

CTIP 환경 구축 및 Build

T2

200511318 김희재

200511326 박현진

200711436 서영주

200913987 이승효

Contents.

1. CTIP Environment

2. Setting

- ◆ Tomcat
- ◆ Hudson

3. Build Setting

- ◆ Hudson
 - Build
 - Project (Ant, SVN)
- ◆ Eclipse
 - Create Project
 - Subversion

4. Build

5. Failure Cases

Contents.

1. CTIP Environment

2. Setting

- ◆ Tomcat
- ◆ Hudson

3. Build Setting

- ◆ **Hudson**
 - Build
 - Project (Ant, SVN)
- ◆ **Eclipse**
 - Create Project
 - Subversion

4. Build

5. Failure Cases

CTIP

◆ *Continuous Integration(CI)*

- 작업을 지속적으로 통합하는 소프트웨어 개발 방법
- 지속적인 통합을 통해 문제의 조기 발견
- 수동적인 반복 작업 감소
- 언제든지 배포 가능한 소프트웨어
- 높은 프로젝트 가시성
- 소프트웨어에 대한 개발자의 자신감 향상

◆ *CTIP(Continuous Test & Integration Platform)*

- CI 서버를 통한 지속적 통합 및 빌드
- 품질 도구들을 통한 코드 품질 검토(테스트 및 정적 분석)
- 빌드 결과의 배포 및 관련자에게 통보(Feedback)

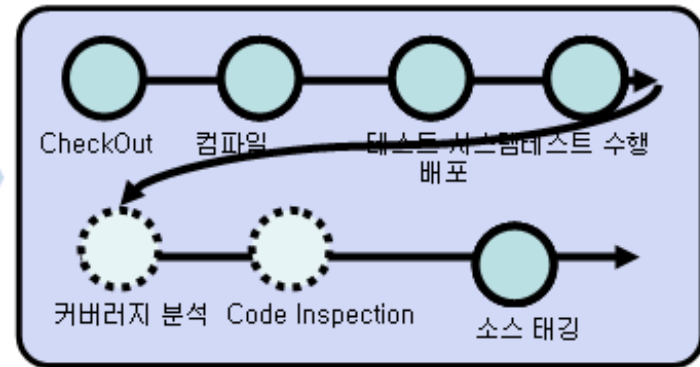
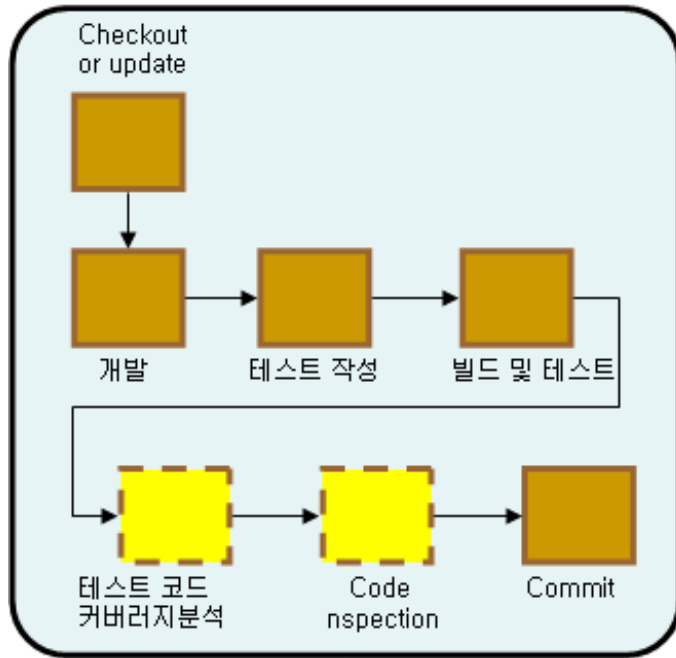
CTIP



개발자



Continuous
Integration
Tools



테스트
결과



커버리지
분석 결과



Code
Inspection
결과

Code Fix
Test 보강



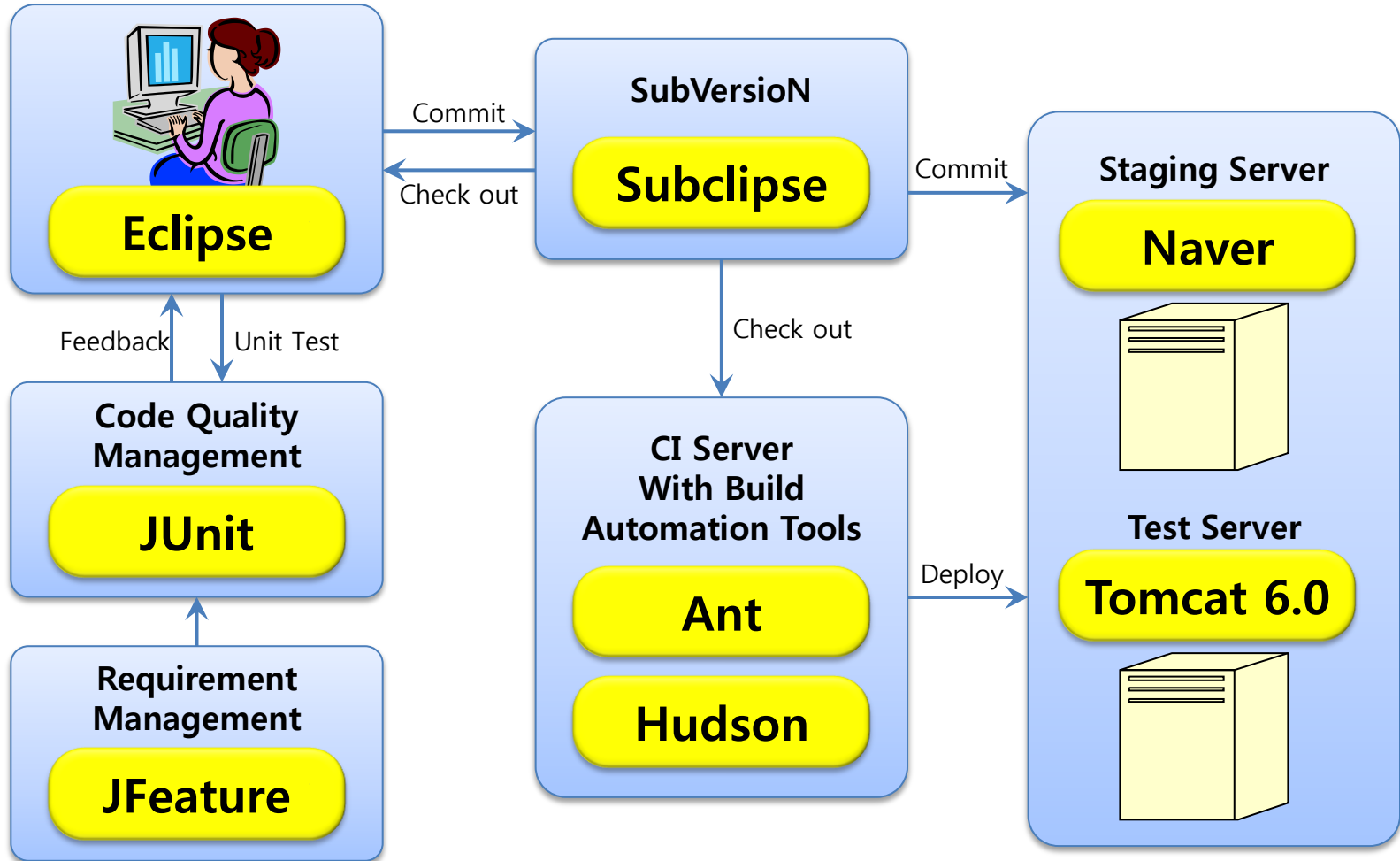
빌드 또는
테스트 실패

이전 성공 버전으로 Revert

CTIP

요건	활동	도구
소스 저장소	소프트웨어 형상관리(SCM)	Subversion, CVS(Current Version System)
빌드 자동화	자동화 빌드	Ant, Maven
테스트 가능한 빌드	테스트, 테스트 커버리지 측정, 테스트 추적성	Xunit Series, TestNG, Corbetura, EMMA, Jfeature, OSRMT
빠른 빌드 수행	단계적 빌드 수행	Ant, Maven
최신 결과물에 대한 쉬운 접근	CI 서버를 통한 결과물 레포트	CruiseControl, Hudson
현 빌드 상태에 대한 쉬운 접근	CI 서버를 이용한 빌드 상태 모니터링	CruiseControl, Hudson

CTIP Environment



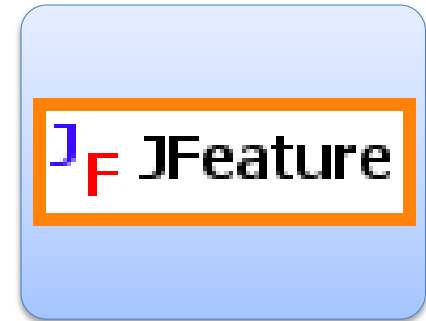
Tools of CTIP



Develop Tool



Unit Testing Tool



RE Tool



CM Tool



Build Automation



CI Server



WAS
(Web Application Server)

Contents.

1. CTIP Environment

2. Setting

- ◆ Tomcat
- ◆ Hudson

3. Build Setting

- ◆ Hudson
 - Build
 - Project (Ant, SVN)
- ◆ Eclipse
 - Create Project
 - Subversion

4. Build

5. Failure Cases

Tomcat Setting


http://tomcat.apache.org/

konkuk.ac.kr


Apache Tomcat - Welcome!

파일(F) 편집(E) 보기(V) 즐겨찾기(A) 도구(T) 도움말(H)

페이지(P) 안전(S) 도구(O) ?



