

개요

각 개발자들이 개인 빌드 과정을 거쳐 작성한 코드들을 통합하여 실행 가능한 산출물로 만들어 내는 일련의 과정을 통합 빌드라 하며 지속적이고 자동화된 통합을 위하여 CI를 사용하며 Continuous Integration이란 여러 명으로 구성된 팀이 작업한 것을 자주-각 팀원이 하루에 한 번 이상, 매일 여러 번-통합하는 것을 가리키는 소프트웨어 개발 Practice를 말한다.

설명

CI 효과와 특징

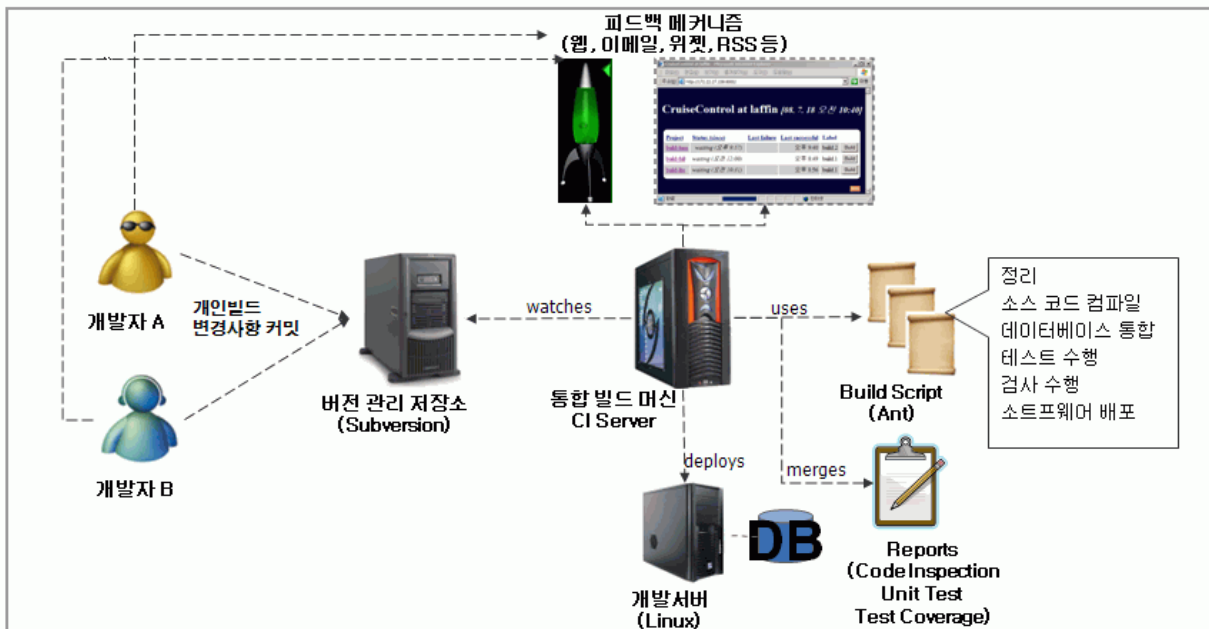
■ 효과

1. 흔히 발생하는 일반적인 위험을 줄여준다.
2. 에러를 초기에 발견할 수 있다. 자주 통합할수록 에러가 발생하는 범위가 좁아 에러를 잡는 것이 수월해진다.
3. 잘 동작하고 에러가 거의 없는 S/W를 유지할 수 있다.
4. 언제 어느 때라도 배포할 수 있는 S/W를 생성해낸다.
5. 반복적인 수작업을 줄여준다.
6. 프로젝트 가시성을 좋게 해준다.
7. 개발 팀이 S/W 제품에 대해 보다 큰 자신감을 갖게 해준다.

■ 특징

1. 소스 코드 일관성 유지 : 소스 관리 시스템이 필요하다.
2. 자동 빌드 : 빌드는 CI 툴에 의해 자동으로 이루어져야 한다.
3. 자동 테스트 : 소스 검증과 함께 기능/비기능 테스트를 매번 검증함으로써 코드의 품질을 높일 수 있다.
4. 일일 체크아웃과 빌드 : 빌드 주기가 길어질수록 개발자 소스의 오류 발생 확률이 높아지고 그 만큼 시간을 낭비하게 된다.

