

Ant&CruiseControl

Team presentation #2

Team 1

200711460 이상열

200711470 정재호

201111344 김재엽

201211350 박주광

Contensts

Chapter 1 - Ant

Chapter 2 - Cruise Control

Chapter 3 - Example


Chapter 1

Ant

Ant

What is Ant?

- ANT(Another Neat Tool)
 - Apache Ant is a software tool for automating software build processes.
 - make, Batch File, shell script 등의 기능을 통합한 것이다.
 - XML 기반이라 어느 운영체제에서도 사용이 가능한 크로스 플랫폼을 지원합니다.



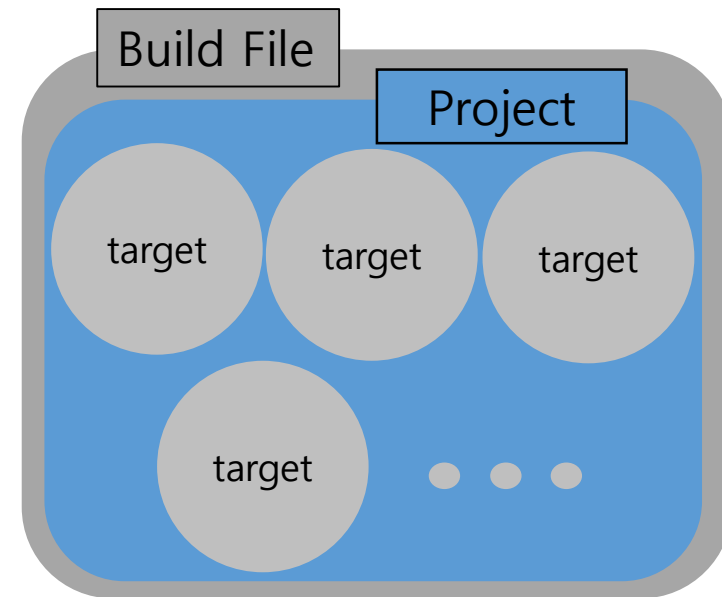
Developer(s)	Apache Software Foundation
Stable release	1.9.0 / March 7, 2013; 26 days ago
Written in	Java
Operating system	Cross-platform
Type	Build Tool
License	Apache License 2.0
Website	ant.apache.org

Ant

Build 파일의 구조

- Build.xml

- Ant의 Build 파일은 XML로 되어 있으며 하나의 Build 파일은 하나의 project와 하나 이상의 target으로 구성되어 있다.
- Target은 task 요소로 구성되어 있다.



Ant

Project

속성(attribute)	설명(description)	필수(requirement)
name	project 이름	아니오
default	target이 설정되지 않았을 때의 기본 target. 이것은 build시에 옵션으로 target을 설정하지 않았을 경우 기본적으로 수행하는 target을 의미	예
basedir	모든 패스(path)의 처리가 실행되는 기본 디렉터리	아니오

- EX) `<project name = "SV_0" default = "init" basedir = "." >`

Ant

Target

속성(attribute)	설명(description)	필수(requirement)
name	target 이름	예
depends	이 target이 하는 target들의 리스트로 왼쪽에서 오른쪽 으로 실행	아니오
if	이 target이 실행하기 위해서 설정되어야 하는 property의 이름	아니오
unless	이 target이 실행하기 위해서 설정되지 않아야 하는 property의 이름	아니오
description	이 target에 대한 설명 부분	아니오

- Ex) `<target name = "get_project" description = "create dir for compile" >`

Ant

Task

- Task의 3가지 형태
 - build-in task : 기본 core task들
 - option task : 추가적인 package 설치 한 경우 사용 가능
 - 사용자 정의 task : 기본 제공 class를 상속 받아서 작성

Ant

Task(cont.)

Task	설명(description)
Property	속성을 선언
copy	파일을 복사
exec	외부파일을 실행
mkdir	디렉토리를 생성
javac	compile하여 class파일을 생성
java	compile된 자바 프로그램을 실행
jar	관련파일을 압축 *.jar 파일을 생성
echo	메시지를 출력

Ant

Export를 이용한 방법

The image shows the Eclipse IDE interface with the 'Export' wizard open. The 'Generate Ant Buildfiles' dialog is the main focus, showing a list of projects to be used for generating buildfiles. The 'SV' project is selected, and the 'Name for Ant buildfile' field is set to 'build.xml'. The 'JUnit output directory' is set to 'junit'.

프로젝트 선택

Build 파일 이름 지정

Ant

Export를 이용한 방법(Cont.)

