Java기반의 CTIP 환경 설정

소프트웨어 검증
컴퓨터공학부
김미셀, 김은정, 엄준용, 한경미

목차

목차

CI(Continuous Integration)

- 여러 명으로 구성된 팀이 작업한 것을 통합
- 대규모의 프로젝트 개발에 통합과정의 문제를 해결하기 위해 나온 해결책

CI(Continuous Integration) 사용

1. Source Repository로 부터 최신 소스 다운로드(Check out)
2. 코드를 작성한 후 자신의 개발기에서 정상적으로 동작하는지 충분히 검토
3. 검증이 끝난 후 개발자는 작업 내용을 Source Repository에 업로드(Check in)
4. 개발자(또는 통합 관리자 또는 자동화된 시스템)는 통합 서버에 방금 작업한 코드가 반영된 전체 코드 내용에 대한 빌드 수행
5. 만약 빌드가 실패할 경우, 실패 원인을 분석하고 문제를 해결하여 빌드를 성공시킴.

CTIP
(Continuous Test & Integration Platform)

- CI 개념을 Java기반의 개발 프로젝트에 쉽게 적용하기 위한 Platform
- 소스 검토(테스트 및 정적 분석), 빌드, 통합, 배포 및 레포트 기능을 제공하는 개발 지원 Platform

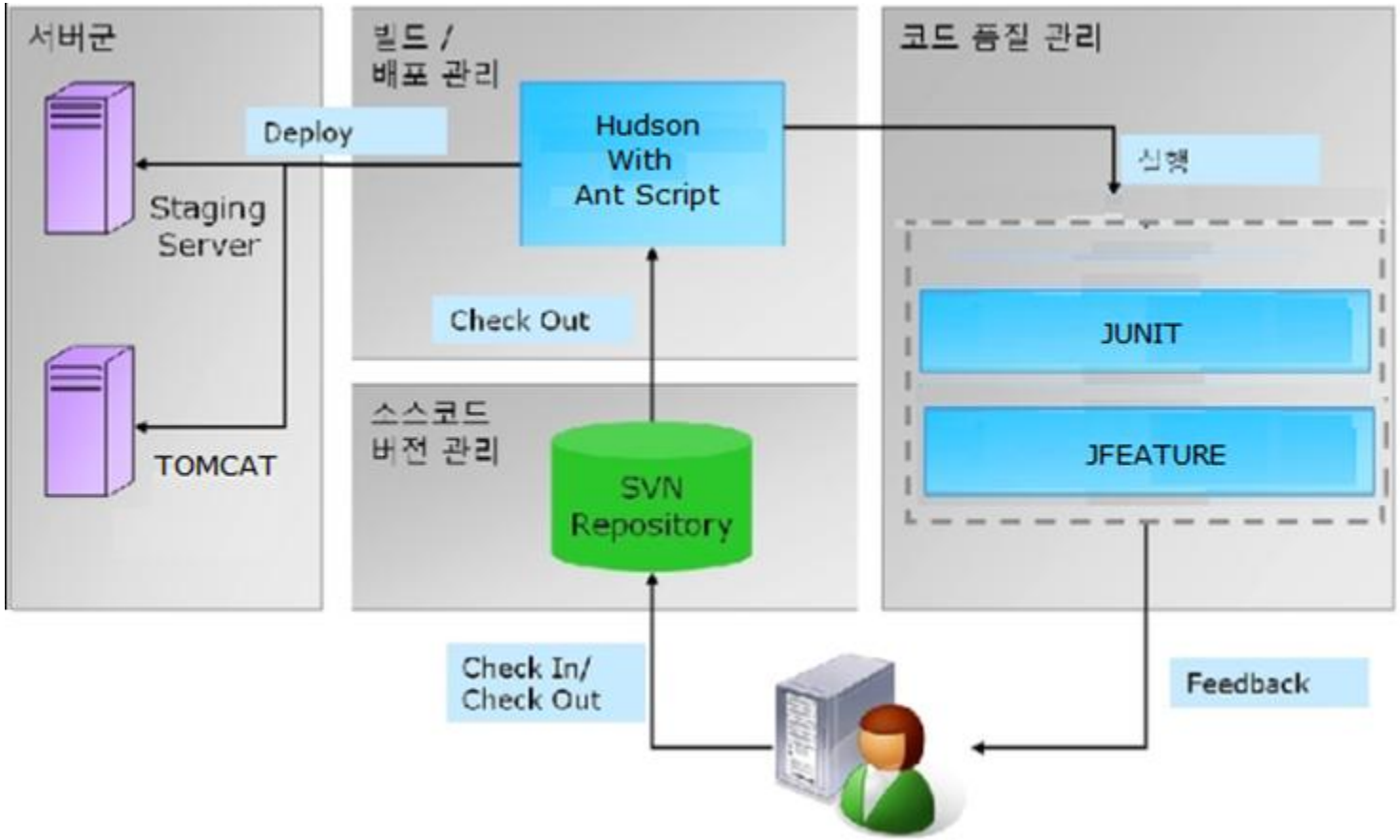
CTIP
(Continuous Test & Integration Platform)

- 단일 소스 저장소 관리
- 빌드 자동화
- 자체적으로 테스트 가능한 빌드
- 빠른 빌드 수행
- 운영환경과 유사한 환경 구성
- 최신결과물에 대한 쉬운 접근
- 손쉬운 빌드 상태 모니터링

CTIP 구성

IDE Tool Eclipse	CI Tool Hudson	Unit Test JUnit
WAS Tomcat 6.0	Build Tortoise SVN	SCM Ant

