000원	2
10	800원	유효하지않은값	4000원	2
11	900원	유효하지않은값	5000원	2

# Pairwise Testing

## AllPairs Testcase Generation

case	초기입력	음료선택	거스름돈	pairings
12	900원	유효한 값	4000원	2
13	1000원	유효한 값	3500원	2
14	1000원	유효하지않은값	400원	2
15	1000원이상	유효하지않은값	3500원	2
16	1000원이상	유효한 값	400원	2
17	500원미만	~유효한 값	3500원	1
18	500원미만	~유효하지않은값	400원	1
19	500원	~유효하지않은값	3500원	1
20	500원	~유효한 값	400원	1
21	600원	~유효한 값	5000원	1
22	600원	~유효하지않은값	4000원	1

# Pairwise Testing

## AllPairs Testcase Generation

case	초기입력	음료선택	거스름돈	pairings
23	700원	~유효하지않은값	5000원	1
24	700원	~유효한 값	4000원	1
25	800원	~유효한 값	3500원	1
26	800원	~유효하지않은값	400원	1
27	900원	~유효하지않은값	3500원	1
28	900원	~유효한 값	400원	1
29	1000원	~유효한 값	5000원	1
30	1000원	~유효하지않은값	4000원	1
31	1000원이상	~유효하지않은값	5000원	1
32	1000원이상	~유효한 값	4000원	1

# Pairwise Testing

## VisualAssert API

### CFIX\_ASSERT(조건식)

- 조건식이 참이면 Pass

### CFIX\_BEGIN\_FIXTURE(Suite name)

- Test Suite 의 시작을 표시

### CFIX\_END\_FIXTURE()

- Test Suite의 끝을 표시

### CFIX\_FIXTURE\_ENTRY(Test Name)

- 수행할 Test의 Entry

# Pairwise Testing

---

## Test Case Code Example

```
static void CFIXCALLTYPE Test21()  
{  
    int input = 600;  
    int selection = 4;  
    int change = 5000;  
  
    int ex_res = 1;  
    int ex_res_selection = 800;  
    int change_array[4] = {1,0,0,0};
```

# Pairwise Testing

## Test Case Code Example

```
int change_res[4] = {0,0,0,0};  
int i;
```

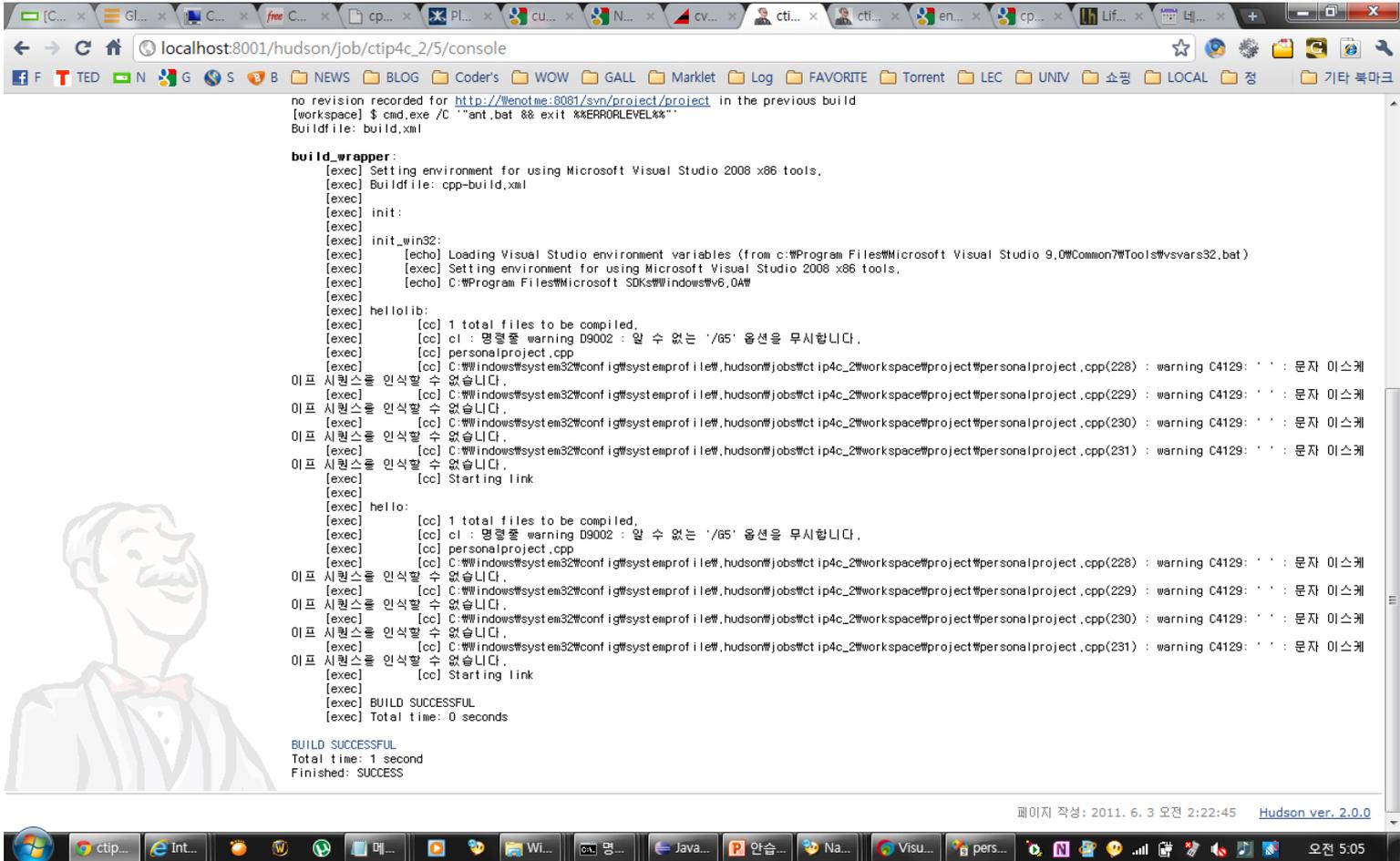
```
CFIX_ASSERT(checkAvailable(input) == ex_res);  
CFIX_ASSERT(selectionToPrice(selection) == ex_res_selection);  
  
makeChange(change_res, change);  
  
for(i = 0 ; i <4; i++)  
    CFIX_ASSERT(change_array[i] == change_res[i]);
```

```
}
```