Apache Tomcat



http://www.apache.org/

Search the Site Search Site

Apache Tomcat

- [Home](#)
- [Taglibs](#)
- [Maven Plugin](#)

Download

Tomcat 7.0

- [Tomcat 7.0](#)
- [Tomcat 6.0](#)
- [Tomcat 5.5](#)
- [Tomcat Connectors](#)
- [Tomcat Native](#)
- [Archives](#)

Documentation

- [Tomcat 7.0](#)
- [Tomcat 6.0](#)
- [Tomcat 5.5](#)
- [Tomcat Connectors](#)
- [Tomcat Native](#)

Apache Tomcat

Apache Tomcat is an open source software implementation of the Java Servlet and JavaServer Pages technologies. The Java Servlet and JavaServer Pages specifications are developed under the [Java Community Process](#).

Apache Tomcat is developed in an open and participatory environment and released under the [Apache License version 2](#). Apache Tomcat is intended to be a collaboration of the best-of-breed developers from around the world. We invite you to participate in this open development project. To learn more about getting involved, [click here](#).

Apache Tomcat powers numerous large-scale, mission-critical web applications across a diverse range of industries and organizations. Some of these users and their stories are listed on the [PoweredBy](#) wiki page.

Apache Tomcat, Tomcat, Apache, the Apache feather, and the Apache Tomcat project logo are trademarks of the Apache Software Foundation.

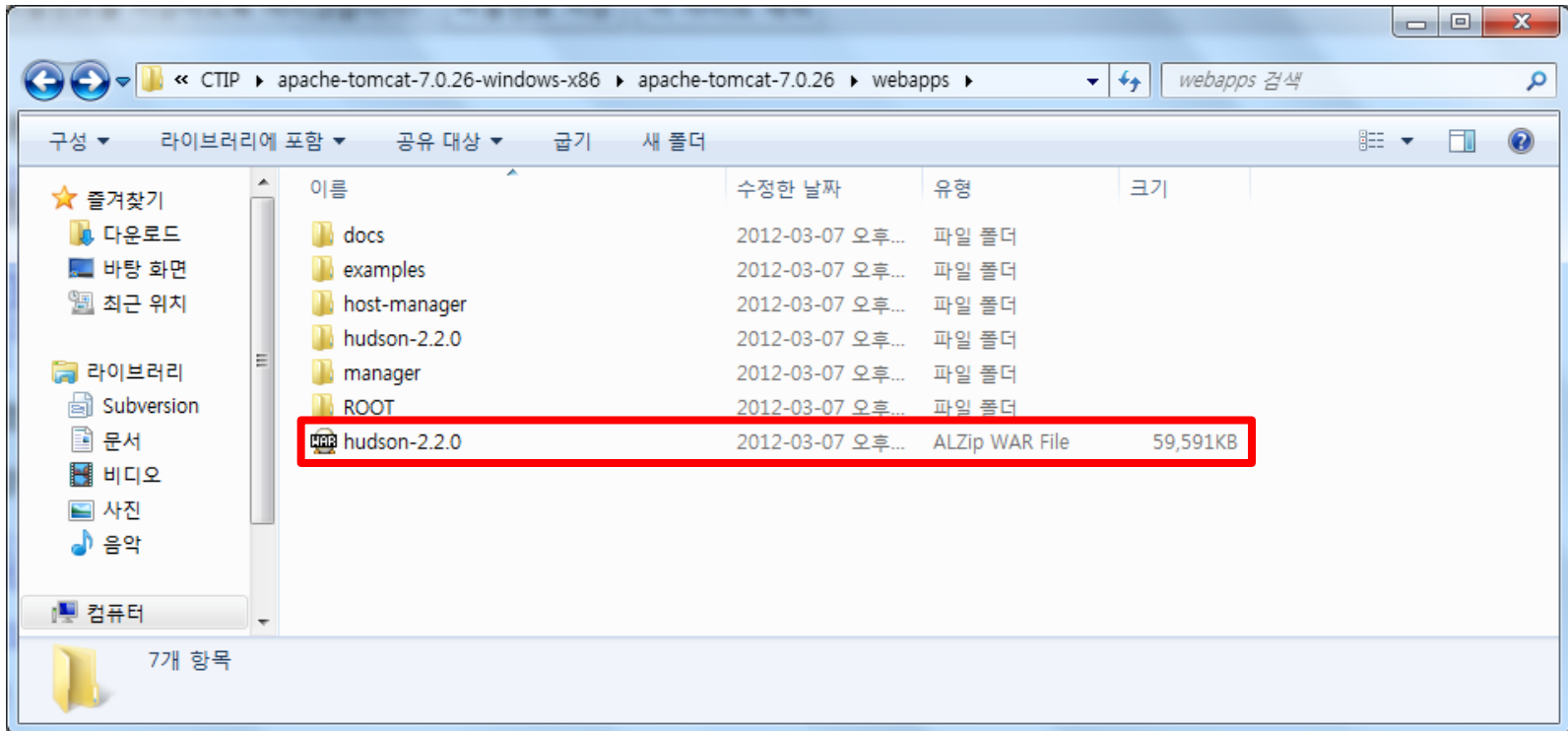
Tomcat 7.0.27 Released

2012-04-05

The Apache Tomcat Project is proud to announce the release of version 7.0.27 of Apache Tomcat. This release includes significant new features as well as a number of bug fixes compared to version 7.0.26. The notable changes include:

- Support for the WebSocket protocol (RFC6455). Both streaming and message based APIs are provided and the implementation currently fully passes the Autobahn test suite. Also included are several examples.
- A number of fixes to the HTTP NIO connector, particularly when using Comet.
- Improve the memory leak prevention and detection code so that it works well with JVMs from IBM.

Hudson Setting



Tomcat 내부의 webapps 폴더 내부에 압축파일(WAR) 을 넣어주면 된다.

Hudson Setting

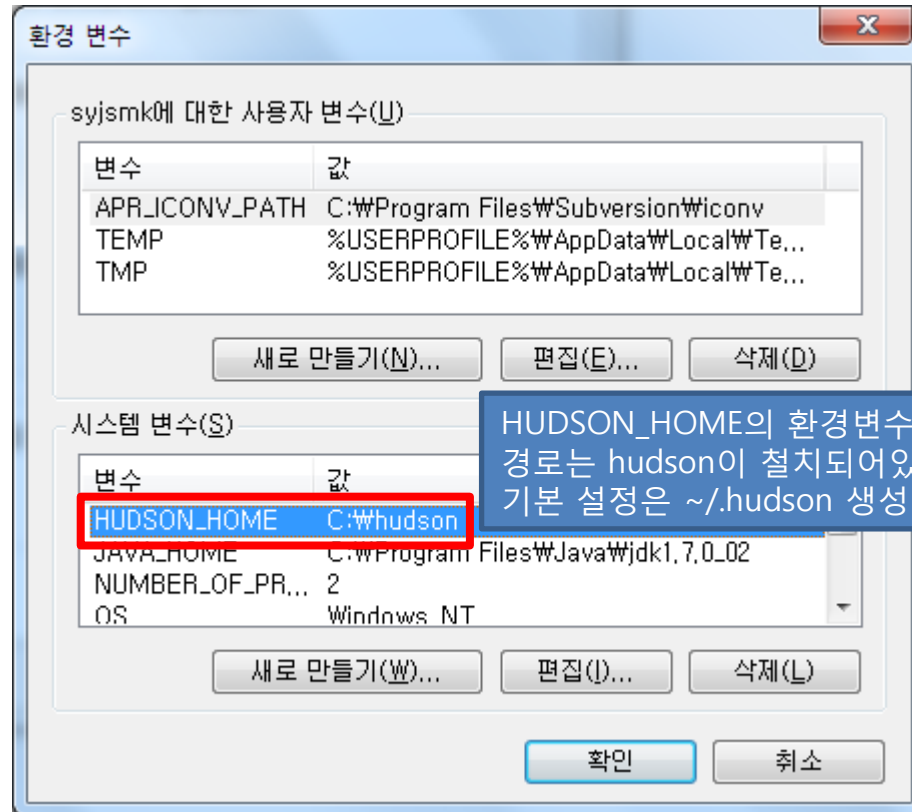
The image shows a Windows Explorer window displaying the file structure of an Apache Tomcat installation. The path is `Software val > CTIP > apache-tomcat-7.0.26-windows-x86 > apache-tomcat-7.0.26 > bin`. A table of files is shown, with the `startup` file highlighted in red. A blue callout box points to it with the text: "Startup 배치 파일을 실행시키면 Tomcat에서 Hudson을 설치한다,".

Below the Explorer window, a Tomcat console window is shown with the following log output:

```
Tomcat
정보: Completed initialization
4월 22, 2012 3:58:09 오후 hudson.TcpSlaveAgentListener <init>
정보: JNLP slave agent listener started on TCP port 5038
4월 22, 2012 3:58:21 오후 com.sun.jersey.server.impl.application.WebApplicationI
mpl_initiate
정보: Initiating Jersey application, version 'Jersey: 1.5 01/14/2011 12:36 PM'
4월 22, 2012 3:58:23 오후 com.sun.jersey.server.impl.application.WebApplicationI
mpl_initiate
정보: Initiating Jersey application, version 'Jersey: 1.5 01/14/2011 12:36 PM'
4월 22, 2012 3:58:23 오후 com.sun.jersey.server.impl.application.WebApplicationI
mpl_initiate
정보: Initiating Jersey application, version 'Jersey: 1.5 01/14/2011 12:36 PM'
4월 22, 2012 3:58:23 오후 com.sun.jersey.server.impl.application.WebApplicationI
mpl_initiate
정보: Initiating Jersey application, version 'Jersey: 1.5 01/14/2011 12:36 PM'
4월 22, 2012 3:58:25 오후 org.hudsonci.rest.plugin.RestPlugin enable
정보: API provider JAX-RS <Jersey> enabled
4월 22, 2012 3:58:25 오후 org.hudsonci.rest.plugin.RestPlugin enable
정보: API provider Bayeux <ConcD> enabled
4월 22, 2012 3:58:25 오후 org.hudsonci.events.ready.ReadyDetector run
정보: Hudson is ready.
```

A red box highlights the final two lines of the log: `4월 22, 2012 3:58:25 오후 org.hudsonci.events.ready.ReadyDetector run` and `정보: Hudson is ready.`. A blue callout box below the console window contains the text: "Hudson is ready 가 나오면, 설치가 완료되고, 사용할 준비가 된다는 뜻."

