
개발환경

개발환경 목차

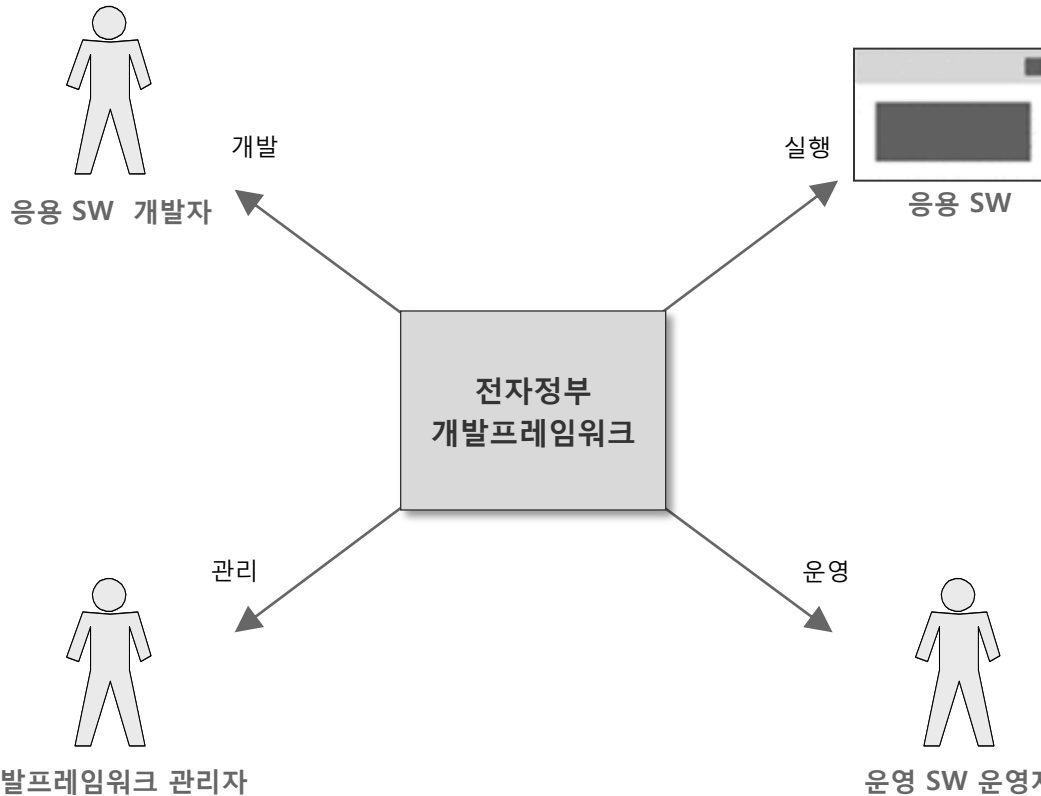
1. 개발환경 개요
2. 구현도구
3. 개인빌드 (Maven)
4. 테스트 도구
5. 형상관리
6. CI 서버
7. Deploy
8. 이슈관리

1. 개발환경 개요
2. 서비스 도출
3. 개발환경 구성요소
4. 적용 오픈소스
5. 서비스별 소개
6. 개발자 개발환경 구성
7. 서버 개발환경 구성

개발 환경은 프레임워크 이해관계자 중 응용 S/W 개발자를 위한 환경

전자정부 개발프레임워크 이해관계자 주요 관심사 분석

- 응용 SW 개발자는 전자정부 개발프레임워크를 기반으로 어플리케이션을 개발하며, 개발 생산성 향상을 위해 전자정부 개발 프레임워크로부터의 개발 지원 기능이 필요함



- 응용 SW는 전자정부 개발 프레임워크를 기반으로 개발되는 어플리케이션으로써 어플리케이션의 효과적인 동작을 위한 전자정부 개발 프레임워크의 실행 지원 기능이 필요함

- 개발프레임워크 관리자는 전자정부 개발프레임워크의 지속적인 개선 및 유지보수를 담당하며, 전자정부 개발프레임워크의 효과적인 관리 지원 기능이 필요함

- 응용 SW 운영자는 응용 소프트웨어의 운영과정에서 어플리케이션의 기반인 전자정부 개발프레임워크로부터의 운영 지원 기능이 필요함

응용 SW 개발자의 역할인 요구 분석자, 분석 설계자, 구현자, 테스터, 배포자, 형상 및 변경 관리자, 프로젝트 관리자 등을 지원하는 사례 분석 결과는 다음과 같음

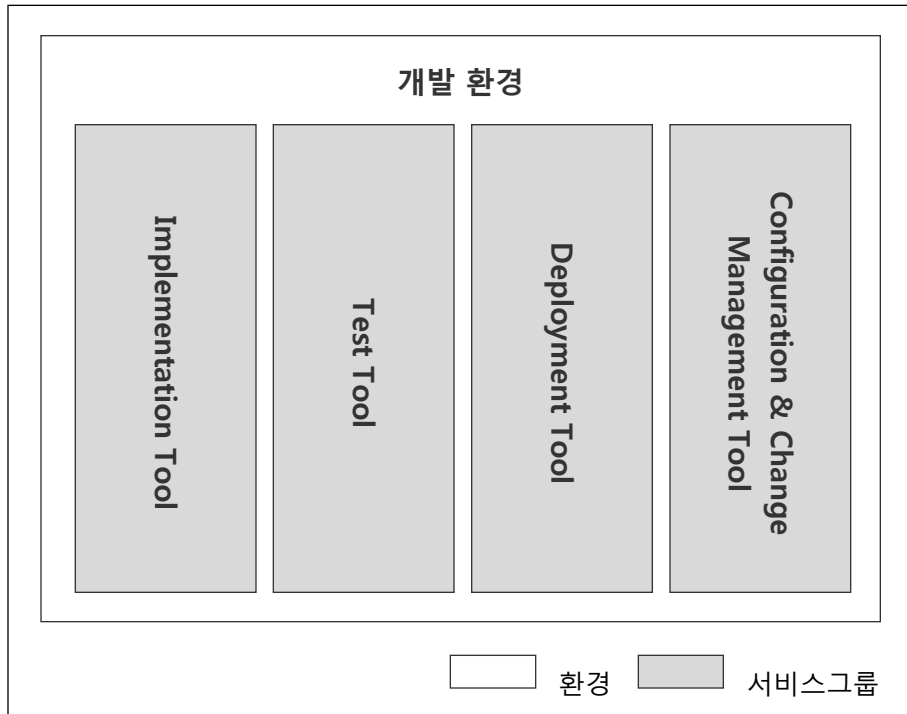
개발 환경 서비스 그룹 사례 분석

개발프레임워크 기술구조 사례 연구

개발프레임워크 (사업자 명)	요구분석자 지원 서비스	분석 설계자 지원 서비스 *	구현자 지원 서비스	테스터 지원 서비스	배포자 지원 서비스	형상/변경관리 자 지원 서비 스	프로젝트관리 자 지원 서비스
SYSTEMiER (삼성 SDS)	N/A	N/A	Code/Conf. Generation, Debugging	Test Automation, Test Reporting	Deployment	형상관리	N/A
Nexcore (SK C&C)	N/A	모델링	IDE, 템플릿 및 개발 표준, 코드 Inspection	단위 테스트, 성 능 테스트	자동 빌드	형상관리	N/A
DevOn (LG CNS)	N/A	모델링, MDS	개발환경 설치, IDE, Prototype, 코드 Inspection	단위 테스트, 성 능 테스트	자동 빌드	형상관리	N/A
Live Framework (현대정보기술)	N/A	N/A	Debugging	Testing	Deployment	Version Control	N/A
한화 프레임워크 (한화 S&C)	N/S	N/A	N/A	N/A	빌드	소스형상관리, 버그 추적관리	N/A
ProFrame (TmaxSoft)	N/A	RSD/RSA Transformatio n	디버깅, EMB Designer, ProMapper, DBIO Editor	Service Tester	Deploy Management	버전 관리, 개발 자 권한 관리	N/A

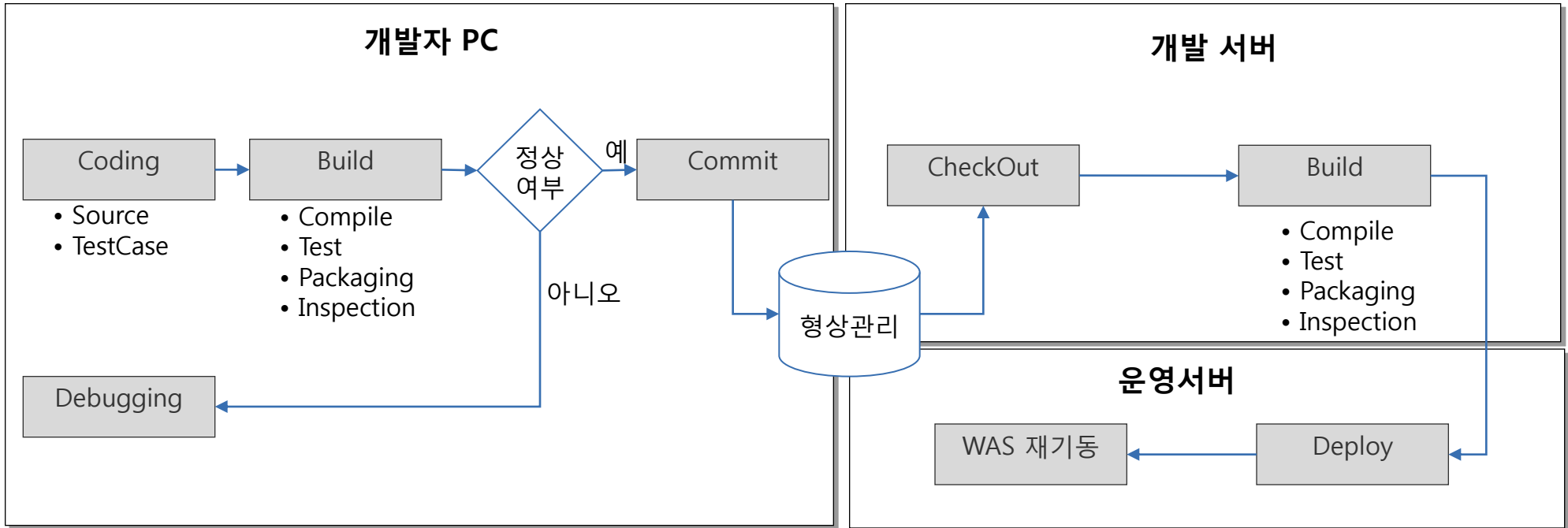
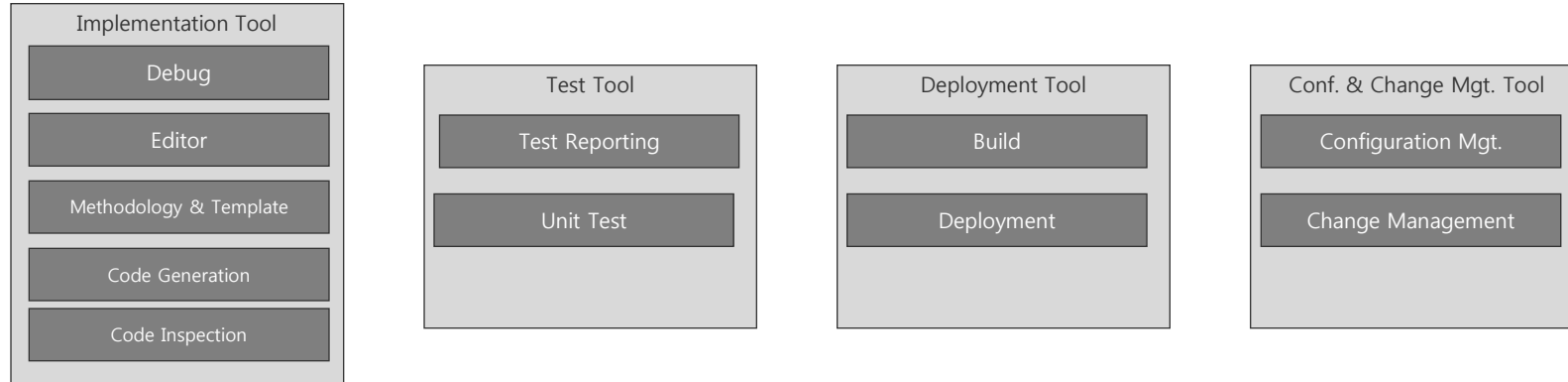
전자정부 개발프레임워크 개발 환경의 서비스 그룹은 **Implementation Tool, Test Tool, Deployment Tool, 그리고 Configuration & Change Management Tool** 등으로 구성됨