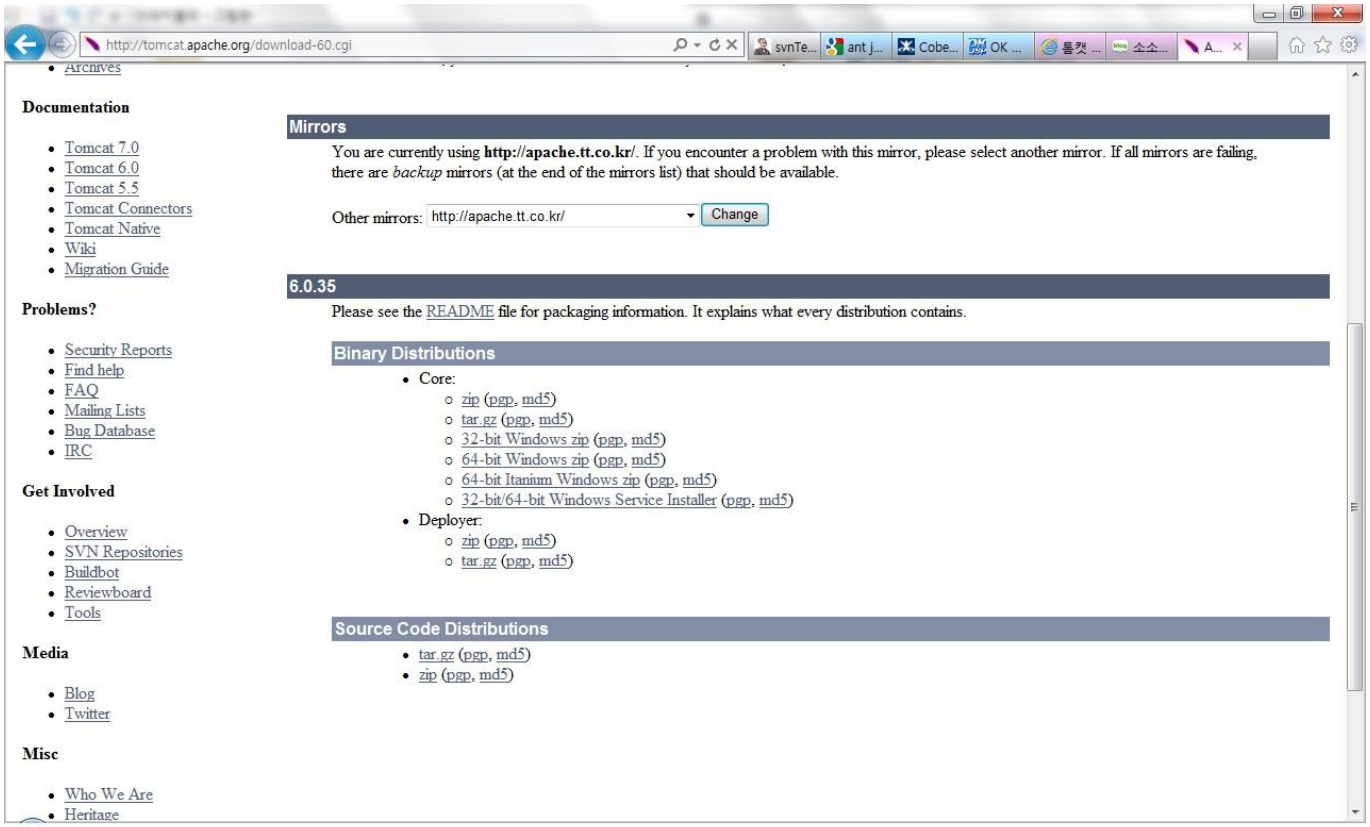
CTIP 구성


서버군
빌드 /
배포 관리
코드 품질 관리
Deploy
Hudson
With
Ant Script
실행
Staging
Server
Check Out
JUNIT
JFEATURE
TOMCAT
스스코드
버전 관리
SVN
Repository
Check In/
Check Out
Feedback