Hudson Setting



Hudson Setting

← → ↻ localhost:8080/hudson-2.2.0/ http://localhost:8080/[hudson 폴더명] 으로 접속
여기선 폴더명이 hudson-2.2.0

Hudson

search

Hudson 자동 재실행 켜기

새 작업 소개 내용 입력

Hudson 관리

개발자

빌드 기록

New View

빌드 대기 목록
빌드 대기 항목이 없습니다.

빌드 실행 상태
Status 0/2
Idle

All +

S	W	작업 ↓	최근 성공	최근 실패	최근 소요 시간	Console
		AntTest	1 mo 10 days (#46)	1 mo 10 days (#50)	1.8 sec	

아이콘: [S](#) [M](#) [L](#)

범례 [모든 것에 대해](#) [실패에 대해](#) [마지막 빌드에 대해](#)

Contents.

1. CTIP Environment

2. Setting

- ◆ Tomcat
- ◆ Hudson

3. Build Setting

- ◆ **Hudson**
 - Build
 - Project (Ant, SVN)
- ◆ **Eclipse**
 - Create Project
 - Subversion

4. Build

5. Failure Cases

Hudson - Build Setting

- Configure System

Hudson ?

[Hudson](#) [자음 재실행 켜기](#)

- [새 작업](#)
- [Hudson 관리](#)
- [개발자](#)
- [빌드 기록](#)
- [New View](#)

빌드 대기 목록

빌드 대기 항목이 없습니다.

빌드 실행 상태

Status 0/2

Idle

Manage Hudson

⚠ Your container doesn't use UTF-8 to decode URLs. If you use non-ASCII characters as a job name etc, this will cause problems. See [Containers](#) and [Tomcat i18n](#) for more details.

- [Configure System](#)
Configure global settings and paths. 빌드를 위한 환경설정
- [Reload Configuration from Disk](#)
Discard all the loaded data in memory and reload everything from file system. Useful when you modified config files directly on disk.
- [Manage Plugins](#)
Add, remove, disable or enable plugins that can extend the functionality of Hudson.
- [System Information](#)
Displays various environmental information to assist trouble-shooting.
- [System Log](#)
System log captures output from `java.util`, logging output related to Hudson.

Hudson - Build Setting

- Configure System

Home directory

System Message

of executors

Quiet period

SCM checkout retry count

- Enable security
- Prevent Cross Site Request Forgery exploits
- Help make Hudson better by sending anonymous usage statistics and crash reports to the Hudson project.

Global properties

- Environment variables

Maven 3 SNAPSHOT Monitor

Maven Repository URL

User

Password

Hudson REST API

- Enable [REST API Reference](#)

JDK

JDK installations

List of JDK installations on this system

Git

Git installations

List of Git installations on this system

Ant

Ant installations

List of Ant installations on this system

Maven

Maven installations

List of Maven installations on this system

Maven 3

Maven 3 installations

List of Maven 3 installations on this system

Maven 2/3 Legacy Project Configuration

Global MAVEN_OPTS

Git plugin

Global Config user.name Value

Global Config user.email Value

CVS

cvsexecutable

.cvspass file

[Check CVS version](#)

Subversion

Subversion Workspace Version

Subversion Revision Policy

Exclusion revprop name

- Validate repository URLs up to the first variable name

Shell

Shell executable

Maven 3 Builder Defaults

Maven 3

Goals

Properties

E-mail Notification

SMTP server

Default User E-mail Suffix

System Admin E-mail Address

Hudson URL

Please set a valid host name, instead of localhost

Test configuration by sending e-mail to System Admin Address

Hudson - Project Setting

The screenshot shows the Hudson web interface at localhost:8080/hudson-2.2.0/view/All/newJob. The page title is 'Hudson' and there is a search bar. The left sidebar contains navigation links: '새 작업' (New Job), 'Hudson 관리' (Hudson Management), '개발자' (Developer), '빌드 기록' (Build History), and 'New View'. Below these are status boxes for '빌드 대기 목록' (Build Queue) showing 'Idle' and '빌드 실행 상태' (Build Execution Status) showing 'Status 0/2'. The main content area is titled '새작업 시작' (Start New Job) and contains a form for creating a new job. The '작업명' (Job Name) field is highlighted with a red box and contains 'CTIPTest'. The 'Build a free-style software project' option is selected with a radio button. Below this option is a description: '이것은 Hudson의 주요 기능입니다. Hudson은 어느 빌드 시스템과 어떤 SCM(형상관리)으로 묶인 당신의 프로젝트를 빌드할 것이고, 소프트웨어 빌드보다 다른 어떤 것에 자주 사용될 수 있습니다.' Other options include 'Build a Maven 2/3 project (Legacy)', 'Monitor an external job', 'Build multi-configuration project', and '기존 작업 복사' (Copy from). An 'OK' button is at the bottom.

localhost:8080/hudson-2.2.0/view/All/newJob

Hudson

Hudson » All

새작업 시작

작업명

Build a free-style software project
이것은 Hudson의 주요 기능입니다. Hudson은 어느 빌드 시스템과 어떤 SCM(형상관리)으로 묶인 당신의 프로젝트를 빌드할 것이고, 소프트웨어 빌드보다 다른 어떤 것에 자주 사용될 수 있습니다.

Build a Maven 2/3 project (Legacy)
Maven 2/3 (Legacy) 프로젝트를 빌드합니다. Hudson은 POM 파일의 이점을 가지고 있고 급격히 설정을 줄입니다.

Monitor an external job
이 유형의 작업은 원격 장비처럼 Hudson 외부에서 동작하는 프로세스의 실행을 기록하는 것을 허용합니다. 그렇게 설계되어서, 기존의 자동 시스템의 대시보드로서 Hudson을 사용할 수 있습니다. 자세한 설명은 [여기\(영문\)](#)을 보세요.

Build multi-configuration project
다양한 환경에서의 테스트, 플랫폼 특성 빌드, 기타 등등 처럼 다수의 서로다른 환경설정이 필요한 프로젝트에 적합함.

기존 작업 복사
Copy from

OK

빌드 대기 목록
빌드 대기 항목이 없습니다.

빌드 실행 상태
Status 0/2
Idle

Hudson - Project Setting

Hudson

Hudson » CTIPTest

[Back to Dashboard](#)

[Status](#)

[Changes](#)

[Workspace](#)

[Build Now](#)

[Delete Project](#)

Configure

Project name: CTIPTest

Cascading Project: None

Description:

Discard Old Builds

This build is parameterized

Disable Build (No new builds will be built)

Execute concurrent builds if necessary (beta)

Advanced Project Options

Source Code Management

- None
- CVS
- Git
- Subversion

Build Triggers

- Build after other projects are built
- Build periodically
- Poll SCM
- Build when Maven dependencies have been updated by Maven 3 integration
- Build when Maven SNAPSHOT dependencies have been updated externally

Build

Add build step ▼

Post-build Actions

- Build other projects
- Aggregate downstream test results
- Publish Javadoc
- Record fingerprints of files to track usage
- Publish JUnit test result report
- Archive the artifacts

소스코드 관리 프로그램 선택 - CVS, Git, SVN

트리거(주기) 선택

프로젝트의 Configure 부분

Hudson - Project Setting

- Ant Setting

- Poll SCM
- Build when Maven dependencies have been updated by Maven 3 integration
- Build when Maven SNAP

Build

Add build step ▼

클릭

Post-build Actions

- Build other projects
- Aggregate downstream
- Publish Javadoc
- Record fingerprints of files to track change
- Publish JUnit test result report
- Archive the artifacts

Build

Add build step ▼

빌드 옵션 선택
Invoke Ant 선택

- Invoke Ant
- Execute shell
- Invoke Maven 2 (Legacy)
- Execute Windows batch command
- Invoke Maven 3

Build

Invoke Ant

Ant Version Ant ▼

Targets ▼

Advanced...

클릭

Delete

Add build step ▼

Build

Invoke Ant

Ant Version Ant ▼ 빌드할 클래스 이름 입력

Targets ▼

Build File C:\Users\syjsmk\workspace\CTIPTest\build.xml

Properties Ant로 Export한 Build.xml 파일의 경로 입력

Java Options ▼

Delete

Add build step ▼

Hudson - Project Setting

- SVN Setting

Source Code Management

- None
 - CVS
 - Git
 - Subversion
- Modules

Repository URL

Local module directory (optional)

Repository depth option

Ignore externals option

Unable to access https://dev.naver.com/svn/syjsmktest :
 svn: OPTIONS /svn/syjsmktest failed (show details)
 (Maybe you need to enter credential?)

외부 저장소는 비밀번호가
 필요하므로 credential에서
 입력한다.

Hudson

Hudson > CTIP > Subversion



Subversion Authentication

Enter the authentication information needed to connect to the Subversion repository. This information will be stored in Hudson.