Assert 문을 이용한 함수 Test

# Test Result

## Build by Hudson



```
no revision recorded for http://jenkins:8081/svn/project/project in the previous build
[workspace] $ cmd.exe /C "ant.bat && exit %%ERRORLEVEL%%"
Buildfile: build.xml

build_wrapper:
[exec] Setting environment for using Microsoft Visual Studio 2008 x86 tools.
[exec] Buildfile: cpp-build.xml
[exec] init:
[exec] init_win32:
[exec] [echo] Loading Visual Studio environment variables (from c:\Program Files\Microsoft Visual Studio 9.0\Common7\Tools\vsvars32.bat)
[exec] [echo] Setting environment for using Microsoft Visual Studio 2008 x86 tools.
[exec] [echo] C:\Program Files\Microsoft SDKs\Windows\v6.0A\
[exec] hellolib:
[exec] [cc] 1 total files to be compiled.
[exec] [cc] cl : 명령줄 warning D9002 : 알 수 없는 '/65' 옵션을 무시합니다.
[exec] [cc] personalproject.cpp
[exec] [cc] C:\Windows\system32\config\systemprofile\jenkins\jobs\ctip4c_2\workspace\project\personalproject.cpp(228) : warning C4129: ' ' : 문자 이스케이프 시퀀스를 인식할 수 없습니다.
[exec] [cc] C:\Windows\system32\config\systemprofile\jenkins\jobs\ctip4c_2\workspace\project\personalproject.cpp(229) : warning C4129: ' ' : 문자 이스케이프 시퀀스를 인식할 수 없습니다.
[exec] [cc] C:\Windows\system32\config\systemprofile\jenkins\jobs\ctip4c_2\workspace\project\personalproject.cpp(230) : warning C4129: ' ' : 문자 이스케이프 시퀀스를 인식할 수 없습니다.
[exec] [cc] C:\Windows\system32\config\systemprofile\jenkins\jobs\ctip4c_2\workspace\project\personalproject.cpp(231) : warning C4129: ' ' : 문자 이스케이프 시퀀스를 인식할 수 없습니다.
[exec] [cc] Starting link
[exec] hello:
[exec] [cc] 1 total files to be compiled.
[exec] [cc] cl : 명령줄 warning D9002 : 알 수 없는 '/65' 옵션을 무시합니다.
[exec] [cc] personalproject.cpp
[exec] [cc] C:\Windows\system32\config\systemprofile\jenkins\jobs\ctip4c_2\workspace\project\personalproject.cpp(228) : warning C4129: ' ' : 문자 이스케이프 시퀀스를 인식할 수 없습니다.
[exec] [cc] C:\Windows\system32\config\systemprofile\jenkins\jobs\ctip4c_2\workspace\project\personalproject.cpp(229) : warning C4129: ' ' : 문자 이스케이프 시퀀스를 인식할 수 없습니다.
[exec] [cc] C:\Windows\system32\config\systemprofile\jenkins\jobs\ctip4c_2\workspace\project\personalproject.cpp(230) : warning C4129: ' ' : 문자 이스케이프 시퀀스를 인식할 수 없습니다.
[exec] [cc] C:\Windows\system32\config\systemprofile\jenkins\jobs\ctip4c_2\workspace\project\personalproject.cpp(231) : warning C4129: ' ' : 문자 이스케이프 시퀀스를 인식할 수 없습니다.
[exec] [cc] Starting link
[exec] BUILD SUCCESSFUL
[exec] Total time: 0 seconds

BUILD SUCCESSFUL
Total time: 1 second
Finished: SUCCESS
```