개발 환경 서비스 그룹 도출



서비스 그룹	설명
Implementation Tool	• 업무 프로그램 구현을 지원하는 도구임
Test Tool	• 구현된 업무 프로그램의 테스트를 지원하는 도구임
Deployment Tool	• 구현 완료된 업무 프로그램을 실행 환경에 배포 가능한 형태로 패키징하고, 패키징된 업무 프로그램을 실행 환경에 배포하는 도구임
Configuration & Change Management Tool	• 형상 및 변경 관리 지원 도구임

개발환경은 개발자 PC에서 개발서버, 운영서버로 연결되는 개발 흐름을 지원함



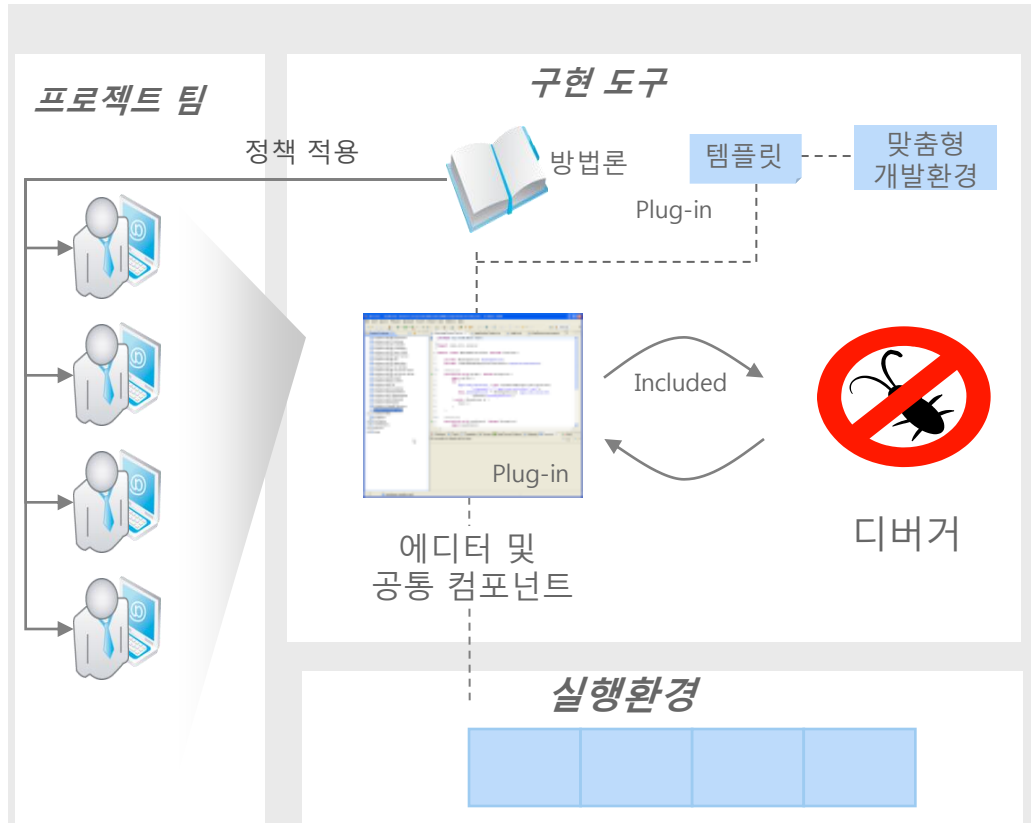
개발환경 오픈소스 SW 1,2차 평가를 통해 선정된 18개의 오픈소스 SW는 다음과 같음

개발환경 오픈소스 SW 선정결과

No	서비스 그룹	서비스	오픈소스	라이선스	
1	구현도구	Editor	Source Editor	Eclipse	EPL & EDL
2			UML Editor	AmaterasUML	EPL 1.0
3			ERD Editor	AmaterasERD	EPL 1.0
4			Web Flow	Spring Web Flow	N/A
5			Code Inspection	PMD, FindBugs	BSD, LGPL
6		Debug	Local Debug	Eclipse	EPL & EDL
7			Remote Debug	Eclipse	EPL & EDL
8	테스트도구	Unit Test	TestCase	Junit	CPL
9			Mock Support	EasyMock	MIT
10			DB Support	DbUnit	LGPL 2.1
11		Test Automation		Ant, Maven	Apache License 2.0
12		Test Reporting		Ant, Maven	Apache License 2.0
13		Test Coverage		EMMA	CPL
14		배포도구	Build	Build Tool	Maven
15	CI Server			Hudson	Creative Commons Attribution Share-Alike License, MIT
16	Nexus			Nexus	GPL
17	형상관리도구	Configuration Management		Subversion	Subversion License
18		Change Management		jTrac	Apache License 2.0

개발자의 코드 작성 및 디버깅을 지원하는 도구로 Eclipse 기반 하에 추가 Plug-in을 제공

주요 메커니즘

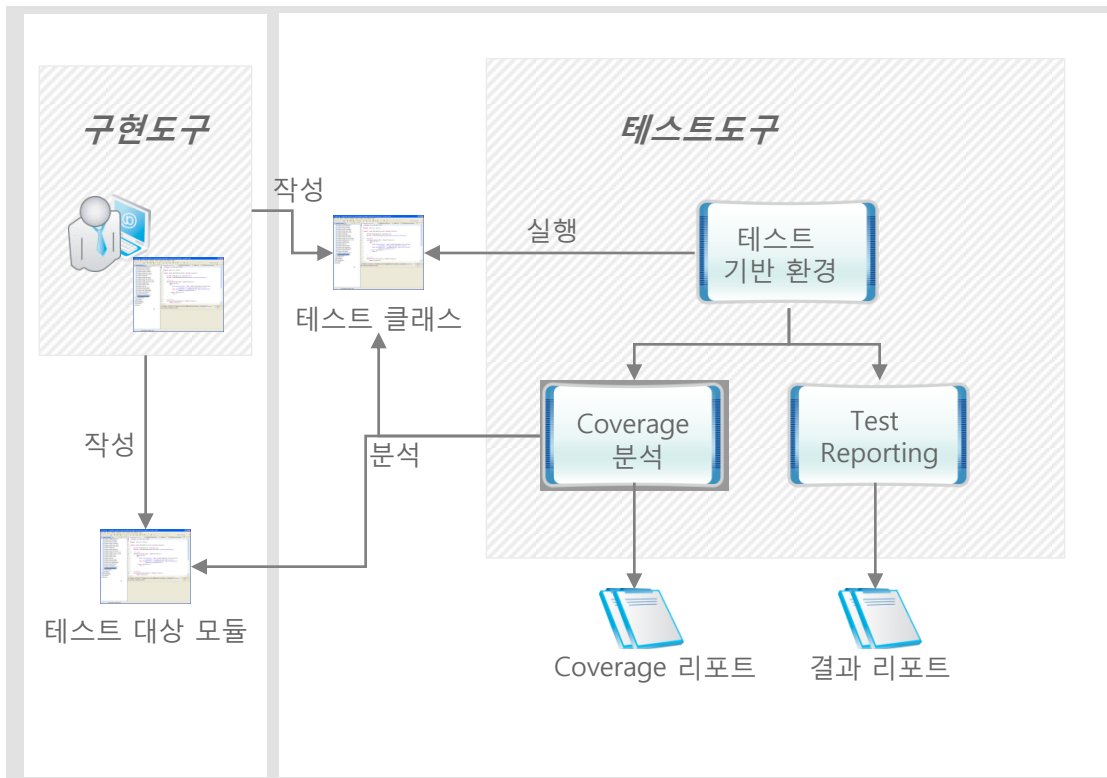


구성요소별 상세

구성요소	설명
에디터	<ul style="list-style-type: none"> • SourceCode Editor로 편리한 소스 코드 작성 환경 제공 • UML, ERD 에디터 제공 • DBIO 에디터를 통해 SQL작성 및 수행결과 확인 가능 • WebFlow Editor를 통해 설정 파일의 유효성 검사 및 에디팅 가능
디버거	<ul style="list-style-type: none"> • 에디터 내에서 편집모드와 디버깅 모드의 전환을 통한 작업의 편리성 • Remote 디버깅 제공
방법론 (개발절차)	<ul style="list-style-type: none"> • 애플리케이션을 개발하기 위한 가이드 제공
템플릿 프로젝트	<ul style="list-style-type: none"> • 애플리케이션 개발 시 자동으로 템플릿을 생성 해주는 마법사 제공
공통 컴포넌트	<ul style="list-style-type: none"> • 애플리케이션 개발 시 마법사를 통해 공통컴포넌트를 추가하는 기능 제공
맞춤형 개발환경	<ul style="list-style-type: none"> • 프로젝트에서 필요로 하는 기능만을 선택적으로 구성할 수 있는 도구 제공

개발자가 자신의 코드를 테스트하기 위해 JUnit 기반으로 TestCase의 작성 및 수행을 지원하고, 자동으로 테스트하고 리포팅하도록 지원하는 도구

주요 메커니즘

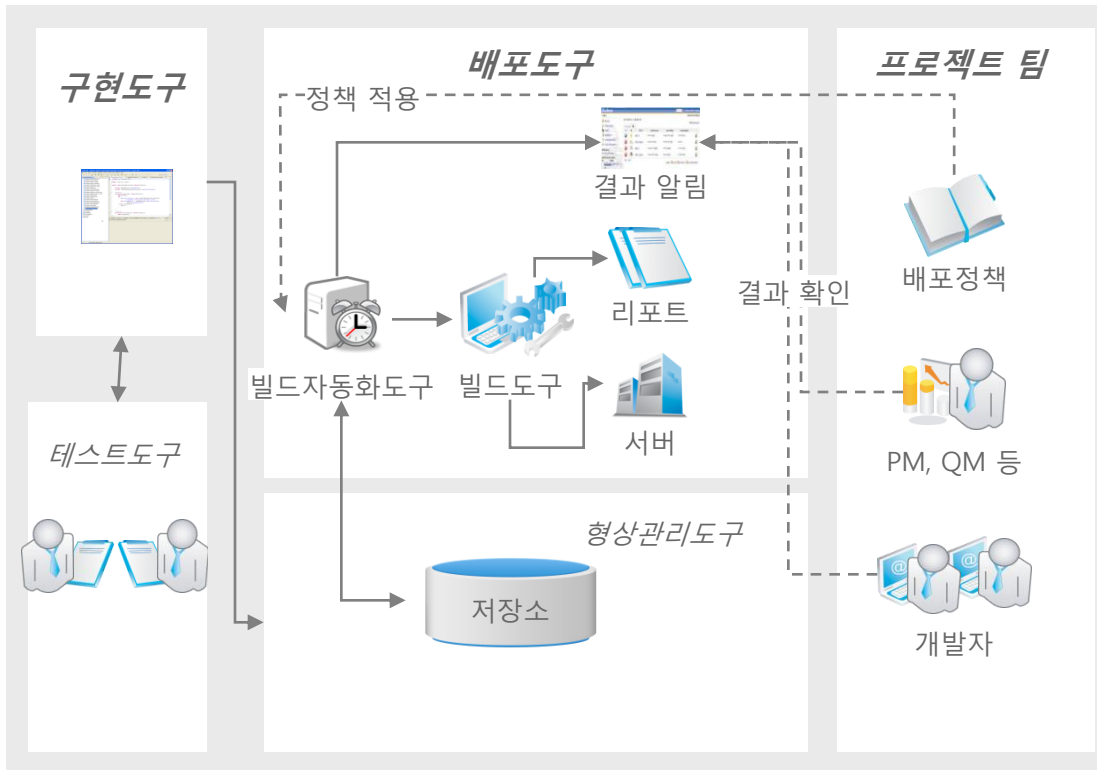


구성요소별 상세

구성요소	설명
테스트 기반환경	<ul style="list-style-type: none"> JUnit 기반의 Unit Test 작성 지원 Mock 및 DB Support
Test Reporting	<ul style="list-style-type: none"> Eclipse 기반의 Green Bar GUI HTML/XML/Excel 형식의 Reporting 지원
Coverage 분석	<ul style="list-style-type: none"> TestCase가 커버하는 정도를 수치 및 코드 영역을 통해 확인