- Eclipse의 Context Menu로서 Ant의 기능을 활용한 방법
- Eclipse로만 Build한다면 GUI로 제어할 수 있는 편리한 방법
- Limitation
 - 개발자가 많을 경우 각 개발자의 Build 환경이 완전히 같기는 힘들다.
 - Cruise Control 내의 Ant를 이용하여 Build하는 환경과는 맞지 않다.

Ant

Build.xml 작성

- 통합된 환경에서 build가 가능.
- 간단한 xml문법을 알아야 한다.

```
1<project name="SV_0" default="all">
2  <target name="all" depends="clean, checkout, compile, test, sleep, jar" />
3
4  <target name="clean">
5    <delete dir="target" quiet="true" />
6  </target>
7
8  <target name="checkout">
9    <exec executable="svn">
10     <arg value="update" />
11     <!--<arg value="svn://127.0.0.1/trunk/SV_0" />
12     <arg value="src" />-->
13   </exec>
14 </target>
15
16 <target name="compile" depends="checkout">
17   <mkdir dir="target/classes"/>
18   <javac srcdir="src/src/myPack" destdir="target/classes">
19     <classpath>
20       <pathelement location="lib/junit.jar" />
21     </classpath>
22   </javac>
23 </target>
24
25 <target name="sleep">
26   <echo message="Huh...!" />
27   <sleep seconds="60" />

```

Chapter 2

Cruise Control

Cruise Control

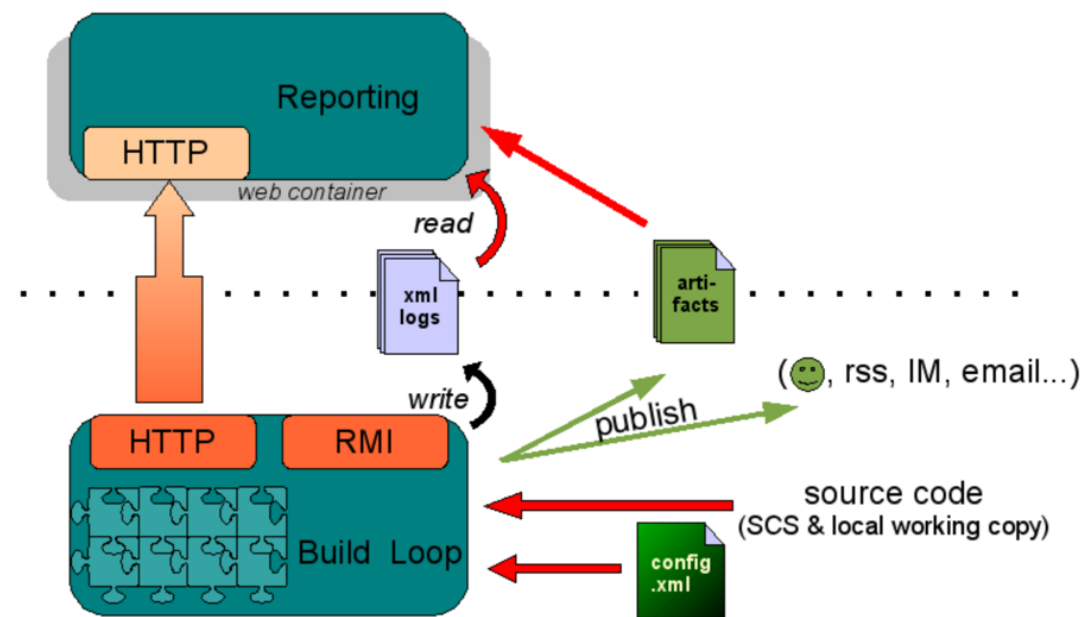
Introduction

- 지속적 통합 (Continuous Integration)을 위한 구성요소 중 하나인 Tool이다.
- 정기적으로 build 스크립트를 수행한다.
- Cruise Control은 java버전과 .Net Framework 버전이 있다.

Cruise Control

Architecture

- CruiseControl에서 Build Loop란 핵심적인 기능이다.
- 소스 코드와 config.xml파일을 읽어들이 처리한다.
- 결과를 web상에서 reporting
- 사용자에게 build 상태를 알려준다.



Cruise Control

Functions

- 여러가지의 버전 관리 프로그램을 지원한다. (GIT, SVN, CVS 등)
- 결과보고서를 통해서 프로젝트의 상태를 확인할 수 있다.
- 대시보드로 등록된 프로젝트의 build상태를 visual하게 보여준다.
- Cruise Control에 plugin으로 통해 기능을 확장할 수 있다.