목차

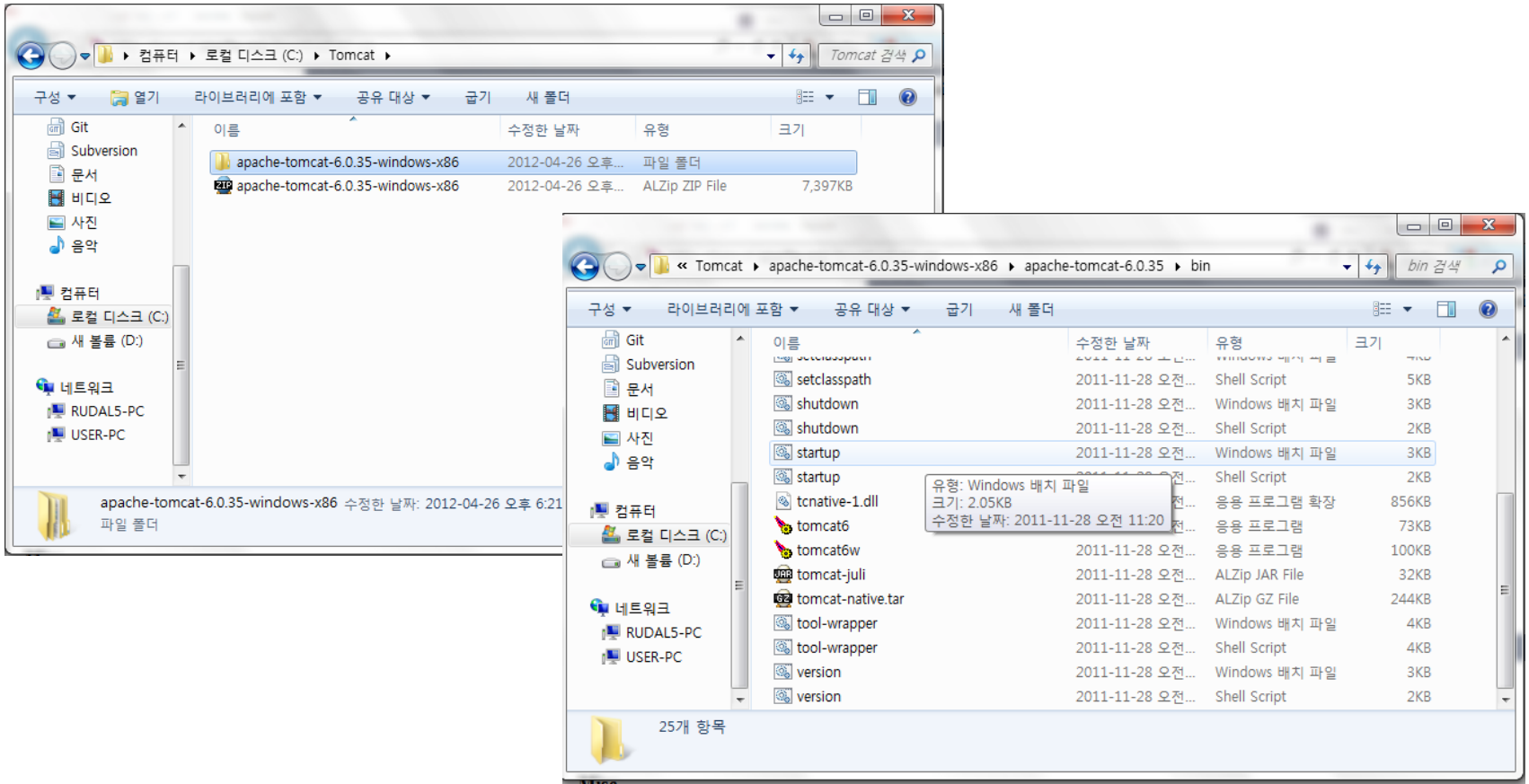
Tomcat 설치

- http://tomcat.apache.org/ 접속 하여 Tomcat 6.0 download



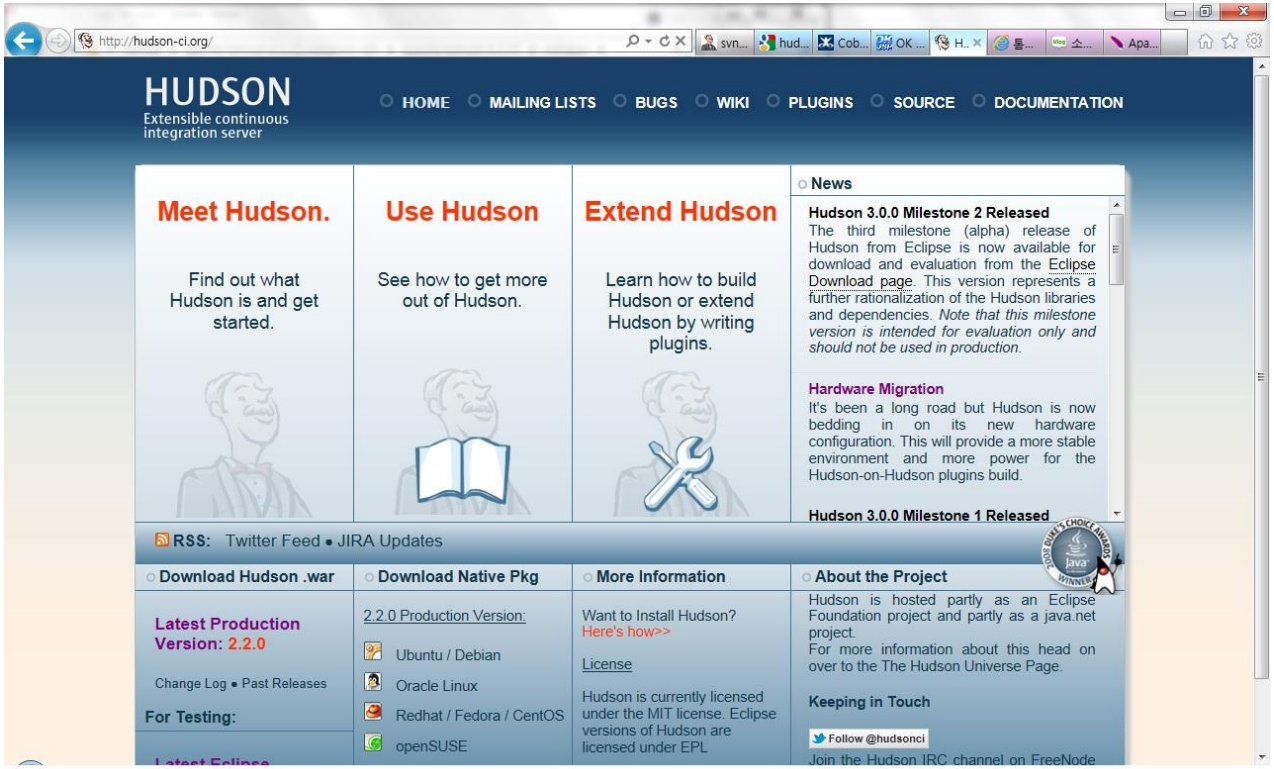
Tomcat 설치

- Tomcat 압축 파일 해제 후 startup 실행



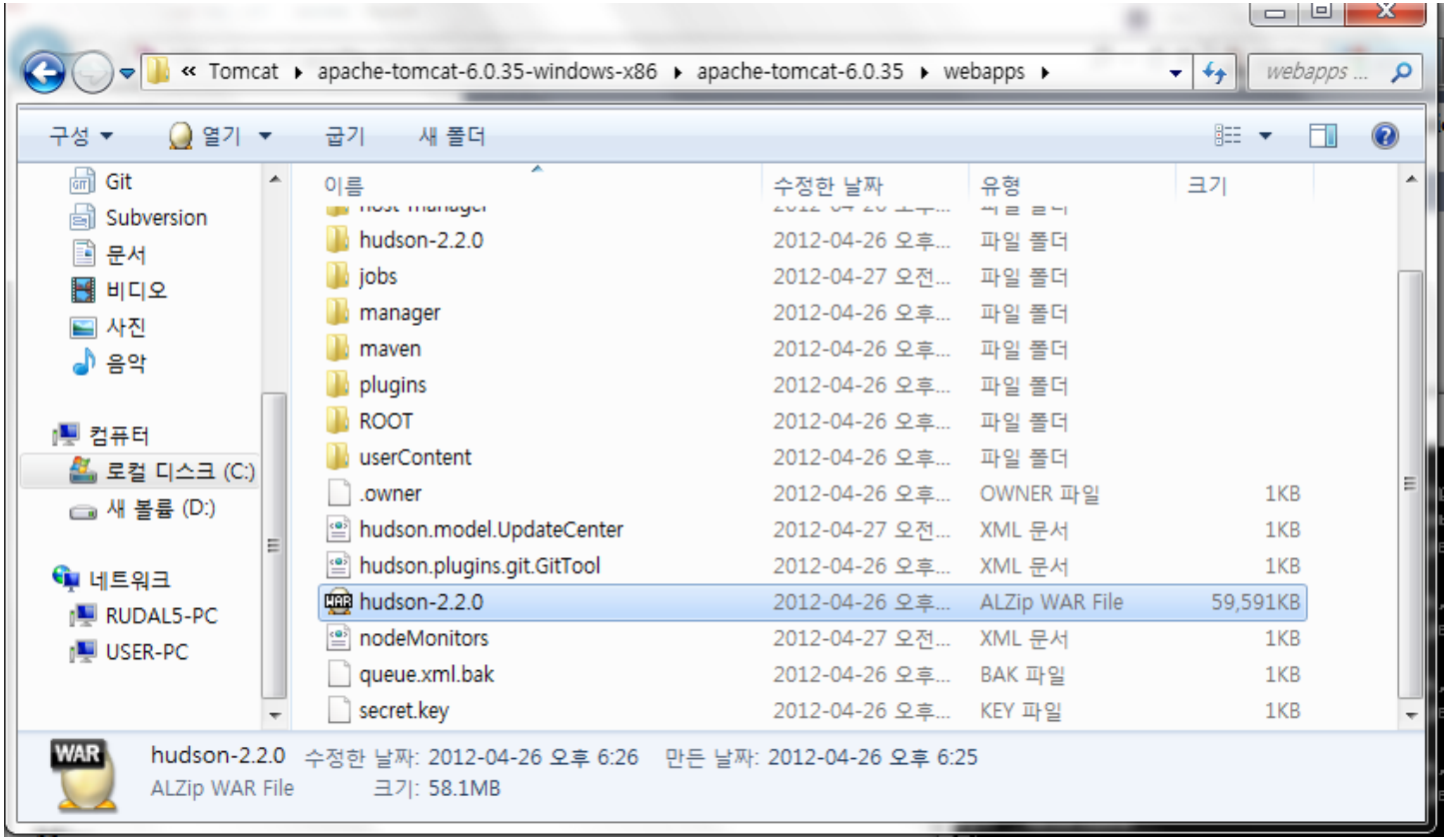
Hudson 설치

- http://hudson-ci.org/ 접속 하여 Hudson Download



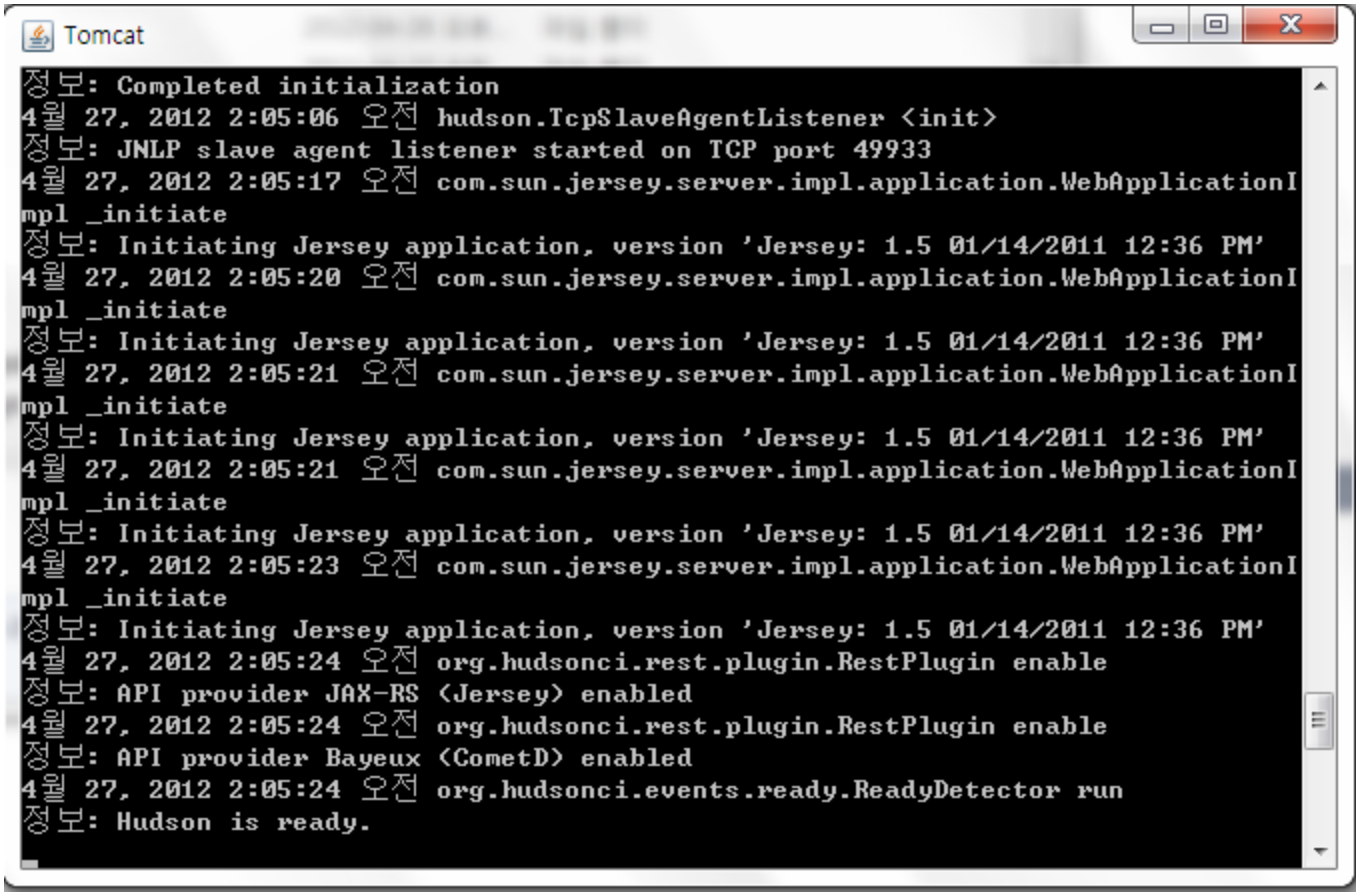
Hudson 설치

- hudson.war 파일을 tomcat 의 webapps 폴더에 저장



Hudson 설치

- Tomcat을 통한 Hudson 실행



Ant

- 자동 생성한 build.xml

*test.java | test_build.xml

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!-- WARNING: Eclipse auto-generated file.
              Any modifications will be overwritten.
              To include a user specific buildfile here, simply create one in the same
              directory with the processing instruction <?eclipse.ant.import?>
              as the first entry and export the buildfile again. -->
<project basedir="." default="build" name="test_89">
    <property environment="env"/>
    <property name="ECLIPSE_HOME" value="../../../../Program Files/eclipse"/>
    <property name="debuglevel" value="source,lines,vars"/>
    <property name="target" value="1.6"/>
    <property name="source" value="1.6"/>
    <path id="test_89.classpath">
        <pathelement location="bin"/>
    </path>
    <target name="init">
        <mkdir dir="bin"/>
        <copy includeemptydirs="false" todir="bin">
            <fileset dir="src">