페이지 작성: 2011. 6. 3 오전 2:22:45 Hudson ver. 2.0.0

# Test Result

## Build by Hudson

Back to Dashboard

Status

Changes

Workspace

Build Now

Delete Project

Configure

### Project ctip4c\_2

Workspace

Recent Changes

Disable Project

#### Build History (trend)

#	Date	Time	Status
#5	2011. 6. 3 오전	2:22:39	Success
#4	2011. 6. 3 오전	2:21:20	Failure
#3	2011. 6. 3 오전	2:21:18	Failure
#2	2011. 6. 3 오전	2:19:58	Failure
#1	2011. 6. 3 오전	2:15:19	Failure

for all for failures

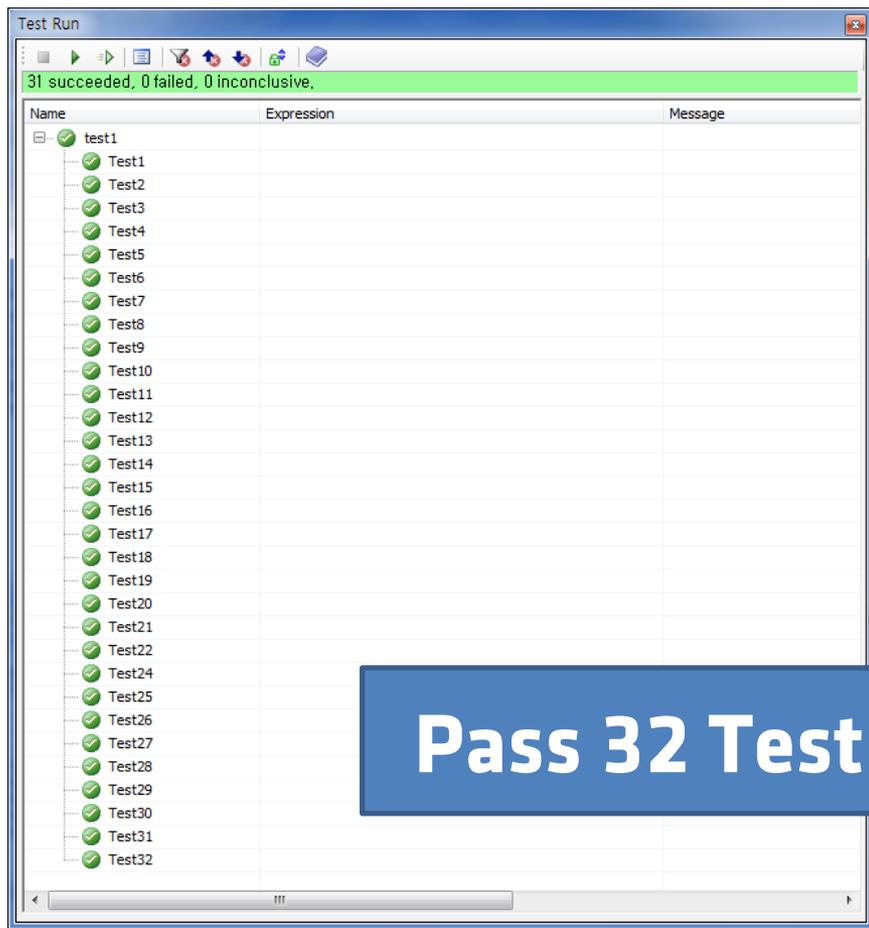
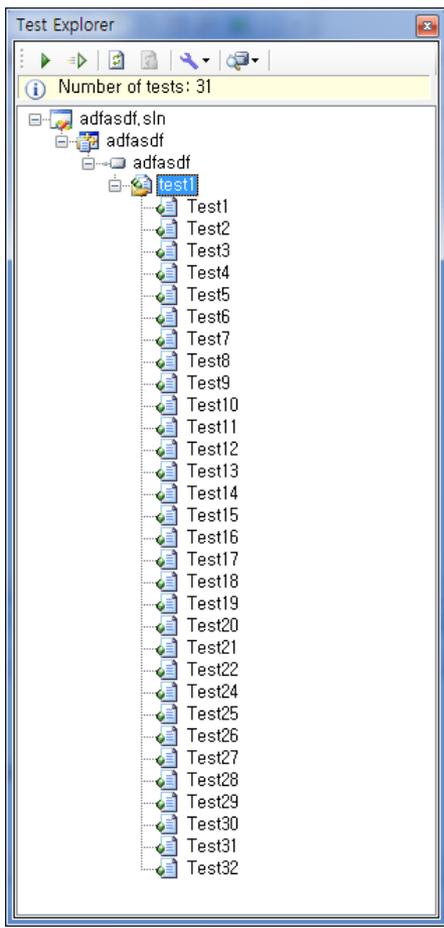
#### Permalinks