CI(Continuous Integration)는 개발자, 버전 관리 저장소, CI시스템, 빌드 스크립트, 의사소통 장치, 통합 빌드 머신 등으로 구성 된다.



CI 서버는 형상관리 서버로부터 개발자들이 commit 한 소스 코드를 check out 받아 CI 서버의 로컬 파일 시스템에 저장하고 그 코드를 바탕으로 빌드를 실행 후 동작하는 코드 산출물(WAR, JAR 등의 working 패키지)을 생성해 낸다.

- 통합빌드
 - 개요
 - 설명
 - 오픈소스 CI서버: Hudson
 - 환경 설정
 - Hudson 설치
 - Hudson 시스템 설정
 - 사용법
 - Hudson 대시보드 (dashboard)
 - Hudson 프로젝트 메뉴
 - Hudson - JUnit 테스트 연동
 - Hudson - 코드 인스펙션 도구 연동
 - Hudson에 소프트웨어 프로젝트 작업 등록하기
 - 참고자료

오픈소스 CI서버: Hudson

Hudson은 오픈 소스 CI 서버로 소스와 빌드 스크립트를 사용한 자동 빌드 및 빌드 결과를 개발자에게 피드백 하는 메커니즘을 제공한다.

Hudson의 특징 및 주요 기능

Hudson 특징

1. 쉬운 Installation : hudson.war 형태로 servlet container에 배포
2. 웹 기반 UI를 통한 쉬운 설정 및 즉각적인 피드백
3. 플러그인 기반의 확장성 제공
4. 깔끔하고 잘 정의된 문서 및 API

Hudson 주요 기능

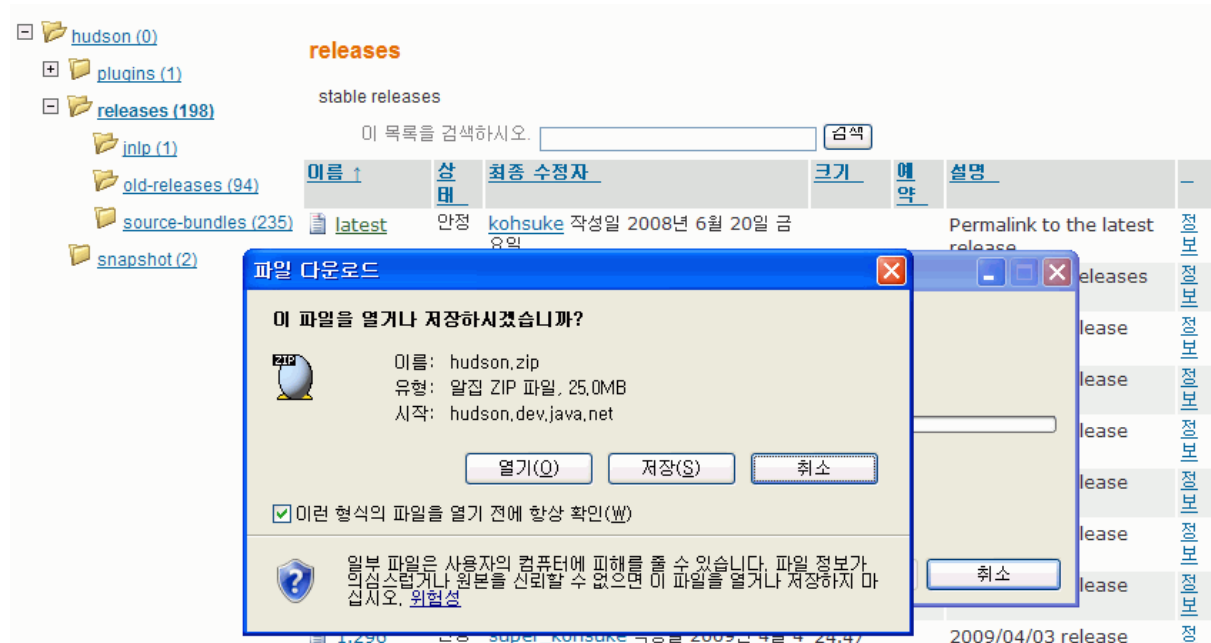
1. 소프트웨어 자동 빌드
 - 빌드 주기에 따른 일일 빌드 또는 주간 빌드 기능 제공
2. 지속적이고 자동화된 빌드 검증
 - SCM 폴링(polling) 기능을 통한 최신 코드 기반의 빌드 수행
3. 지속적이고 자동화된 빌드 테스트
 - 테스트 스위트 실행을 통한 코드 품질 검증
4. 빌드 후속 절차 자동화
 - 컴파일 된 코드의 패키징 및 테스트 리포팅 기능 제공