Maven 기반의 개발자 빌드와 CI서버를 통한 지속적인 통합을 지향

주요 메커니즘

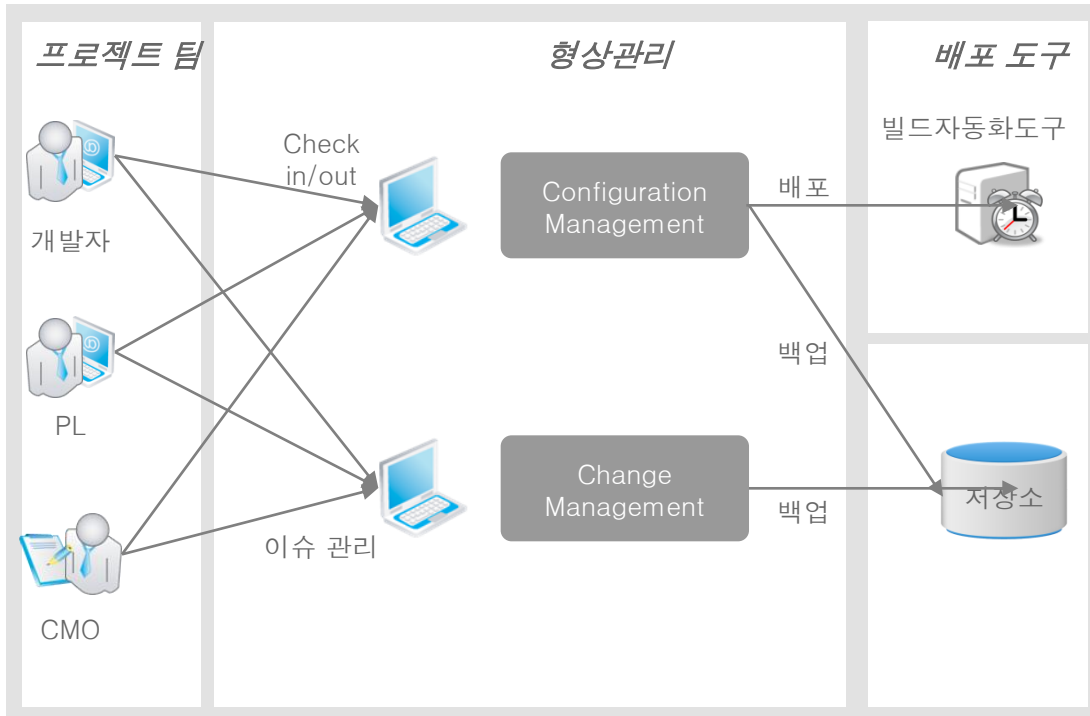


구성요소별 상세

구성요소	설명
빌드 도구	<ul style="list-style-type: none"> Maven 기반의 빌드 수행 3rd Party 라이브러리 관리 편의성을 위한 Nexus Repository 제공
빌드 자동화 도구	<ul style="list-style-type: none"> 형상관리도구와 연계하여 소스가 변경된 경우에만 빌드를 수행 배포 정책에 따라 설정된 주기에 자동 수행 빌드 결과에 대한 리포트 및 피드백 기능 제공 서버 이관 및 WAS 기동/중지

형상요소의 식별, 이력관리를 지원하는 형상도구와 JAVA기반의 이슈관리시스템 제공

주요 메커니즘



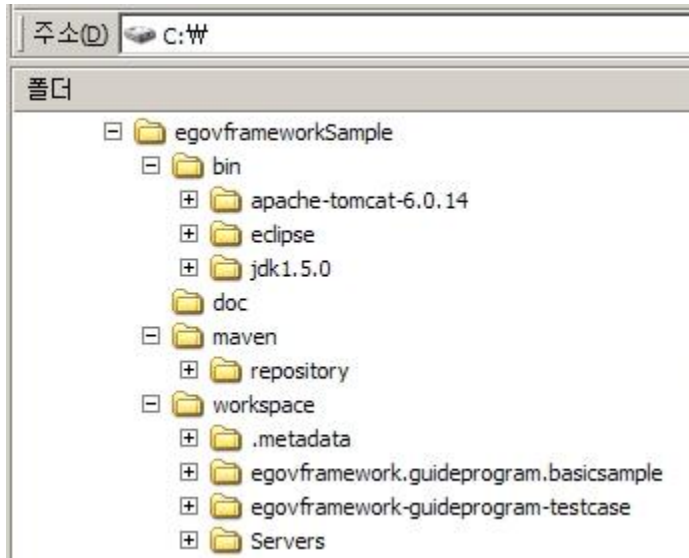
구성요소별 상세

구성요소	설명
Config. Mgmt	<ul style="list-style-type: none"> • Suversion을 그대로 활용 • 이슈ID 등록을 통한 이슈 연계
Change Mgmt	<ul style="list-style-type: none"> • JAVA기반의 이슈관리 시스템 • 이슈 등록, 검색, 처리 기능 제공

□ 설치

- eGovFrameDev-2.0.0.zip 파일을 C:\ 드라이브에서 압축을 푼다.

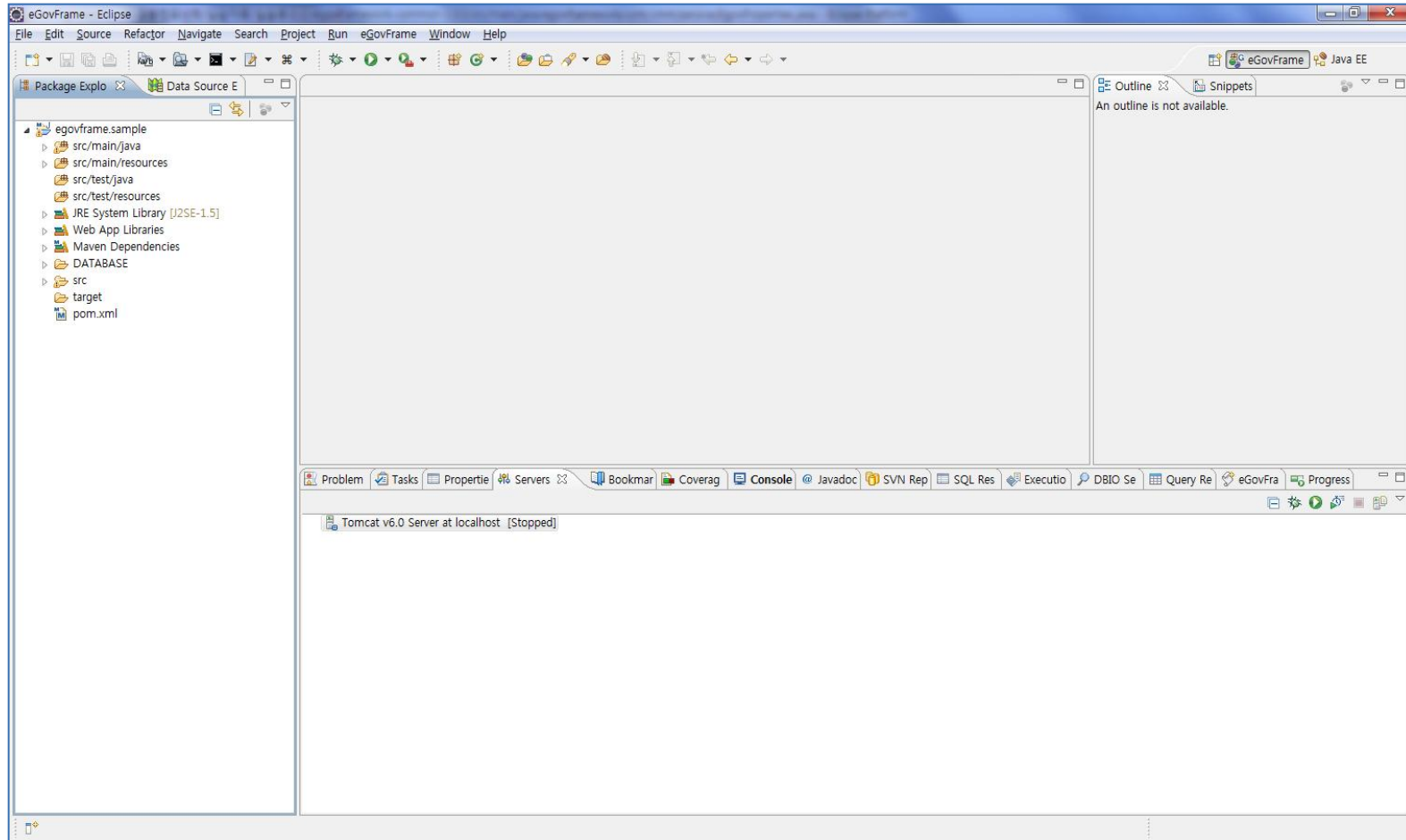
□ 디렉토리 정보



디렉토리	설명
bin	실행 파일 디렉토리
apache-tomcat-6.0.14	Apache Tomcat 6
jdk1.5.0 / jdk1.6.0	JDK_HOME
eclipse	개발환경이 포함된 Eclipse JEE Helios Sr2
maven/repository	Local Maven Repository
workspace	Eclipse Workspace

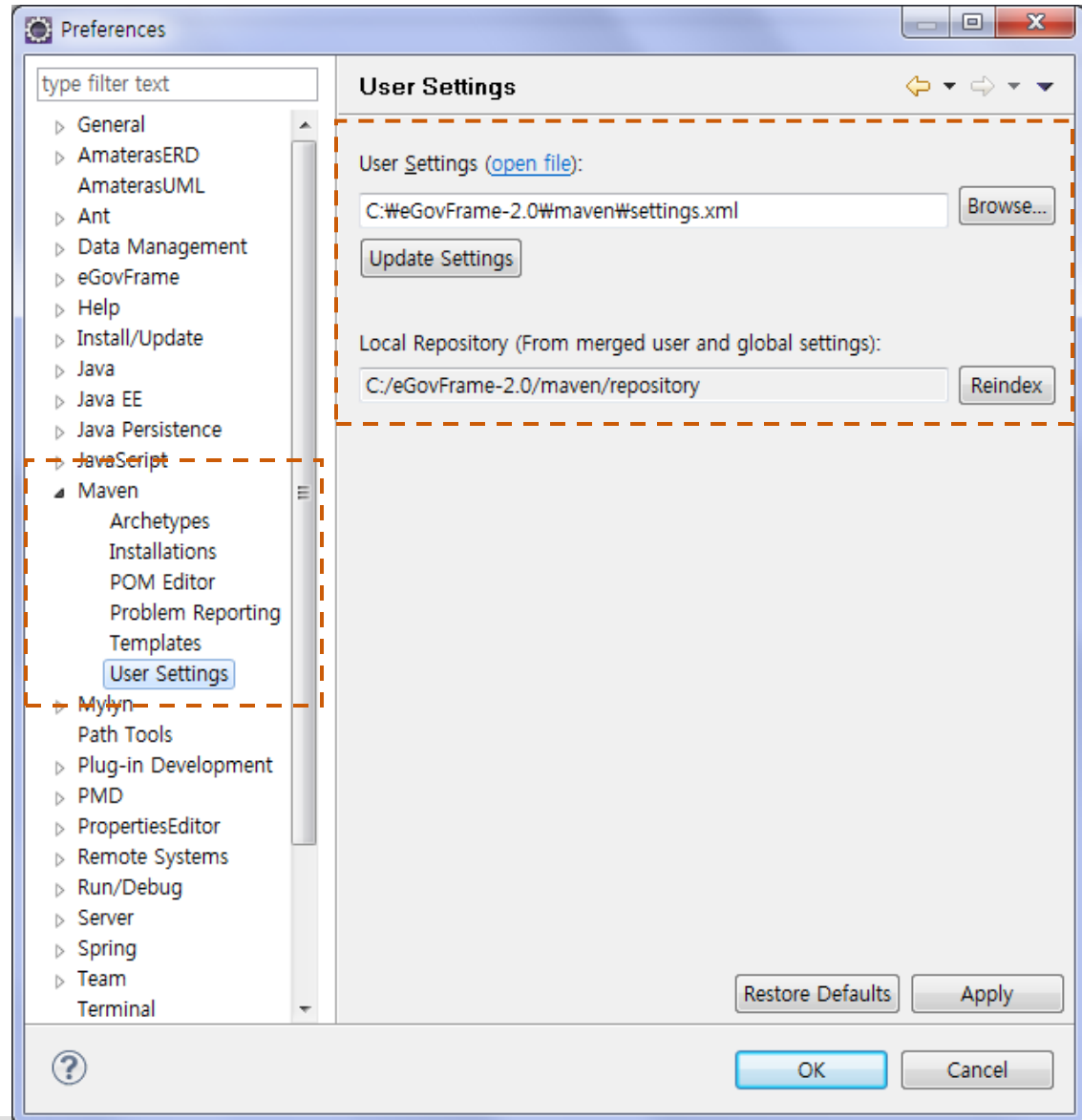
□ 실행

- “C:\eGovFrame-2.0\egovframe.edu 시작” 를 더블클릭하여 Eclipse를 실행 한다.