Repository URL

User name/password authentication

SSH public key authentication (svn+ssh)

HTTPS client certificate

Override global credentials Yes No

클릭

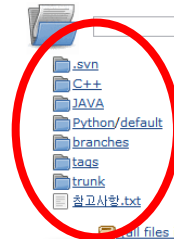
Hudson

Hudson > CTIPTest

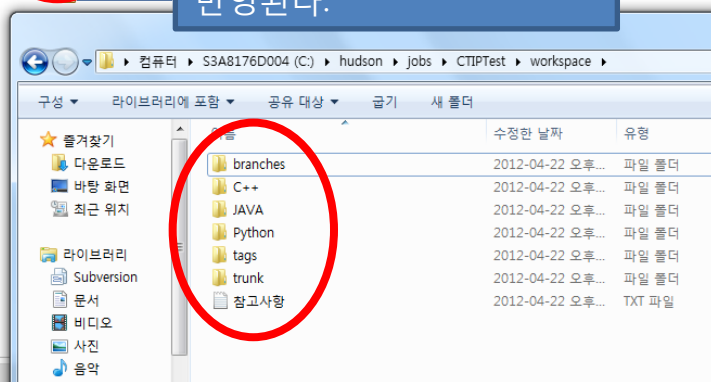
- [Back to Dashboard](#)
- [Status](#)
- [Changes](#)
- Workspace**
- [Wipe Out Workspace](#)
- [Build Now](#)
- [Delete Project](#)
- [Configure](#)

Build History (trend)

#	Time	Status
#7	2012. 4. 22 오후 5:18:00	Success
#6	2012. 4. 22 오후 5:06:01	Success
#5	2012. 4. 22 오후 5:04:00	Success
#3	2012. 4. 22 오후 4:56:12	Success
#2	2012. 4. 22 오후 4:53:49	Success
#1	2012. 4. 22 오후 4:50:39	Success

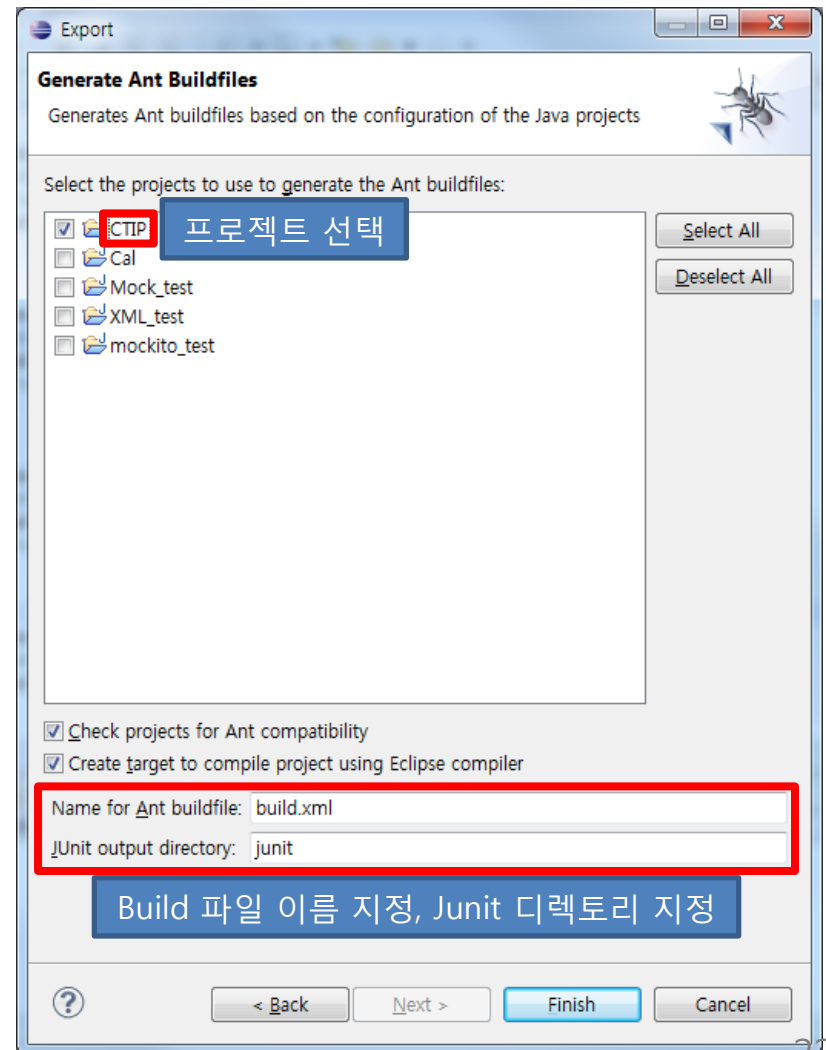
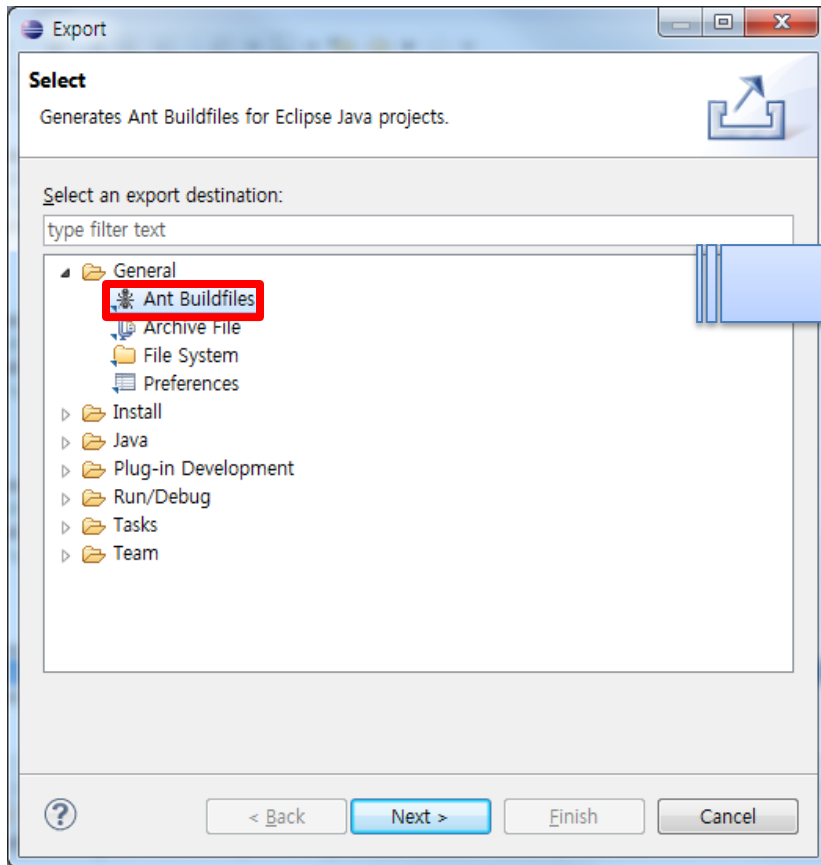


SVN연동시 외부저장소에
 있던 내용이 workspace에
 반영된다.



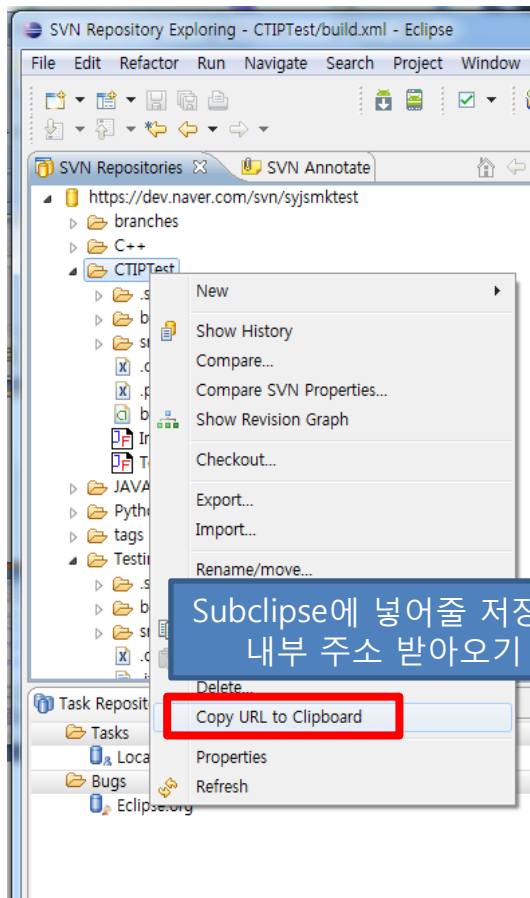
Eclipse - Create Project

- Ant를 이용한 XML 파일 추출



Eclipse - Create Project

- 외부저장소 주소 받아오기



이슈 > Subversion(SVN)에 대한 문서 도 참고하실 수 있습니다.

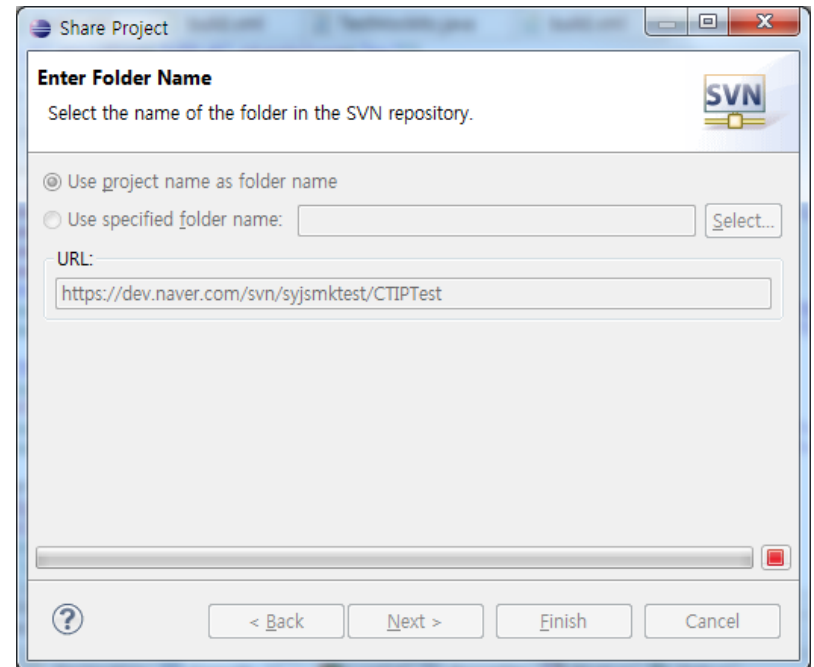
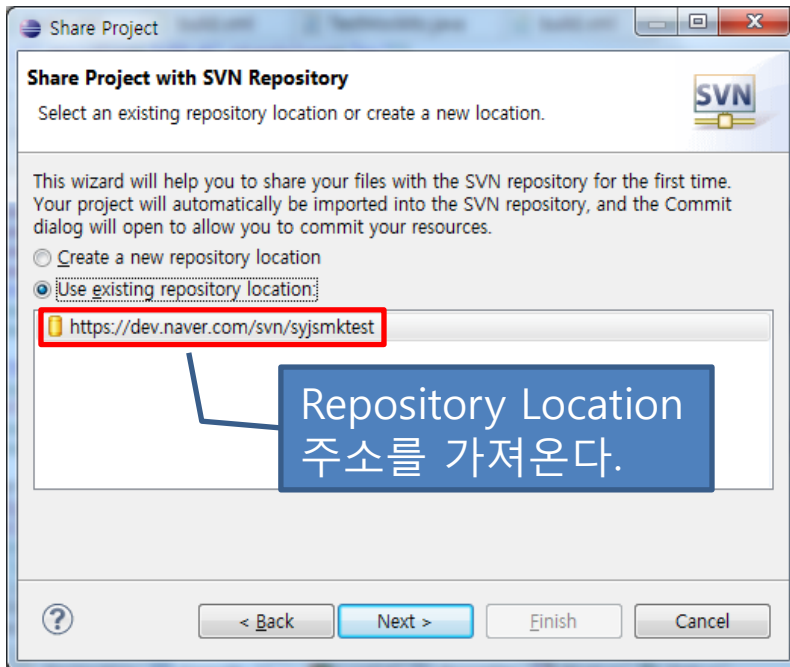
코드 > **코드 트리 보기**
커밋 로그
코드 통계