환경 설정

Hudson 설치

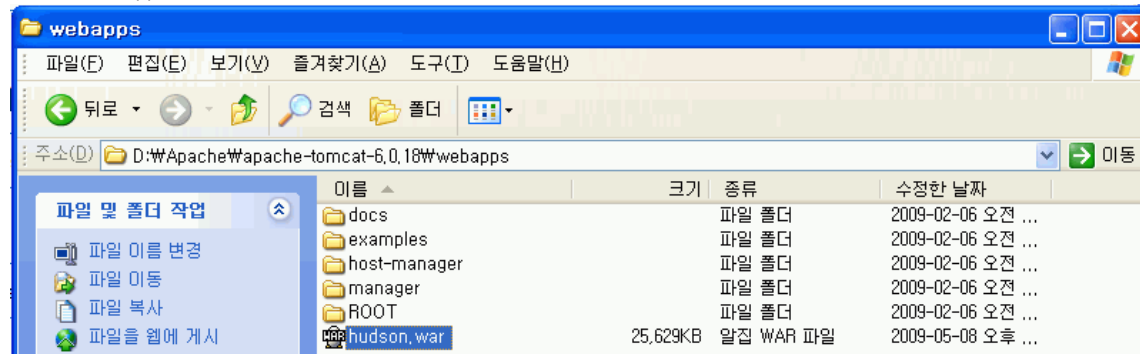
1. hudson.war을 다운로드 받는다.

 Hudson 다운로드



2. hudson.zip 파일로 받은 파일의 확장자를 .war로 변경한다.

3. 톰캣의 webapp 디렉터리 밑에 다운 받은 hudson.war 파일을 위치시킨다.



4. startup.bat 파일로 톰캣 서버를 구동시킨다.

Hudson 시스템 설정

- Configure System 메뉴를 통해 소프트웨어 빌드 작업 및 피드백 작업을 시작하기 위한 JDK 및 Ant, Maven 등의 인스톨 위치를 지정한다.

기본 설정으로 JDK, Maven, Ant, 빌드 결과 피드백 메커니즘 설정을 한다.

1. JDK Home 설정

JDKs

JDK installations	name	<input type="text" value="JAVA_HOME"/>
	JAVA_HOME	<input type="text" value="D:\egovframework\java\jdk1.5.0_17"/>
		<input type="button" value="Delete"/>
	<input type="button" value="Add"/>	

List of JDK installations on this system

2. Maven Home 설정

Maven 빌드를 위한 Maven Home 위치 지정

Maven

Maven installation	name	<input type="text" value="M2_HOME"/>
	MAVEN_HOME	<input type="text" value="D:\egovframework\apache\apache-maven-2.0.9"/>
		<input type="button" value="Delete"/>
	<input type="button" value="Add"/>	

List of Maven installations on this system. Both maven 1 and 2 are supported.

3. Ant Home 설정

Ant 작업 수행을 위한 Ant Home 위치 지정

Ant

Ant installation	name	<input type="text" value="ANT_HOME"/>
	ANT_HOME	<input type="text" value="D:\apache-ant-1.7.0"/>
		<input type="button" value="Delete"/>
	<input type="button" value="Add"/>	

List of Ant installations on this system

4. 피드백 설정을 위한 E-mail Notification 설정

SMTP 서버, Hudson 시스템 Admin 이메일 주소, 설치된 Hudson의 URL

E-mail Notification

[Test configuration by sending e-mail to System Admin Address](#)

SMTP server	<input type="text" value="smtp.gmail.com"/>	<input type="button" value="?"/>
Default user e-mail suffix	<input type="text"/>	<input type="button" value="?"/>
System Admin E-mail Address	<input type="text" value="jason@xxx.net"/>	<input type="button" value="?"/>