Cruise Control

Settings

- 새로운 프로젝트 xml의 구성

```
<project name=" SV_0" >
```

```
<listeners>
```

```
  <currentbuildstatuslistener file=" logs/${project.name}/status.txt" />
```

```
</listeners>
```

- Project name: 이름 지정
- Currentbuildstatuslistener: 현재 build 상태와 언제 변경 됐는지에 대한 시간 정보를 파일에 기록

Cruise Control

Settings(Cont.)

```
<bootstrappers>
```

```
<svnbootstrapper localWorkingCopy=" projects/${project.name}/src" />
```

```
</bootstrappers>
```

- Bootstrappers: build 하기 전에 처리되는 항목
- Svnbootstrapper: build를 하기 전에 Subversion Repository에서 build.xml 또는 build.properties파일을 updating

Cruise Control

Settings(Cont.)

```
<modificationset quietperiod=" 30" >  
  <svn RepositoryLocation=" svn://localhost/trunk/${project.name}"  
    username=" user1" password=" test1" />  
</modificationset>
```

- Modificationset: 일정한 시간마다 하위요소에 지정한 handle에 따라서 프로젝트의 바뀐점을 check해준다.
- SVN RepositoryLocation: svn서버에서 프로젝트의 내용들을 기존의 프로젝트와 비교하여 바뀐 점이 있는지 check해준다.

Cruise Control

Settings(Cont.)

```
<schedule interval=" 30" >  
  <ant anthome=" apache-ant-1.7.0"  
        buildfile=" project/${project.name}/build.xml"  />  
</schedule>
```

- Schedule interval: 일정한 시간마다 하위요소에 지정한 build요소를 실행한다. (modificationset에서 바뀐 내용이 있는 경우에만 build를 실행한다)
- Ant anthome: ant라는 자동화 빌드 툴을 이용하여 build를 수행하게 한다.

Cruise Control

Settings(Cont.)

```
<log encoding=" utf-8" >  
<merge dir=" projects/${project.name}/target/test-results" />  
</log>
```

- Log encoding: cruisecontrol log 파일을 저장한다. 이 log파일들을 합치거나 압축 또는 삭제를 하위 요소에 따라서 처리할 수 있다.
- Merge: 지정한 directory에 있는 log들을 합친다

Cruise Control

Settings(Cont.)

```
<publishers>
```

```
<onsuccess>
```

```
  <artifactspublisher dest=" artifacts/${project.name}"
```

```
    file=" project/${project.name}/target/${project.name}.jar"  />
```

```
</onsuccess>
```

```
</publishers>
```

- Publishers: build가 complete했을 때 작동된다.
- Onsuccess: build가 성공했을 때 작동된다.
- Artifactspublisher: build된 products를 대상 디렉토리에 복사해온다.

Cruise Control

Dashboard

Dashboard Server : 192.168.0.24

crucruisecontrol.

Dashboard Administration

Dashboard Builds Administration

Summary

2 project build(s)

Summary

2 project build(s)

0 project build(s) failed

2 project build(s) succeeded

0 project(s) building

0 project(s) discontinued

0 project(s) inactive

100% of projects passing

Tools

RSS Feed

for CCTray

for CC-Config







connectfour passed 6 days ago

SV_0 passed about 2 hours ago

- 주소: <http://localhost:8080/dashboard>

Cruise Control

Dashboard(Cont.)

아이콘	설명
	24시간 이상 빌드가 성공적으로 진행 중이며 현대도 빌드 대기 상태
	24시간 이내 빌드가 성공적으로 진행중
	24시간 이상 빌드가 실패한 상태가 계속됨
	24시간 이내 빌드가 실패했으며 현재 중단된 상태
	현재 빌드 진행 중
	비활성화 되었거나 중단된 프로젝트

Cruise Control

Dashboard(Cont.)

- 빌드 결과를 확인할 수 있다.

Build Error Message

No error message

▼ Errors and Warnings

Test UnitTest.CalcTest FAILED
Huh...!

Stacktrace

No stacktrace

- 테스트 결과를 볼 수 있다.

⌵ Expand All ⌵ Collapse All

▼ Test Suites (2)

UnitTest.CalcTest
Tests: 4, Failures: 1, Errors: 0, Duration: 0.031
UnitTest.MyTest
Tests: 1, Failures: 0, Errors: 0, Duration: 0.0060

Test Failures (1)

▼ **UnitTest.CalcTest (1)**
TestName: testDivision Duration: 0.0040
expected:<2> but was:<8>

junit.framework.AssertionFailedError: expected:<2> but was:<8>
at UnitTest.CalcTest.testDivision(Unknown Source)

Cruise Control

Report Page

Error 404 – Not Found.