다운로드 >

파일	최종 로그 메시지	작성자	최종변경일
C++/	oo	syjsmk	2 월
CTIPTest/	CTIPTest commit	syjsmk	16 시간
JAVA/	oo	syjsmk	2 월
Python/	python up	syjsmk	7 월
Testing/		syjsmk	15 시간
branches/	Initial	anonsvn	9 월
tags/	Initial	anonsvn	9 월
trunk/	Initial	anonsvn	9 월
참고사항.txt	참고사항	syjsmk	7 월

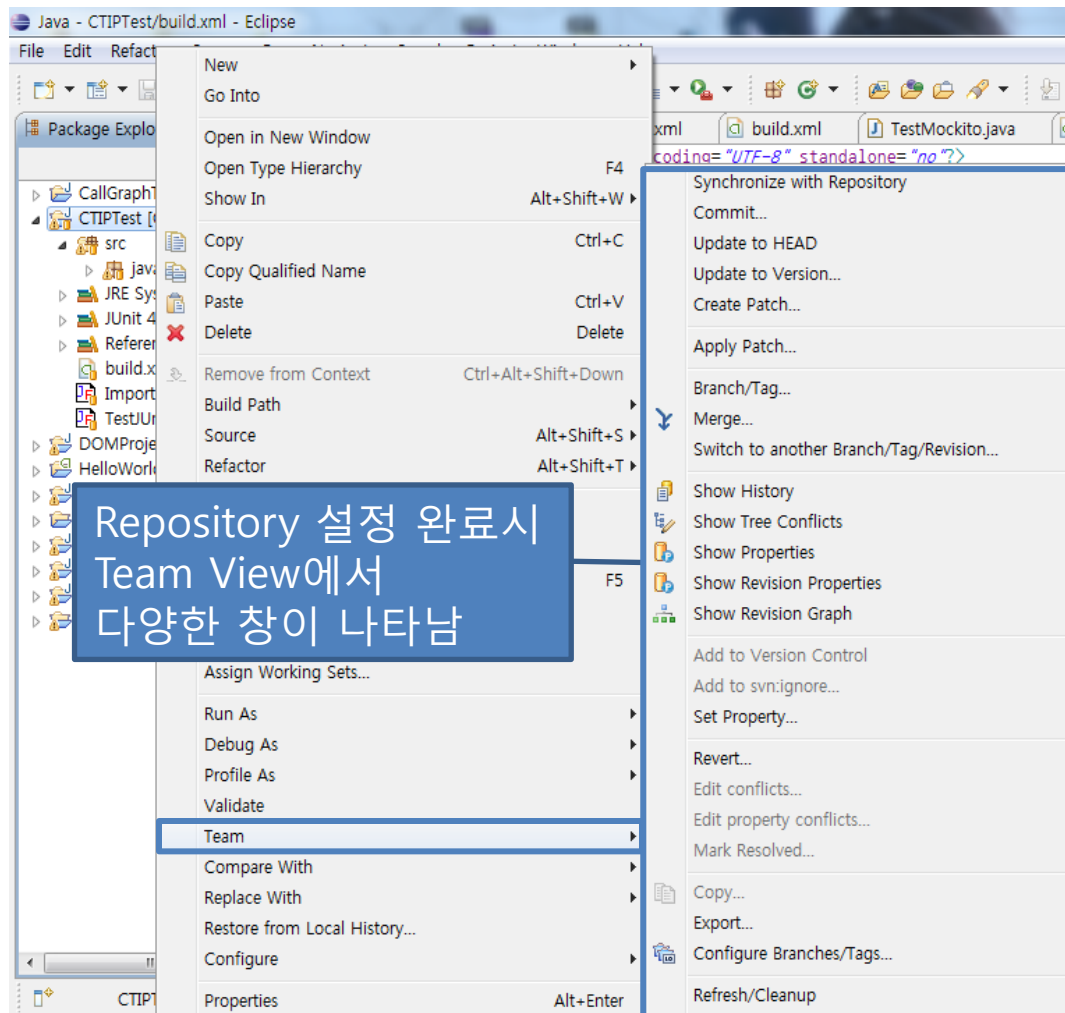
Eclipse - Subversion Setting

- Eclipse Project를 SVN으로 외부저장소와 연동



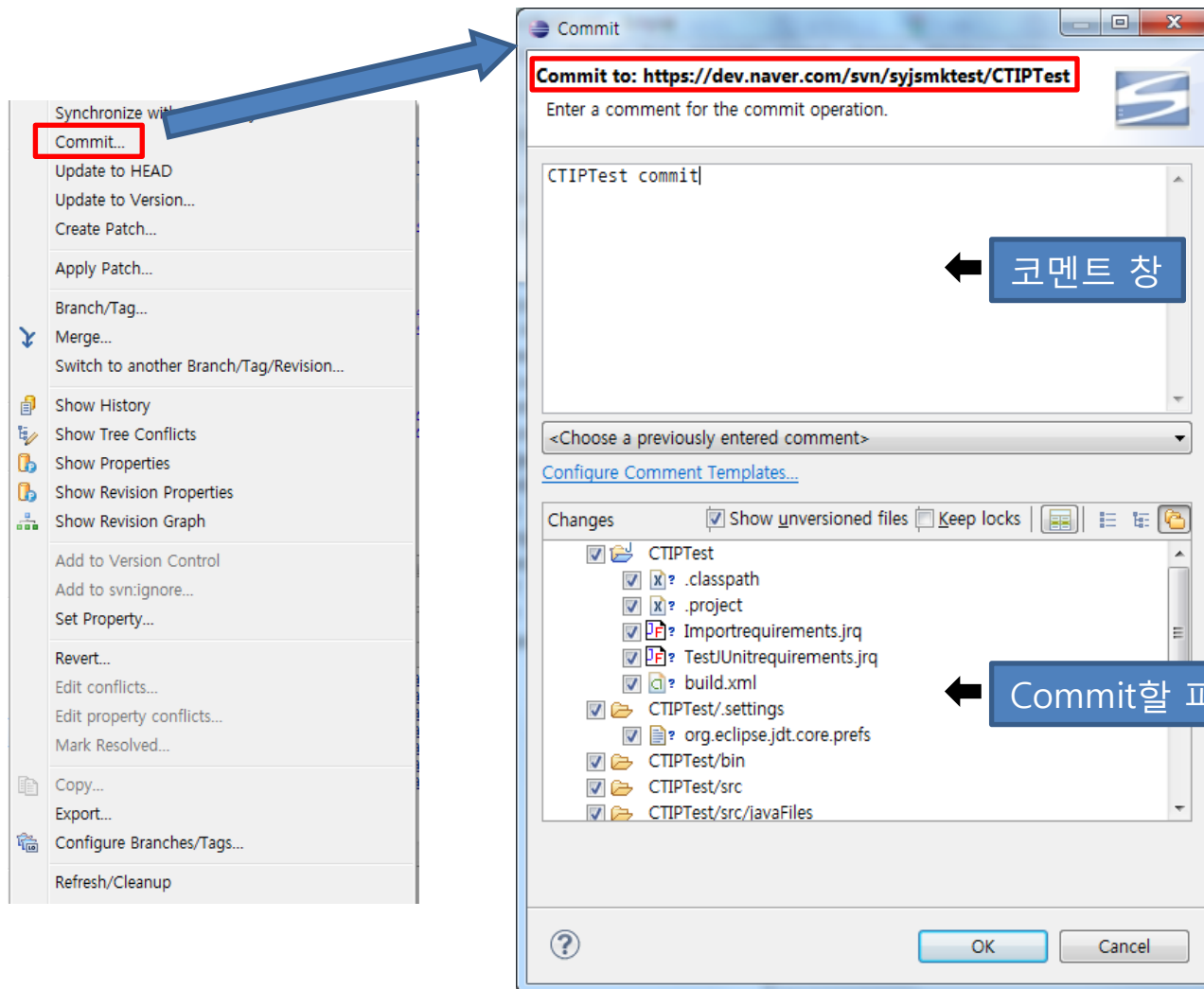
Eclipse - Subversion Setting

- Eclipse Project를 SVN으로 외부저장소와 연동



Eclipse - Subversion Setting

- Commit 하여 SVN Repository에 업로드



Eclipse - Subversion Setting

- Commit 하여 SVN Repository에 업로드

The image shows the Eclipse SVN Commit dialog and the Subversion client interface. The Commit dialog is on the left, and the Subversion client interface is on the right. Annotations in Korean explain the steps.