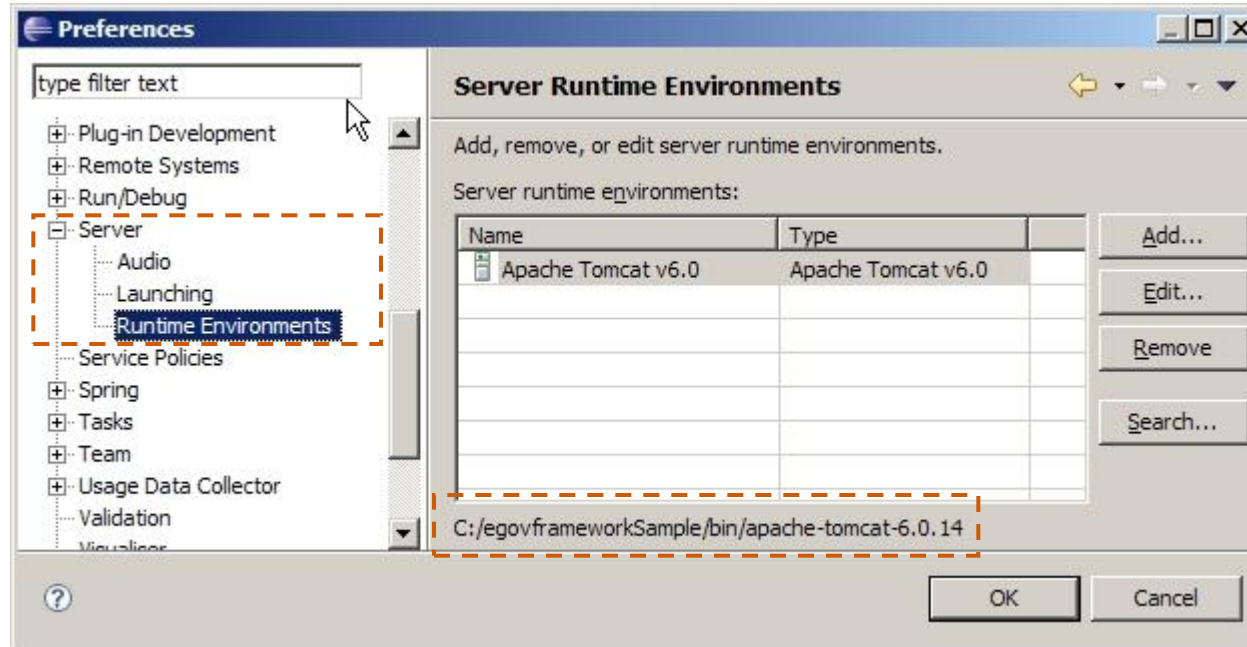
□ 설정

- Maven Installations 설정 확인



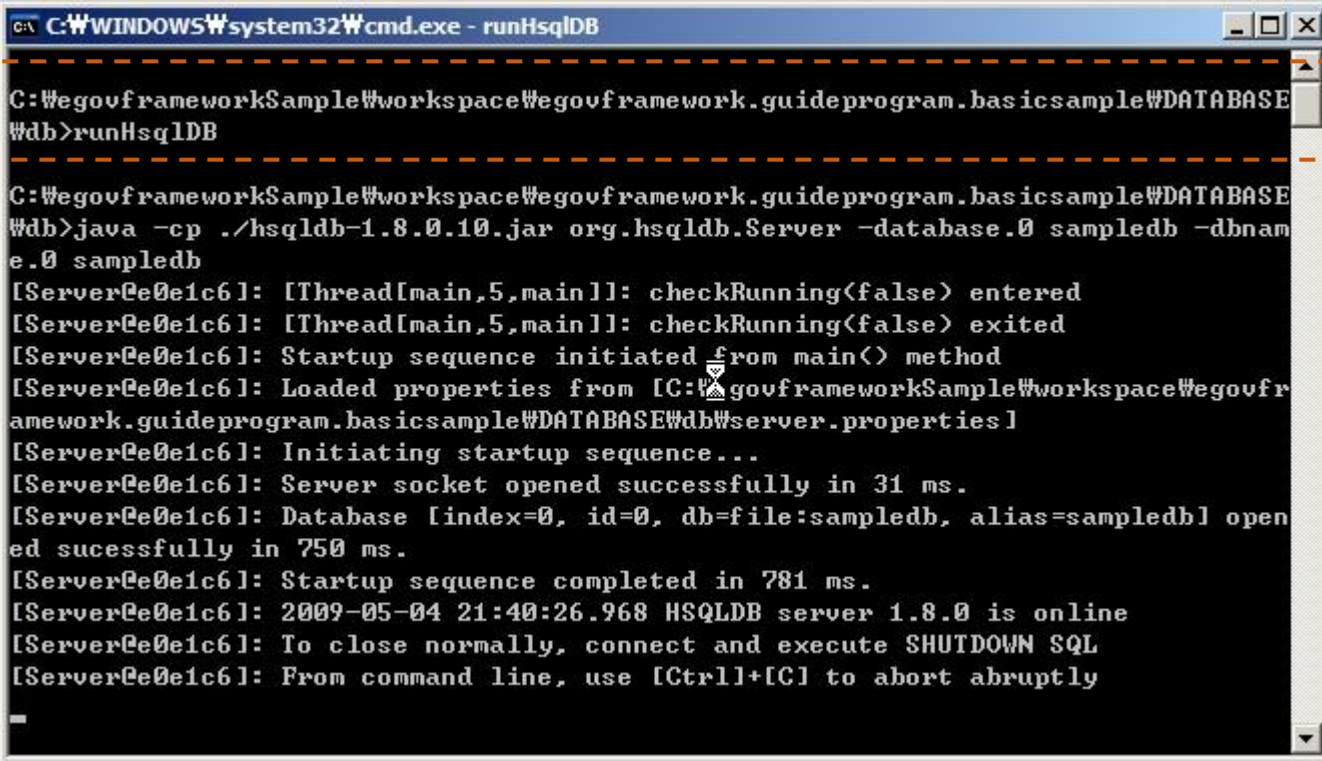
□ 설정

- Apache Tomcat 6 설정



□ 설정

- Sample 프로젝트 생성 후 DB 구동을 확인한다. (project name. egovframe.sample)
- C:\egovFrame-2.0\workspace.edu\egovframe.sample\DATABASE\db\runHsqlDB.cmd” 를 더블클릭하여 Sample용 DB를 띄운다.



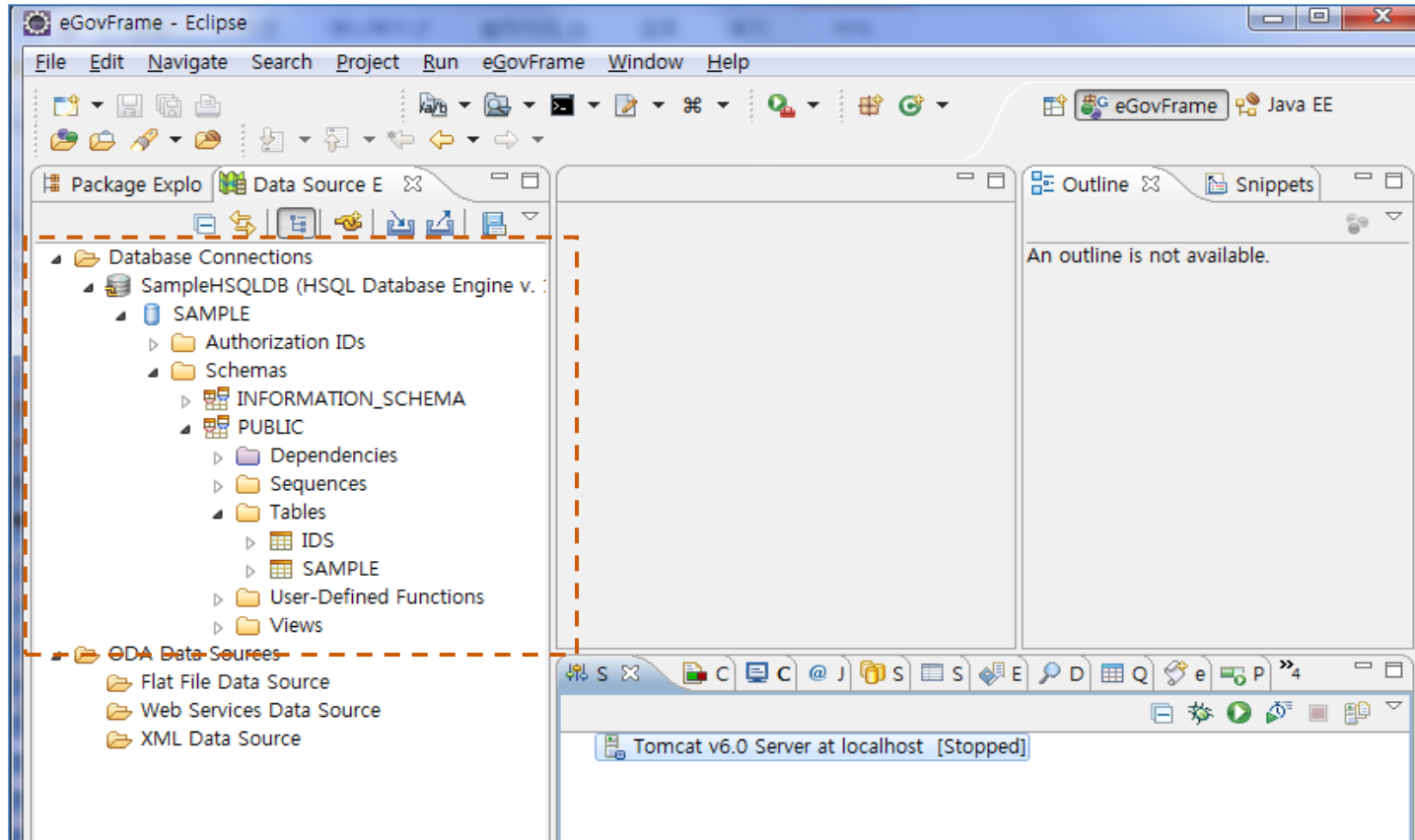
```
C:\WINDOWS\system32\cmd.exe - runHsqlDB

C:\egovframework\Sample\workspace\egovframework\guideprogram\basic\sample\DATABASE\Wdb>runHsqlDB

C:\egovframework\Sample\workspace\egovframework\guideprogram\basic\sample\DATABASE\Wdb>java -cp ./hsqldb-1.8.0.10.jar org.hsqldb.Server -database.0 sampledb -dbname.0 sampledb
[Server@e0e1c6]: [Thread[main,5,main]]: checkRunning(false) entered
[Server@e0e1c6]: [Thread[main,5,main]]: checkRunning(false) exited
[Server@e0e1c6]: Startup sequence initiated from main() method
[Server@e0e1c6]: Loaded properties from [C:\egovframework\Sample\workspace\egovframework\guideprogram\basic\sample\DATABASE\Wdb\server.properties]
[Server@e0e1c6]: Initiating startup sequence...
[Server@e0e1c6]: Server socket opened successfully in 31 ms.
[Server@e0e1c6]: Database [index=0, id=0, db=file:sampled, alias=sampled] opened successfully in 750 ms.
[Server@e0e1c6]: Startup sequence completed in 781 ms.
[Server@e0e1c6]: 2009-05-04 21:40:26.968 HSQLDB server 1.8.0 is online
[Server@e0e1c6]: To close normally, connect and execute SHUTDOWN SQL
[Server@e0e1c6]: From command line, use [Ctrl]+[C] to abort abruptly
```

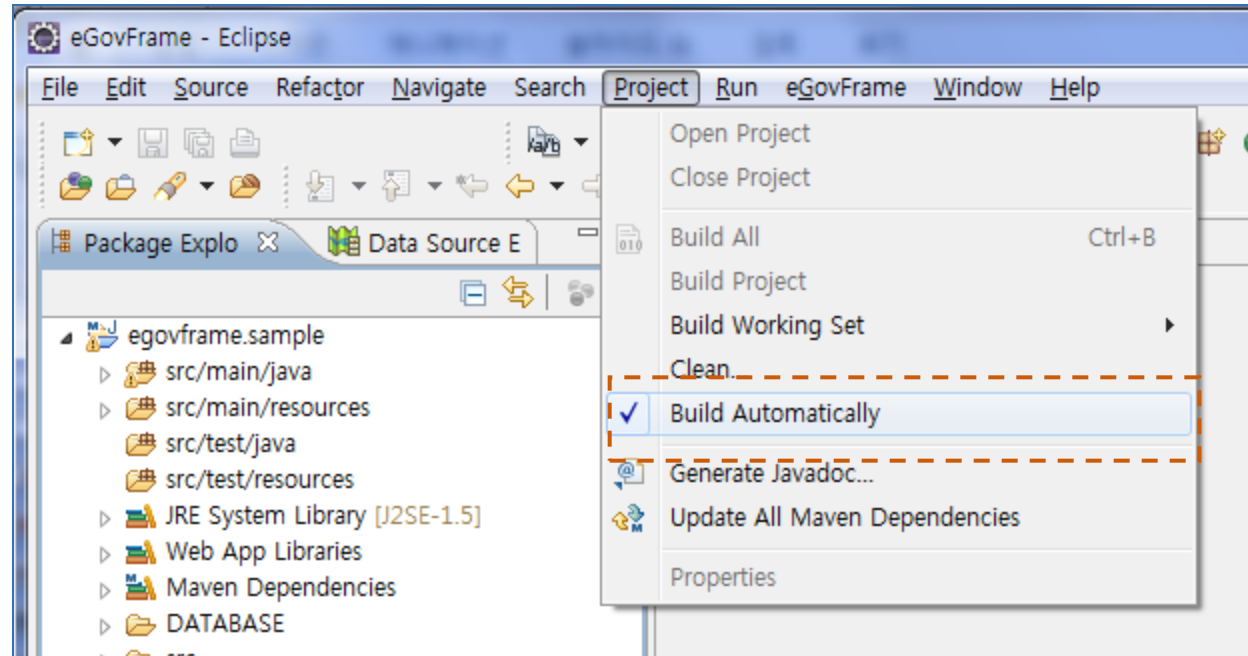
□ 설정

- Eclipse의 “Data Source Explorer” View에서 “SampleHSQLDB”에서 오른쪽 버튼의 “connect”를 선택하여 다음과 같이 테이블이 설정되어 있는지 확인한다.