No context on this server matched or handled this request.
Contexts known to this server are:

- [/cc-config ---> org.mortbay.jetty.webapp.WebAppContext@606e8697{/cc-config,file:/Users/papimomi/Downloads/cruisecontrol-bin-2.8.4/webapps/cc-config/}](#)
- [/cruisecontrol ---> org.mortbay.jetty.webapp.WebAppContext@784654f6{/cruisecontrol,file:/Users/papimomi/Downloads/cruisecontrol-bin-2.8.4/webapps/cruisecontrol/}](#)
- [/dashboard ---> org.mortbay.jetty.webapp.WebAppContext@5174202a{/dashboard,file:/Users/papimomi/Downloads/cruisecontrol-bin-2.8.4/webapps/dashboard/}](#)
- [/documentation ---> org.mortbay.jetty.webapp.WebAppContext@1eebcf2d{/documentation,file:/Users/papimomi/Downloads/cruisecontrol-bin-2.8.4/webapps/documentation/}](#)

1. 서버에 접속
cruisecontrol 페이지 선택



\$(build.version) \$(build.version.info)

192.168.0.24 [13. 4. 4 오후 6:31]

Project	Status (since)	Last failure	Last successful	Label	
connectfour	waiting (오후 6:27)		13. 3. 29	build.1	Build
SV 0	← waiting (오후 6:30)		오후 5:50	build.13	Build

[RSS](#)

2. Build Result를 볼
프로젝트를 선택한다.

Cruise Control

Report Page(Cont.)

Build Results | Test Results | XML Log File | Metrics | Config | Control Panel

BUILD COMPLETE - build.13

Date of build: 2013-04-04T08:50:54
Time to build: 1 minute 0 seconds
Last changed: 2013-04-04T08:50:24
Last log entry: Calc Add

[Build Artifacts](#)

Errors/Warnings: (2)

Test `UnitTest.CalcTest` FAILED

Huh...!

Unit Tests: (5)

failure	testDivision	UnitTest.CalcTest

Unit Test Error Details: (1)

Test: `testDivision`
Class: `UnitTest.CalcTest`

```
java.lang.AssertionError: junit.framework.AssertionFailedError: expected:<2> but was:<8>  
    at UnitTest.CalcTest.testDivision(Unknown Source)
```

Modifications since last successful build: (1)

modified	user1	/trunk/SV_0/src/myPack/Calc.java	16	2013-04-04T08:50:24	Calc Add

Deployments by this build: (1)

Building jar: /Users/papimomi/Downloads/cruisecontrol-bin-2.8.4/projects/SV_0/target/SV_0.jar

3. Build Result

Build에 대한 기본 정보

build된 products를 대상 디렉토리에 복사해온다.