Notification e-mails from Hudson to project owners will be sent with this address in the from header. This can be just "foo@acme.org" or it could be something like "Hudson Daemon <foo@acme.org>"

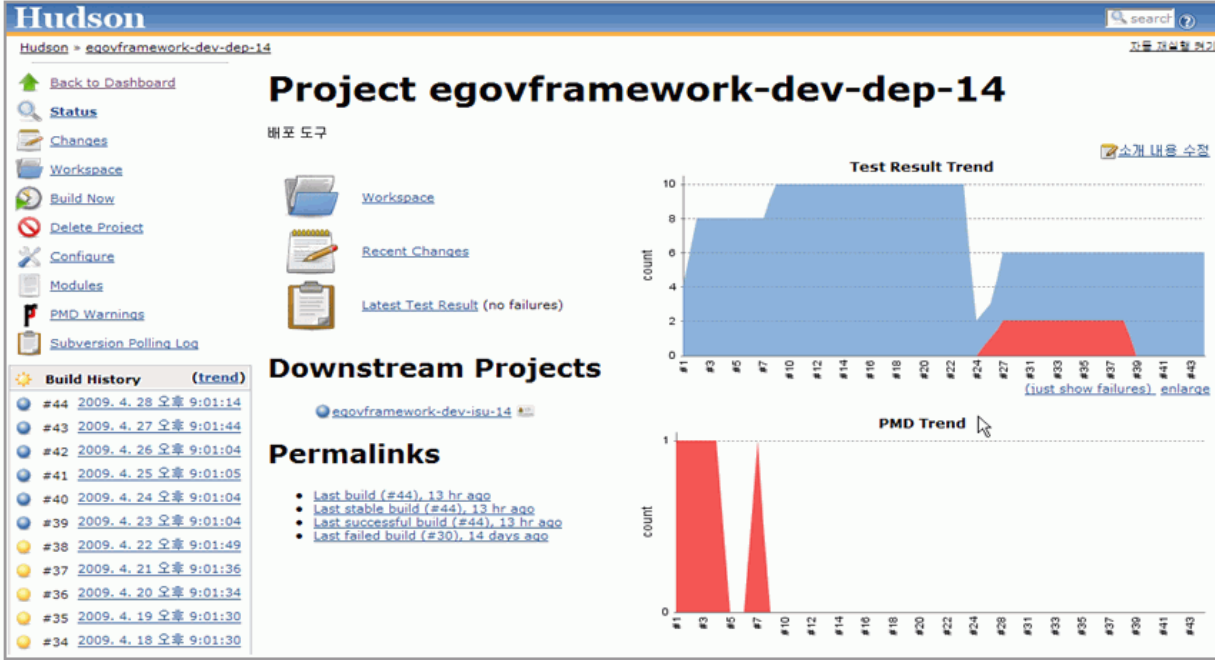
Hudson URL	<input type="text" value="http://192.168.100.16:8080/hudson/"/>	<input type="button" value="?"/>
------------	---	----------------------------------

Optionally specify the HTTP address of the Hudson installation, such as `http://yourhost.yourdomain/hudson/`. This value is used to put links into e-mails generated by Hudson.

This is necessary because Hudson cannot reliably detect such a URL from within itself.

Hudson 대시보드 (dashboard)

Hudson은 빌드 결과 및 테스트 결과등을 보여주는 대시보드 기능을 제공하고 있다.



Hudson 프로젝트 메뉴

Hudson은 등록된 프로젝트의 소스 코드 변경 내용 추적용 메뉴와 수동 빌드 메뉴 그리고 프로젝트 빌드 상태를 보여주는 메뉴 등을 제공하고 있다.