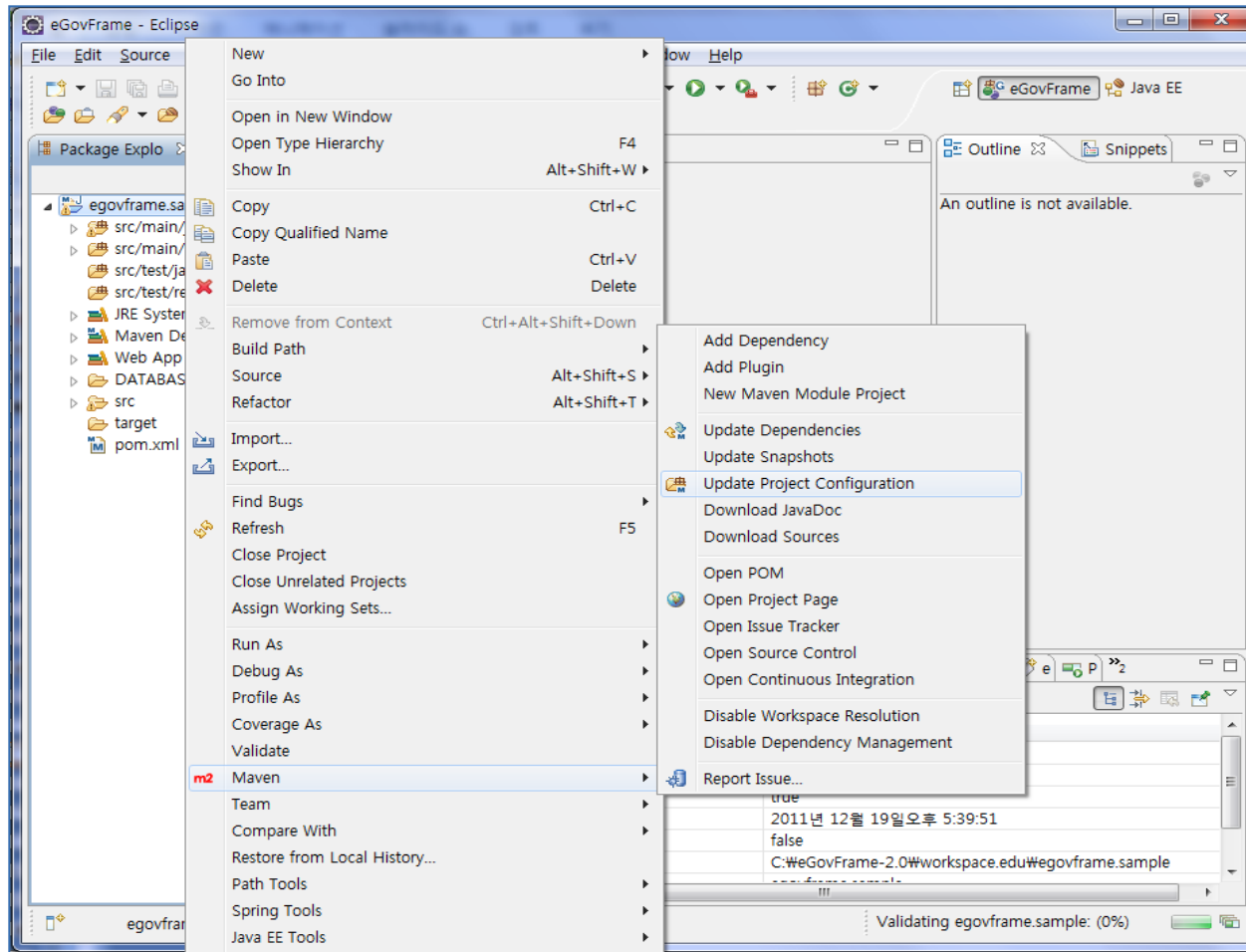
□ 설정

- Eclipse에서 “Build Automatically”가 선택되어 있다.



□ 설정

- 프로젝트 앞에 빨간 아이콘이 있어 오류가 있다면 Update Maven Configuration 을 실행해본다.



□ 다운로드

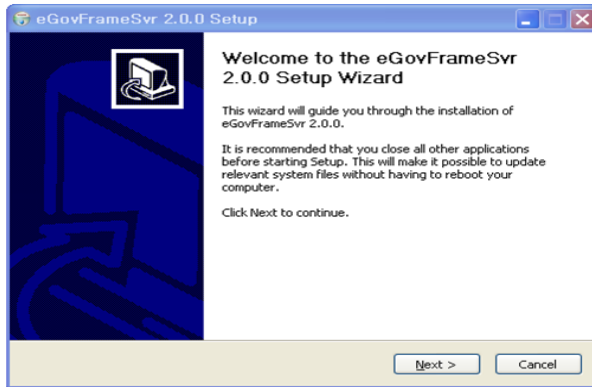
- eGovFrame 포털 (<http://www.egovframe.go.kr/>, <http://www.egovframe.org/>, <http://www.egovframe.kr/>)에 접속하여 “다운로드 > 개발환경”에서 제공하는 서버용 개발환경(Windows, Unix계열) 설치파일을 다운로드 한다.
- 다운로드 받은 파일을 C:\ 또는 D:\ 드라이브에서 압축을 해제한다.



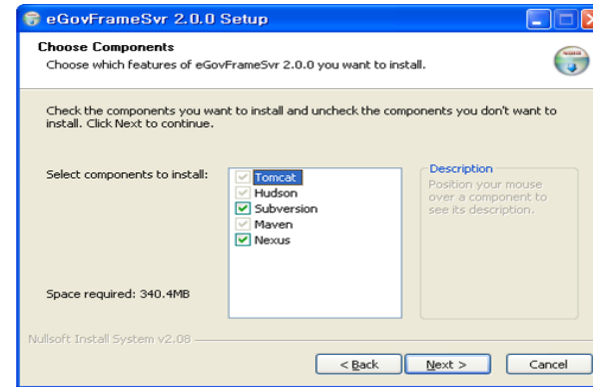
□ 설치

- eGovFrameSvr.exe 파일을 실행하여 설치를 진행한다.

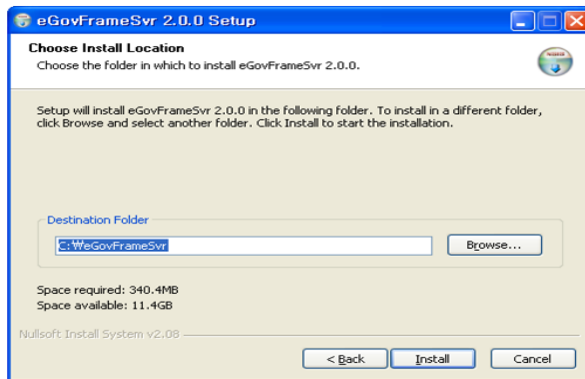
[설치화면 1/4 - eGovFrameSvr.exe 파일 실행]



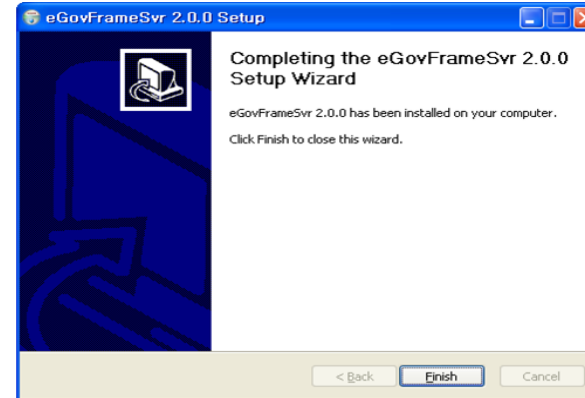
[설치화면 2/4 - 필수/선택요소 확인]



[설치화면 3/4 - 서버개발환경 설치 폴더 선택]

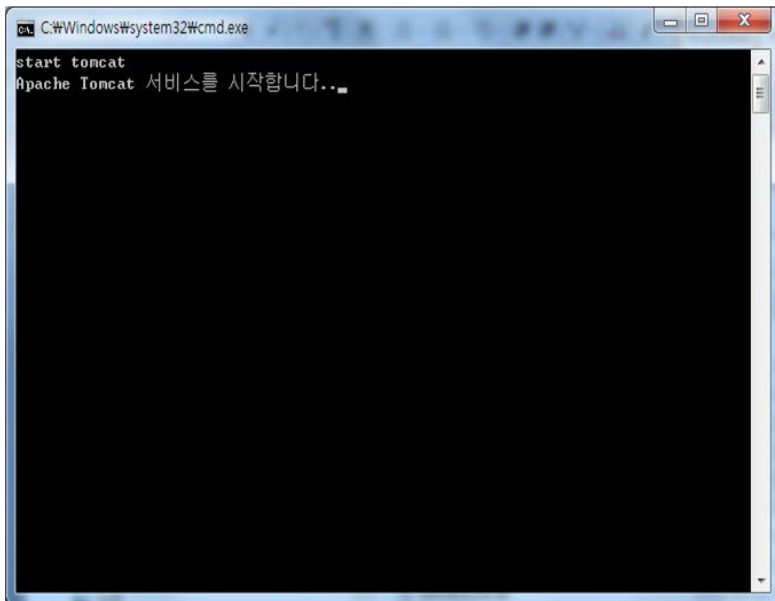


[설치화면 4/4]

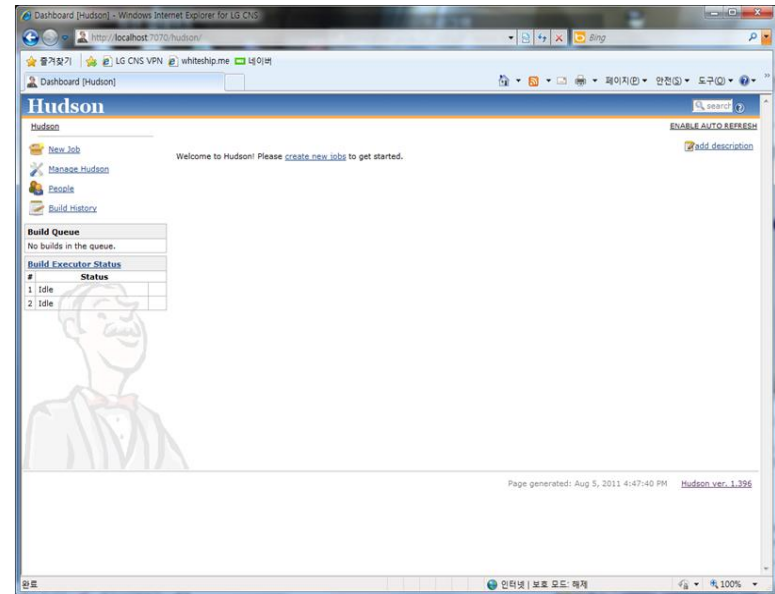


□ 실행

- 설치가 완료되면 필수요소 및 선택요소가 윈도우 서비스로 등록된다.
- 서비스 실행은 설치폴더\bin\start.bat 파일을 실행하여 진행한다. (Subversion 설치 시 해당 서비스도 수행됨)
- 브라우저 주소 입력창에 SVN, Nexus가 설치된 URL을 입력하여 설치 여부를 확인할 수 있다.