Error/Warnings에 대한 정보와 콘솔에 출력된 내용을 표기

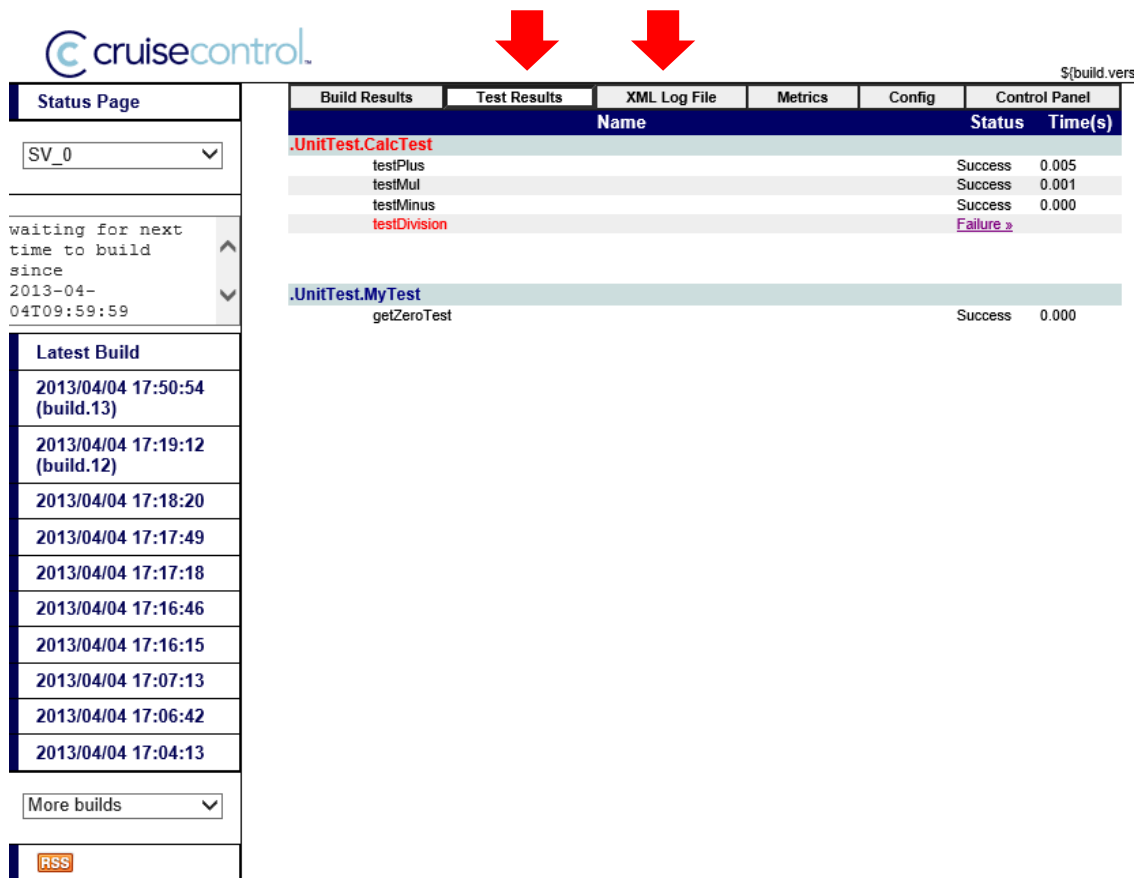
Unit Tests에서는 어디서 fail이 발생했는지 알 수 있다.

Unit Test Error들의 자세한 정보를 제공

가장 최근에 성공적이었던 build와의 차이점을 보여준다.

Cruise Control

Report Page(Cont.)



The screenshot shows the Cruise Control report page. On the left is a sidebar with a 'Status Page' section containing a dropdown menu set to 'SV_0', a status message 'waiting for next time to build since 2013-04-04T09:59:59', and a 'Latest Build' list with timestamps from 2013/04/04 17:50:54 to 17:04:13. The main content area has a navigation bar with tabs for 'Build Results', 'Test Results', 'XML Log File', 'Metrics', 'Config', and 'Control Panel'. Below this is a table of test results. Two red arrows point to the 'Test Results' tab and the table header. The table has columns for 'Name', 'Status', and 'Time(s)'. The first section is '.UnitTest.CalcTest' with rows for 'testPlus' (Success, 0.005), 'testMul' (Success, 0.001), 'testMinus' (Success, 0.000), and 'testDivision' (Failure). The second section is '.UnitTest.MyTest' with a row for 'getZeroTest' (Success, 0.000).

Name	Status	Time(s)
.UnitTest.CalcTest		
testPlus	Success	0.005
testMul	Success	0.001
testMinus	Success	0.000
testDivision	Failure	
.UnitTest.MyTest		
getZeroTest	Success	0.000

4. Test Result

← Class 별, method 별
Unit Test 결과를 보여준다.