Commit Dialog:

- Commit to:** `https://dev.naver.com/svn/syjsmktest/CTIPTest` (Annotated: Commit to: `https://dev.naver.com/svn/syjsmktest/CTIPTest`)
- Comment:** `CTIPTest commit` (Annotated: 코멘트)
- Changes:** A list of files and folders to be committed, including `CTIPTest`, `.classpath`, `.project`, `Importrequirements.jrq`, `TestJUnitrequirements.jrq`, `build.xml`, `CTIPTest/settings`, `org.eclipse.jdt.core.prefs`, `CTIPTest/bin`, `CTIPTest/src`, and `CTIPTest/src/javaFiles`. (Annotated: CTIPTest 프로젝트 선택)

Subversion Client Interface:

- Code:** `코드를 불러 보기` (Annotated: 코드 불러 보기)
- Directory Location:** `CTIPTest/` (Annotated: CTIPTest 내부 파일 목록)
- File List:** A list of files and folders in the `CTIPTest` directory, including `.상위디렉토리`, `.settings/`, `bin/`, `src/`, `.classpath`, `.project`, `Importrequirements.jrq`, `TestJUnitrequirements.jrq`, and `build.xml`.

Eclipse - Subversion Setting

- 네이버 개발자 센터 프로젝트 디렉토리

syjsmktest 

프로젝트 홈 >

마일스톤 >

게시판 >

이슈 >

코드 

[코드 트리 보기](#)

[커밋 로그](#)

[코드 통계](#)

다운로드 >

디렉토리 위치: **CTIPTest** CTIPTest 프로젝트가 Commit 되었음을 확인

파일 개수 : 5개 디렉토리 리비전 : 95 / 103 스티키 리비전 :

파일 ^	최종 로그 메세지	작성자	최종변경일
..상위디렉토리			
 .settings/	CTIPTest commit	syjsmk	17 시간
 bin/	CTIPTest commit	syjsmk	17 시간
 src/	CTIPTest commit	syjsmk	17 시간
 .classpath	CTIPTest commit	syjsmk	17 시간
 .project	CTIPTest commit	syjsmk	17 시간
 Importrequirements.jrq	CTIPTest commit	syjsmk	17 시간
 TestJUnitrequirements.jrq	CTIPTest commit	syjsmk	17 시간
 build.xml	CTIPTest commit	syjsmk	17 시간

Hudson 환경구성도

- Hudson

Ant

Ant installations

Ant	Name	Ant
ANT_HOME	C:\apache-ant-1.8.3	

Install automatically

Add Ant

List of Ant installations on this system

Subversion

Subversion Workspace Version	1.5
Subversion Revision Policy	Queue time

Exclusion revprop name

Validate repository URLs up to the first variable name

Shell

Shell executable

Maven

Maven installations

Maven	Name	Maven
MAVEN_HOME	C:\syjsmk\Univ\4-1\Software	

Install automatically

Install from Apache

Version 3.0.4

Add Installer ▼

Add Maven

List of Maven installations on this system

Maven 3 Builder Defaults

Maven 3

(Bundled)

Goals

clean install

Properties

Sonar

Sonar installations

Add Sonar

List of Sonar installations

E-mail Notification

Contents.

1. CTIP Environment

2. Setting

- ◆ Tomcat
- ◆ Hudson

3. Build Setting

- ◆ **Hudson**
 - Build
 - Project (Ant, SVN)
- ◆ **Eclipse**
 - Create Project
 - Subversion

4. Build

5. Failure Cases

Hudson - Build

- 결과레포트 추출 경로 설정

Post-build Actions

- Build other projects
- Aggregate downstream test results
- Publish Javadoc
- Record fingerprints of files to track usage

Publish JUnit test result report

Test report XMLs

Fileset 'includes' setting that specifies the generated raw XML report files, such as 'myproject/target/test-reports/*.xml'. Basedir of the fileset is [the workspace root](#).

Retain long standard output/error

Additional test report features

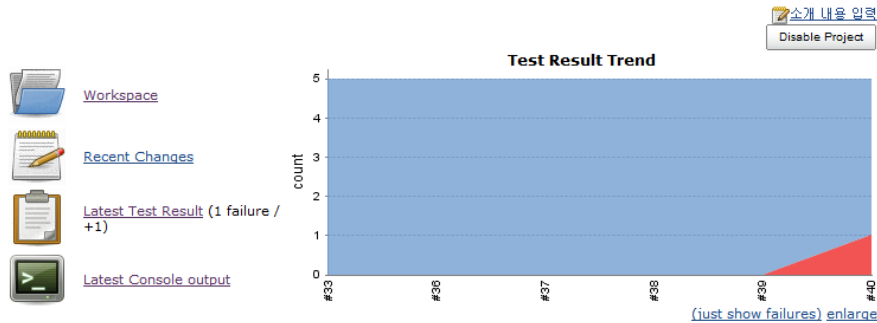
Publish test attachments

- Archive the artifacts
- Record fingerprints of Maven 3 artifacts
- Archive Maven 3 artifacts
- Publish testing tools result report
- Git Publisher
- Sonar
- E-mail Notification

Hudson - Build

Test Failure

Project CTIPTest



Test Result : TestTargetTest

1 failures (+1)



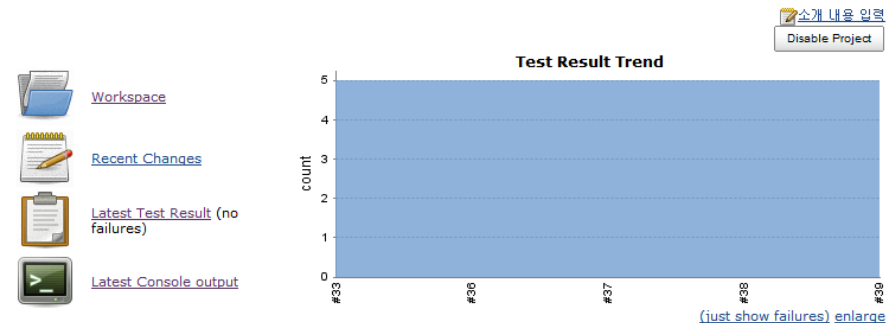
All Tests

Test name	Duration	Status
testMinus	0 ms	Passed
testPlus	7 ms	Passed
testReturnOne	1 ms	Passed
testReturnString	9 ms	Passed
testReturnTrue	74 ms	Regression

테스트 실패

Test Success

Project CTIPTest



Test Result

0 failures (± 0)



All Tests

Package	Duration	Fail	(diff)	Skip	(diff)	Total	(diff)
javaFiles	18 ms	0		0		5	

Hudson - Build

Test Failure

콘솔 출력

```
Started by user anonymous
Updating https://dev.naver.com/svn/svismktest/CTIPTest revision: 2012. 4. 25 오후 10:49:37 depth:infinity ignoreExternals: false
At revision 103
no change for https://dev.naver.com/svn/svismktest/CTIPTest since the previous build
[workspace] $ cmd.exe /C "C:\apache-ant-1.8.3\bin\ant.bat -file build.xml TestTargetTest && exit %%ERRORLEVEL%%"
Buildfile: C:\hudson\jobs\CTIPTest\workspace\build.xml
```

```
Test Target Test:
[junit] Running javaFiles.TestTargetTest
[junit] Tests run: 5, Failures: 1, Errors: 0, Time elapsed: 0,573 sec
[junit] Output:
[junit] BeforeClass
[junit] Before
[junit] After
[junit] Before
[junit] After
[junit] Before
[junit] After
[junit] Before
[junit] After
[junit] Before
[junit] After
[junit] Before
[junit] After
[junit] AfterClass
[junit] Test javaFiles.TestTargetTest FAILED
```

```
BUILD SUCCESSFUL
Total time: 7 seconds
Recording test results
[DEBUG] Skipping watched dependency update: build not configured with trigger: CTIPTest #40
Finished: UNSTABLE
```

Finished : UNSTABLE

Test Success

콘솔 출력

```
Started by user anonymous
Updating https://dev.naver.com/svn/svismktest/CTIPTest revision: 2012. 4. 24 오후 10:40:08 depth:infinity ignoreExternals: false
At revision 103
no change for https://dev.naver.com/svn/svismktest/CTIPTest since the previous build
[workspace] $ cmd.exe /C "C:\apache-ant-1.8.3\bin\ant.bat -file build.xml TestTargetTest && exit %%ERRORLEVEL%%"
Buildfile: C:\hudson\jobs\CTIPTest\workspace\build.xml
```

