

SOFTWARE VERIFICATION

Ant, Cruise Control

2005 | 1342 | 이도현
2006 | 1000 | 안병욱
2007 | 10069 | 전창완
2009 | 2432 | 김다영

INDEX



| Index

| CI

| Ant

| Cruise Control

CI

CI

! CI(Continuous Integration)

/ 개발자 자신을 위한 자동화 방법론

/ 통합개발 환경 도입

/ 절차 자동화 배포 도구 등 등장

/ 프로세스 관리 통합 도구를 이용, 개발프로세스 자동화

/ 개발 라이브러리에 대한 연관관계를 도구로 자동화

CI

CI(Continuous Integration)

- / 사전적 의미의 **지속적**이라는 말은 어떠한 상태가 오랫동안 계속 유지되는 것, 지속적 통합의 **지속적**이라는 뜻은 어떤 행동을 수시로 반복 수행한다는 의미가 더 큼
- / 소프트웨어에 통합을 반복하고, 수시로 수행 및 테스트 하는 것.

CI

CI도구의 주요 기능

/ 프로젝트 관리 기능

/ 버전관리, 시스템에 연결하여 변경사항 인지

/ 빌드 스크립트 실행, 실행 결과 확인

/ 특정 이벤트 발생 시 빌드 스크립트 실행

/ 프로젝트 빌드 후 생성되는 산출물과 연동할 수 있는 기능

/ 빌드 스크립트의 실행 결과를 구성원에게 피드백 할수 있는 기능

CI

CI도구의 수행 과정

- / Code저장소에 Code를 Commit
- / CI서버가 변경사항 없는지 저장소를 확인(Interval 설정가능)
- / Commit이 되면 CI서버가 저장소에서 변경내역 확인
- / 최신소스를 가져와 빌드 스크립트를 실행, 소프트웨어 통합
- / CI서버는 프로젝트 구성원에게 메일로 빌드 결과를 보냄
- / CI서버는 저장소에 변경사항을 계속해서 확인

CI

┆ 지속적 통합에서 빌드의 의미

┆ 개발자가 형상 관리에 Commit한 소스를 가장 먼저 테스트 하는 품질 검사 활동을 의미

┆ 테스트와 검사의 차이

┆ 테스트는 소프트웨어 동작과 관련된 부분을 살펴 봄

┆ 검사는 소프트웨어 구조가 올바르게 작성되었는지를 살펴 봄

┆ 즉, 테스트는 결과가 올바른지 판단, 검사는 구조적인 면을 판단

CI

CI 실천방법

/ 개인 빌드를 돌릴 것

/ 코드를 자주 Commit 할 것

/ 깨진 코드를 Commit하지 말고, 깨진 코드를 받지 말 것

/ 빌드가 깨지면 즉시 수정할 것

/ 자동화된 테스트를 작성할 것

/ 테스트와 검사 모두 통과 할 것

CI

CI 도입시 주의사항

- / 개발방법을 적용하여 얻을수 있는 목표를 크게 잡지 말 것
- / 확인 가능한 각종 결과물 및 피드백을 100% 신뢰하지 말 것
 - / 개발자는 완전한 코드를 Commit하지 않음
 - / 모든 통합 절차를 통과하였어도 문제가 발생
 - / 모든 절차를 CI에 포함시키는 것은 위험할 수 있음
- / 세상에 자동화란 일부일 뿐, 사람없는 자동화는 불가능

CI

| CI 장점

/ 위험을 줄여줌

/ 반복적인 작업을 줄여줌

/ 언제 어느 때라도 배포할 수 있는 소프트웨어를 생성

/ 프로젝트 가시성을 보기 좋게 해줌

CI

| CI 도구들

/ IBM Rational Build Forge

/ TeamCity (JetBrains)

/ Hudson

/ CruiseControl

/ ...

Ant

Ant

| Ant?

/' James Duncan Davidson이 만든 Java Build Tool

/' Another Neat Tool (ANT)

Ant

┆ Ant, 빌드도구로서의 의미

┆ XML 사용하여 문법이 간결

┆ 기존 Make를 대체

┆ Cross Platform 지원, 포팅가능한 방법으로 Java의 Classpath와 File Directory 구조를 그대로 사용가능

┆ JUnit Test Framework와 밀접하게 통합

┆ FTP, Telnet, SQL 등의 명령어들을 배치 자동화에 사용가능

┆ Apache Tomcat 등과 같은 많은 오픈소스의 업계표준으로 정착

Ant

Ant 설치법

/ ant.apache.org

/ Binary Distributions

/ apache-ant-1.9.0-bin.tar.gz

The screenshot shows the Apache Ant website's 'Binary Distributions' page. The browser address bar shows 'ant.apache.org/bin/download.cgi'. The page features a navigation menu on the left with categories like 'Apache Ant', 'Documentation', 'Download', 'Contributing', 'Sponsorship', and 'Project Management'. The 'Download' section is circled in red, with 'Binary Distributions' highlighted. The main content area includes a 'Binary Distributions' header, a description of Apache Ant, and instructions for downloading. A 'Mirror' section shows the current mirror as 'http://apache.tt.co.kr/'. A 'Current Release of Ant' section states that Apache Ant 1.9.0 is the best available version. A red 'Note' box indicates that Ant 1.9.0 was released on 07-Mar-2013 and may not be available on all mirrors. A red box titled 'Tar files may require gnu tar to extract' contains a warning about long file names. At the bottom, a list of download links for .zip, .tar.gz, and .tar.bz2 archives is shown, with the .tar.gz link 'apache-ant-1.9.0-bin.tar.gz' circled in red. The 'Old Ant Releases' section is partially visible at the bottom.

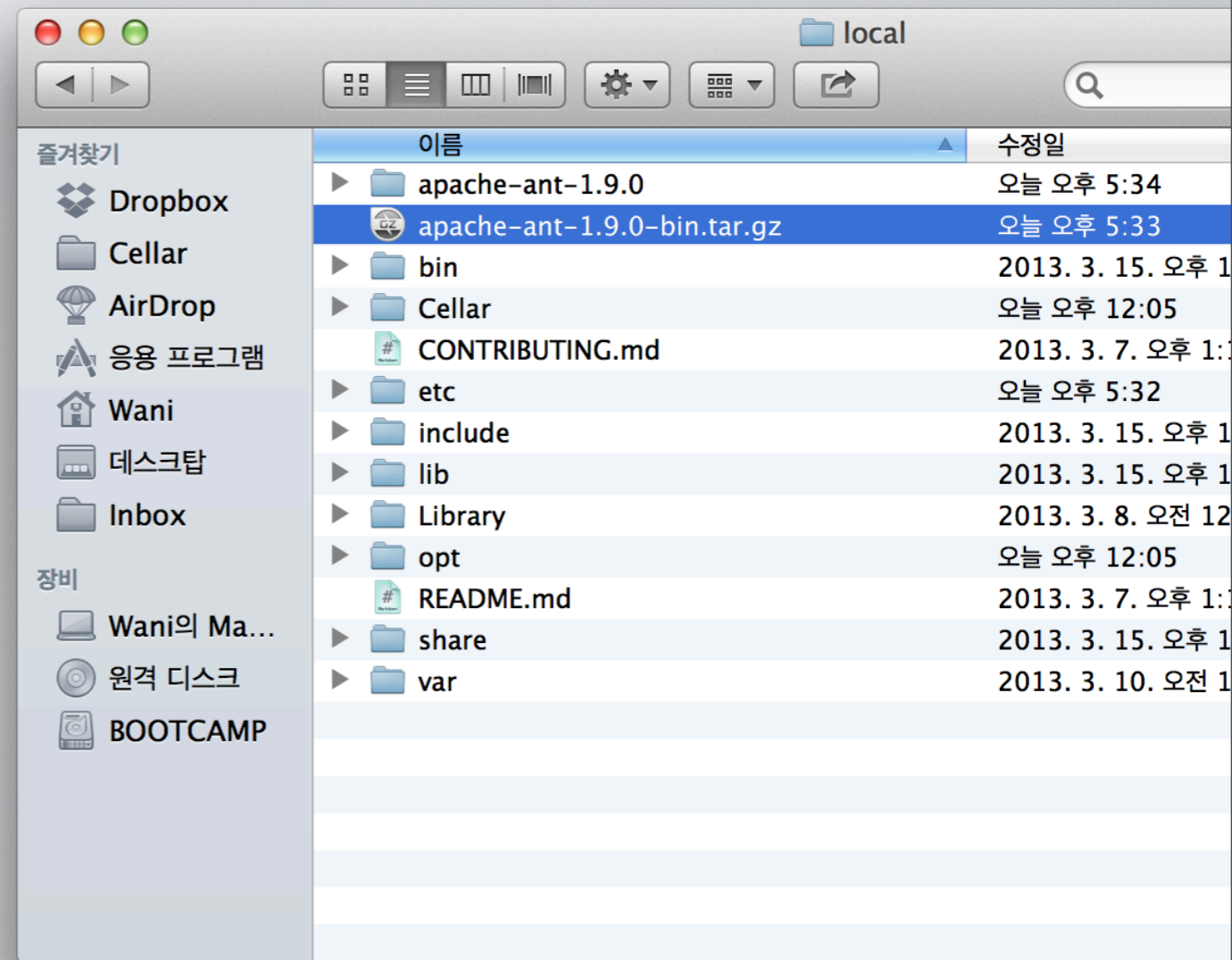
Ant

Ant 설치법

/usr/local/
(Mac)

C:\Program Files\Java\
(Window)

압축해제



Ant

Ant 설치법(Mac)

vi ~/.bash_profile

export ANT_HOME="/usr/local/apache-ant-1.9.0"
export PATH="{ANT_HOME}/bin:\$PATH" 추가

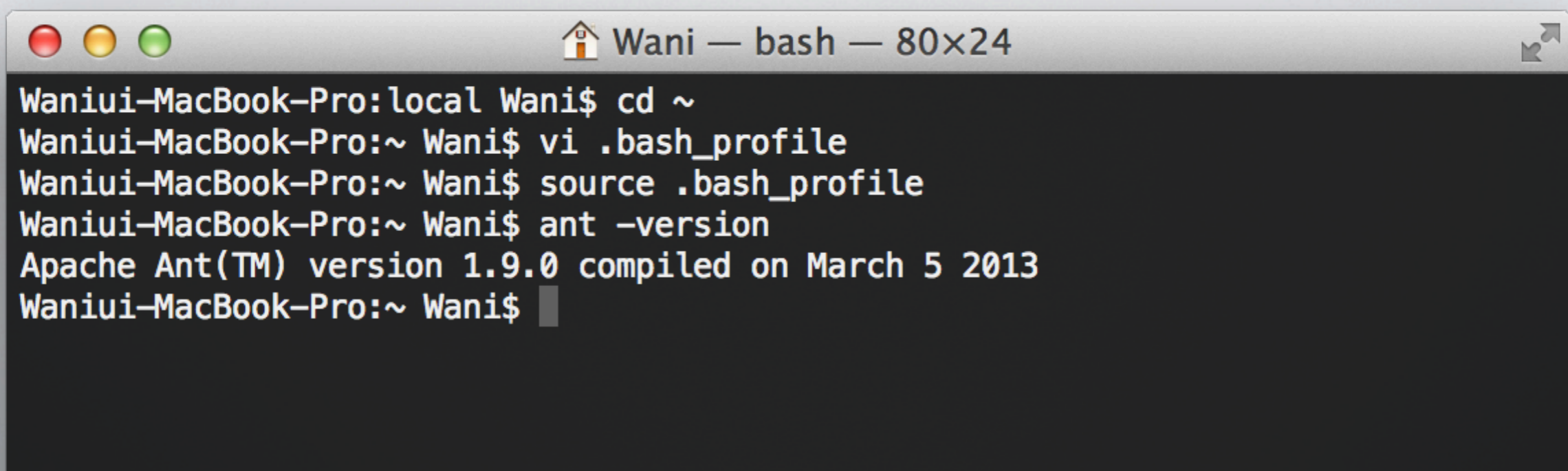
```
1 export CC_HOME="/usr/local/cruisecontrol-2.8.4"
2 export ANT_HOME="/usr/local/apache-ant-1.9.0"
3 export PATH="{ANT_HOME}/bin:$PATH"
4 export PATH="{ANT_HOME}/bin:$PATH"
5 export PATH="{CC_HOME}/bin:$PATH"
6
7 export CLICOLOR=1
8 export LSCOLORS=GxFxCxDxBxegedabagaced
9
10 alias ls='ls -la'
11 alias proj='cd /Users/Wani/Desktop/Project'
12 alias mamp='cd /Users/Wani/Desktop/Project/Mamp'
```

Ant

└ Ant 설치법(Mac)

└ source .bash_profile

└ ant -version , 설치완료!

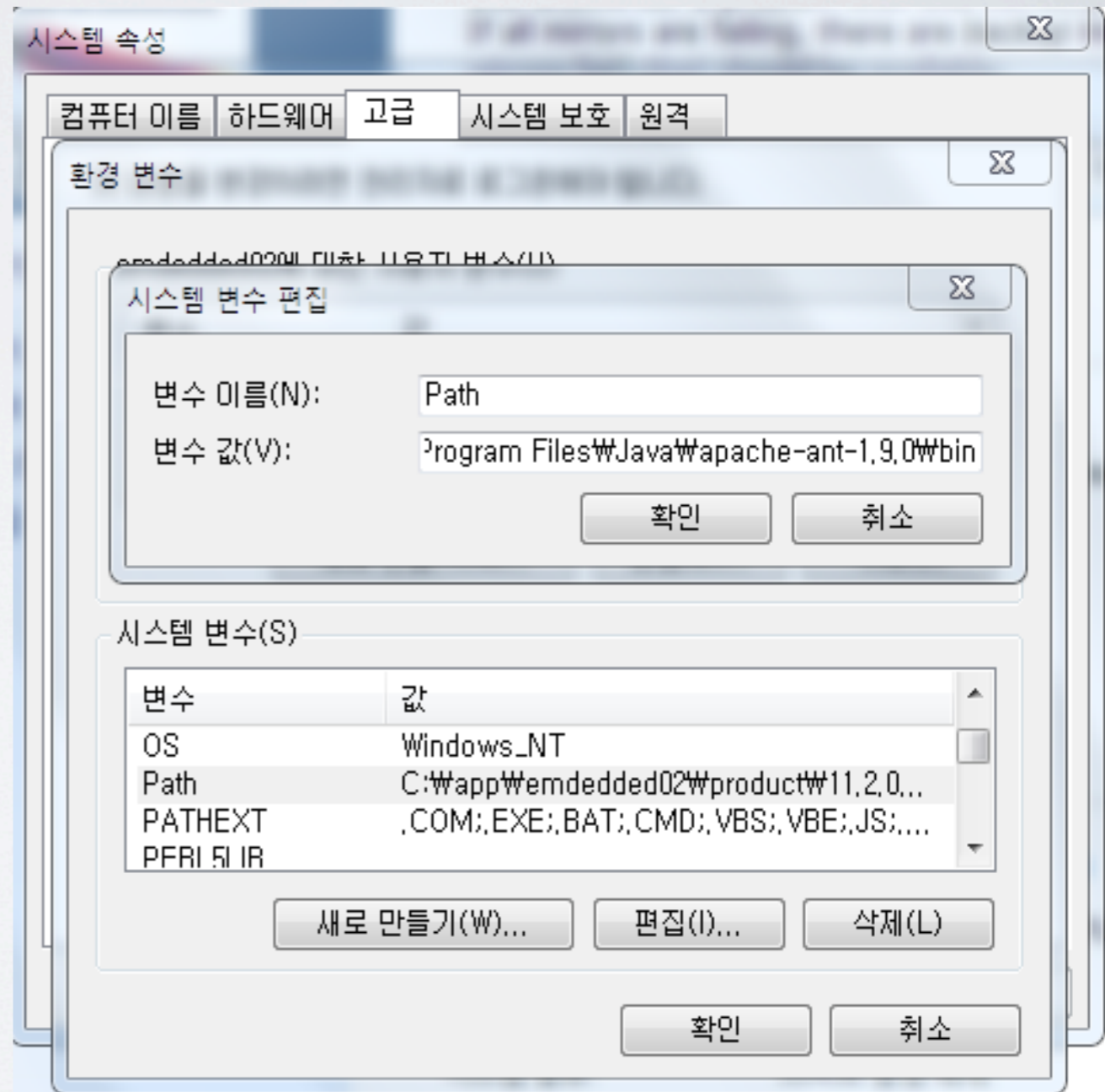


```
Wani — bash — 80x24
Waniui-MacBook-Pro:local Wani$ cd ~
Waniui-MacBook-Pro:~ Wani$ vi .bash_profile
Waniui-MacBook-Pro:~ Wani$ source .bash_profile
Waniui-MacBook-Pro:~ Wani$ ant -version
Apache Ant(TM) version 1.9.0 compiled on March 5 2013
Waniui-MacBook-Pro:~ Wani$
```