Class가 성공 시 파란색
실패 시 빨간색으로 표시

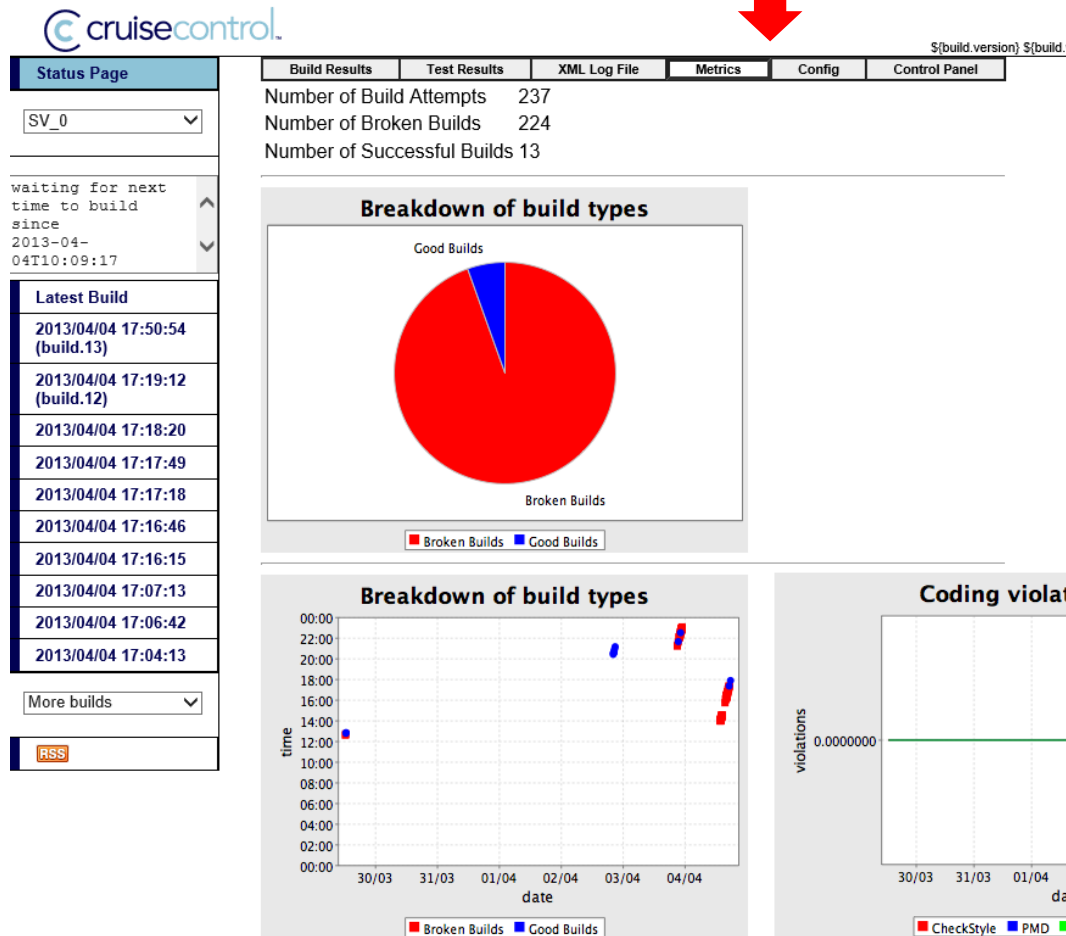
method는 실패 시에만
빨간색으로 표시된다.

5. XML Log File

Cruise Control에서 자동으로
작성되는 XML Log File이다.

Cruise Control

Report Page(Cont.)



6. Metrics

← Build 횟수에 대한 정보

Breakdown of build types
Breakdown of build types
Coding violations 에 대한
그래프 제공

Cruise Control

Report Page(Cont.)

The screenshot displays the Cruise Control web interface. At the top, it says 'Ready for Alpha-level testing' and '\$(build.version)'. The main content area is divided into sections: 'Listeners' (with 'currentbuildstatuslistener' and an 'Add listener' form), 'Bootstrappers' (with 'svnbootstrapper' and an 'Add bootstrapper' form), 'Source Controls', 'Schedule', 'Log', and 'Publishers'. A 'listeners Configuration' dialog box is open, showing an 'OK' button. The left sidebar contains 'Status Page' (SV_0), a log of build times, 'Latest Build' (a list of build timestamps), and 'More builds'. At the bottom, a navigation bar includes 'Build Results', 'Test Results', 'XML Log File', 'Metrics', 'Config' (highlighted with a red arrow), and 'Control Panel'.

7. Config

← 본래 xml로 구성되어 있는 config 파일을 GUI환경을 통해 수정 할 수 있는 탭

Cruise Control

Report Page(Cont.)

Build Results Test Results XML Log File Metrics Config **Control Panel**

JMX Control Panel

CruiseControl JMX Console

Server view MBean view Timers Monitors Relations MLet

MBean CruiseControl Project:name=SV_0
Description Controller for a CruiseControl project