- Last build (#5), 2 hr 43 min ago
- Last stable build (#5), 2 hr 43 min ago
- Last successful build (#5), 2 hr 43 min ago
- Last failed build (#4), 2 hr 44 min ago
- Last unsuccessful build (#4), 2 hr 44 min ago

페이지 작성: 2011. 6. 3 오전 5:05:54 Hudson ver. 2.0.0

# Test Result

## Test Case Execution



**Pass 32 Test Cases**

# Test Result

## DevPartner Coverage Report

The screenshot shows the DevPartner Coverage Report interface. On the left, a tree view displays the project structure: 'All (88.1% of 134 lines)', 'JANG-PC - 0 (adfasdf)', and 'Source(88.1% of 134 lines)'. Below this are categories for 'Methods Not Covered', 'Methods Less Than 20% Covere', 'Over 30 Lines, Less Than 10% C', and 'Removed Methods'. On the right, two progress bars indicate overall coverage: '118 of 134 lines executed (88,1%)' and '6 of 7 methods called (85,7%)'. Below these is a table with columns for 'Method Name', '% Covered', 'Called', and '# Lines Not Execut...'. The table lists methods: 'checkOrder' (0.0% covered, 0 called, 4 lines), 'main' (83.8% covered, 34 called, 12 lines), 'printAvailable' (100.0% covered, 42 called, 0 lines), 'printMenu' (100.0% covered, 42 called, 0 lines), 'checkAvailable' (100.0% covered, 42 called, 0 lines), 'selectionToPrice' (100.0% covered, 42 called, 0 lines), and 'makeChange' (100.0% covered, 34 called, 0 lines).

Method Name	% Covered	Called	# Lines Not Execut...
checkOrder	0.0	0	4
main	83.8	34	12
printAvailable	100.0	42	0
printMenu	100.0	42	0
checkAvailable	100.0	42	0
selectionToPrice	100.0	42	0
makeChange	100.0	34	0

**Code Coverage : 88.1 %**  
**Method Called 85.7 %**

# References

---

- **Advanced Software Testing(Ane Mette /Jonassen Hass)**
- 소프트웨어 테스트링 마이크로소프트에선 이렇게 한다.(앨런 페이지/켄존스톤/비제이롤리슨) - 에이콘
- 자바의 또 다른 멋진 도구 Ant(에릭 해쳐/스티브라우란) - 인포북
- [http://www.code1.co.kr/bbs/board.php?bo\\_table=devpartner&wr\\_id=39](http://www.code1.co.kr/bbs/board.php?bo_table=devpartner&wr_id=39)
- <http://open-tube.com/10-code-coverage-tools-c-c>

# References

---

- <http://ankhsvn.open.collab.net/>
- <http://www.bullseye.com/productInfo.html>
- <http://www.visualassert.com/unit-testing-framework/>
- <http://cunit.sourceforge.net/>
- [http://en.wikipedia.org/wiki/List\\_of\\_unit\\_testing\\_frameworks](http://en.wikipedia.org/wiki/List_of_unit_testing_frameworks)
- <http://www.testcocoon.org/>
- <http://www.testwell.fi/ctcdesc.html>