[그림 1] Tomcat이 설치된 모습



[그림 2] Hudson이 설치된 모습

□ 설치

- 서버 개발환경을 설치할 폴더에 eGovFrameSvr_2.0.0.tar 파일의 압축을 해제한다.
- 서버별 JDK를 설정한다.
- 서버 IP와 포트를 설정한다.
- Hudson 설치 확인 및 SVN, Nexus 설치 여부를 선택한다.(Hudson은 필수요소이며 SVN, Nexus는 선택요소임)

```
*****
*                                     *
*   eGovFrameSvr_2.0.0               *
*   (AIX Version)                    *
*                                     *
*****

Choose the bit of JDK 1.5
1: 32bit
2: 64bit
```

AIX버전의 JDK설정 화면

```
*****
*                                     *
*   eGovFrameSvr_2.0.0               *
*   (Solaris Version)                *
*                                     *
*****

Choose the bit of JDK 1.5
1: 32bit
2: 64bit
-
```

Solaris버전의 JDK설정 화면

```
*****
*                                     *
*   eGovFrameSvr_2.0.0               *
*   (linux Version)                  *
*                                     *
*****

Choose the bit of JDK 1.5
1: 32bit
2: 64bit
```

Linux버전의 JDK설정 화면

□ 설치

- 설치완료 후 정상적으로 설치가 되었는지 확인하고 서버를 실행한다.

```
[was:dev] /user/dev/ciserv/bin > ls
apr-1-config      install.jar
apu-1-config      install_aix.sh    start.sh          svnlook
config.ini        install_linux.sh  status.sh         svnservice
cvs               install_solaris.sh stop.bat          svnsync
cvshug            jython.jar       stop.sh           svnversion
getuid.pl         rcs2log          svn
install.bat       service.bat      svnadmin
```

정상적으로 파일이 설치되어있는지 확인한다.

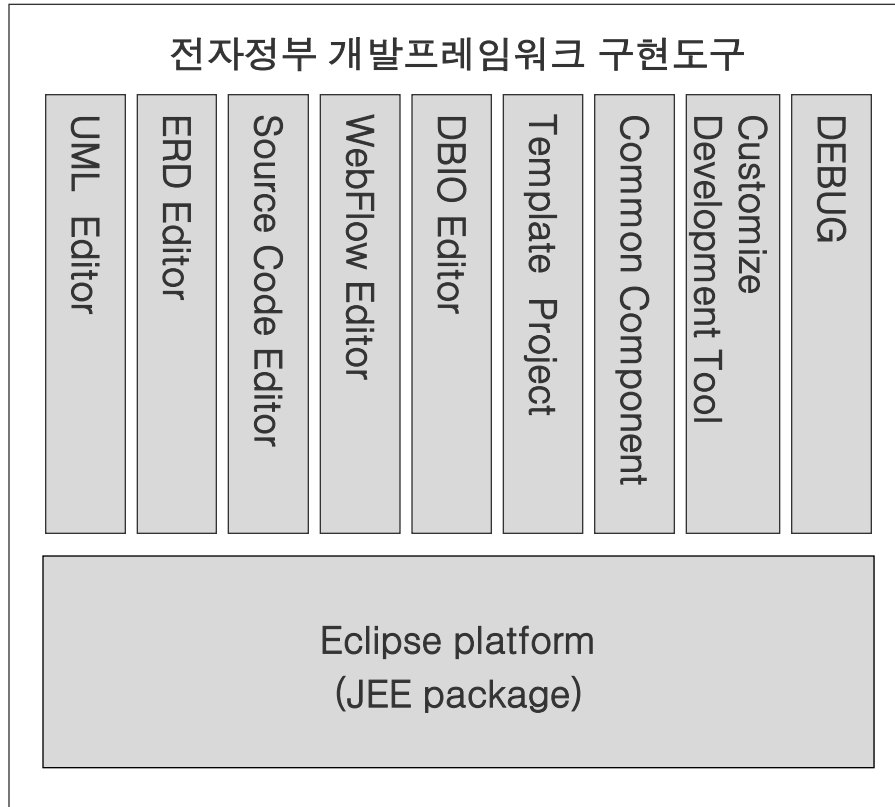
```
[was:dev] /user/dev/ciserv/bin > sh start.sh
start tomcat server e.g. hudson, syneton
Using CATALINA_BASE:   /user/dev/ciserv/tomcat6
Using CATALINA_HOME:   /user/dev/ciserv/tomcat6
Using CATALINA_TMPDIR: /user/dev/ciserv/tomcat6/temp
Using JRE_HOME:        /user/dev/ciserv/jdk1.5
done.
```

서버를 실행한다.

```
[was:dev] /user/dev/ciserv/bin >
```

1. 개요
2. eGovFrame IDE
3. Source Code Editor
4. eGovFrame Template Project
5. Common Component
6. Customize Development Tool
7. UML 작성
8. ERD 작성
9. DBIO 작성
10. Code Generation
11. Code Inspection
12. Server Connection Management
13. DEBUG - 로컬 디버깅
14. DEBUG - 리모트 디버깅
15. 참고자료

구현도구는 UML, ERD, Source Code, WebFlow, DBIO Editor와 Template Project, Common Component, Customize Development Tool 그리고 DEBUG로 구성됨



서비스 그룹	설명
UML	<ul style="list-style-type: none"> UseCase Diagram, Class Diagram 작성기능을 제공 AmaterasUML 선정
ERD	<ul style="list-style-type: none"> ER 논리모델과 물리모델 작성을 지원하며 DDL 생성, 테이블 명세서 생성, Reverse engineering 기능 AmaterasERD 선정
Source Code Editor	<ul style="list-style-type: none"> eGovFrame기반의 어플리케이션 개발 지원을 위한 프로젝트 생성 마법사, Perspective 및 Menu 등 기능 Eclipse 선정
WebFlow Editor	<ul style="list-style-type: none"> 복잡한 웹 어플리케이션의 흐름을 관리하기 위한 훌륭한 도구로 설정 파일의 유효성 검사와 에디팅 기능 제공 Spring Web Flow 선정
DBIO Editor	<ul style="list-style-type: none"> iBATIS 실행환경을 지원하기 위한 SQL Map 기반의 DBIO 작성 Editor
Template Project	<ul style="list-style-type: none"> 어플리케이션 개발시 편의성을 제공하기 위해 자동으로 템플릿을 생성 해주는 마법사 제공
Common Component	<ul style="list-style-type: none"> 전자정부 표준 프레임워크에서 제공하는 공통컴포넌트의 설치와 관련 파일 복사 및 공통컴포넌트 설정 정보를 관리해 주는 기능 제공
Customize Development Tool	<ul style="list-style-type: none"> 개발환경 구성 시 위저드를 통해 쉽고 빠르게 필요한 기능을 설치 할 수 있는 기능 제공
DEBUG	<ul style="list-style-type: none"> 로컬 또는 원격으로 실행 중인 프로그램에서 오류를 발견하고 진단할 수 있는 디버거 기능 Eclipse 선정

1. 개요 - Package 구성

2. 구현 도구

구현도구는 Eclipse Java EE를 기반으로 Package와 UML, ERD, DBIO Editor 와 Template Project, Common Component, Customize Development Tool 플러그인으로 구성됨

□ 구현도구가 Eclipse JEE 를 기반으로 하고 있으므로

아래와 같은 패키지가 포함되어 있다.

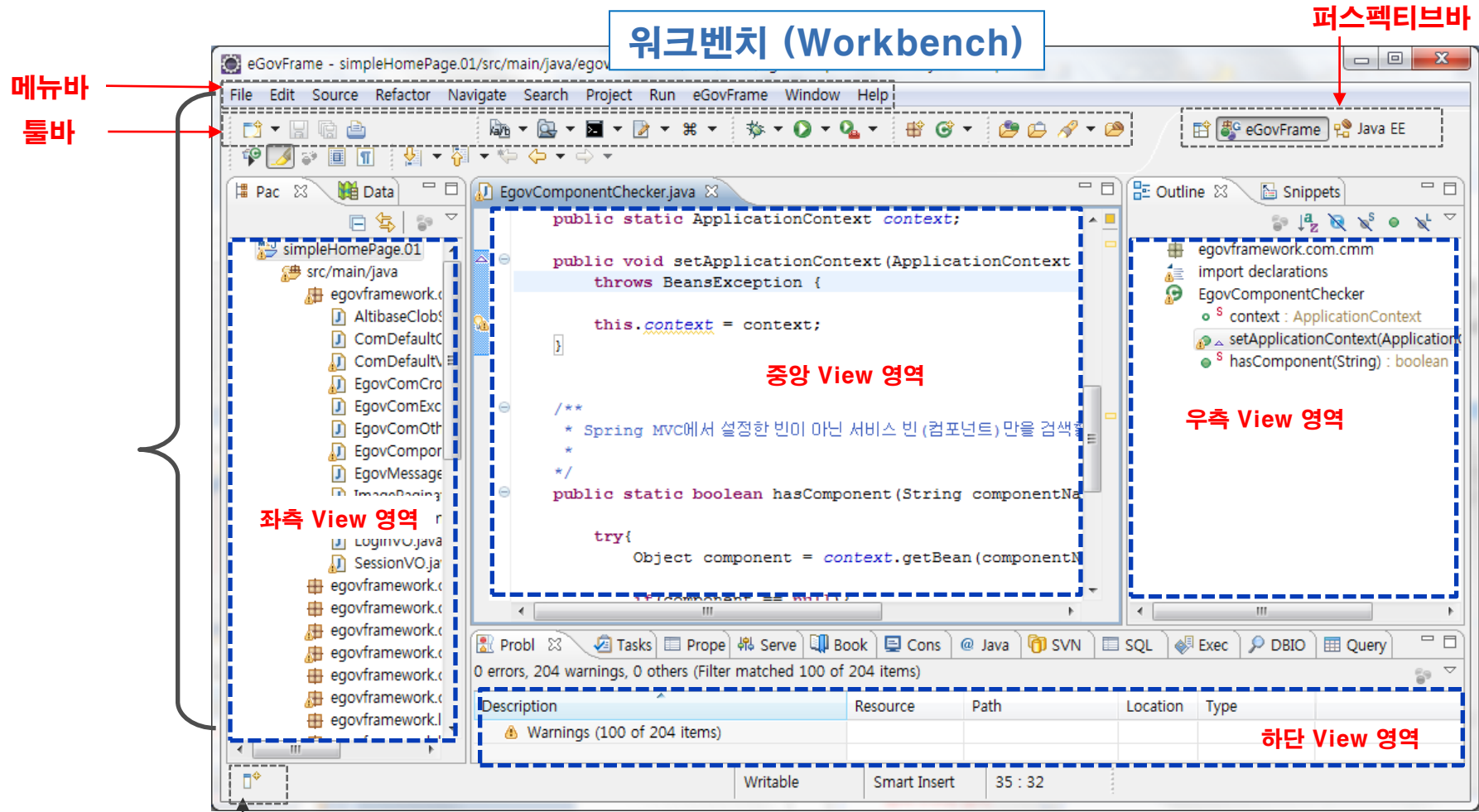
- RCP/Platform(Rich Client Platform)
- CVS(Concurrent Versions System)
- EMF(Eclipse Modeling Framework)
- GEF(Graphical Editing Framework)
- JDT(Eclipse Java development tools)
- Mylyn
- WTP (Web Tools Platform)
 - : Web Tools, Java EE Tools , XML Tools
- RSE(RSE, Target Management)
- Eclipse Link
- PDE(The Plug-in Development Environment)
- Datatools(DTP, Data Tools Platform)

	Java	Java EE	C/C++	C/C++ Linux	RCP/RAP	Modeling	JEE BIRT Reporting	Parallel	Scout	Testers	Javascript	Classic
RCP/Platform	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
CVS	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
EGit			✓	✓	✓	✓						
EMF	✓	✓				✓	✓					
GEF	✓	✓				✓	✓					
JDT	✓	✓			✓	✓	✓		✓			✓
Mylyn	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
Web Tools		✓					✓				✓	
Linux Tools			✓	✓				✓				
Java EE Tools		✓					✓					
XML Tools	✓	✓			✓		✓	✓				
RSE		✓	✓	✓			✓	✓				
EclipseLink		✓					✓			✓		
PDE		✓			✓	✓	✓		✓			✓
Datatools		✓					✓					
CDT			✓	✓				✓				
BIRT							✓					
GMF						✓						
PTP								✓				
MDT						✓						
Scout									✓			
Jubula										✓		
RAP					✓							
WindowBuilder	✓											
Maven	✓											

Legend:

✓ Included (with Source) ✓ Included ✓ Partially Included

Eclipse 기본 화면은 메뉴바, 툴바, 퍼스펙티브바, 4개의 View 영역으로 크게 구분된다. 이 4개의 뷰를 포함하는 전체 영역을 퍼스펙티브(Perspective) 라고 하고, 이 Eclipse Window 전체를 워크벤치(Workbench) 라 부른다.