Attributes

Name	Description	Type	Value	New Value
BuildInterval	The build interval in milliseconds. Changes to this attribute are not persisted	long	30000	<input type="text" value="30000"/>
BuildStartTime	Start Time of the last build, using the format 'yyyyMMddHHmmss'	java.lang.String	20130404192313	<input type="text" value="Read-only attribute"/>
Label	The current build label	java.lang.String	build.14	<input type="text" value="build.14"/>
LastBuild	Time of the last build, using the format 'yyyyMMddHHmmss'	java.lang.String	20130404175054	<input type="text" value="20130404175054"/>
LastBuildSuccessful	Attribute exposed for management	boolean	true	<input type="text" value="Read-only attribute"/>
LastSuccessfulBuild	Time of the last successful build, using the format 'yyyyMMddHHmmss'	java.lang.String	20130404175054	<input type="text" value="20130404175054"/>
LogDir	The directory where the log files for this project are written to. Changes to this attribute are not persisted	java.lang.String	logs/SV_0	<input type="text" value="logs/SV_0"/>
LogLabels	A list with the names of the available log files.	java.util.List	View collection	<input type="text" value="Read-only attribute"/>
OutputLoggerID	A unique (for this VM) identifying string for a logger instance. Intended to allow reporting apps (eg: Dashboard) to check if the live output log file has been reset and to start asking for output from the first line of the current output file if the	java.lang.String	377042242_0_1_2_3_4_5_6	<input type="text" value="Read-only attribute"/>

8. Control Panel

BuildInterval 설정
BuildStartTime 확인 등

CruiseControl project의
Controller에 대한 정보를
확인 및 수정 할 수 있다.

Chapter 3

Example

Example

Build.xml 작성

- Clean
 - Delete를 이용한 폴더 삭제
 - Mkdir을 통한 폴더 생성
- Checkout
 - Exec를 이용하여 SVN실행
 - Update를 통해 checkout
- Comile
 - Javac를 이용하여 컴파일

```
1<project name="SV_0" default="all">
2  <target name="all" depends="clean, checkout, compile, test, sleep, jar" />
3
4  <target name="clean">
5    <delete dir="target" quiet="true" />
6  </target>
7
8  <target name="checkout">
9    <exec executable="svn">
10     <arg value="update" />
11     <!--<arg value="svn://127.0.0.1/trunk/SV_0" />
12     <arg value="src" />-->
13   </exec>
14 </target>
15
16 <target name="compile" depends="checkout">
17   <mkdir dir="target/classes"/>
18   <javac srcdir="src/src/myPack" destdir="target/classes">
19     <classpath>
20       <pathelement location="lib/junit.jar" />
21     </classpath>
22   </javac>
23 </target>
24
25 <target name="sleep">
26   <echo message="Huh...!" />
27   <sleep seconds="60" />

```

Example

Build.xml 작성(Cont.)

- test

- mkdir을 통하여 디렉토리 생성
- javac로 컴파일 pathelement location으로 외부 library를 추가 할 수 있다.
- junit을 Unit Test를 실행
- haltonfailure = yes일 경우 실패가 하나라도 있으면 중지
- printsummary = on일 경우 testcase에 대한 statistics를 보여준다.