Ant

Ant 설치법(Window)

환경변수 추가

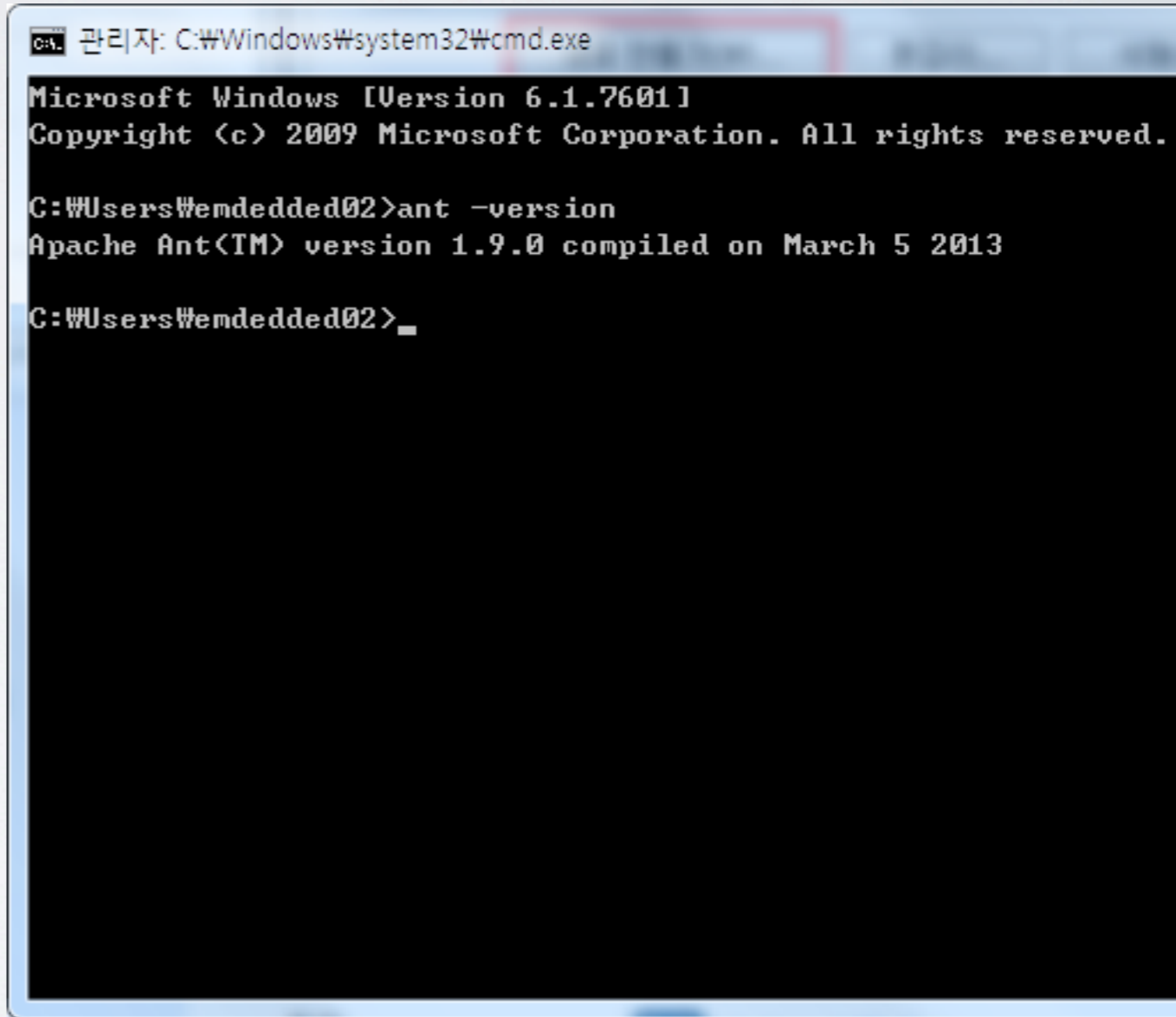


Ant

| Ant 설치법(Window)

/ 시작 -> cmd

/ ant -version , 설치완료!



```
C:\> 관리자: C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\emdedded02>ant -version
Apache Ant(TM) version 1.9.0 compiled on March 5 2013

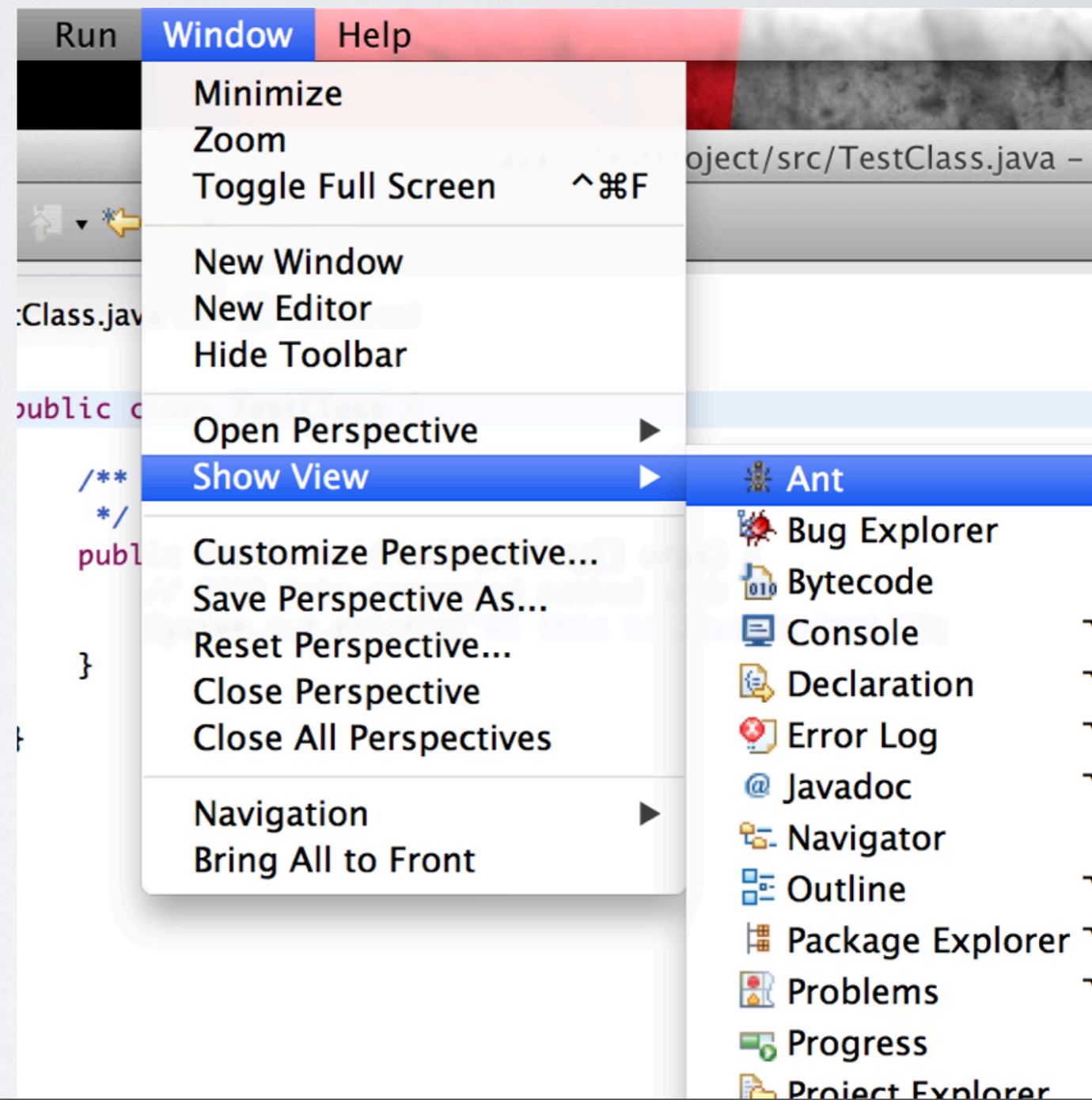
C:\Users\emdedded02>_
```