                <exclude name="**/*.launch"/>
                <exclude name="**/*.java"/>
            </fileset>
```

Ant

- 자동 생성한 build.xml

*test.java | test_build.xml

```
        </copy>
    </target>
    <target name="clean">
        <delete dir="bin"/>
    </target>
    <target depends="clean" name="cleanall"/>
    <target depends="build-subprojects,build-project" name="build"/>
    <target name="build-subprojects"/>
    <target depends="init" name="build-project">
        <echo message="${ant.project.name}: ${ant.file}"/>
        <javac debug="true" debuglevel="${debuglevel}" destdir="bin" source="${source}" target="${target}">
            <src path="src"/>
            <classpath refid="test_89.classpath"/>
        </javac>
    </target>
    <target description="Build all projects which reference this project. Useful to propagate changes." name="build-refprojects"/>
    <target description="copy Eclipse compiler jars to ant lib directory" name="init-eclipse-compiler">
        <copy todir="${ant.library.dir}">
            <fileset dir="${ECLIPSE_HOME}/plugins" includes="org.eclipse.jdt.core_*.jar"/>
        </copy>
        <unzip dest="${ant.library.dir}">
            <patternset includes="jdtCompilerAdapter.jar"/>
            <fileset dir="${ECLIPSE_HOME}/plugins" includes="org.eclipse.jdt.core_*.jar"/>
        </unzip>
    </target>
```

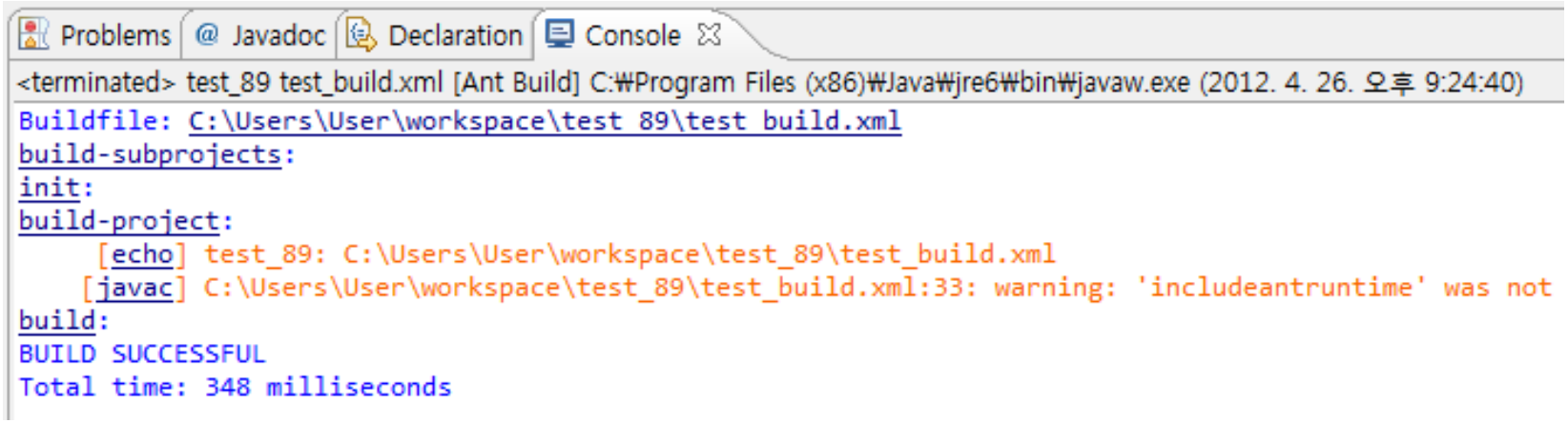
Ant

- 자동 생성한 build.xml