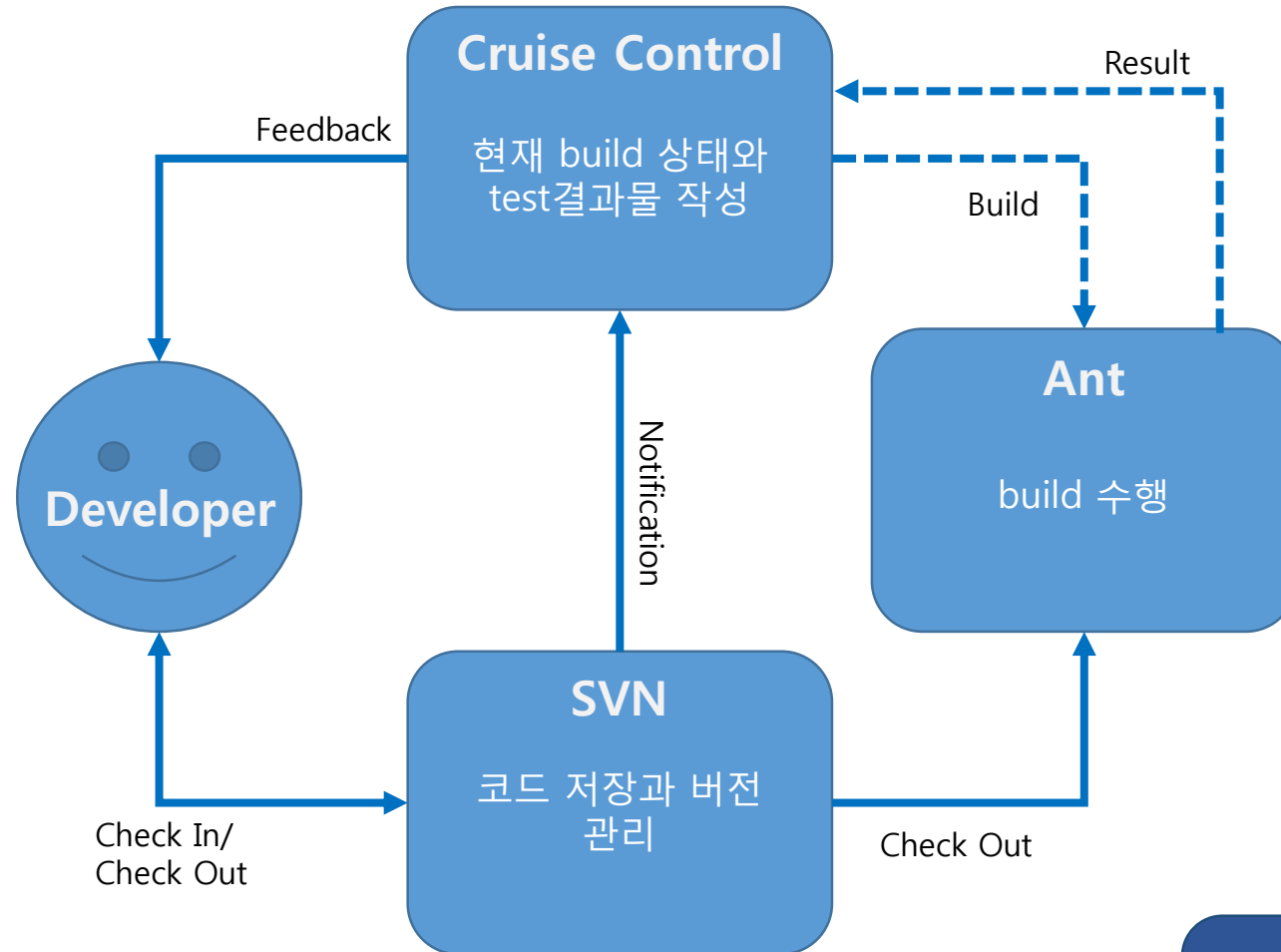
- Jar

- jar를 통하여 관련파일들을 압축 *.jar 파일을 생성

```
30 <target name="test" depends="compile">
31   <mkdir dir="target/test-classes"/>
32   <javac srcdir="src/src/UnitTest" destdir="target/test-classes">
33     <classpath>
34       <pathelement location="target/classes" />
35       <pathelement location="lib/junit.jar" />
36     </classpath>
37   </javac>
38
39   <mkdir dir="target/test-results" />
40   <junit haltonfailure="no" printsummary="on">
41     <classpath>
42       <pathelement location="target/classes" />
43       <pathelement location="lib/junit.jar" />
44       <pathelement location="target/test-classes" />
45     </classpath>
46     <formatter type="brief" usefile="false" />
47     <formatter type="xml" />
48     <batchtest todir="target/test-results">
49       <fileset dir="target/test-classes" includes="**/*Test.class"/>
50     </batchtest>
51   </junit>
52 </target>
53
54 <target name="jar" depends="compile">
55   <jar jarfile="target/SV_0.jar" basedir="target/classes" />
56 </target>
57 </project>
```

Example

Stream Flow



Reference

- Ant

- <http://ant.apache.org/manual/>
- <http://www.sereform.com/?m=20090303>
- <http://bit.ly/14Geucf>
- http://en.wikipedia.org/wiki/Apache_Ant
- <http://blog.naver.com/bell0701?Redirect=Log&logNo=30019755022>
- https://groups.google.com/forum/?fromgroups=#!topic/ksug/_-nBweRwV-Y
- 에릭 해처, [자바의 또 다른 멋진 도구 Ant], (인포북, 2003)
- 이상민, [자바 개발자도 쉽고 즐겁게 배우는 테스트 이야기], (한빛미디어, 2009)

Reference

- **Cruise Control**

- <http://www.build-doctor.com/2008/07/27/cruisecontrol-best-practices-bootstrap-with-a-bootstrapper/>
- <http://techie-notebook.blogspot.kr/2007/08/cruise-control-with-svn-automated-build.html>
- <http://www.sereform.com/?tag=continuous-integration>
- <http://www.okjsp.pe.kr/seq/91899>
- <http://cruisecontrol.sourceforge.net/main/configxml.html>
- <http://www.dahliabock.com/blog/2010/07/16/setting-up-svn-update-in-cruisecontrol/>
- <http://pragmaticstory.com/306>

Thank You!

Q & A