Ant

Ant와 Eclipse

/ 기본으로 내장되어있음

/ Window -> Show View ->
Ant

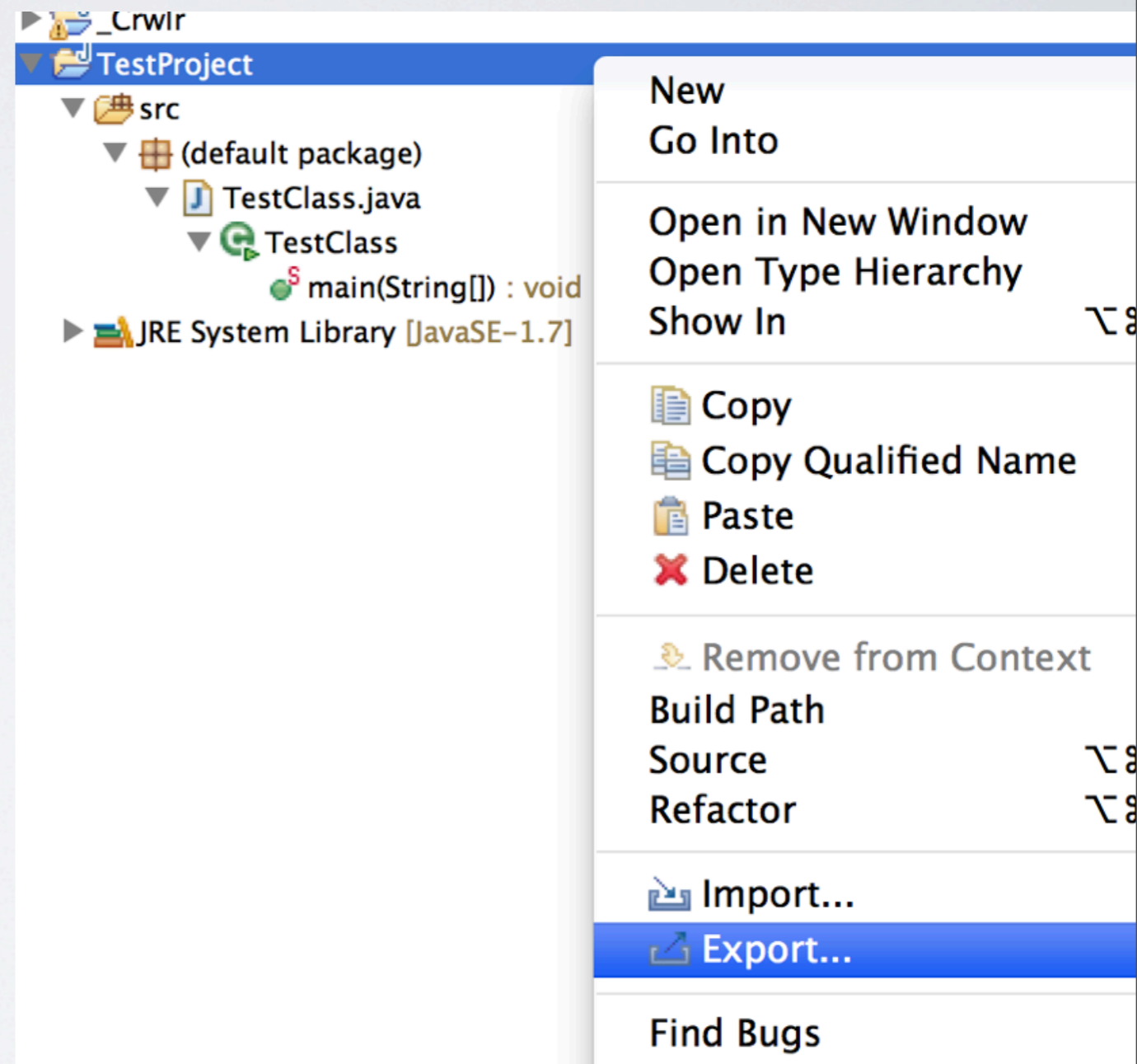


Ant

| Ant와 Eclipse

/ 작성한 프로젝트에서 우클릭

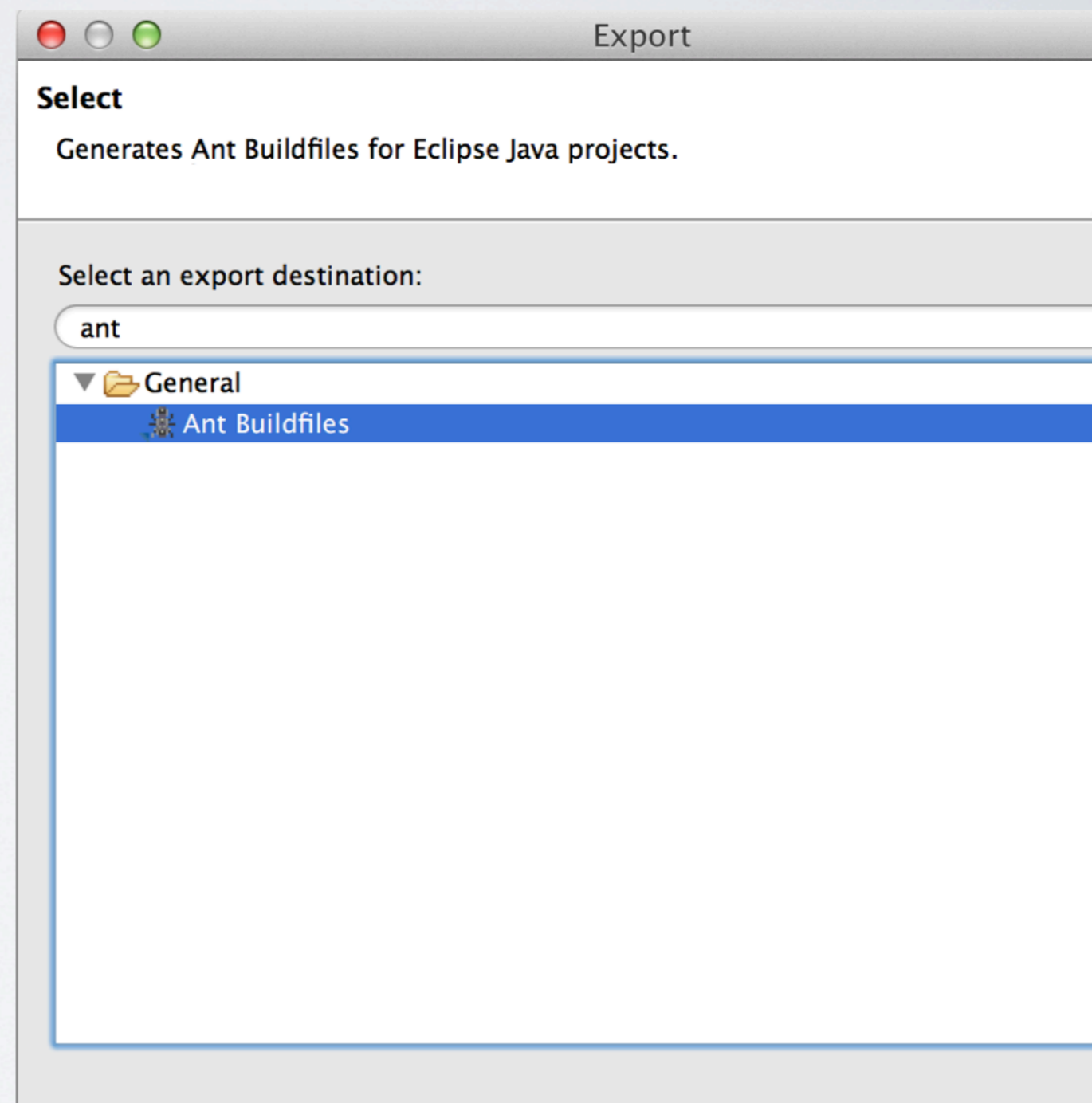
/ Export



Ant

| Ant와 Eclipse

/ Ant, Next, Generate



Ant

Ant와 Eclipse

build.xml 이 생성!

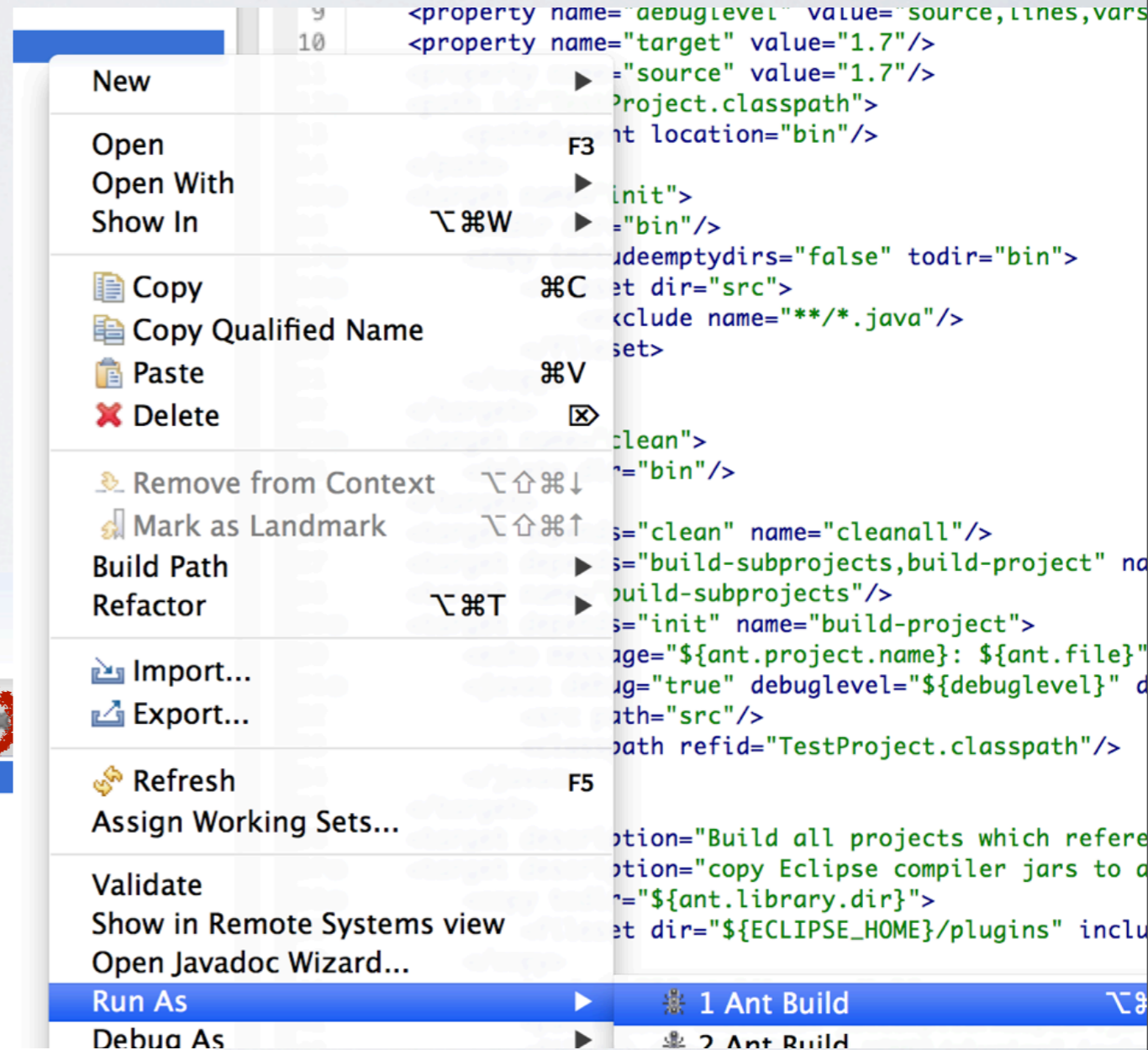
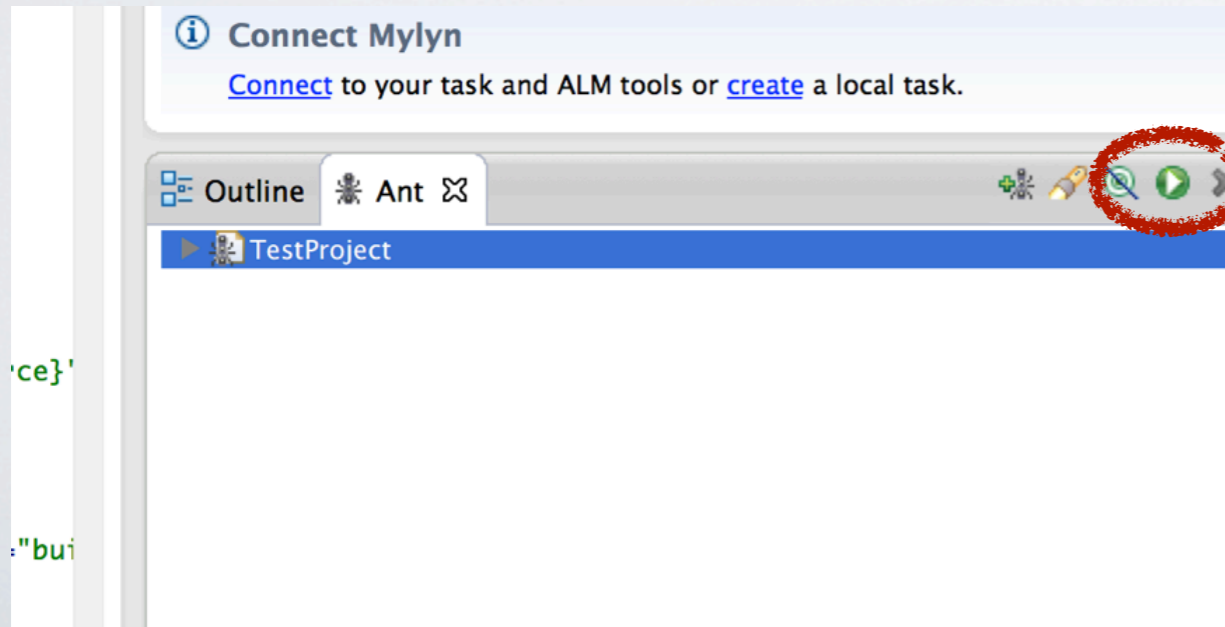
```
TestClass.java build.xml
1 <?xml version="1.0" encoding="UTF-8" standalone="no"?>
2 <!-- WARNING: Eclipse auto-generated file.
3      Any modifications will be overwritten.
4      To include a user specific buildfile here, simply create o
5      directory with the processing instruction <?eclipse.ant.im
6      as the first entry and export the buildfile again. --><pro
7 <property environment="env"/>
8 <property name="ECLIPSE_HOME" value="../../../../Applications/eclips
9 <property name="debuglevel" value="source,lines,vars"/>
10 <property name="target" value="1.7"/>
11 <property name="source" value="1.7"/>
12 <path id="TestProject.classpath">
13     <pathelement location="bin"/>
14 </path>
15 <target name="init">
16     <mkdir dir="bin"/>
17     <copy includeemptydirs="false" todir="bin">
18         <fileset dir="src">
19             <exclude name="**/*.java"/>
20         </fileset>
21     </copy>
22 </target>
23 <target name="clean">
24     <delete dir="bin"/>
25 </target>
26 <target depends="clean" name="cleanall"/>
27 <target depends="build-subprojects,build-project" name="build"/>
28 <target name="build-subprojects"/>
29 <target depends="init" name="build-project">
30     <echo message="${ant.project.name}: ${ant.file}"/>
31     <javac debug="true" debuglevel="${debuglevel}" destdir="bin" inc
32         <src path="src"/>
33         <classpath refid="TestProject.classpath"/>
34     </javac>
35 </target>
```

Ant

Ant와 Eclipse

Ant 창에 Run클릭

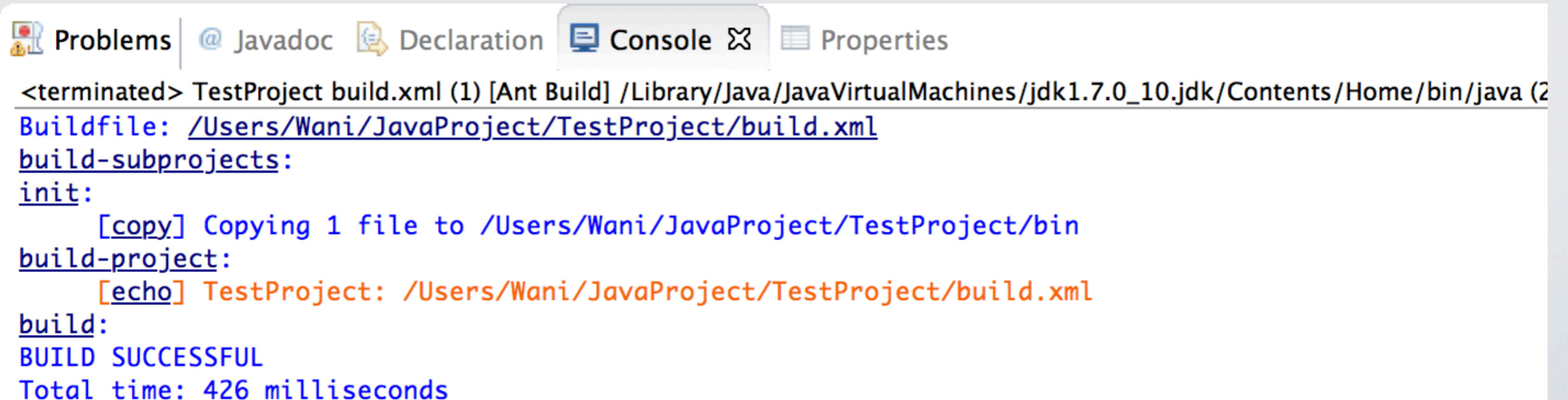
또는, Runs As -> Ant Build



Ant

Ant와 Eclipse

아래 콘솔창에 Build 과정을 볼 수 있다.