```
    <target description="compile project with Eclipse compiler" name="build-eclipse-compiler">
        <property name="build.compiler" value="org.eclipse.jdt.core.JDTCompilerAdapter"/>
        <antcall target="build"/>
    </target>
    <target name="test (1)">
        <java classname="test" failonerror="true" fork="yes">
            <classpath refid="test_89.classpath"/>
        </java>
    </target>
</project>
```

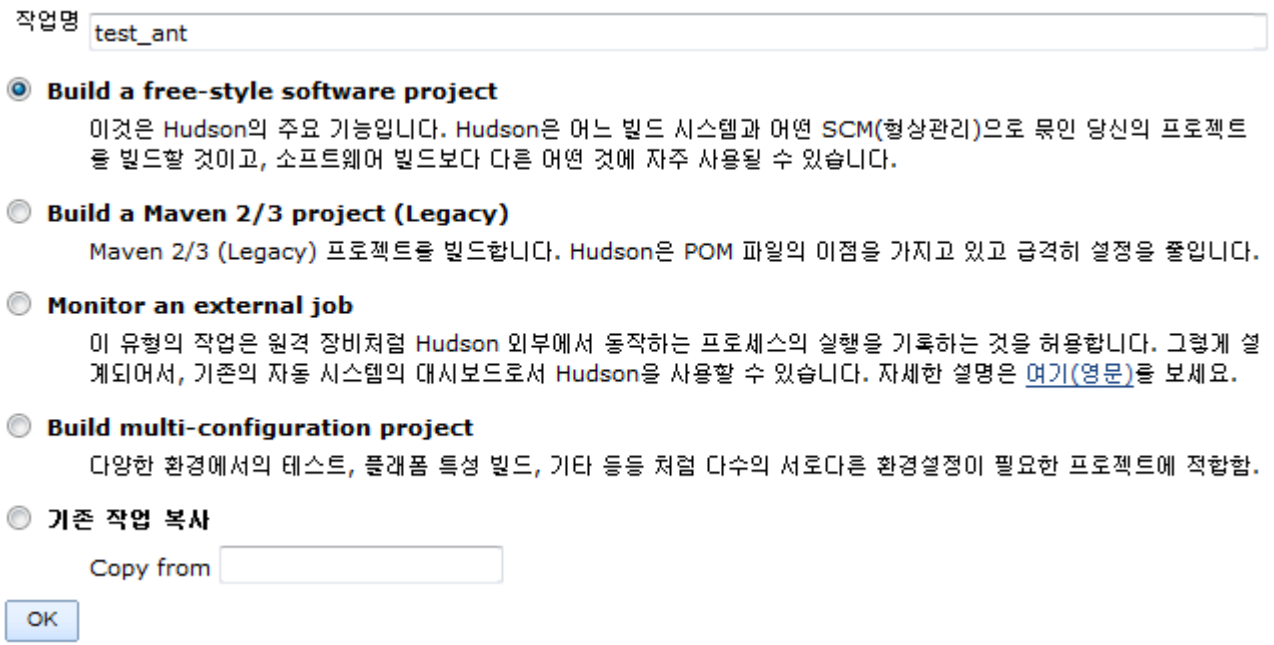
Ant

- build.xml 파일의 Build 결과



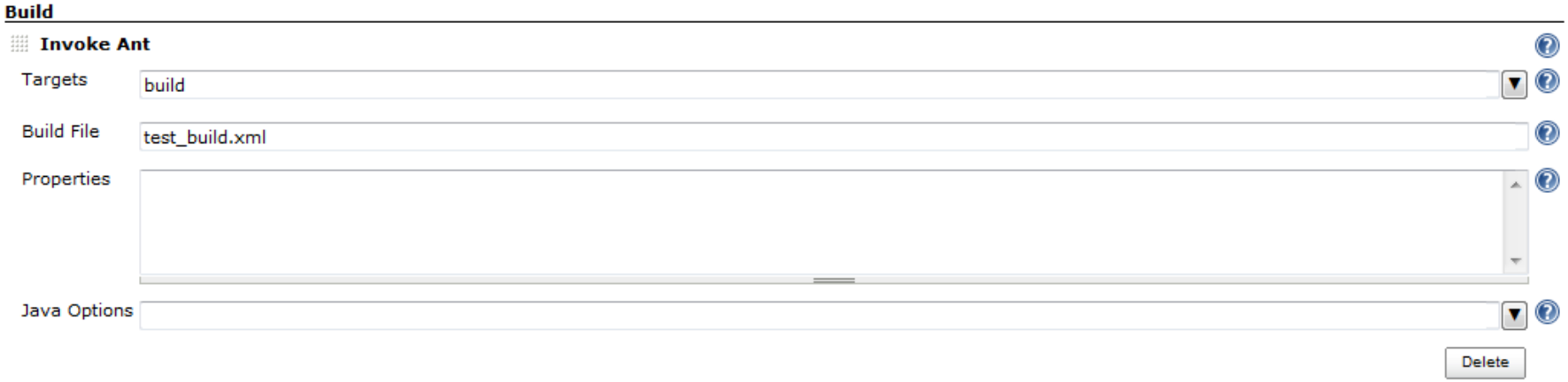