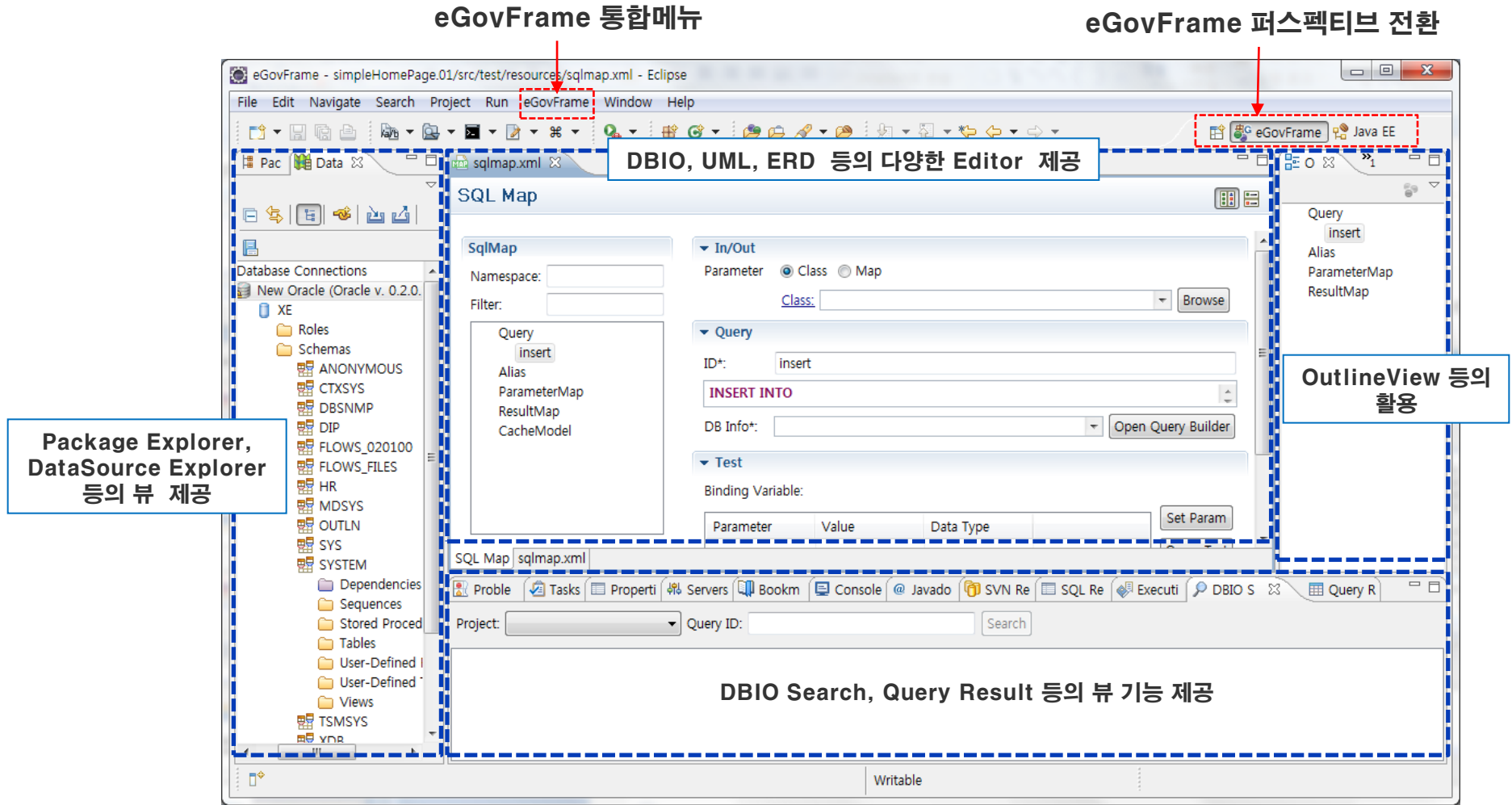


Fast 뷰 : 잘 사용하지 않는 화면들을 최소화 시켜놓고 필요한 시점에만 잠깐 활성화 시킬 때 사용된다.

1. 개요 - 화면 구성

2. 구현 도구

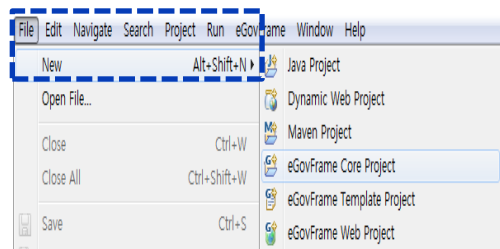
구현도구는 개발자가 개발환경을 쉽고 편리하게 개발할 수 있도록 특성화된 퍼스펙티브, 통합메뉴, 뷰, 에디터 등을 제공한다.



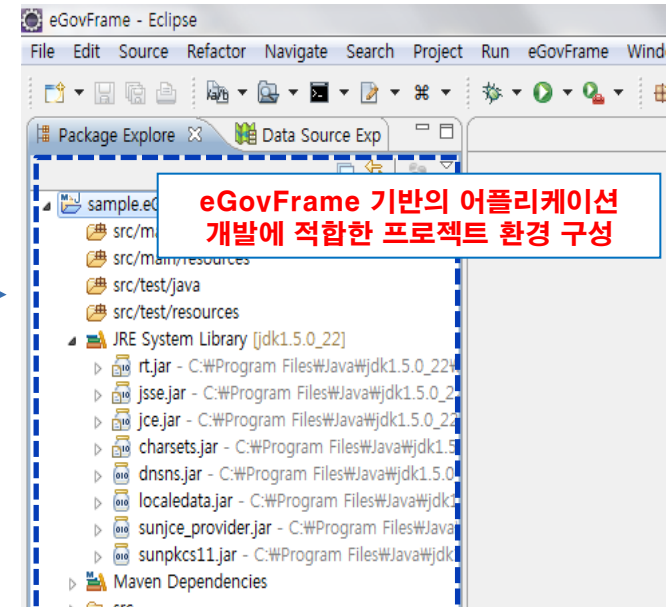
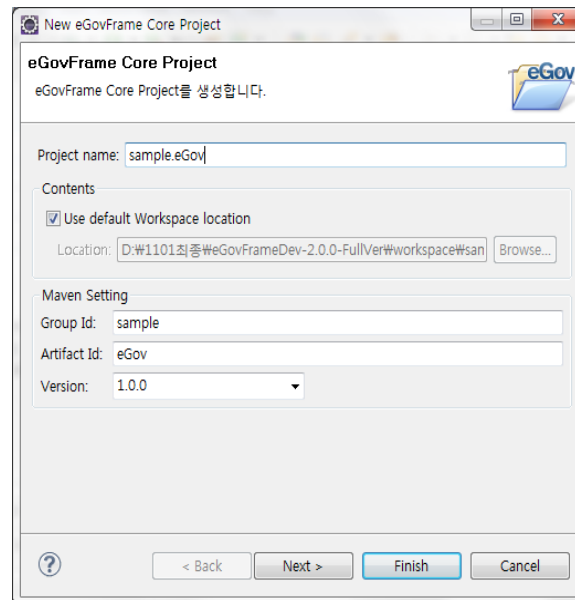
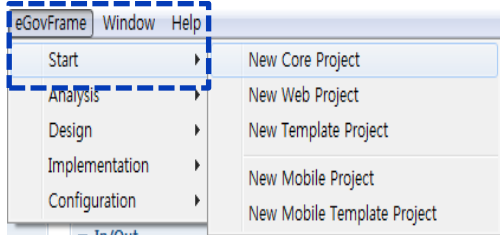
구현도구는 실행환경 기반의 어플리케이션을 개발하는 개발자의 편의성을 위하여 **Eclipse** 기반의 **Perspective, Menu, 프로젝트 생성 마법사** 등을 제공한다.

❑ 프로젝트 생성 마법사

- eGovFrame Core Project : 일반적인 JAVA Project 생성
- eGovFrame Web Project : Web 기반 프로젝트 생성



2가지 생성 방식



□ eGovFrame Perspective

- eGovFrame기반의 어플리케이션 개발을 위한 최적의 View 초기 세트와 레이아웃을 제공

eGovFrame 통합메뉴

eGovFrame 퍼스펙티브 전환

The screenshot displays the Eclipse IDE with the eGovFrame perspective. The interface includes a menu bar with 'eGovFrame' highlighted, a toolbar with various icons, and several views: Package Explorer on the left, SQL Map editor in the center, and OutlineView on the right. A bottom toolbar contains 'DBIO Search' and 'Query Result' buttons. Red callout boxes highlight specific features: 'eGovFrame 통합메뉴' points to the menu bar; 'eGovFrame 퍼스펙티브 전환' points to the 'eGovFrame' button in the toolbar; 'DBIO, UML, ERD 등의 다양한 Editor 제공' points to the central editor area; 'Package Explorer, Data Source Explorer 등의 뷰 제공' points to the Package Explorer; 'OutlineView 등의 활용' points to the OutlineView; and 'DBIO Search, Query Result 등의 뷰 기능 제공' points to the bottom toolbar.

DBIO, UML, ERD 등의 다양한 Editor 제공

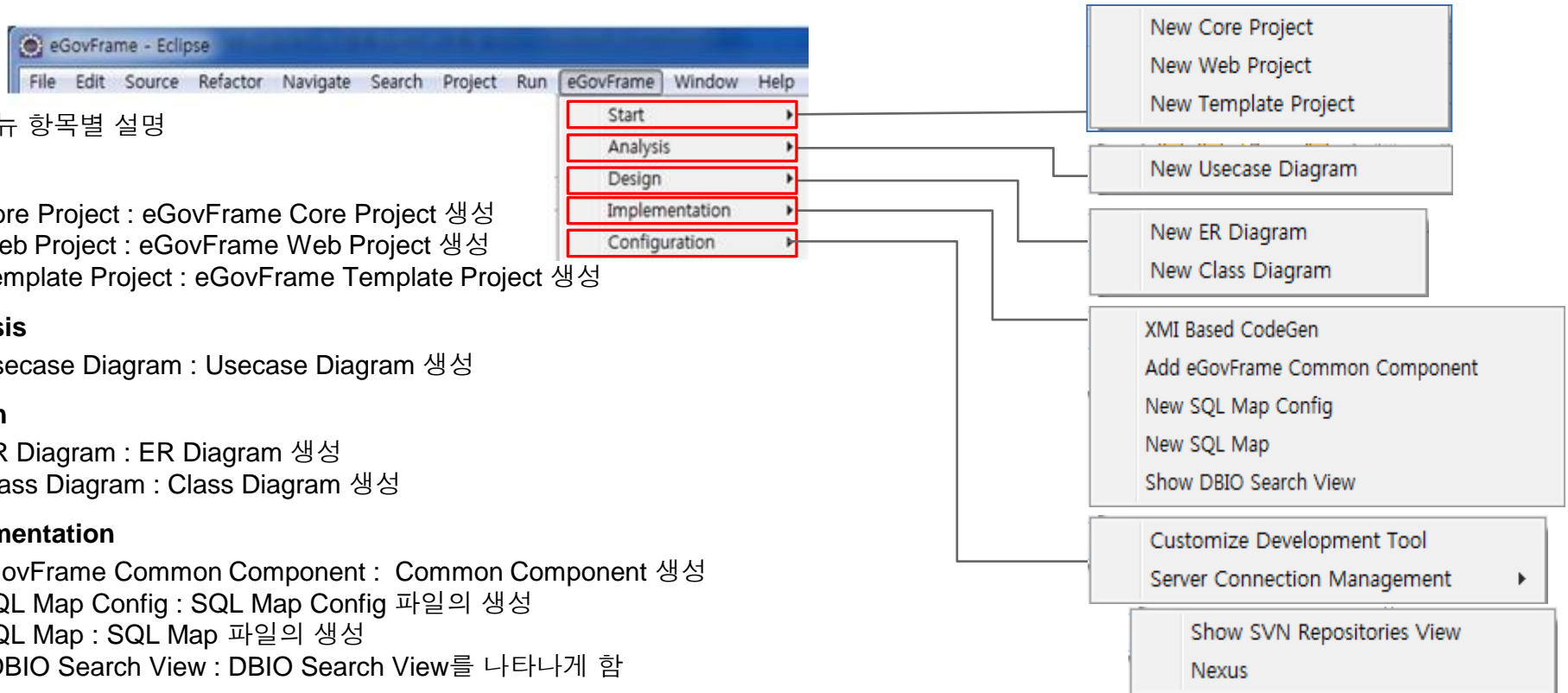
Package Explorer,
Data Source Explorer
등의 뷰 제공