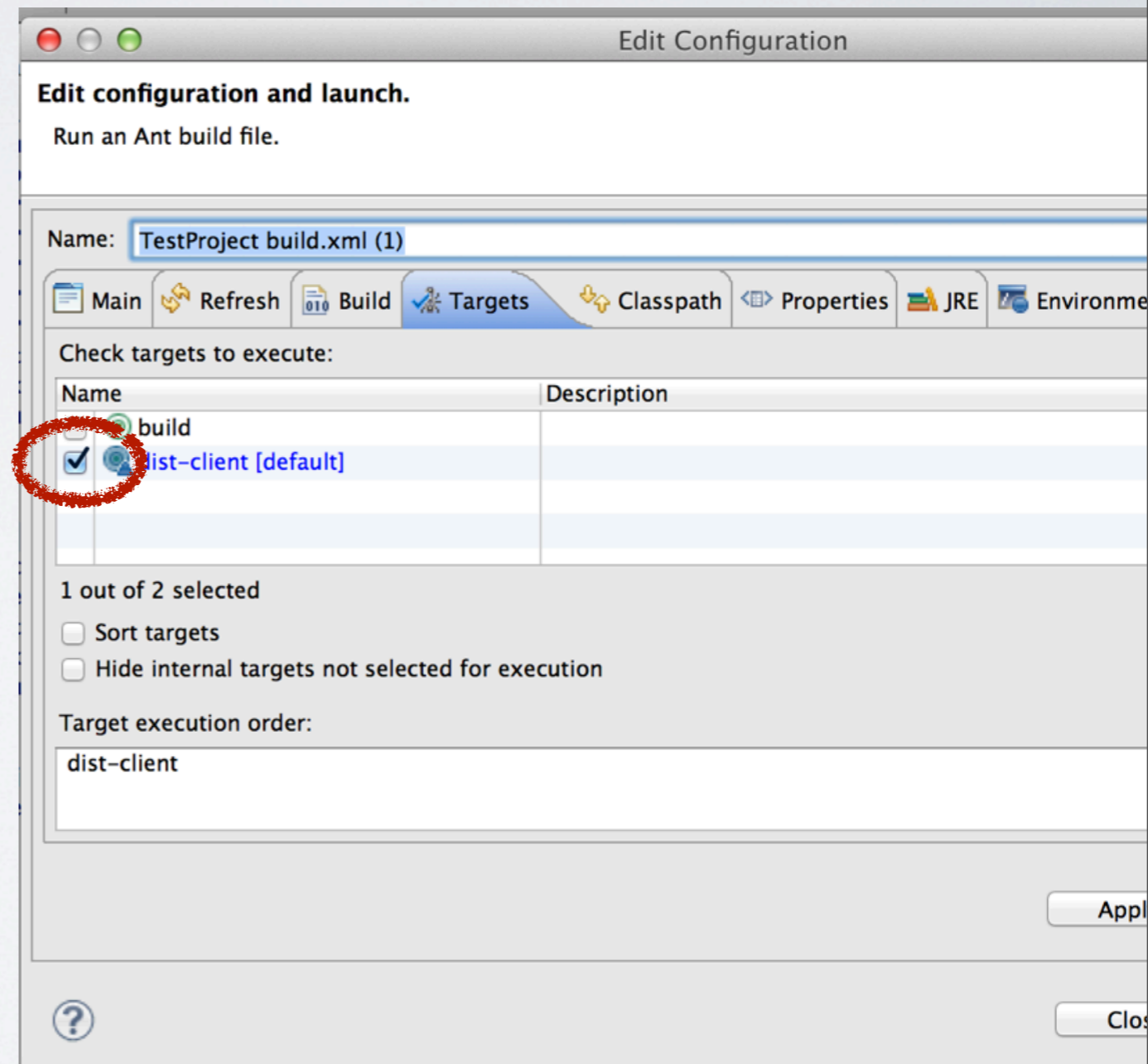
```
Problems | @ Javadoc | Declaration | Console ✕ | Properties
<terminated> TestProject build.xml (1) [Ant Build] /Library/Java/JavaVirtualMachines/jdk1.7.0_10.jdk/Contents/Home/bin/java (2
Buildfile: /Users/Wani/JavaProject/TestProject/build.xml
build-subprojects:
init:
  [copy] Copying 1 file to /Users/Wani/JavaProject/TestProject/bin
build-project:
  [echo] TestProject: /Users/Wani/JavaProject/TestProject/build.xml
build:
BUILD SUCCESSFUL
Total time: 426 milliseconds
```

Ant

Ant와 Eclipse

Runs As -> Ant Build...

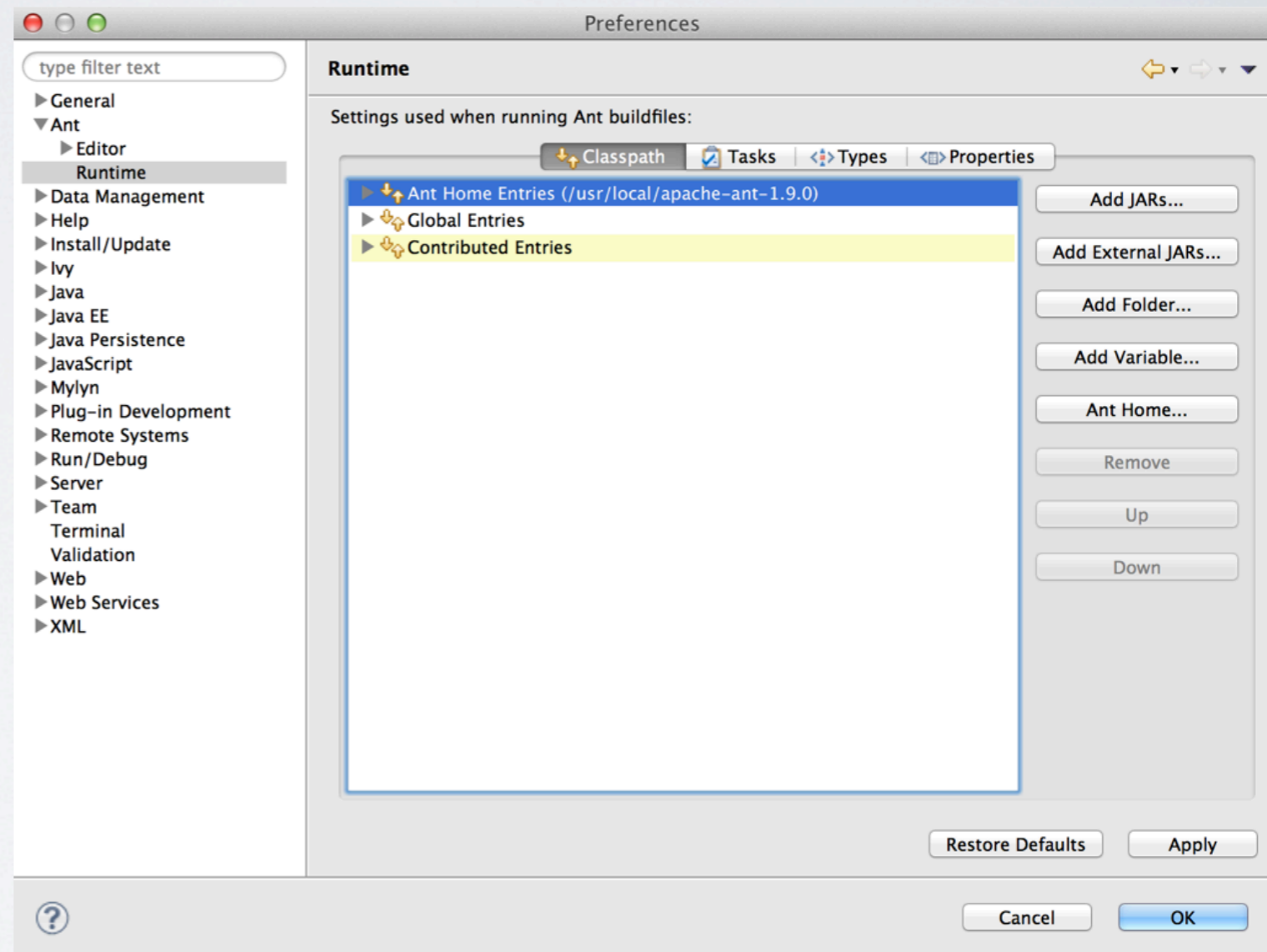
target 선택가능



Ant

Ant와 Eclipse

Ant 최신버전을 Eclipse
에서 사용하려면..

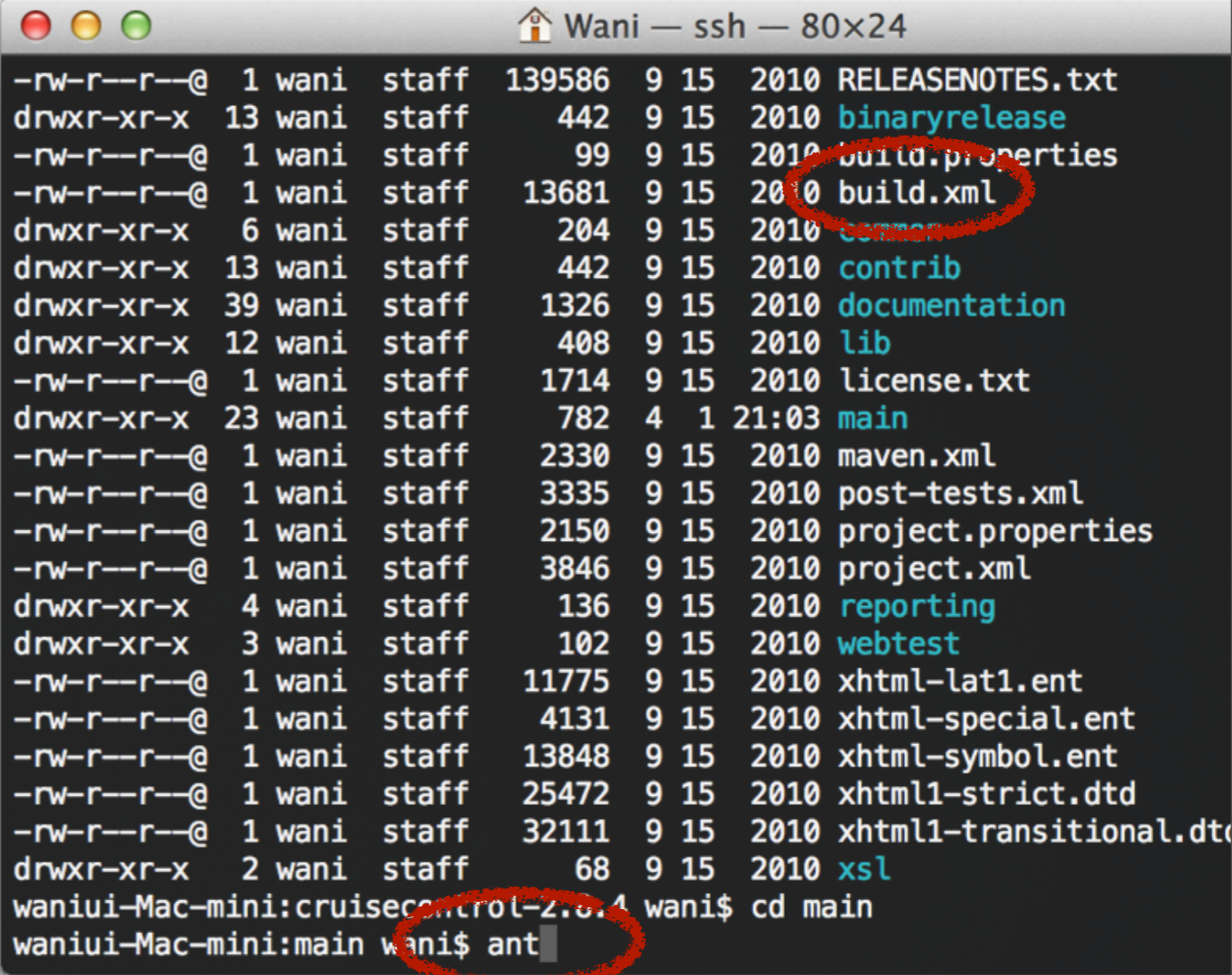


Ant

| Command line tool

/ build.xml 확인

/ ant [Enter!]

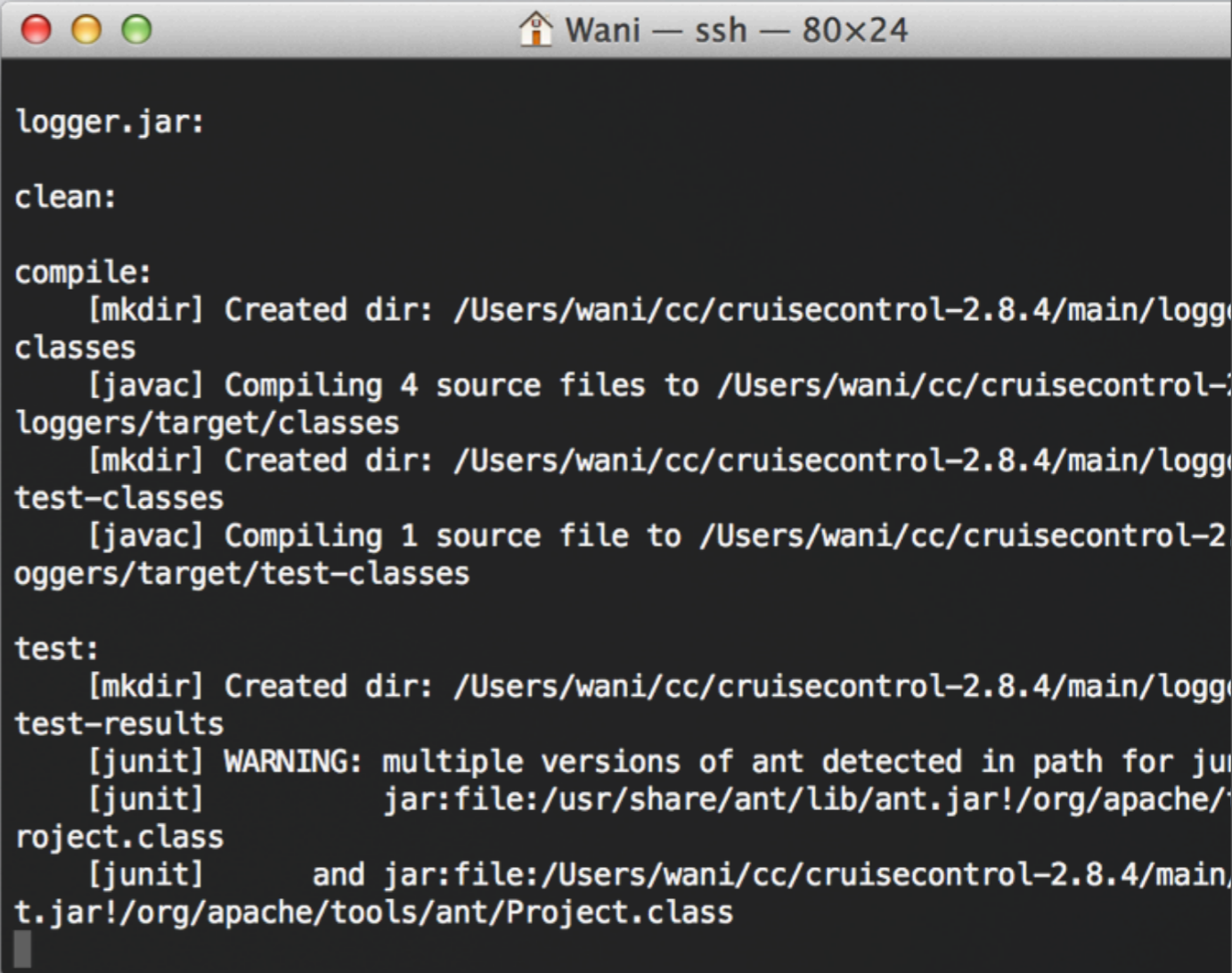


```
Wani — ssh — 80x24
-rw-r--r--@ 1 wani  staff  139586  9 15  2010  RELEASENOTES.txt
drwxr-xr-x 13 wani  staff    442  9 15  2010  binaryrelease
-rw-r--r--@ 1 wani  staff    99  9 15  2010  build.properties
-rw-r--r--@ 1 wani  staff  13681  9 15  2010  build.xml
drwxr-xr-x  6 wani  staff   204  9 15  2010  contrib
drwxr-xr-x 13 wani  staff   442  9 15  2010  contrib
drwxr-xr-x 39 wani  staff  1326  9 15  2010  documentation
drwxr-xr-x 12 wani  staff   408  9 15  2010  lib
-rw-r--r--@ 1 wani  staff  1714  9 15  2010  license.txt
drwxr-xr-x 23 wani  staff   782  4  1  21:03  main
-rw-r--r--@ 1 wani  staff  2330  9 15  2010  maven.xml
-rw-r--r--@ 1 wani  staff  3335  9 15  2010  post-tests.xml
-rw-r--r--@ 1 wani  staff  2150  9 15  2010  project.properties
-rw-r--r--@ 1 wani  staff  3846  9 15  2010  project.xml
drwxr-xr-x  4 wani  staff   136  9 15  2010  reporting
drwxr-xr-x  3 wani  staff   102  9 15  2010  webtest
-rw-r--r--@ 1 wani  staff  11775  9 15  2010  xhtml-lat1.ent
-rw-r--r--@ 1 wani  staff   4131  9 15  2010  xhtml-special.ent
-rw-r--r--@ 1 wani  staff  13848  9 15  2010  xhtml-symbol.ent
-rw-r--r--@ 1 wani  staff  25472  9 15  2010  xhtml1-strict.dtd
-rw-r--r--@ 1 wani  staff  32111  9 15  2010  xhtml1-transitional.dtd
drwxr-xr-x  2 wani  staff    68  9 15  2010  xsl
waniui-Mac-mini:cruisecontrol-2.0.4 wani$ cd main
waniui-Mac-mini:main wani$ ant
```