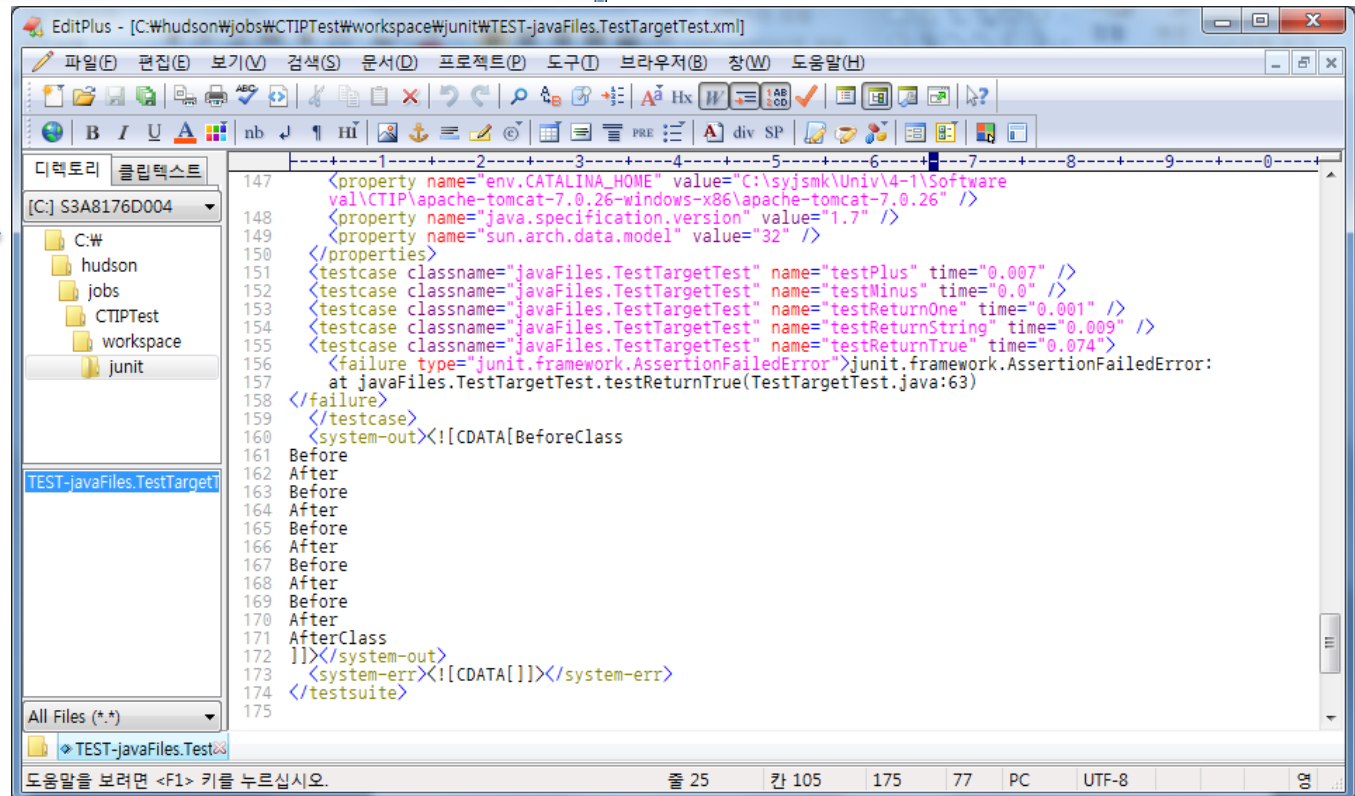
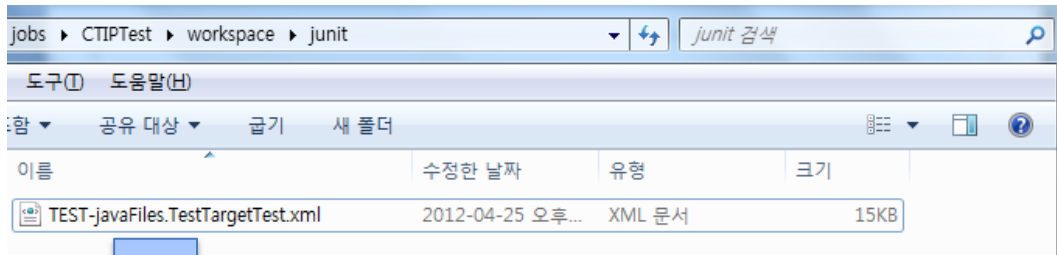
```
Test Target Test:
[junit] Running javaFiles.TestTargetTest
[junit] Tests run: 5, Failures: 0, Errors: 0, Time elapsed: 0,309 sec
[junit] Output:
[junit] BeforeClass
[junit] Before
[junit] After
[junit] Before
[junit] After
[junit] Before
[junit] After
[junit] Before
[junit] After
[junit] Before
[junit] After
[junit] AfterClass
[junit]
```

```
BUILD SUCCESSFUL
Total time: 7 seconds
Recording test results
[DEBUG] Skipping watched dependency update: build not configured with trigger: CTIPTest #39
Finished: SUCCESS
```

Finished : SUCCESS

Hudson - Build

- 결과레포트 확인



Contents.

1. CTIP Environment

2. Setting

- ◆ Tomcat
- ◆ Hudson

3. Build Setting

- ◆ **Hudson**
 - Build
 - Project (Ant, SVN)
- ◆ **Eclipse**
 - Create Project
 - Subversion

4. Build

5. Failure Cases

#1. JUnit 인식 에러

콘솔 출력

```
Started by user anonymous
Updating https://dev.naver.com/svn/svismktest/Testing revision: 2012. 4. 24 오후
2:37:32 depth:infinity ignoreExternals: false
At revision 103
no change for https://dev.naver.com/svn/svismktest/Testing since the previous build
[workspace] $ cmd.exe /C ""C:\wapache-ant-1.8.3\bin\ant.bat -file build.xml
TestTargetTest && exit %%ERRORLEVEL%%"
Buildfile: C:\wudson\jobs\Testing\workspace\build.xml
```

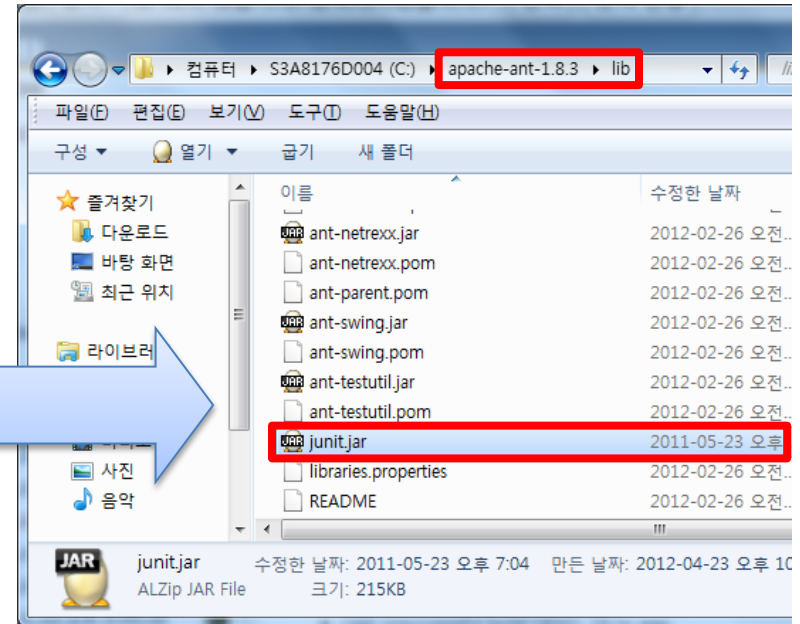
Test Target Test :

Junit.jar파일 위치를 제대로 인식하지 못할 경우 발생하는 에러.

BUILD FAILED

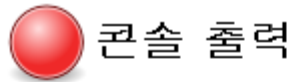
C:\wudson\jobs\Testing\workspace\build.xml:81: The <classpath> for <junit> must include junit.jar if not in Ant's own classpath

```
Total time: 0 seconds
Recording test results
[DEBUG] Skipping watched dependency update for build: Testing #36 due to result:
FAILURE
Finished: FAILURE
```



파일 경로. Hudson이 인식하기 위해서 ant/lib에 junit.jar 파일을 넣어주는게 좋음.

#3. 타겟 클래스 인식 에러



```
Started by user anonymous
Updating https://dev.naver.com/svn/syismktest/CTIPTest revision: 2012. 4. 24 오후 3:06:33 depth:infinity ignoreExternals: false
At revision 103
no change for https://dev.naver.com/svn/syismktest/CTIPTest since the previous build
[workspace] $ cmd.exe /C "C:\wapache-ant-1.8.3\bin\ant.bat -file build.xml TestTargetTest && exit %%ERRORLEVEL%%"
Buildfile: C:\hudson\jobs\CTIPTest\workspace\build.xml
```

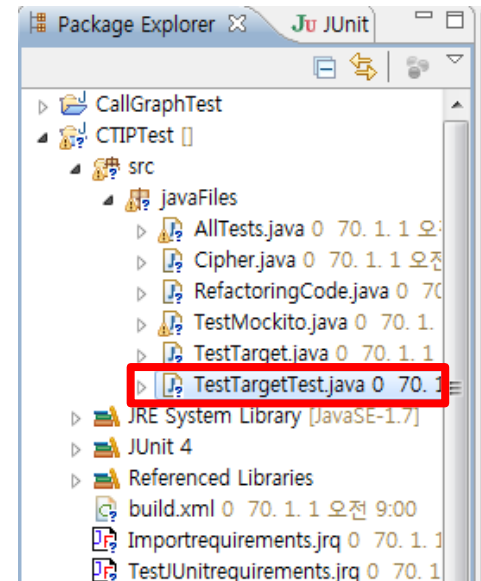
BUILD FAILED

Target "TestTargetTest" does not exist in the project "CTIPTest"

Total time: 0 seconds

[DEBUG] Skipping watched dependency update for build: CTIPTest #26 due to result: FAILURE
Finished: FAILURE

"Test Target Test" 클래스가 있는데 인식을 못함



#3. 타겟 클래스 인식 에러

The image displays two side-by-side screenshots of an IDE showing Ant build.xml files. The left screenshot shows a 'build' target with a 'javac' task. The right screenshot shows a 'JUnit' target with a 'TestTargetTest' target highlighted in red. A blue text box explains that the 'TestTargetTest' class exists but is not listed in the build.xml target name.

```
32 <delete dir="${dir}" />
33 </target>
34 <target depends="clean" name="cleanall" />
35 <target depends="build-subprojects,build-proje" />
36 <target name="build-subprojects" />
37 <target depends="init" name="build-project">
38 <echo message="${ant.project.name}: ${ant.
39 <javac debug="true" debuglevel="${debuglev
40 <src path="src" />
41 <classpath refid="CTIPTest.classpath" />
42 </javac>
43 </target>
44 <target description="Build all projects which
45 <target description="copy Eclipse compiler jar
46 <copy todir="${ant.library.dir}">
47 <fileset dir="${ECLIPSE_HOME}/plugins" />
48 </copy>
49 <unzip dest="${ant.library.dir}">
50 <patternset includes="jdtCompilerAdapt
51 <fileset dir="${ECLIPSE_HOME}/plugins" />
52 </unzip>
53 </target>
54 <target description="compile project with Ecl
55 <property name="build.compiler" value="org
56 <antcall target="build" />
57 </target>
58 </project>
59
```

```
67 </junit>
68 </target>
69 <target name="JUnitTest">
70 <mkdir dir="${junit.output.dir}" />
71 <junit fork="yes" printsummary="withOutAnc
72 <formatter type="xml" />
73 <test name="javaFiles.AllTests" todir=
74 <test name="javaFiles.TestMockito" toc
75 <test name="javaFiles.TestTargetTest"
76 <classpath refid="JUnitTest.classpath
77 </junit>
78 </target>
79 <target name="TestTargetTest">
80 <mkdir dir="${junit.output.dir}" />
81 <junit fork="yes" printsummary="withOutAnc
82 <formatter type="xml" />
83 <test name="javaFiles.TestTargetTest"
84 <classpath>
85 <pathelement location = "${ECLIPSE
86 </classpath>
87 </junit>
88 </target>
89 <target name="JUnitReport">
90 <report format="frames" todir="${juni
91
92
93
94 </report format="frames" todir="${juni
```

TestTargetTest 클래스가 있는데,
Build.xml 파일에는 target name에 없다.