Hudson search 고급 검색 및 필터

Hudson > [DEV](#) > [egovframework-dev-dep-14](#)

[Back to Dashboard](#)

Status

[Changes](#)

[Workspace](#)

[Build Now](#)

[Delete Project](#)

[Configure](#)

[Modules](#)

[PMD Warnings](#)

[Subversion Polling Log](#)

배포 도구

[Workspace](#)

[Recent Changes](#)

[Latest Test Result \(no failures\)](#)

Project egovfram

Test Result Trend

count

#1 #3 #5 #7 #10 #12 #14 #16 #18 #20 #22 #24 #27 #31 #33 #35 #37 #39 #41 #43

(just show failures). enlarge

Downstream Projects

[egovframework-dev-isu-14](#)

Permalinks

- Last build (#47), 2 days 17 hr ago
- Last stable build (#47), 2 days 17 hr ago
- Last successful build (#47), 2 days 17 hr ago
- Last failed build (#30), 19 days ago

Build History (trend)

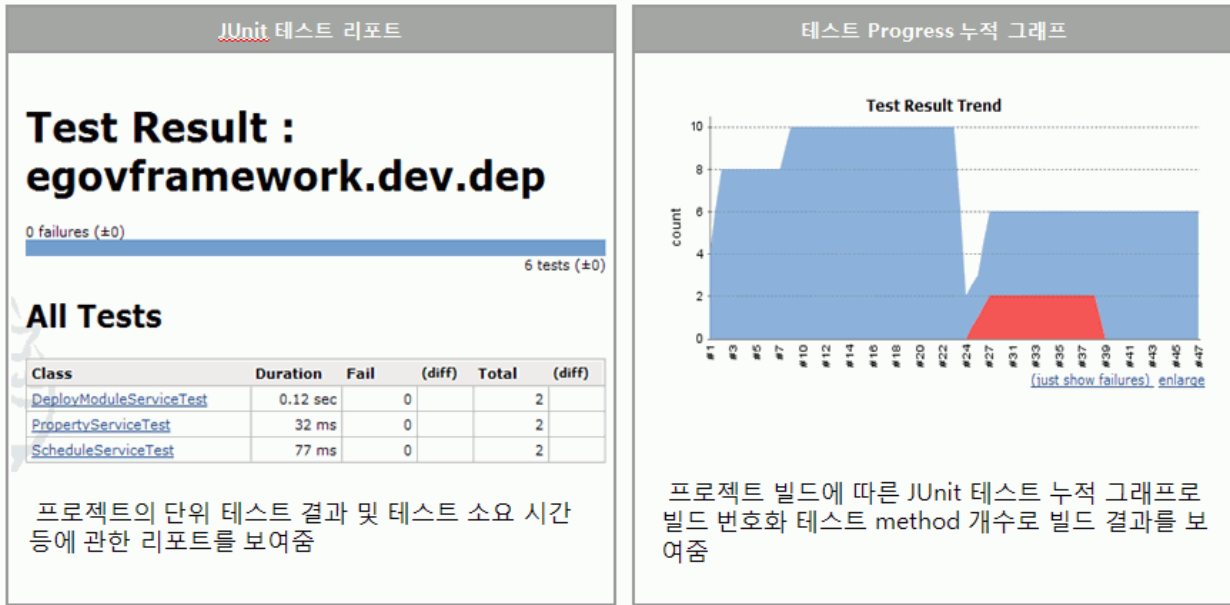
Build Number	Time
#47	2009. 5. 1 오후 9:01:51
#46	2009. 4. 30 오후 9:01:30
#45	2009. 4. 29 오후 9:01:30
#44	2009. 4. 28 오후 9:01:14
#43	2009. 4. 27 오후 9:01:44
#42	2009. 4. 26 오후 9:01:04

기본 메뉴

- Changes**
: 빌드 버전별로 SCM에서 지난 버전에 비해서 변경된 내용에 대해 변경자와 커밋시에 개발자가 추가한 comment를 확인할 수 있음
- Workspace**
: 프로젝트의 빌드 디렉터리로 브라우저를 통해 빌드에 사용된 파일 등을 확인할 수 있음
- Build Now**
: 스케줄에 상관없이 지금 강제로 빌드
- Build History**
: 빌드 수행 시기와 빌드 상태 및 성공 여부를 보여줌

Hudson - JUnit 테스트 연동

Hudson은 기본 설정으로 빌드 스크립트를 이용해 생성한 JUnit 테스트 리포트를 브라우저에 출력해 주는 기능을 제공하고 있으며 테스트의 Progress 누적 그래프를 Hudson 프로젝트의 초기 화면에 보여주는 기능을 제공한다.