Ant

| Command line tool

/ build되는 것을 볼 수 있다.



```
Wani — ssh — 80x24
logger.jar:
clean:
compile:
  [mkdir] Created dir: /Users/wani/cc/cruisecontrol-2.8.4/main/loggers/target/classes
  [javac] Compiling 4 source files to /Users/wani/cc/cruisecontrol-2.8.4/main/loggers/target/classes
  [mkdir] Created dir: /Users/wani/cc/cruisecontrol-2.8.4/main/loggers/target/test-classes
  [javac] Compiling 1 source file to /Users/wani/cc/cruisecontrol-2.8.4/main/loggers/target/test-classes
test:
  [mkdir] Created dir: /Users/wani/cc/cruisecontrol-2.8.4/main/loggers/target/test-results
  [junit] WARNING: multiple versions of ant detected in path for junit
  [junit]          jar:file:/usr/share/ant/lib/ant.jar!/org/apache/ant/Project.class
  [junit]          and jar:file:/Users/wani/cc/cruisecontrol-2.8.4/main/loggers/target/test-results/ant.jar!/org/apache/tools/ant/Project.class
```

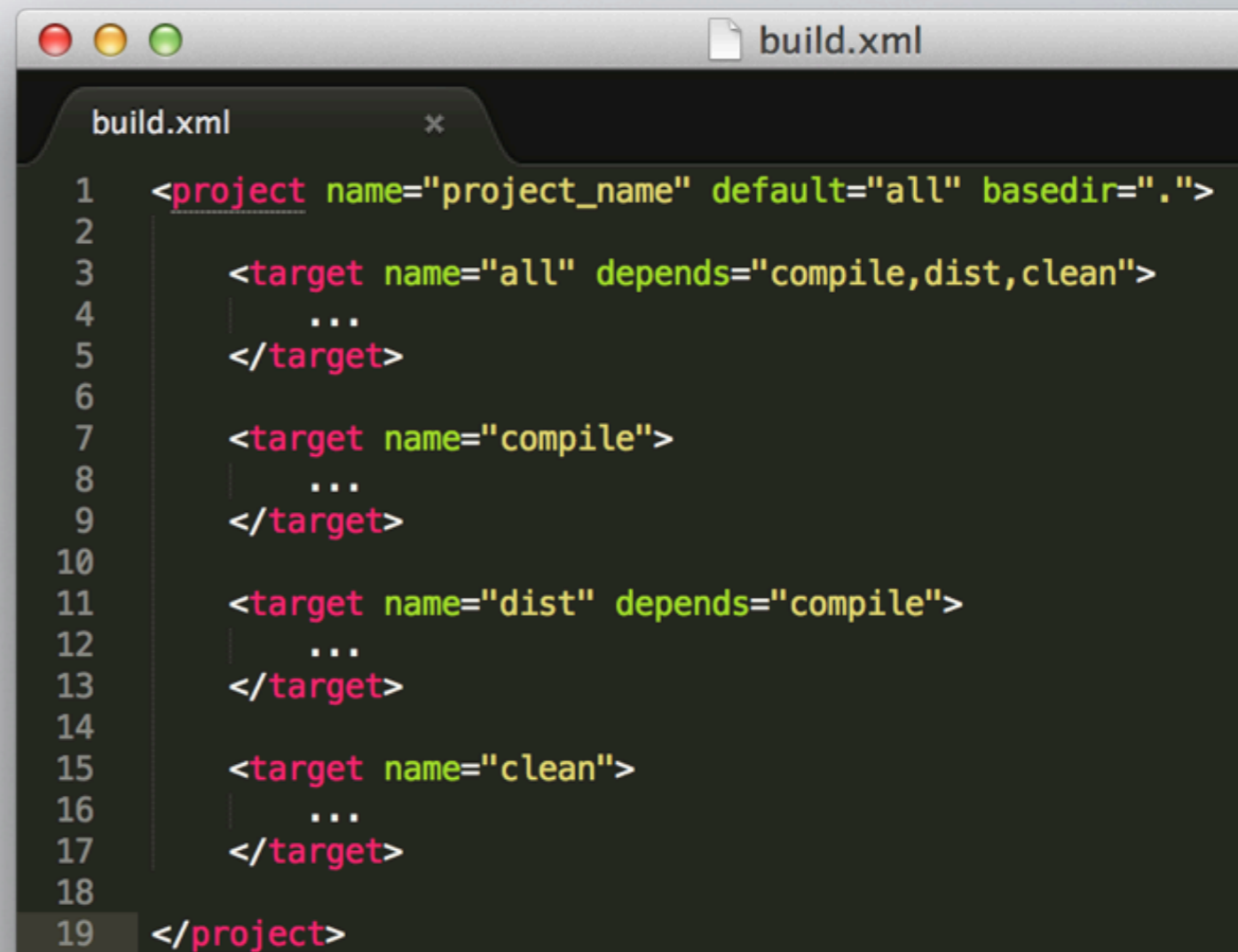
Ant

Ant 사용법

/ 하나의 project 요소

/ project요소는
target요소를 포함

/ target요소는
여러개의 태스크 요소 포함



```
build.xml
1 <project name="project_name" default="all" basedir=".">
2
3     <target name="all" depends="compile,dist,clean">
4         ...
5     </target>
6
7     <target name="compile">
8         ...
9     </target>
10
11    <target name="dist" depends="compile">
12        ...
13    </target>
14
15    <target name="clean">
16        ...
17    </target>
18
19 </project>
```


Ant

Ant 사용법

Target요소가 가질 수 있는 대표 태스크

property	속성지정
mkdir	새로운 디렉터리 생성
copy	파일, 디렉터리 복사
javac	컴파일
jar	jar파일 생성
javadoc	javadoc 생성
delete	파일, 디렉터리 삭제

Ant

| property

/ 이름과 값으로 속성을 지정한다. 속성에서는 대소문자를 구별한다.

```
<property name="foo.src" value="src" />
```

/ 다른 부분에서 이 속성을 참조하려면 "\${foo.src}"와 같이 쓰면 된다.

```
<javac srcdir="${foo.src}" destdir="${build}" />
```

/ 다음과 같이 파일을 읽어 속성을 설정할 수도 있다.

```
<property file="foo.properties" />
```

Ant

| mkdir

/ 새로운 디렉터리를 만들 때 사용되며 다음과 같이 사용한다.

```
<mkdir dir="{dist}" />
```

```
<mkdir dir="{dist}/lib" />
```

Ant

| copy

/ 파일 하나를 복사할 때는 다음과 같이 한다.

```
<copy file="myfile.txt" tofile="mycopy.txt" />
```

/ 디렉토리를 다른 디렉토리로 복사하는 경우는 다음과 같다.

```
<copy todir="../new/dir">  
  <fileset dir="src_dir" />  
</copy>
```

Ant

javac

컴파일 하는데 사용하며, 소스 디렉토리는 재귀적으로 탐색하고 class파일이 없거나 class파일이 java파일보다 오래된 경우에만 컴파일한다.

`${src}`에서 `${build}`로 컴파일하며 클래스 패스에는 a.jar가 포함될 경우 다음과 같다.

```
<javac srcdir="${src}" destdir="${build}"  
    classpath="a.jar" />
```

`${src}`, `${src2}`를 `${build}` 로 컴파일하며, 컴파일할 때 package/p1 과, package/p2에 있는 파일만 사용하고, package/p1/foo는 제외한다.

```
<javac srcdir="${src}:${src2}" destdir="${build}"  
    include="package/p1/**,package/p2/**"  
    exclude="package/p1/foo/**"  
    classpath="a.jar" />
```

Ant

| javac

/ 또는 다음과 같이 표현할 수 있다.

```
<javac destdir="${build}" classpath="a.jar">  
  <src path="${src}" />  
  <src path="${src2}" />  
  <include name="package/p1/**" />  
  <include name="package/p2/**" />  
  <exclude name="package/p1/foo/**" />  
</javac>
```

Ant

└ jar

└ 지정된 파일들을 jar로 묶는다.

└ `${build}/classes` 밑에 있는 파일을 `app.jar`로 묶는다.

```
<jar destfile="${dist}/lib/app.jar"  
    basedir="${build}/classes" />
```

└ `${build}/classes`와 `${src}/resources` 밑에 있는 파일을 `app.jar`로 묶되, `Test.class`는 제외한다.

```
<jar destfile="${dist}/lib/app.jar">  
    <fileset dir="${build}/classes" exclude="**/Test.class" />  
    <fileset dir="${src}/resources" />  
</jar>
```

Ant

| javadoc

/ javadoc 문서를 생산한다.

/ src디렉토리 밑에 있는 소스파일을 읽어 javadoc문서를 생성하여 `${doc}` 디렉토리에 저장한다.

```
<javadoc destdir="${doc}">  
  <fileset dir="${src}"></fileset>  
</javadoc>
```


Ant

delete

/ 하나의 파일 또는 디렉토리와 그 하위 디렉토리를 삭제한다.

/ 파일 하나를 삭제할 때는 다음과 같다.

```
<delete file="/lib/ant.jar" />
```

/ lib디렉토리와 그 하위 디렉토리를 삭제한다.

```
<delete dir="lib" />
```

/ 현재 디렉토리와 그 하위 디렉토리에 확장자가 bak인 파일을 삭제한다.

```
<delete>
```

```
  <fileset dir="." include="**/*.bak" />
```

```
</delete>
```

/ build디렉토리와 그 하위 디렉토리를 삭제한다.

```
<delete includeEmptyDirs="true">
```

```
  <fileset dir="build" />
```

```
</delete>
```

CruiseControl

CruiseControl

| CruiseControl?

/ 배포자 : ThoughtWorks

/ 플랫폼 : Java2

/ 빌드도구 : Ant, Maven, NAnt

/ 버전관리시스템 : ClearCase, CM, CVS, Subversion..

CruiseControl

| CruiseControl?

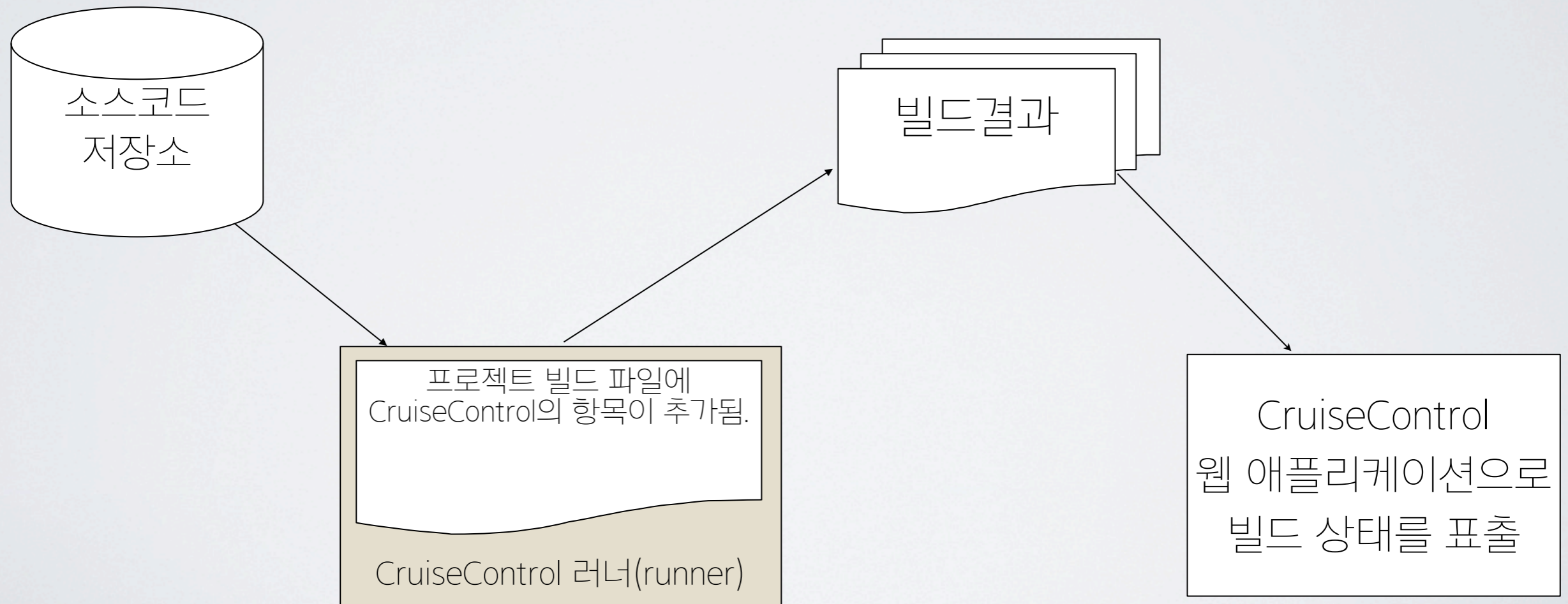
/ 자바용 CI통합서버

/ 백그라운드 Process로 소스 처리,
Java Web Application 기반의 UI제공.

/ 빌드 자동화 시스템,
작업시간의 효율적 사용, 버전관리 가능

CruiseControl

! CruiseControl?