#3. 타겟 클래스 인식 에러

```
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 one in the same
5 directory with the processing instruction <?eclipse.ant.import?>
6 as the first entry and export the buildfile again. -->
7 <project basedir="." default="build" name="JUnitTest">
8 <property environment="env"/>
9 <property name="ECLIPSE_HOME" value="C:/Desktop/eclipse-java-indigo-SR1-win32/eclipse-
10 <property name="junit.output.dir" value="junit"/>
11 <property name="lines,vars"/>
12 <property name="lines,vars"/>
13 <path
14 <path id="JUnitTest.classpath">
15 <path element location="${ECLIPSE_HOME}/plugins/org.junit_4.8.2.v4_8_2_v2011032
16 <path element location="${ECLIPSE_HOME}/plugins/org.hamcrest.core_1.1.0.v2009050
17 </path>
18 <path id="JUnitTest.classpath">
19 <path element location="bin"/>
20 <path refid="JUnit 4.libraryclasspath"/>
21 <path element location=".../Desktop/eclipse-java-indigo-SR1-win32/eclipse/Ext
22 <path element location=".../Desktop/eclipse-java-indigo-SR1-win32/eclipse/Ext
23 </path>
24 <target name="init">
25 <mkdir dir="bin"/>
26 <copy includeemptydirs="false" todir="bin">
27 <fileset dir="src">
28 <exclude name="**/*.java"/>
29 </fileset>
30 </copy>
31 </target>
32 <target name="clean">
33 <delete dir="bin"/>
34 </target>
35 <target depends="clean" name="cleanall"/>
36 <target depends="build-subprojects,build-project" name="build"/>
37 <target name="build-subprojects"/>
38 <target depends="init" name="build-project">
39 <echo message="${ant.project.name}: ${ant.file}"/>
40 <javac debug="true" debuglevel="${debuglevel}" destdir="bin" source="${source}
41 <src path="src"/>
42 <classpath refid="JUnitTest.classpath"/>
43 </javac>
```

두 xml 문서간의 차이

```
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 one in the
5 directory with the processing instruction <?eclipse.ant.import?>
6 as the first entry and export the buildfile again. -->
7 <project basedir="." default="build" name="CTIPTest">
8 <property environment="env"/>
9 <property name="ECLIPSE_HOME" value=".../Users/syjsmk/Desktop/eclipse-
10 <property name="debuglevel" value="source,lines,vars"/>
11 <property name="target" value="1.7"/>
12 <property name="source" value="1.7"/>
13 <path id="JUnit 4.libraryclasspath">
14 <path element location="${ECLIPSE_HOME}/plugins/org.junit_4.8.2.v4_8_2_v201
15 <path element location="${ECLIPSE_HOME}/plugins/org.hamcrest.core_1.1.0.v20
16 </path>
17 <path id="CTIPTest.classpath">
18 <path element location="bin"/>
19 <path refid="JUnit 4.libraryclasspath"/>
20 <path element location=".../Users/syjsmk/Desktop/eclipse-java-indig
21 <path element location=".../Users/syjsmk/Desktop/eclipse-java-indig
22 </path>
23 <target name="init">
24 <mkdir dir="bin"/>
25 <copy includeemptydirs="false" todir="bin">
26 <fileset dir="src">
27 <exclude name="**/*.java"/>
28 </fileset>
29 </copy>
30 </target>
31 <target name="clean">
32 <delete dir="bin"/>
33 </target>
34 <target depends="clean" name="cleanall"/>
35 <target depends="build-subprojects,build-project" name="build"/>
36 <target name="build-subprojects"/>
37 <target depends="init" name="build-project">
38 <echo message="${ant.project.name}: ${ant.file}"/>
39 <javac debug="true" debuglevel="${debuglevel}" destdir="bin" source="${sou
40 <src path="src"/>
41 <classpath refid="CTIPTest.classpath"/>
42 </javac>
43 </target>
```

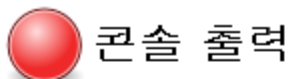
#3. 타겟 클래스 인식 에러

```
52 <fileset dir="${ECLIPSE_HOME}/plugins" includes="org.eclipse.jdt.core_*.j
53 </unzip>
54 </target>
55 <target description="compile project with Eclipse compiler" name="build-eclipse-c
56 <property name="build.compiler" value="org.eclipse.jdt.core.JDTCompilerAdapte
57 <antcall target="build"/>
58 </target>
59 <target name="AllTests (1)">
60 <mkdir dir="${junit.output.dir}"/>
61 <junit fork="yes" printsummary="withOutAndErr">
62 <formatter type="xml"/>
63 <test name="javaFiles.AllTests" todir="${junit.output.dir}"/>
64 <classpath refid="JUnitTest.classpath"/>
65 </junit>
66 </target>
67 <target name="JUnitTest">
68 <mkdir dir="${junit.output.dir}"/>
69 <junit fork="yes" printsummary="withOutAndErr">
70 <formatter type="xml"/>
71 <test name="javaFiles.AllTests" todir="${junit.output.dir}"/>
72 <test name="javaFiles.TestMockito" todir="${junit.output.dir}"/>
73 <test name="javaFiles.TestTargetTest" todir="${junit.output.dir}"/>
74 <classpath refid="JUnitTest.classpath"/>
75 </junit>
76 </target>
77 <target name="TestTargetTest (1)">
78 <mkdir dir="${junit.output.dir}"/>
79 <junit fork="yes" printsummary="withOutAndErr">
80 <formatter type="xml"/>
81 <test name="javaFiles.TestTargetTest" todir="${junit.output.dir}"/>
82 <classpath refid="JUnitTest.classpath"/>
83 </junit>
84 </target>
85 <target name="junitreport">
86 <junitreport todir="${junit.output.dir}"/>
87 <fileset dir="${junit.output.dir}"/>
88 <include name="TEST-*.xml"/>
89 </fileset>
90 <report format="frames" todir="${junit.output.dir}"/>
91 </junitreport>
92 </target>
93 </project>
94
```

두 xml 문서간의 차이
: JUnitTest 부분이 없음
Classpath 부분을 수정하면,
빌드가 정상적으로 됨

```
17 <path id="CTIPtest.classpath">
18 <pathelement location="bin"/>
19 <path refid="JUnit 4.libraryclasspath"/>
20 <pathelement location="../../../../Users/syjsmk/Desktop/eclipse-java-indig
21 <pathelement location="../../../../Users/syjsmk/Desktop/eclipse-java-indig
22 </path>
23 <target name="init">
24 <mkdir dir="bin"/>
25 <copy includeemptydirs="false" todir="bin">
26 <fileset dir="src">
27 <exclude name="**/*.java"/>
28 </fileset>
29 </copy>
30 </target>
31 <target name="clean">
32 <delete dir="bin"/>
33 </target>
34 <target depends="clean" name="cleanall"/>
35 <target depends="build-subprojects,build-project" name="build"/>
36 <target name="build-subprojects"/>
37 <target depends="init" name="build-project">
38 <echo message="${ant.project.name}: ${ant.file}"/>
39 <javac debug="true" debuglevel="${debuglevel}" destdir="bin" source="${sou
40 <src path="src"/>
41 <classpath refid="CTIPtest.classpath"/>
42 </javac>
43 </target>
44 <target description="Build all projects which reference this project. Useful t
45 <target description="copy Eclipse compiler jars to ant lib directory" name="in
46 <copy todir="${ant.library.dir}"/>
47 <fileset dir="${ECLIPSE_HOME}/plugins" includes="org.eclipse.jdt.core_
48 </copy>
49 <unzip dest="${ant.library.dir}"/>
50 <patternset includes="jdtCompilerAdapter.jar"/>
51 <fileset dir="${ECLIPSE_HOME}/plugins" includes="org.eclipse.jdt.core_
52 </unzip>
53 </target>
54 <target description="compile project with Eclipse compiler" name="build-eclips
55 <property name="build.compiler" value="org.eclipse.jdt.core.JDTCompilerAda
56 <antcall target="build"/>
57 </target>
58 </project>
59
```

#4. 결과레포트 에러



```
Started by user anonymous
Updating https://dev.naver.com/svn/svismktest/Testing revision: 2012, 4, 24 오후 3:34:38 depth:infinity ignoreExternals: false
At revision 103
no change for https://dev.naver.com/svn/svismktest/Testing since the previous build
[workspace] $ cmd.exe /C "C:\apache-ant-1.8.3\bin\ant.bat -file build.xml TestTargetTest && exit %%ERRORLEVEL%%"
Buildfile: C:\hudson\jobs\Testing\workspace\build.xml
```

```
Test Target Test :
[junit] Running javaFiles.TestTargetTest
[junit] Tests run: 1, Failures: 0, Errors: 1, Time elapsed: 0 sec
[junit] Test javaFiles.TestTargetTest FAILED
```

```
BUILD SUCCESSFUL
Total time: 2 seconds
Recording test results
None of the test reports contained any result
```

테스트가 아예 failure일 경우,
빌드는 success인데 결과 레포트가 나오지 않는다.

```
[DEBUG] Skipping watched dependency update for build: Testing #41 due to result: FAILURE
Finished: FAILURE
```