Hudson - 코드 인스펙션 도구 연동

Hudson은 코드 인스펙션 도구인 PMD 플러그인을 제공하며 빌드 스크립트를 통해 수행한 PMD 결과 리포트를 PMD 플러그인 연동을 통해 Hudson 화면에 보여주는 기능을 제공한다.

Build Settings

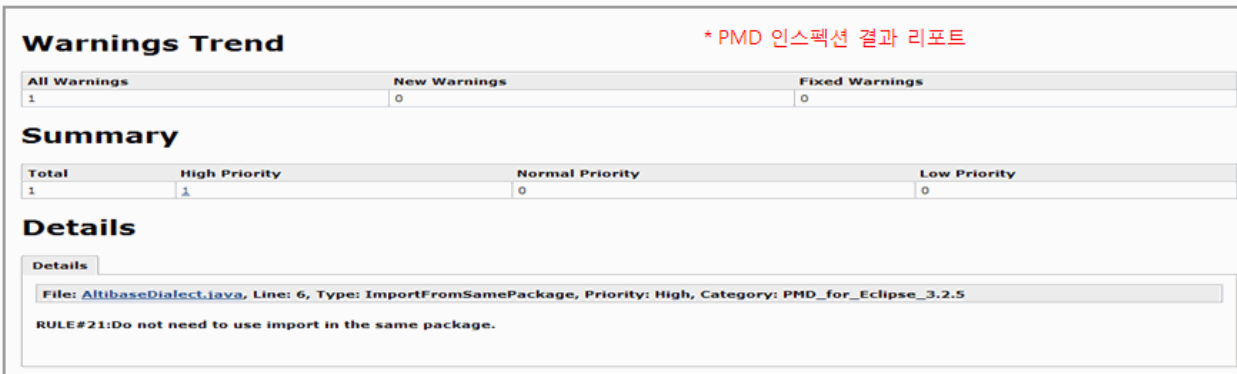
☐ E-mail Notification

☒ Publish PMD analysis results

* 프로젝트의 'Configure; 메뉴에서 PMD 결과 생성 설정

Hudson understands [PMD](#) analysis report XML format. When this option is configured and your build runs the maven goal `pmd:pmd` then Hudson provides useful information about analysis results, such as historical result trend, module and package statistics, web UI for viewing analysis reports and warnings, and so on.