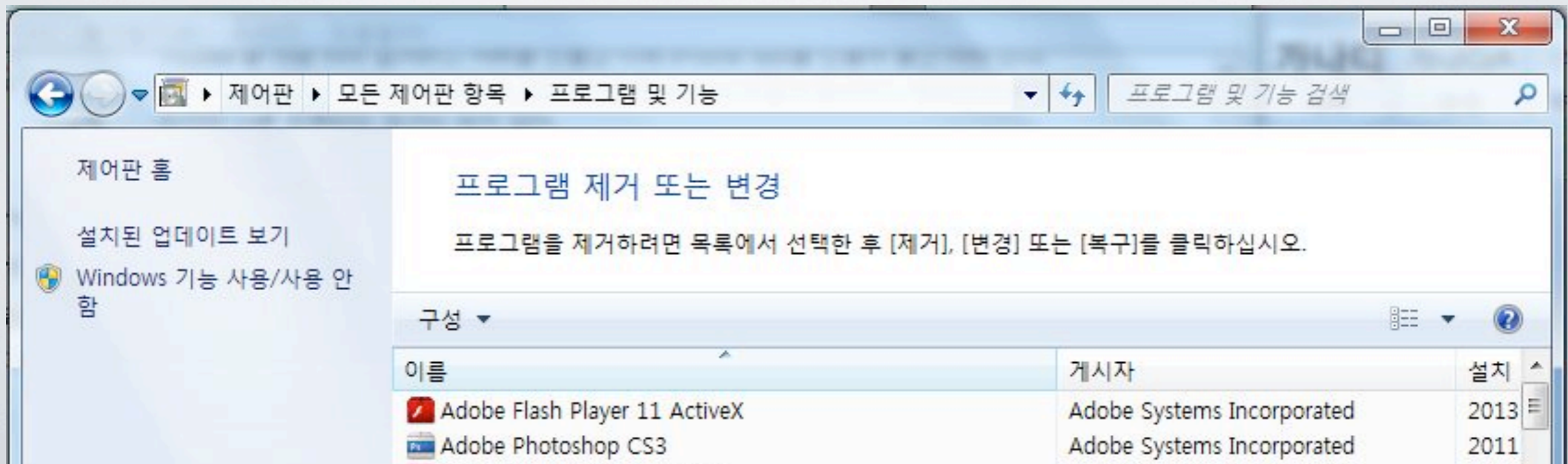


CruiseControl

설치법

/ 제어판 -> 프로그램 및 기능

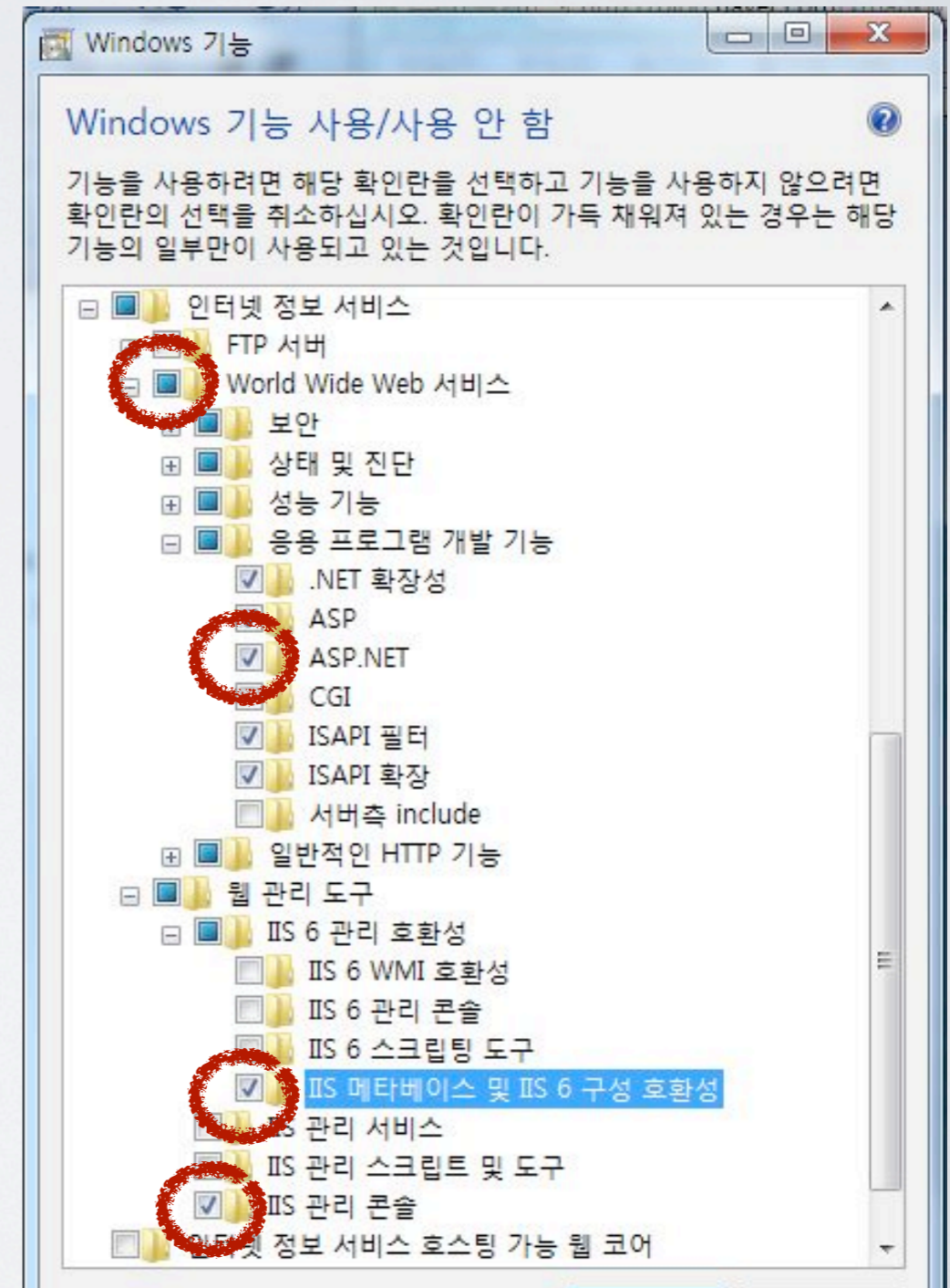
/ Windows 기능 사용/사용 안함



CruiseControl

설치법

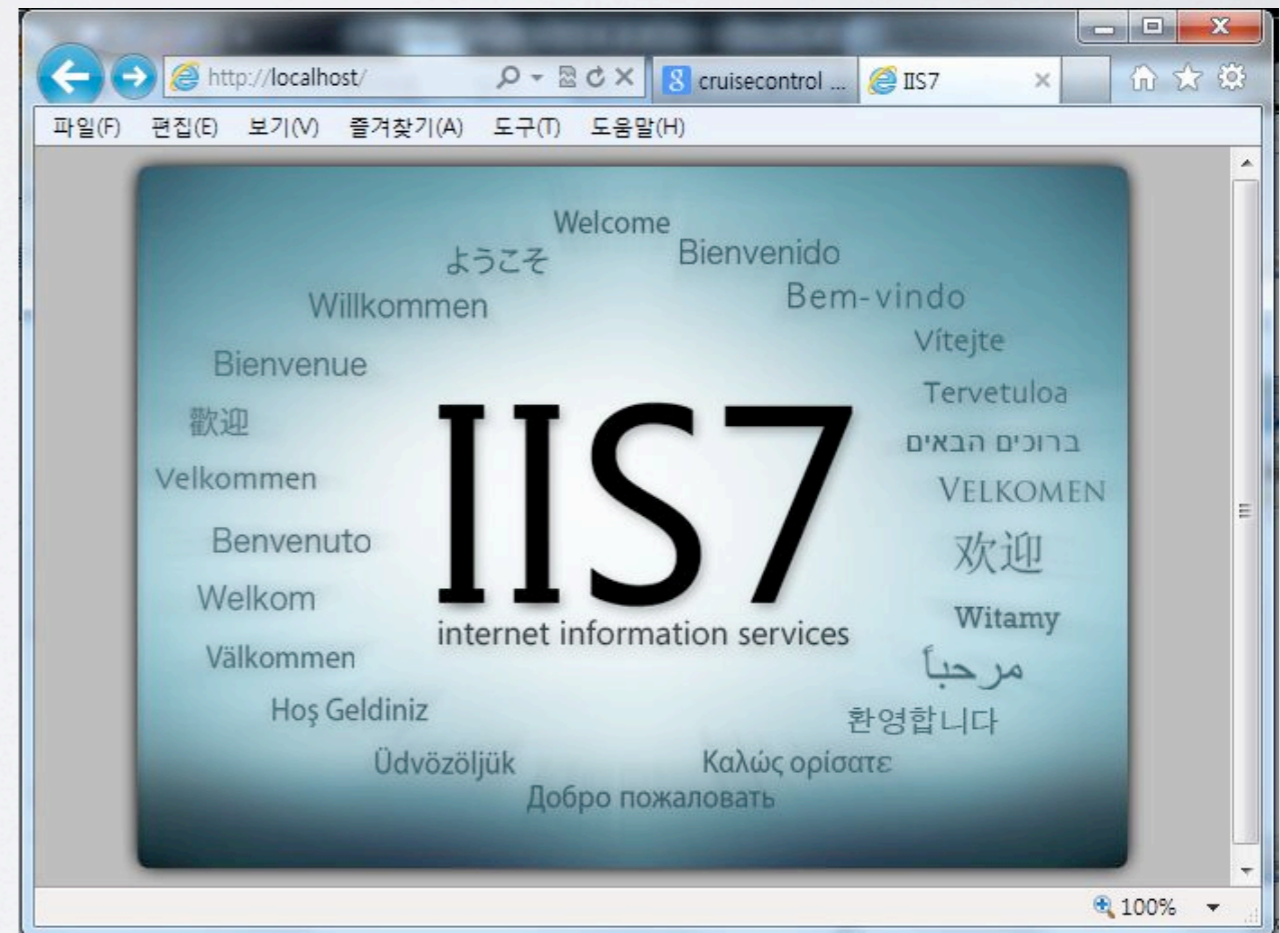
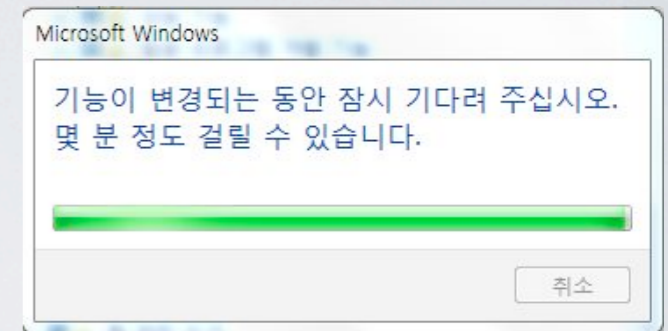
/ 다음과 같이 체크 한후 확인



CruiseControl

| 설치법

/ IIS 설치 완료



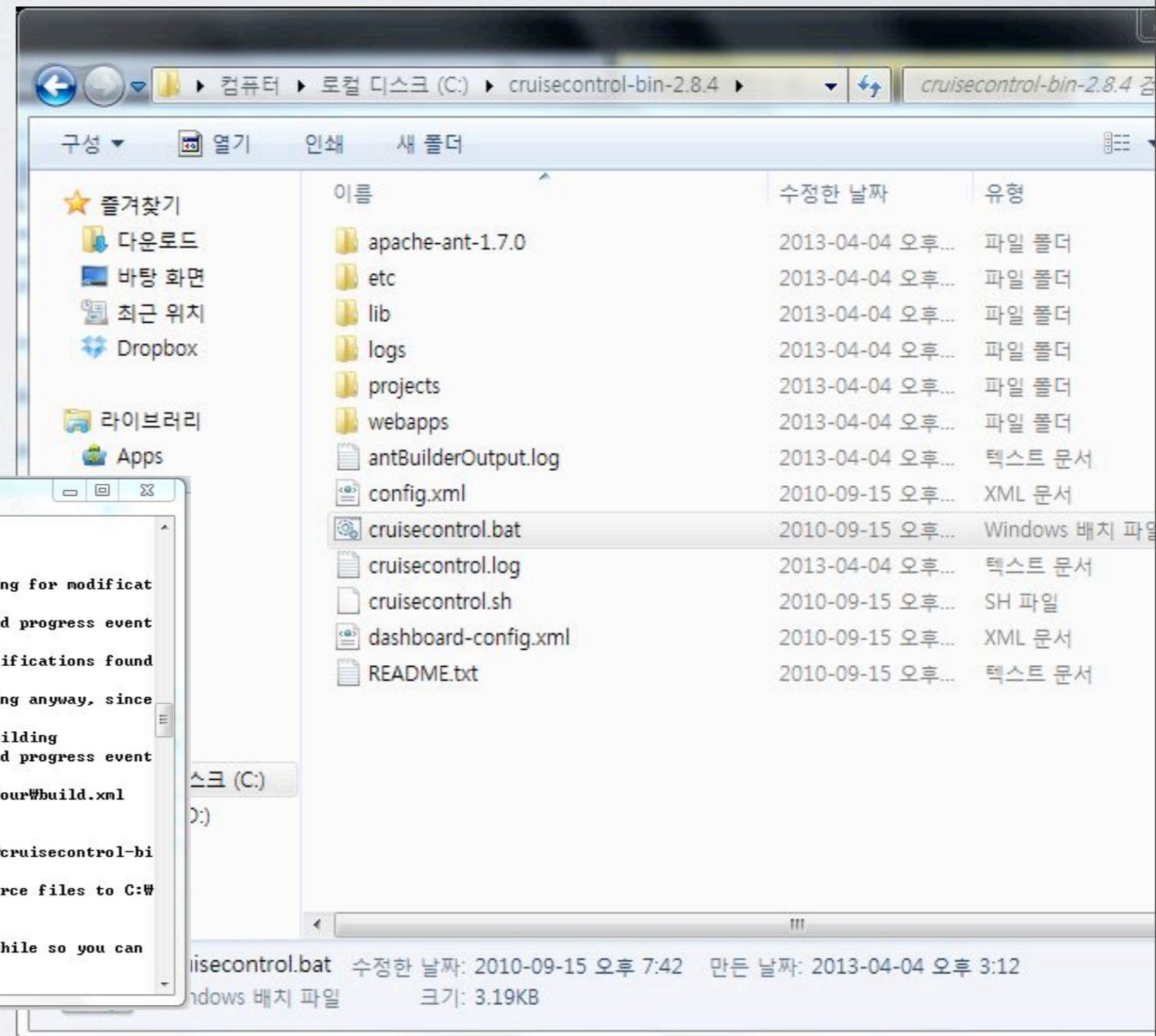
CruiseControl

설치법

/ 압축풀고,

/ cruisecontrol.bat 실행

```
C:\Windows\system32\cmd.exe
[cc14]-04 15:13:42 ScriptRunner - BUILD SUCCESSFUL
[cc14]-04 15:13:42 ScriptRunner - Total time: 0 seconds
[cc14]-04 15:13:42 ntBootstrapper- Bootstrap successful.
[cc14]-04 15:13:42 Project      - Project connectfour: checking for modifications
[cc14]-04 15:13:42 jectController- connectfour Controller: build progress event
: checking for modifications
[cc14]-04 15:13:42 Project      - Project connectfour: No modifications found
, build not necessary.
[cc14]-04 15:13:42 Project      - Project connectfour: Building anyway, since
build was explicitly forced.
[cc14]-04 15:13:42 Project      - Project connectfour: now building
[cc14]-04 15:13:42 jectController- connectfour Controller: build progress event
: now building
[cc14]-04 15:13:42 ScriptRunner - Buildfile: projects\connectfour\build.xml
[cc14]-04 15:13:43 ScriptRunner - ccAntProgress -- clean
[cc14]-04 15:13:43 ScriptRunner - ccAntProgress -- compile
[cc14]-04 15:13:43 ScriptRunner - [mkdir] Created dir: C:\cruisecontrol-bin-2.8.4\projects\connectfour\target\classes
[cc14]-04 15:13:43 ScriptRunner - [javac] Compiling 10 source files to C:\cruisecontrol-bin-2.8.4\projects\connectfour\target\classes
[cc14]-04 15:13:45 ScriptRunner - ccAntProgress -- sleep
[cc14]-04 15:13:45 ScriptRunner - [echo] Sleeping for a while so you can see the build in the new dashboard
```