Ant

- New works



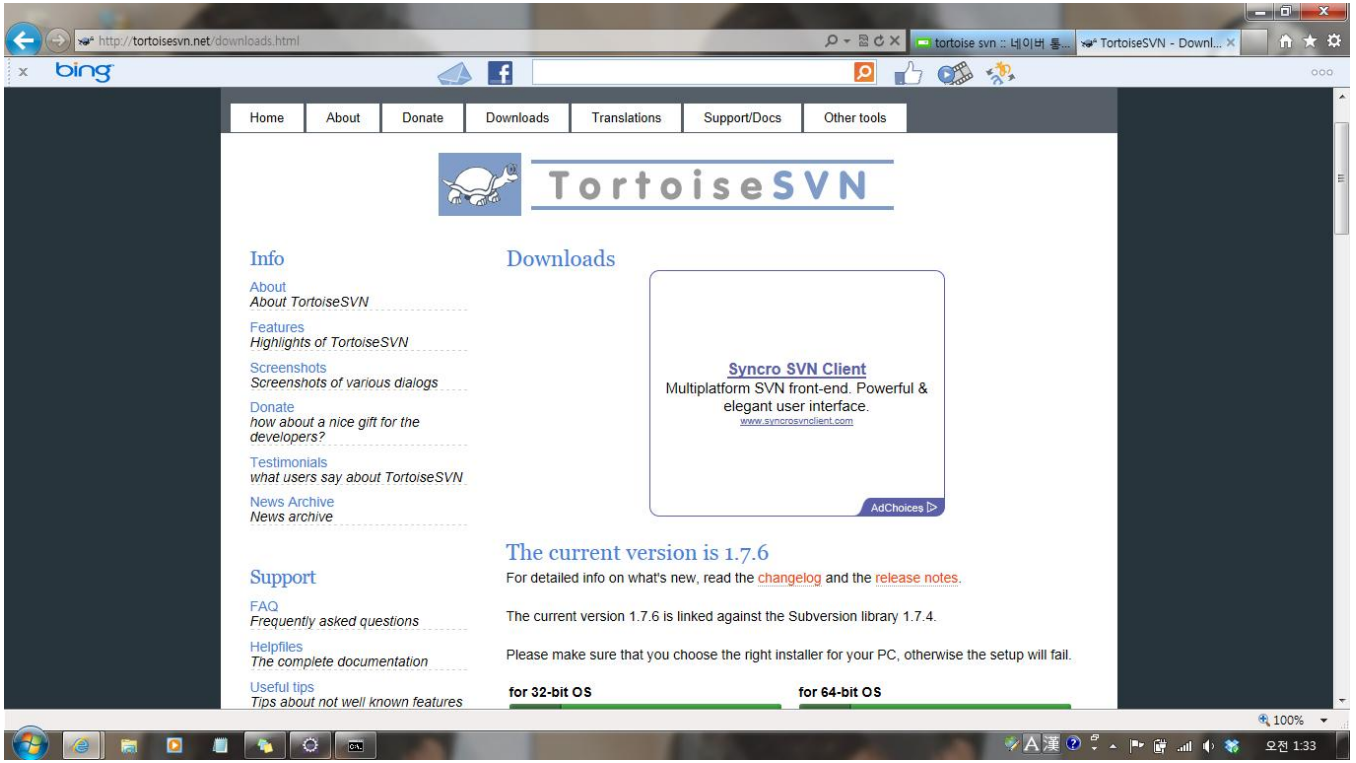
Ant

- Invoke Ant



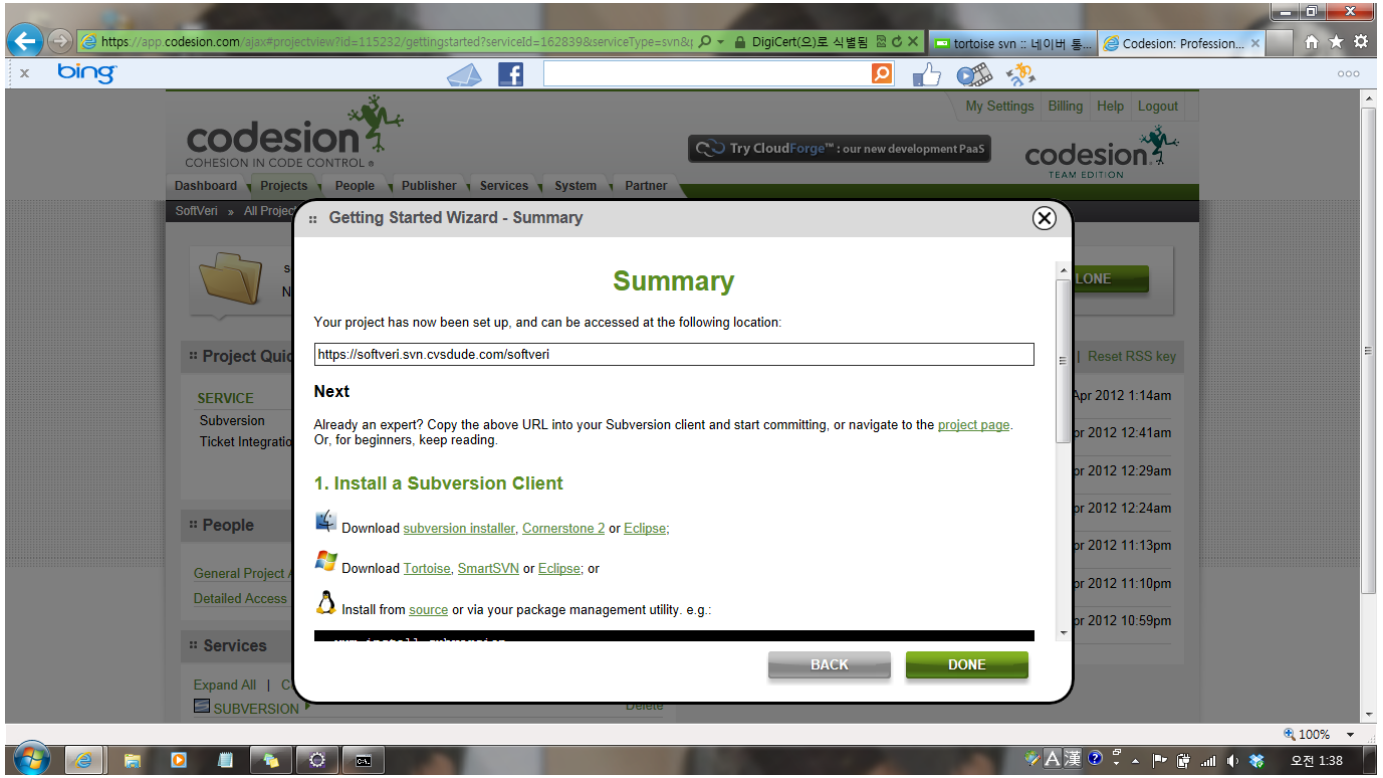
Tortoise SVN

- Tortoise SVN 홈페이지로 접속 후 최신버전 Download



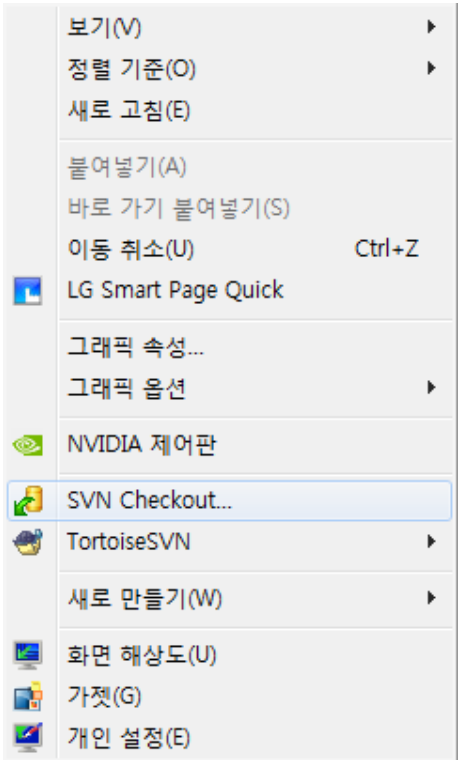
Tortoise SVN

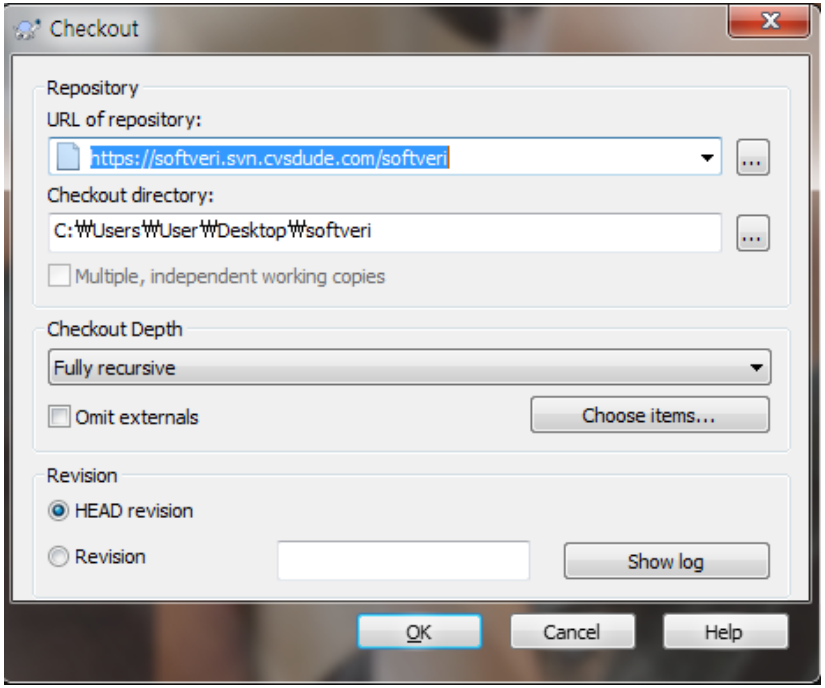
- Codesion.com 사이트를 통해 SVN 저장소 생성



Tortoise SVN

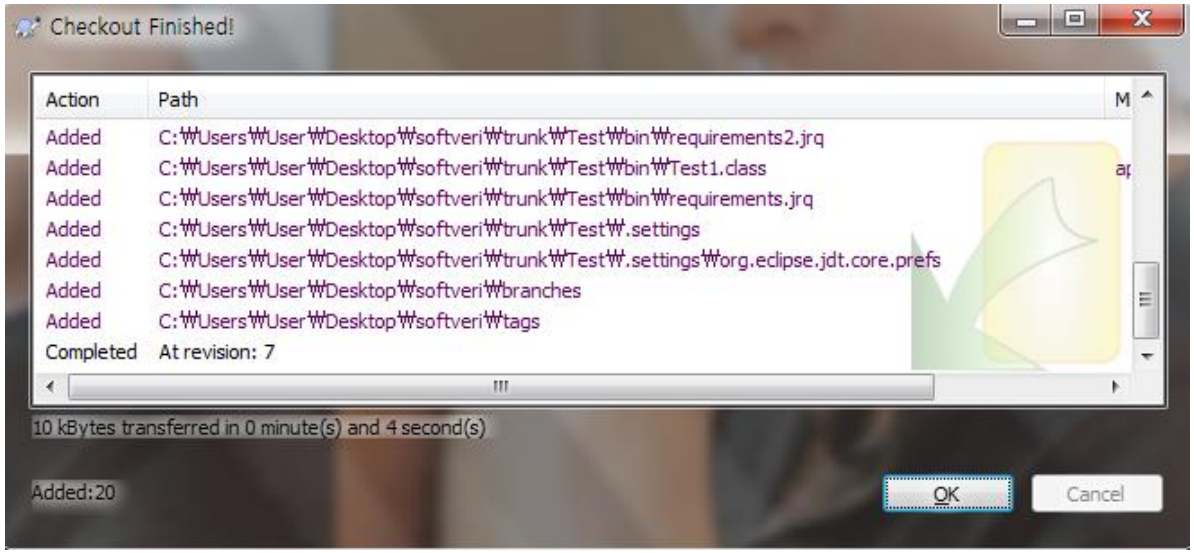
- SVN Checkout을 통해 내컴퓨터에 SVN폴더 생성(URL에는 SVN 저장소의 주소 입력)





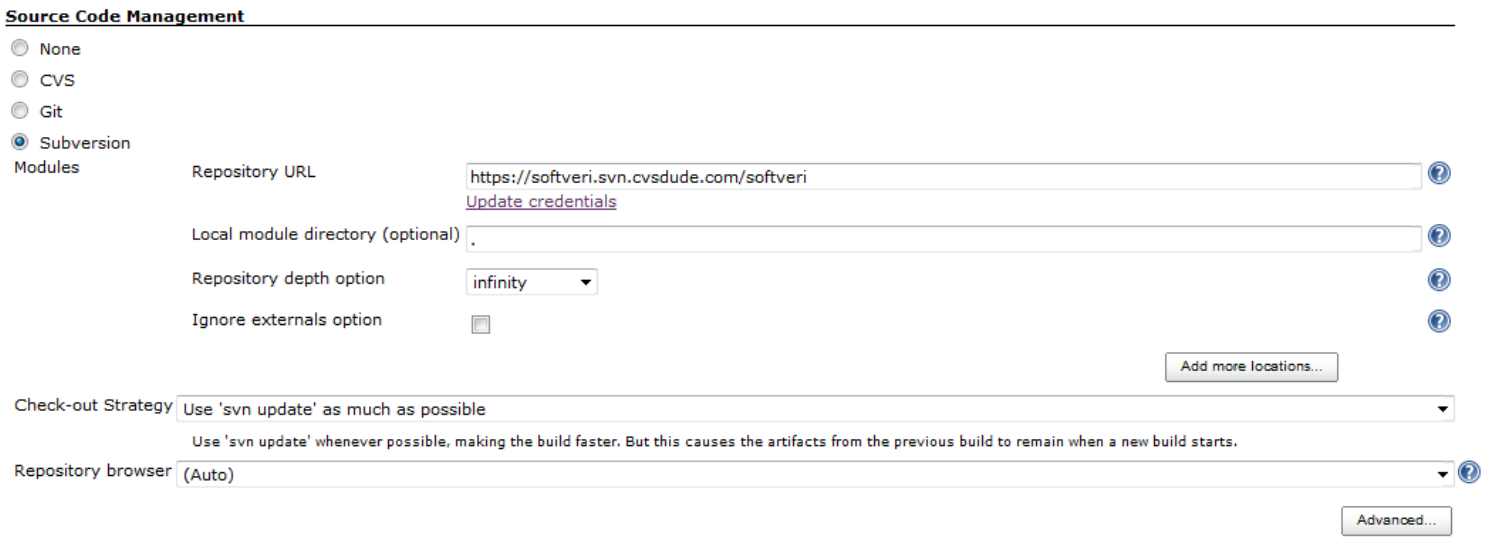
Tortoise SVN

- Checkout이 완료 된 후, 저장되었던 파일 update