프로젝트의 Configure 메뉴 선택 후 PMD 설정



Hudson 화면에서 PMD 결과 확인

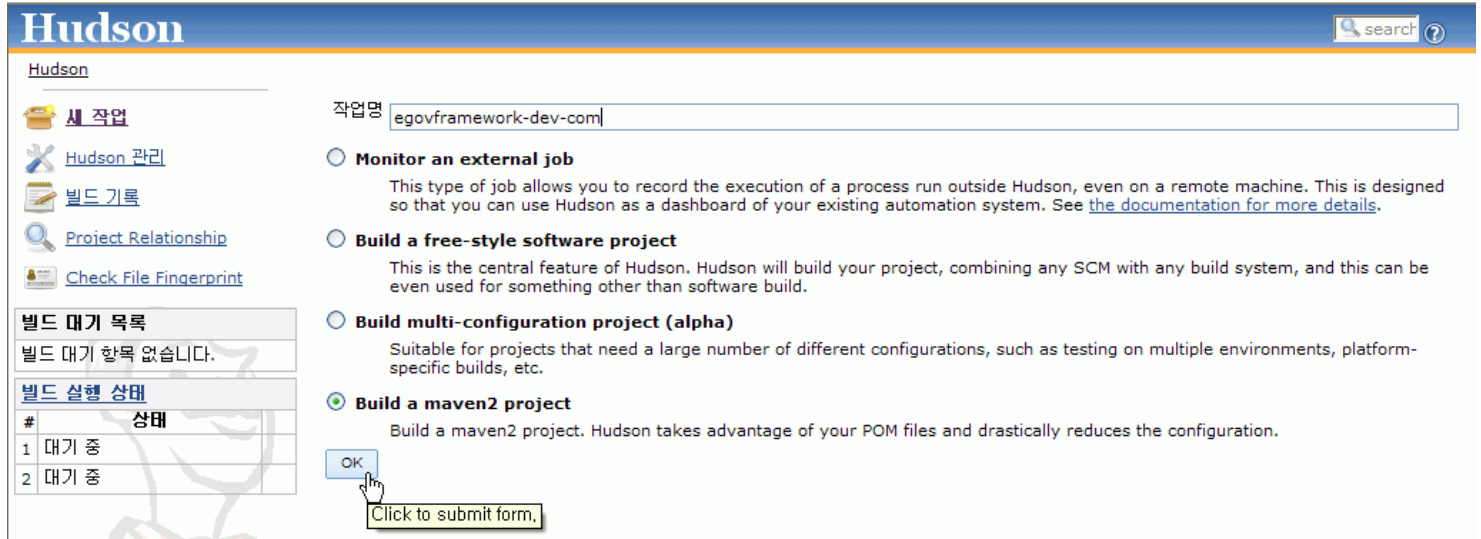
Hudson에 소프트웨어 프로젝트 작업 등록하기

1. 설정 전제 조건

1. 접근 가능한 소스 코드 저장소(Repository)가 존재해야 한다.
2. 저장소는 등록하려고 하는 작업(프로젝트)의 소스 코드를 담고 있어야 한다.
3. 프로젝트는 소스 자동 빌드를 위한 스크립트를 반드시 포함하고 있어야 한다.

2. 작업 등록 시나리오

1. 등록할 작업명을 입력한다.
2. Build a maven2 project를 선택한다. (전자정부 표준 프레임워크는 메이븐 프로젝트를 기본으로 함)
3. OK 버튼을 클릭 후 Hudson 작업을 생성한다.



The screenshot shows the Hudson 'New Job' page. The job name 'egovframework-dev-com' is entered. The 'Build a maven2 project' option is selected. A tooltip points to the 'OK' button with the text 'Click to submit form.'.

Hudson

Jobs

새 작업

Hudson 관리

빌드 기록

Project Relationship

Check File Fingerprint

빌드 대기 목록

빌드 대기 항목 없습니다.

빌드 실행 상태

#	상태
1	대기 중
2	대기 중

작업명: egovframework-dev-com

☐ Monitor an external job
This type of job allows you to record the execution of a process run outside Hudson, even on a remote machine. This is designed so that you can use Hudson as a dashboard of your existing automation system. See [the documentation for more details](#).

☐ Build a free-style software project
This is the central feature of Hudson. Hudson will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

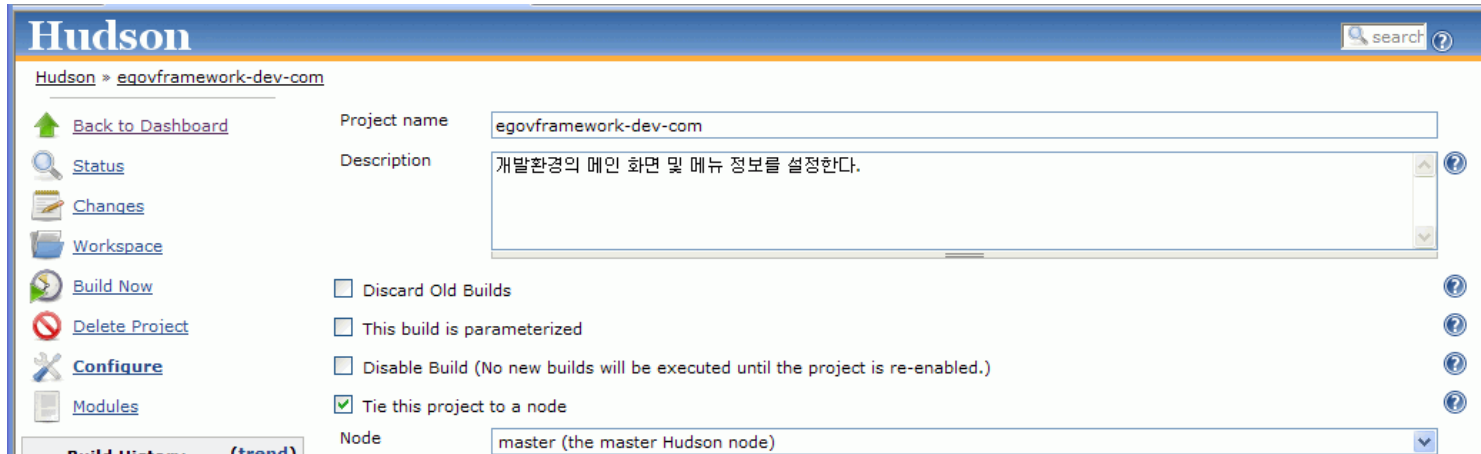
☐ Build multi-configuration project (alpha)
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

☒ Build a maven2 project
Build a maven2 project. Hudson takes advantage of your POM files and drastically reduces the configuration.