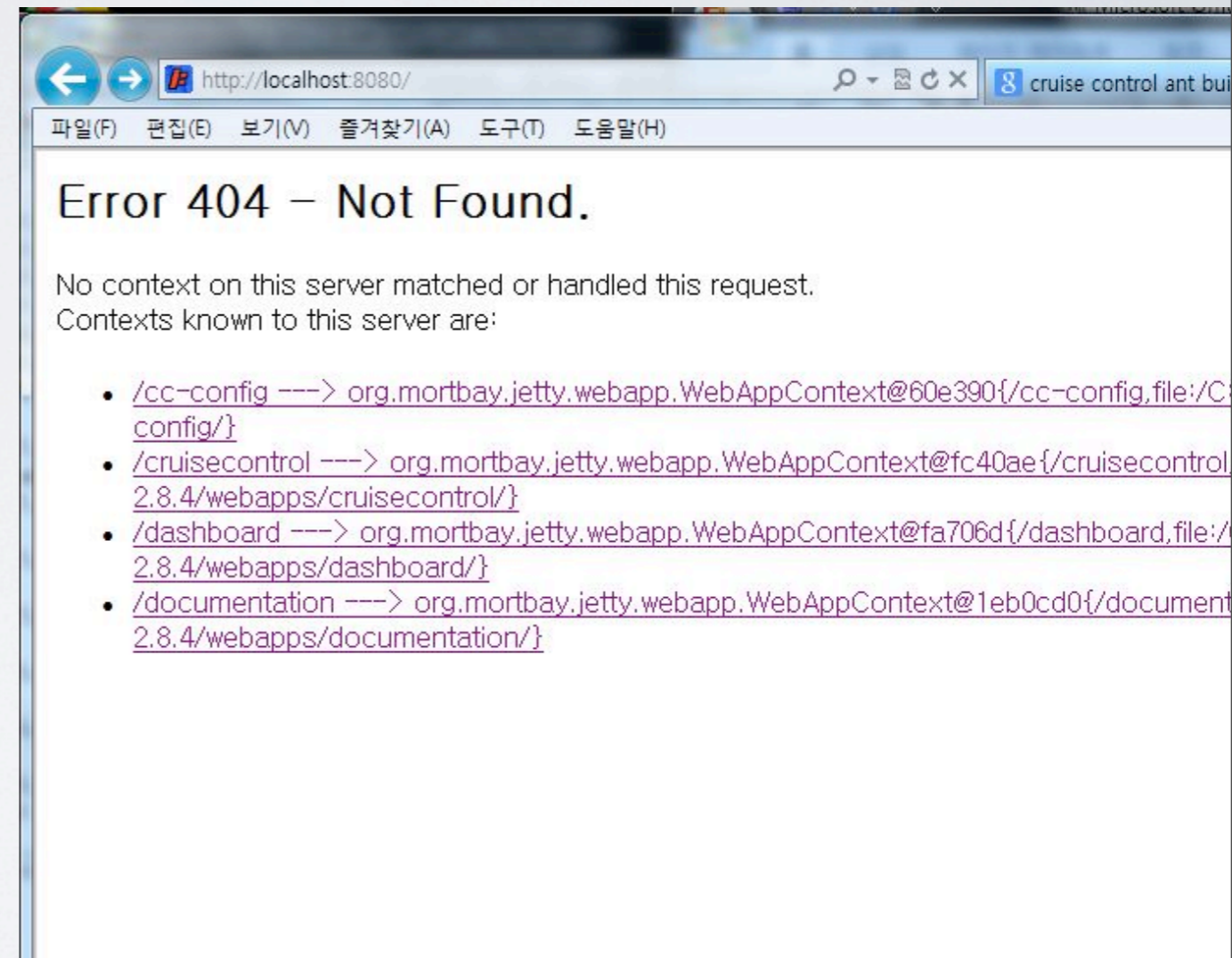


CruiseControl

| 설치법

/ localhost:8080

/ 다음과 같은 환경 뜨면 성공



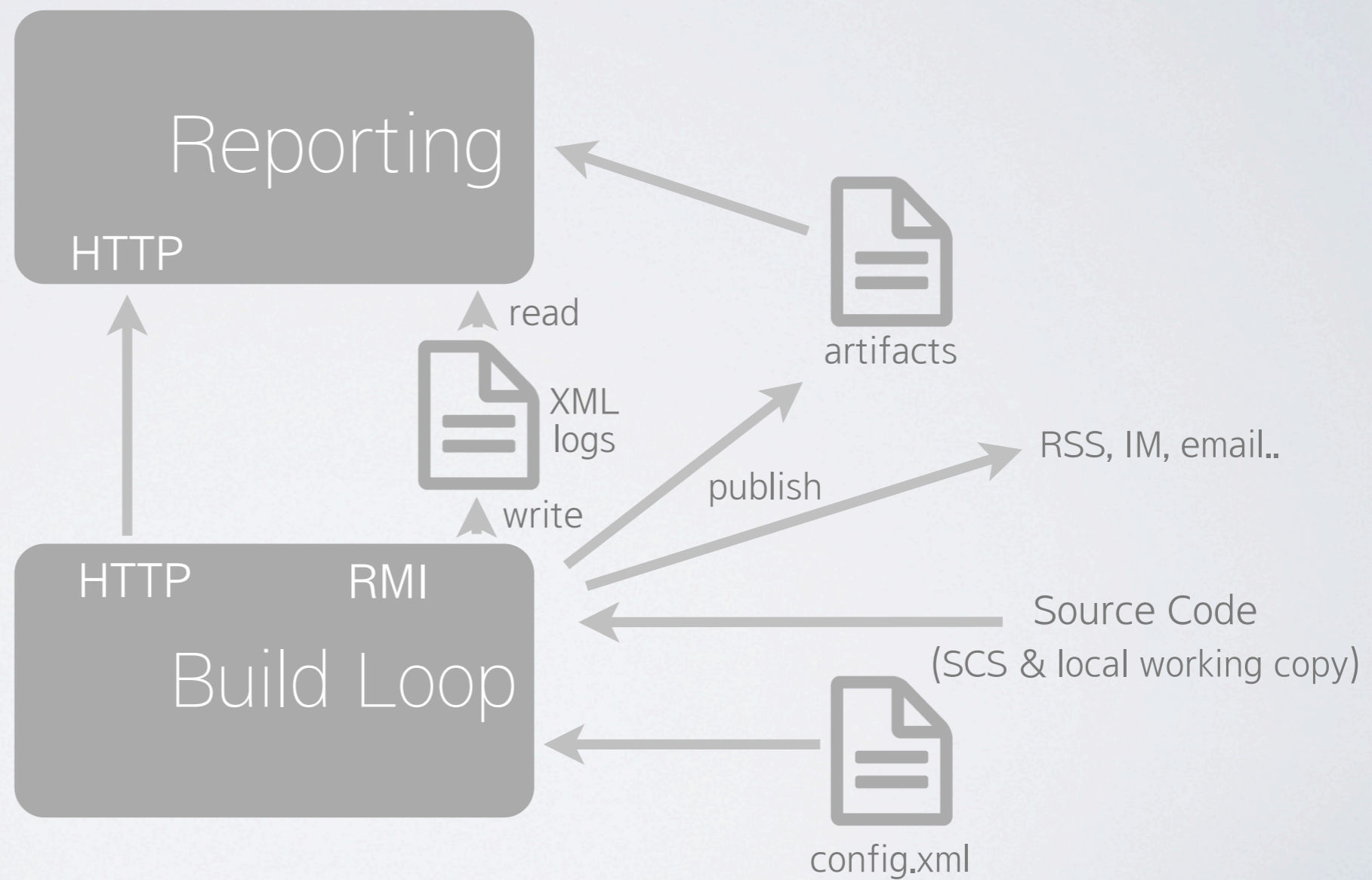
CruiseControl

| Main Modules

/ Build Loop

/ Build Result JSP

/ Dashboard



CruiseControl

| Build Loop

/ 빌드 루프는 정기적으로 사용자의 코드베이스에 대한 변경사항에 대한 소스 제어 도구를 확인, 필요한 경우 구축하고 빌드의 상태를 알린다.

CruiseControl

Build Result JSP

- / CruiseControl Build Loop의 결과를 보여주는 것.
- / 왼쪽은 현재 프로젝트가 빌드되어 있는지 여부에 대한 정보를 알 수 있으며 이전 빌드의 세부 사항에 대한 링크를 제공한다.
- / 오른쪽은 컴파일 오류, 테스트 결과 그리고 마지막 빌드 이후 변경한 내용을 파일에 대한 세부 정보를 포함하여 빌드의 결과를 보여준다.

The screenshot shows the CruiseControl web interface for a project named 'helloworld'. The page is titled 'CruiseControl Build Result' and is accessed via 'localhost:8080/cruisecontrol/buildresults/helloworld'. The interface includes a navigation menu with options like 'Build Results', 'Test Results', 'XML Log File', 'Metrics', 'Config', and 'Control Panel'. The main content area displays the status of the latest build, which is 'BUILD COMPLETE - build.7'. It provides details such as the date of build (2013-04-04T09:41:50), time to build (0 seconds), and last changed time (2013-04-04T09:41:20). There is also a section for 'Errors/Warnings' showing one warning: 'warning: 'includeantruntime' was not set, defaulting to build.sysclasspath=last; set to false for repeatable builds'. The 'Unit Tests' section shows 'No Tests Run'. The 'Modifications since last successful build' section shows a change by 'User' to 'projects/helloworld/DS_Store' on '2013-04-04T09:41:20'. On the left side, there is a 'Latest Build' table listing previous builds from build.1 to build.7 with their respective timestamps.

Latest Build
2013/04/04 18:41:50 (build.7)
2013/04/04 18:39:14 (build.6)
2013/04/04 18:38:42 (build.5)
2013/04/04 18:36:46 (build.4)
2013/04/04 18:36:02 (build.3)
2013/04/04 18:34:54 (build.2)
2013/04/04 18:34:14 (build.1)

CruiseControl

Dashboard

/ 프로젝트 상태를 시각화 해서 보여주는 도구이다. 빌드 결과는 프로젝트의 현재 상태를 아이콘으로 보여준다.

The screenshot displays the CruiseControl Dashboard interface. At the top, it shows the server address: **Dashboard Server : 172.30.93.123**. The main content area features a 'Dashboard' tab and a 'Builds' tab. A project card for 'Project: helloworld' is visible, showing a green checkmark icon, the server address '172.30.93.123', and a status of 'Waiting'. Below this, it indicates the last build result was 'Passed' and the last build date was 'about 1 hour ago'. A link to 'Open Build' is provided. On the right side, a 'Summary' section provides an overview: 1 project build(s) total, with 0 failed, 1 succeeded, 0 building, 1 discontinued, and 0 inactive. It also states '100% of projects passing'. Below the summary, a 'Tools' section lists links for 'RSS Feed', 'for CCTray', and 'for CC-Config'. A smaller inset window at the bottom shows a 'Builds' tab with a project card for 'Project: cceqa-config-manager' on server '10.18.2.3', with a status of 'Queued' and a last build date of '7 days ago'.

CruiseControl

Dashboard



이것은 24시간 성공적으로 빌드되어진 것을 가르키며 현재는 빌드되어지기 위해 대기큐에 있다는 것을 의미한다.



이것은 프로젝트 빌드가 24시간 미만동안 성공적이었다는 것을 의미한다.



이것은 프로젝트 빌드가 24시간 이상내에 실패했었다는 것을 의미한다.



이것은 프로젝트 빌드가 24시간 미만내에 실패했었으며 현재는 멈춰져있다는 것을 의미한다.



이것은 이 프로젝트가 지난번 실행시에 통과했고 현재는 빌드중이라는 것을 의미한다.



이것은 중지 혹은 프로젝트와의 연결이 끊겼다는 것을 의미한다.

CruiseControl

Dashboard - Builds

/ 각 프로젝트 빌드 및 상태를 볼 수 있다.

Dashboard Server : 172.30.93.123

Dashboard Builds Administration

Summary

- 0 project build(s) failed
- 1 project build(s) succeeded
- 0 project(s) building
- 1 project(s) discontinued
- 0 project(s) inactive

100% of projects passing

Tools

- RSS Feed
- for CCTray
- for CC-Config

hell last passed 5 minutes ago

helloworld passed less than a minute ago

- cce-windows** passed about 3 hours ago
- cceqa-config-manager** passed 7 days ago
- cceqa-source-dist-windows** passed about 3 hours ago
Elapsed 00:01:38 Remaining 00:05:15
- cceqa-windows-ie6** failed about 3 hours ago

CruiseControl

Dashboard - Build Detail

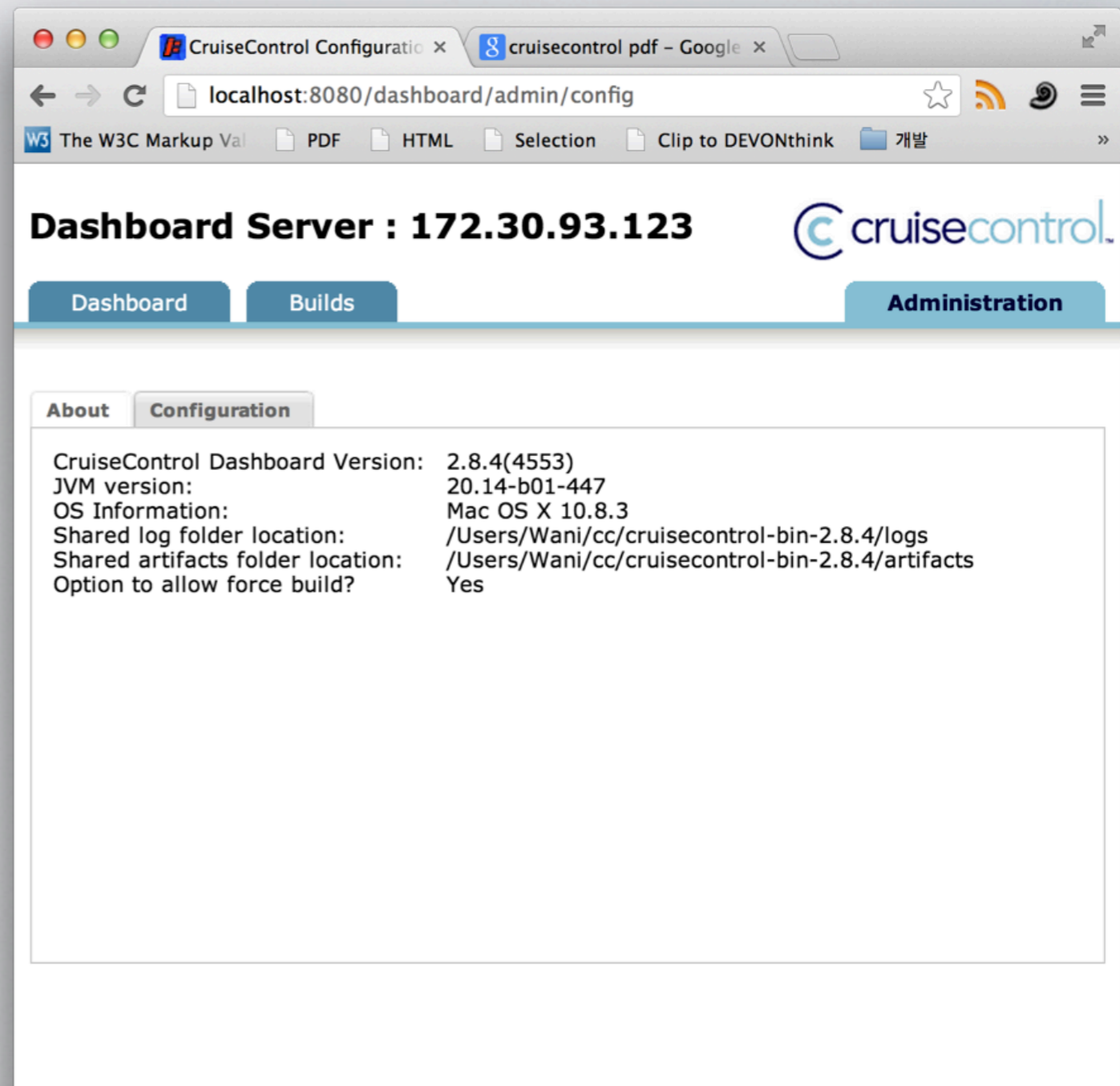
/ 마지막으로 성공한 빌드,
빌드 번호 등의 필드 정보
를 보여준다.