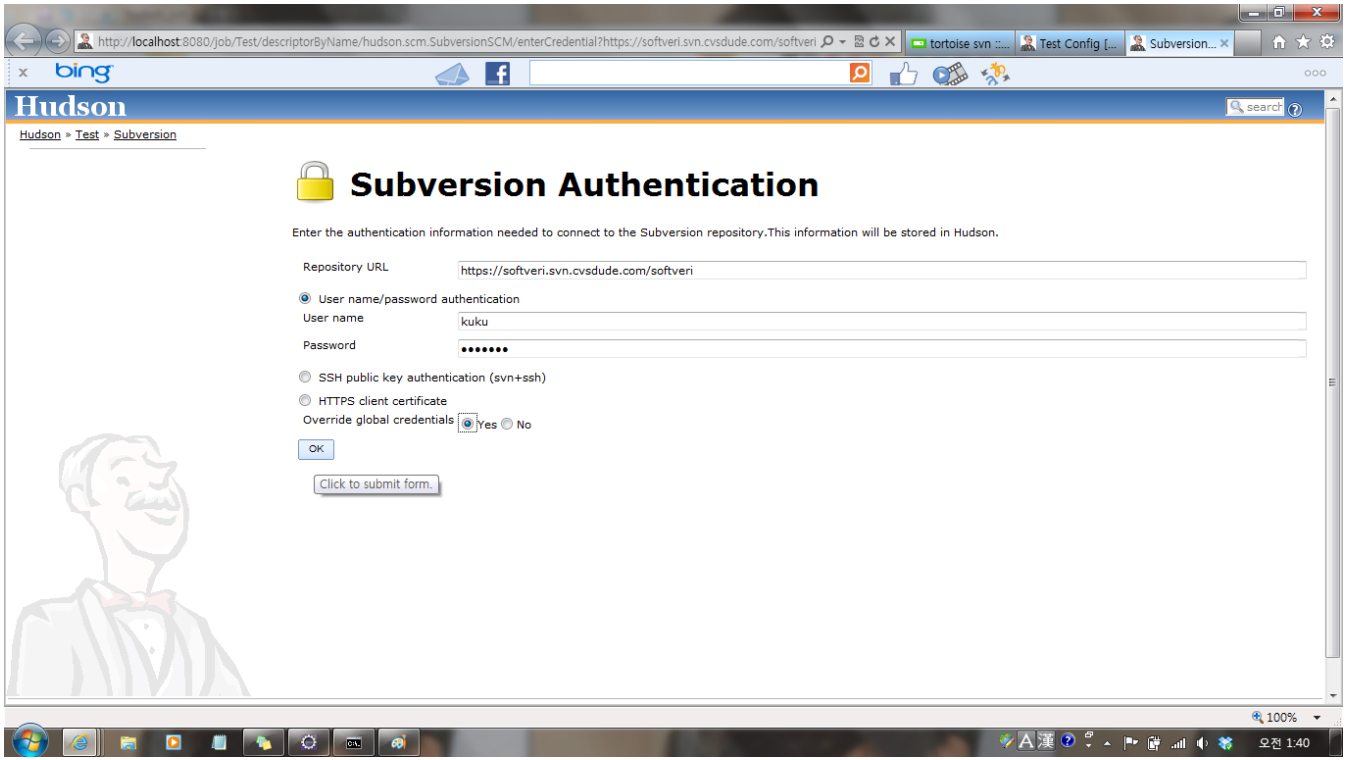
Tortoise SVN

- SVN URL에 SVN 저장소의 주소를 등록



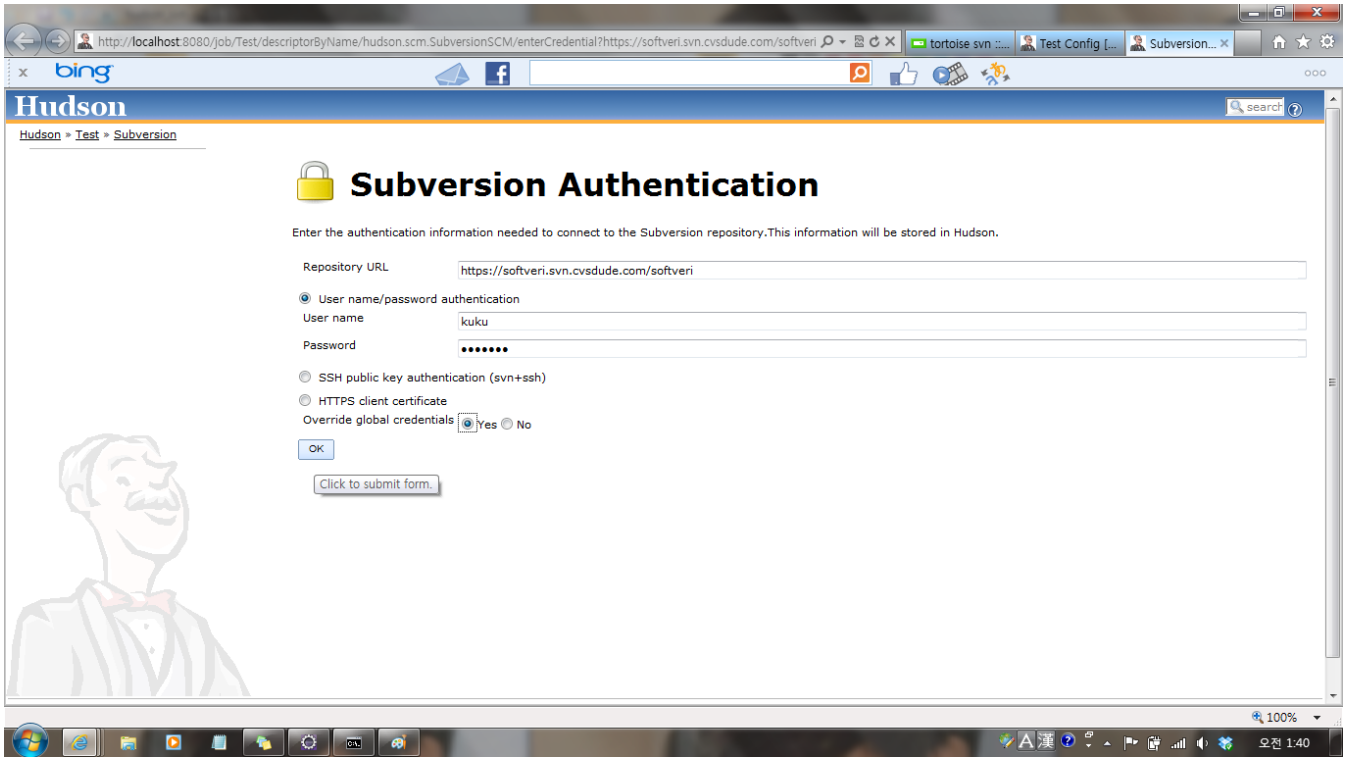
Tortoise SVN

- SVN 저장소에 로그인이 필요할 경우 인증을 해줘야 함



Tortoise SVN

- SVN 저장소에 로그인이 필요할 경우 인증을 해줘야 함



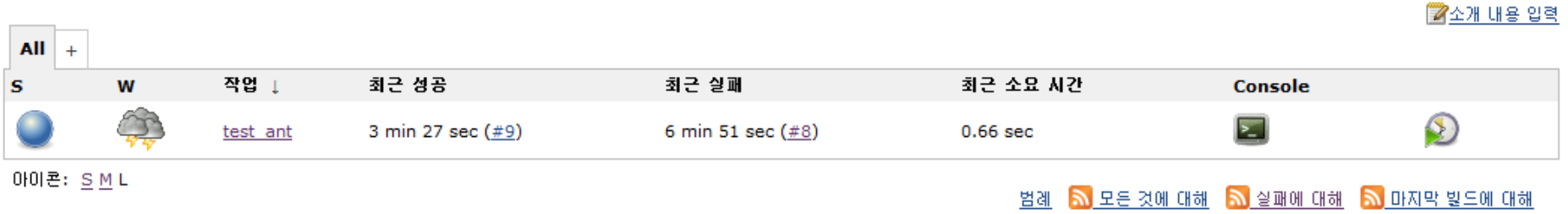
목차

Result - Ant

콘솔 출력

```
Started by user anonymous
[workspace] $ cmd.exe /C '"ant.bat -file test_build.xml build && exit %%ERRORLEVEL%%"'
Buildfile: C:\apache-tomcat-6.0.35\webapps\jobs\test_ant\workspace\test_build.xml

build-subprojects:

init:

build-project:
     [echo] test_89: C:\apache-tomcat-6.0.35\webapps\jobs\test_ant\workspace\test_build.xml
    [javac] C:\apache-tomcat-6.0.35\webapps\jobs\test_ant\workspace\test_build.xml:33: warning:

build:

BUILD SUCCESSFUL
Total time: 0 seconds
[DEBUG] Skipping watched dependency update; build not configured with trigger: test_ant #9
Finished: SUCCESS
```

Result - Ant



“ 20% 만 성공 ”

-> Origin Source Code 에서 의도적으로
오류가 발생하도록 작성하였기 때문

Result - Tortoise SVN

- SVN에 저장된 코드를 hudson을 통해 빌드 성공한 화면

감사합니다