OK
Click to submit form.

3. 등록된 작업 설정

3.1 등록된 작업(프로젝트)의 이름 확인 및 프로젝트 요약 설명 작성한다.



The screenshot shows the Hudson 'Configure' page for the job 'egovframework-dev-com'. The project name and description are set. The 'Tie this project to a node' checkbox is checked, and the node is set to 'master (the master Hudson node)'.

Hudson

Hudson » egovframework-dev-com

Back to Dashboard

Status

Changes

Workspace

Build Now

Delete Project

Configure

Modules

Build History (trend)

Project name: egovframework-dev-com

Description: 개발환경의 메인 화면 및 메뉴 정보를 설정한다.

☐ Discard Old Builds

☐ This build is parameterized

☐ Disable Build (No new builds will be executed until the project is re-enabled.)

☒ Tie this project to a node

Node: master (the master Hudson node)

3.2 프로젝트의 코드가 보관된 SCM(저장소)를 선정한다.

- 프로젝트에서 사용중인 SCM을 선택한다. - CVS 또는 SVN 등
- 전자정부 표준 프레임워크에서는 SVN을 개발환경의 도구로 제공하고 있다.

3.3 SCM 종류 선택 후 Hudson 서버에 등록할 프로젝트가 저장된 저장소(repository) URL을 설정한다.

- URL은 repository에 등록된 프로젝트 name 레벨까지 적어준다.
- 예) <http://192.168.100.11:8090/svn/eGovFramework/DEV/egovframework-dev-com>



The screenshot shows the 'Source Code Management' configuration page. 'Subversion' is selected as the SCM. The repository URL is 'http://192.168.100.11:8090/svn/eGovFramework/DEV/egovframework-dev-com'. The 'Use update' checkbox is checked. The repository browser is set to '(Auto)'.

Source Code Management

☐ None

☐ CVS

☒ Subversion

Modules

Repository URL: <http://192.168.100.11:8090/svn/eGovFramework/DEV/egovframework-dev-com>

Local module directory (optional): egovframework-dev-com

Add more locations...

Use update: ☒

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

Repository browser: (Auto)

Advanced...

3.4 주기적 통합을 위한 자동 빌드 주기를 설정한다.

- Hudson은 cron expression 기반의 스케줄링을 제공하고 있다
- 예) 매일 21시에 빌드 실행 - 0 21 * * *

Build Triggers

☒ Build whenever a SNAPSHOT dependency is built ?

☐ Build after other projects are built ?

☒ Build periodically ?

Schedule

☐ Poll SCM ?

3.5 빌드를 설정한다.

- Root POM - Maven 프로젝트의 정보를 담고 있는 pom.xml을 입력한다.
- Maven 빌드 생명 주기 단계 및 option 사항을 입력한다.
- 예) clean install pmd:pmd emma:emma

3.6 빌드 피드백 매커니즘을 설정한다.

Build

Root POM ?

Goals and options ?

Build Settings

☒ E-mail Notification ?

Recipients

Whitespace-separated list of recipient addresses. E-mail will be sent when a build fails.

☒ Send e-mail for every unstable build

☐ Send separate e-mails to individuals who broke the build ?

3.7 빌드 후 인스펙션 도구인 pmd 및 emma 커버리지 리포트 생성 설정을 한다.

☒ Publish PMD analysis results

Post-build Actions

☐ Deploy artifacts to Maven repository

☐ Aggregate downstream test results

☒ Build other projects

Projects to build

☐ Trigger even if the build is unstable

☒ Record Emma coverage report

Emma XML report

Specify the path to the Emma's XML report file, relative to [the workspace root](#).

참고자료

 Martin Fowler - Continous Integration