The screenshot displays the CruiseControl web interface. At the top, the browser address bar shows the URL `localhost:8080/dashboard/tab/build/detail/helloworld`. The page header includes the text "Dashboard Server : 172.30.93.123" and the CruiseControl logo. Below the header, there are navigation tabs for "Dashboard" and "Builds", with "Builds" being the active tab. A prominent green notification box states "helloworld passed (2 minutes ago)" with a green checkmark icon. Below this, it provides details: "Build Time: 4 Apr 2013 18:39 GMT +09:00", "Duration: 0 seconds", and "Build: build.6". To the right, a "Latest Builds" sidebar lists several previous builds, each with a green checkmark and a timestamp (e.g., "2 minutes ago build.6", "2 minutes ago build.5", etc.). Below the main notification, there are tabs for "Artifacts", "Modifications", "Build Log", "Tests", and "Errors and Warnings". The "Artifacts" tab is selected, showing the text "No artifacts found." in a large, empty container.

CruiseControl

Dashboard - Administration

이 페이지에선
Configuration 에 대한 정
보를 확인할 수 있다.



The screenshot shows a web browser window displaying the CruiseControl Dashboard Administration page. The browser's address bar shows the URL `localhost:8080/dashboard/admin/config`. The page header includes the text "Dashboard Server : 172.30.93.123" and the CruiseControl logo. Below the header, there are three tabs: "Dashboard", "Builds", and "Administration", with "Administration" being the active tab. Under the "Administration" tab, there are two sub-tabs: "About" and "Configuration", with "Configuration" being the active sub-tab. The "Configuration" sub-tab displays the following information:

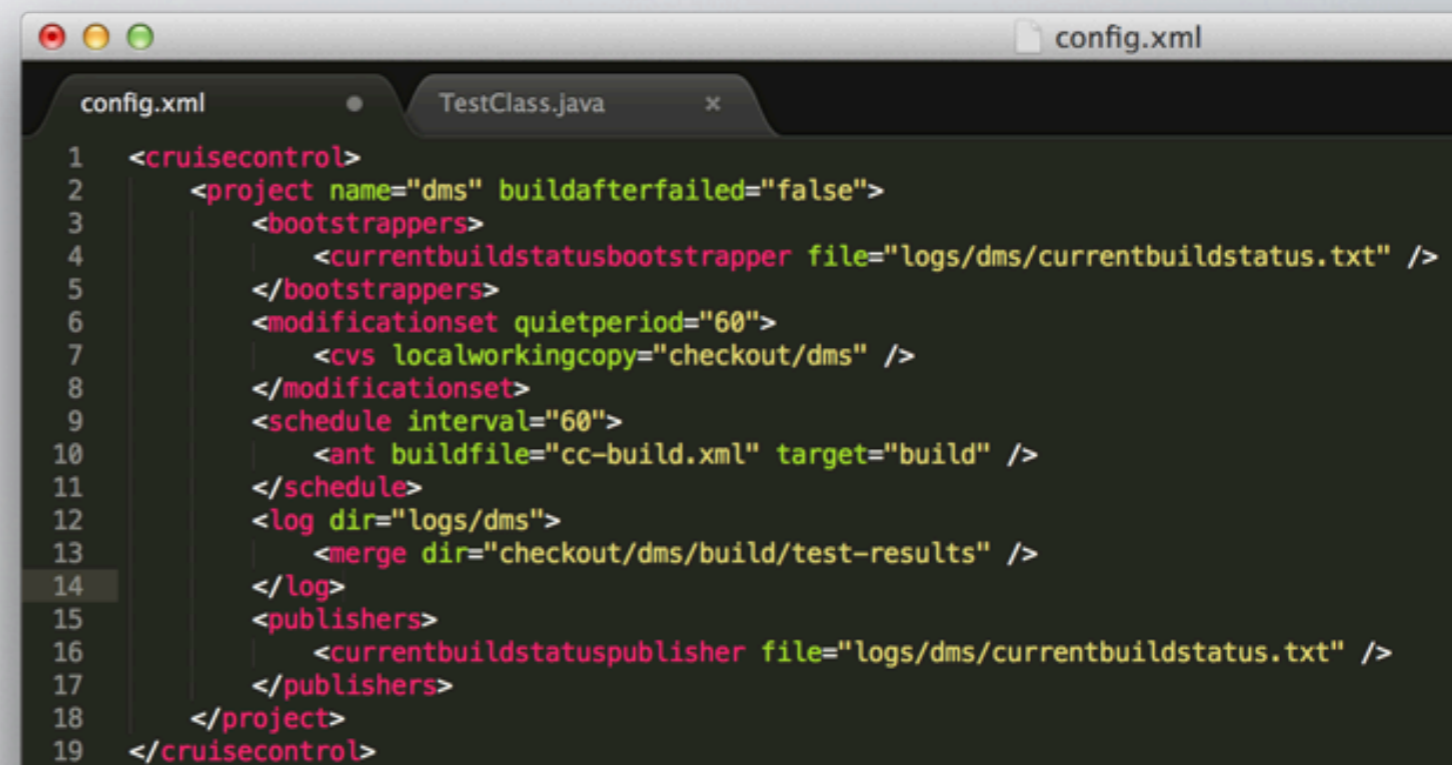
CruiseControl Dashboard Version:	2.8.4(4553)
JVM version:	20.14-b01-447
OS Information:	Mac OS X 10.8.3
Shared log folder location:	/Users/Wani/cc/cruisecontrol-bin-2.8.4/logs
Shared artifacts folder location:	/Users/Wani/cc/cruisecontrol-bin-2.8.4/artifacts
Option to allow force build?	Yes

CruiseControl

| config.xml

/ CruiseControl Project를
관리하는 파일

/ <cruisecontrol> 아래에
<project>들이 위치한다.



```
1 <cruisecontrol>
2   <project name="dms" buildafterfailed="false">
3     <bootstrappers>
4       <currentbuildstatusbootstrapper file="logs/dms/currentbuildstatus.txt" />
5     </bootstrappers>
6     <modificationset quietperiod="60">
7       <cvss localworkingcopy="checkout/dms" />
8     </modificationset>
9     <schedule interval="60">
10      <ant buildfile="cc-build.xml" target="build" />
11    </schedule>
12    <log dir="logs/dms">
13      <merge dir="checkout/dms/build/test-results" />
14    </log>
15    <publishers>
16      <currentbuildstatuspublisher file="logs/dms/currentbuildstatus.txt" />
17    </publishers>
18  </project>
19 </cruisecontrol>
```

CruiseControl

| config.xml

/ <bootstrappers>

/ 빌드가 시작되기전에 하는 과정.

```
<bootstrappers>  
  <currentbuildstatusbootstrapper file="logs/dms/currentbuildstatus.txt" />  
</bootstrappers>  
<modificationset quietperiod="60">
```

CruiseControl

| config.xml

/ <modificationset>

/ 일정시간(quietperiod)마다 변화된걸 확인.

```
</bootstrappers>  
<modificationset quietperiod="60">  
  <cv s localworkingcopy="checkout/dms" />  
</modificationset>  
<schedule interval="60">
```

CruiseControl

┆ config.xml

┆ <schedule>

┆ 마지막으로 빌드를 실행하는 시간간격을 정의하고, 빌드를 어떻게 시도해야 하는지를 정의한다.

┆ 여기서는 interval시간마다 modificationset에 지정되어있는 변경사항을 확인하고 변경사항이 발생했을 때 크루즈 컨트롤이 빌드를 시도하게 된다.

```
</modificationset>  
<schedule interval="60">  
  <ant buildfile="cc-build.xml" target="build" />  
</schedule>  
<log dir="logs/dms">
```

CruiseControl

| config.xml

| <log>

| 빌드를 실행할 때마다 로그를 저장하는 것을 정의, 기존의 로그 파일과 병합되어야 해서 <merge>요소를 사용하였다.

```
</schedule>  
<log dir="logs/dms">  
  <merge dir="checkout/dms/build/test-results" />  
</log>  
<publishers>
```

CruiseControl

| config.xml

/ <publishers>

/ 빌드 과정이 종료된 다음에 보고할 사항들을 정해줄 수 있다. 여기에는 단순히 <currentbuildstatuspublisher>가 빌드 과정이 끝났다는 메시지를 current..에 기록하는 일을 한다.

```
</log>  
<publishers>  
  <currentbuildstatuspublisher file="logs/dms/currentbuildstatus.txt" />  
</publishers>  
project>
```


CruiseControl

config.xml

http://

cruisecontrol.sourceforge.net/
main/configxml.html

The screenshot shows a web browser window displaying the 'CruiseControl Configuration Reference' page. The browser's address bar shows the URL 'cruisecontrol.sourceforge.net/main/configxml.html'. The page features a navigation sidebar on the left with links for 'home', 'download', 'getting started', 'documentation' (with 'config.xml' selected), 'faq', 'wiki', 'contributing', 'developers', 'mailing lists', 'source repository', and 'license'. Below the sidebar, the 'Release: 2.8.4' is noted. The main content area is titled 'Configuration Reference' and contains an introductory paragraph about XML configuration files. Below this, there are two sections: 'Hierarchical Index' and 'Alphabetical Index', each containing a list of XML tags with links to their respective documentation pages.

Configuration Reference

CruiseControl configuration files are written in XML. This document describes the XML elements and attributes for a valid configuration file.

The use of [plugins](#) means that other elements not documented here can also be used in the configuration. At a minimum, though, the config file contains a single top level `<cruisecontrol>` element, with one or more child `<project>` elements.

Hierarchical Index

- [<cruisecontrol>](#)
- [<property/>](#)
- [<dashboard/>](#)
- [<include.projects/>](#)
- `<system>`
 - `<configuration>`
 - [<threads/>](#)
 - `</configuration>`
- `</system>`
- [<plugin/>](#)
- [<project>](#)
 - [<property/>](#)
 - [<plugin/>](#)
 - [<cvslabelincrementer>](#)
 - [<emptylabelincrementer>](#)
 - [<formattedlabelincrementer>](#)
 - [<labelincrementer/>](#)
 - [<pi4changelistlabelincrementer>](#)
 - [<propertyfilelabelincrementer>](#)
 - [<svnlabelincrementer>](#)
 - [<listeners>](#)
 - [<cmsynergysessionmonitor/>](#)
 - [<currentbuildstatusftp listener/>](#)
 - [<currentbuildstatus listener/>](#)
 - [<currentbuildstatuspagelister/>](#)
 - [<lockfile listener/>](#)
 - `</listeners>`
 - [<bootstrappers>](#)
 - [<accurevbootstrapper/>](#)
 - [<alienbrainbootstrapper/>](#)
 - [<antbootstrapper/>](#)
 - [<clearcasebootstrapper/>](#)
 - [<clearcaseviewstrapper/>](#)
 - [<cmsynergybootstrapper/>](#)
 - [<cmsynergybaselinebootstrapper/>](#)
 - [<cvsgitbootstrapper/>](#)
 - [<darcsbootstrapper/>](#)
 - [<execbootstrapper/>](#)
 - [<gitbootstrapper/>](#)
 - [<harvestbootstrapper/>](#)
 - [<lockfilebootstrapper/>](#)
 - [<mercurialbootstrapper/>](#)
 - [<pi4bootstrapper/>](#)
 - [<plasticscmbootstrapper/>](#)
 - [<snapshotcmbootstrapper/>](#)
 - [<starteambootstrapper/>](#)

Alphabetical Index

- [<accurev>](#)
- [<accurevbootstrapper>](#)
- [<alienbrain>](#)
- [<alienbrainbootstrapper>](#)
- [<alwaysbuild>](#)
- [<ant>](#)
- [<antbootstrapper>](#)
- [<antpublisher>](#)
- [<artifactspublisher>](#)
- [<bootstrappers>](#)
- [<composite>](#)
- [<buildstatus>](#)
- [<clearcase>](#)
- [<clearcasebaselinepublisher>](#)
- [<clearcasebootstrapper>](#)
- [<clearcaseviewstrapper>](#)
- [<cmsynergy>](#)
- [<cmsynergybaselinepublisher>](#)
- [<cmsynergybaselinebootstrapper>](#)
- [<cmsynergybootstrapper>](#)
- [<cmsynergysessionmonitor>](#)
- [<cmsynergystaskpublisher>](#)
- [<compound>](#)
- [<compoundpublisher>](#)
- [<cruisecontrol>](#)
- [<currentbuildstatusftp listener>](#)
- [<currentbuildstatus listener>](#)
- [<currentbuildstatuspagelister>](#)
- [<cvsgit>](#)
- [<cvsgitbootstrapper>](#)
- [<cvslabelincrementer>](#)
- [<darcs>](#)
- [<darcsbootstrapper>](#)
- [<dashboard>](#)
- [<delete>](#)
- [<deleteartifacts>](#)
- [<email>](#)
- [<emptylabelincrementer>](#)
- [<exec>](#)
- [<execbootstrapper>](#)
- [<execute>](#)
- [<filesystem>](#)
- [<forceonly>](#)
- [<formattedlabelincrementer>](#)
- [<ftppublisher>](#)
- [<git>](#)

exit;_