OutlineView 등의
활용

DBIO Search, Query Result 등의 뷰 기능 제공

□ eGovFrame Menu

- eGovFrame Perspective에서만 활성화되는 메뉴로 eclipse내에서 분산되어 있는 플러그인들의 기능 (eGovFrame에서 필히 사용되어지는 기능)을 빠르게 접근할 수 있는 통합 메뉴를 제공



* 세부 메뉴 항목별 설명

□ Start

- New Core Project : eGovFrame Core Project 생성
- New Web Project : eGovFrame Web Project 생성
- New Template Project : eGovFrame Template Project 생성

□ Analysis

- New Usecase Diagram : Usecase Diagram 생성

□ Design

- New ER Diagram : ER Diagram 생성
- New Class Diagram : Class Diagram 생성

□ Implementation

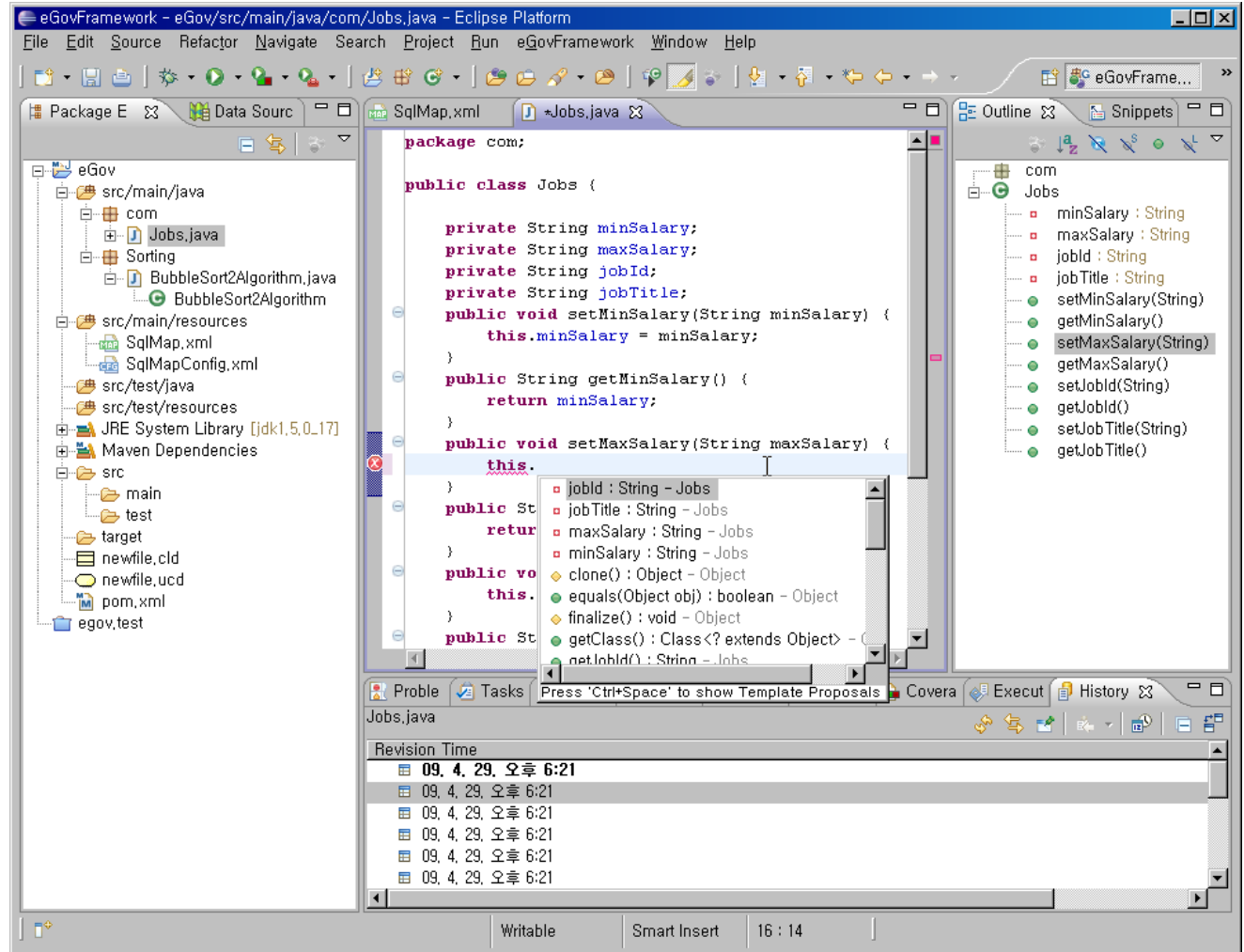
- Add eGovFrame Common Component : Common Component 생성
- New SQL Map Config : SQL Map Config 파일의 생성
- New SQL Map : SQL Map 파일의 생성
- Show DBIO Search View : DBIO Search View를 나타나게 함

□ Configuration

- Customize Development Tool : 필요한 기능을 선택적으로 설치
- Server Connection Management
 - Show SVN Repositories View : SVN Repositories View를 나타나게 함
 - Nexus : Nexus Repository 정보 관리

Source Code Editor는 Eclipse 내 Eclipse Java development tools(JDT)를 통해 지원

- ❑ Code Assist : Code 도움말 기능
- ❑ Quick Fix : Code 에러 빠른 수정 가능
- ❑ 코드 스타일 적용
- ❑ 코드 템플릿 처리
- ❑ Quick Type Hierarchy : 상속구조 표시
- ❑ Quick Outline : 코드 개요 표시
- ❑ 소스 코드 네비게이션
- ❑ Mark Occurrences
 - 커서가 위치한 지역변수, 상수, 필드, 메소드, 클래스 등의 코드 요소가 나타나는 곳이 모두 하이라이트된다
- ❑ 로컬 히스토리 : file 수정 이력 비교/복구



eGovFrame Template Project는 유형별 템플릿 프로젝트를 자동 생성하는 기능을 제공한다.

□ Simple Homepage

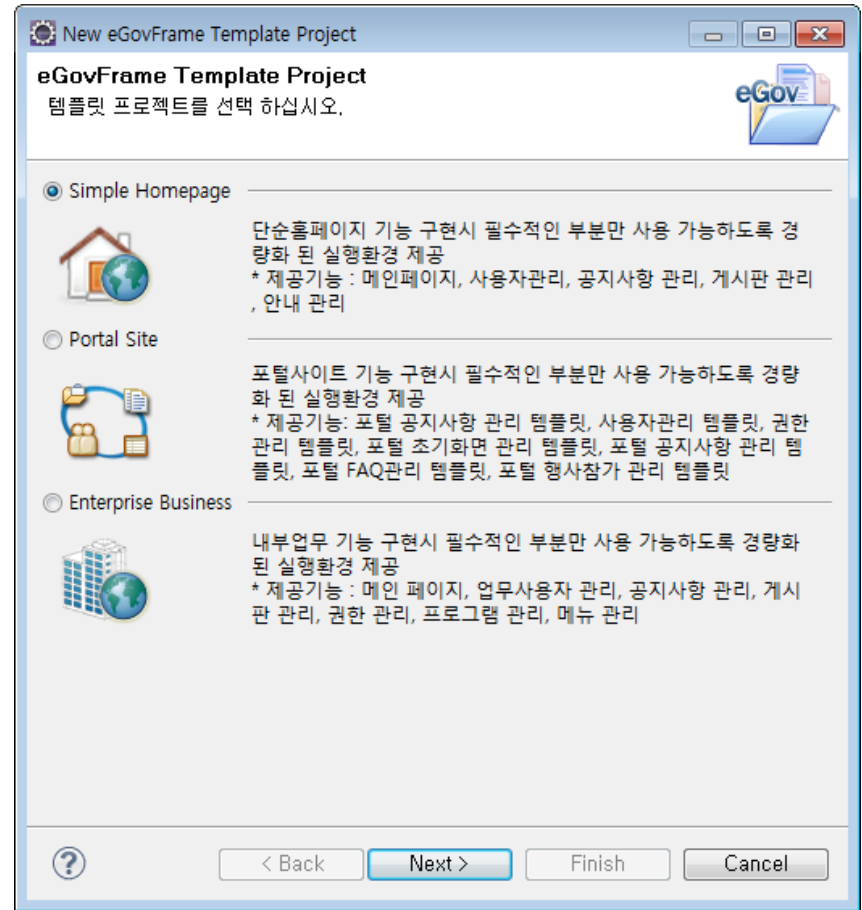
- 메인페이지, 사용자관리, 공지사항 관리, 게시판 관리, 안내 관리 기능 제공

□ Portal Site

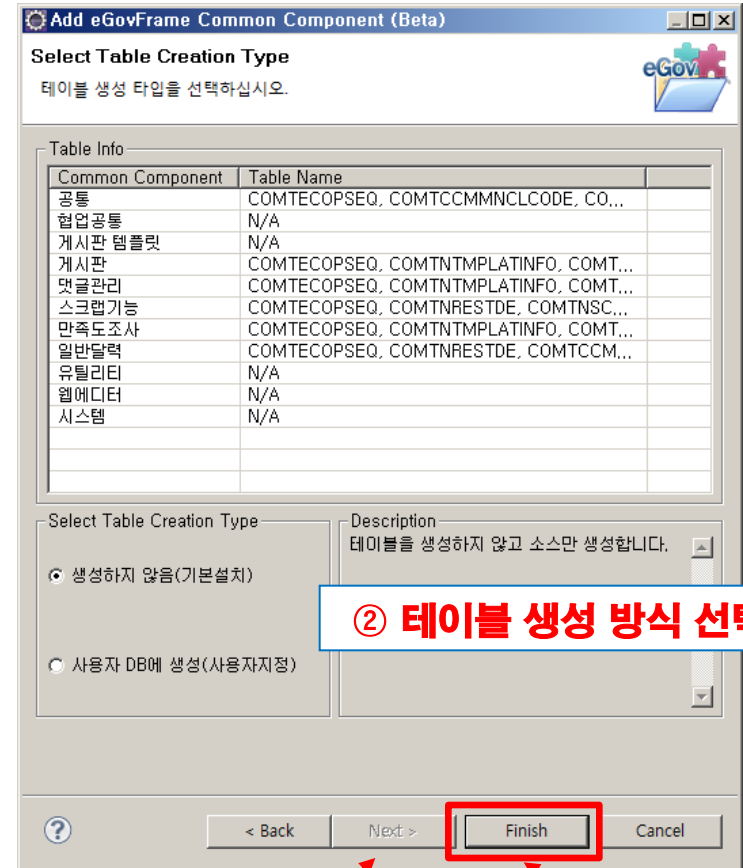
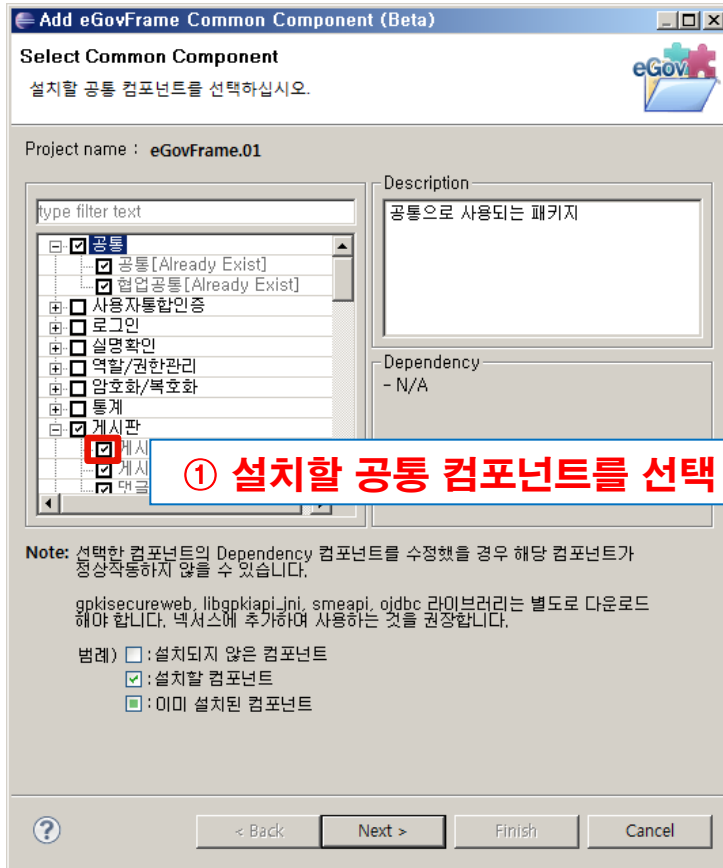
- 포털 공지사항 관리, 사용자 관리, 권한 관리, 포털 초기화면 관리, 포털 공지사항 관리, 포털 FAQ관리, 포털 행사참가 관리 기능 제공

□ Enterprise Business

- 메인 페이지, 업무 사용자 관리, 공지사항 관리, 게시판 관리, 권한 관리, 메뉴 관리 기능 제공



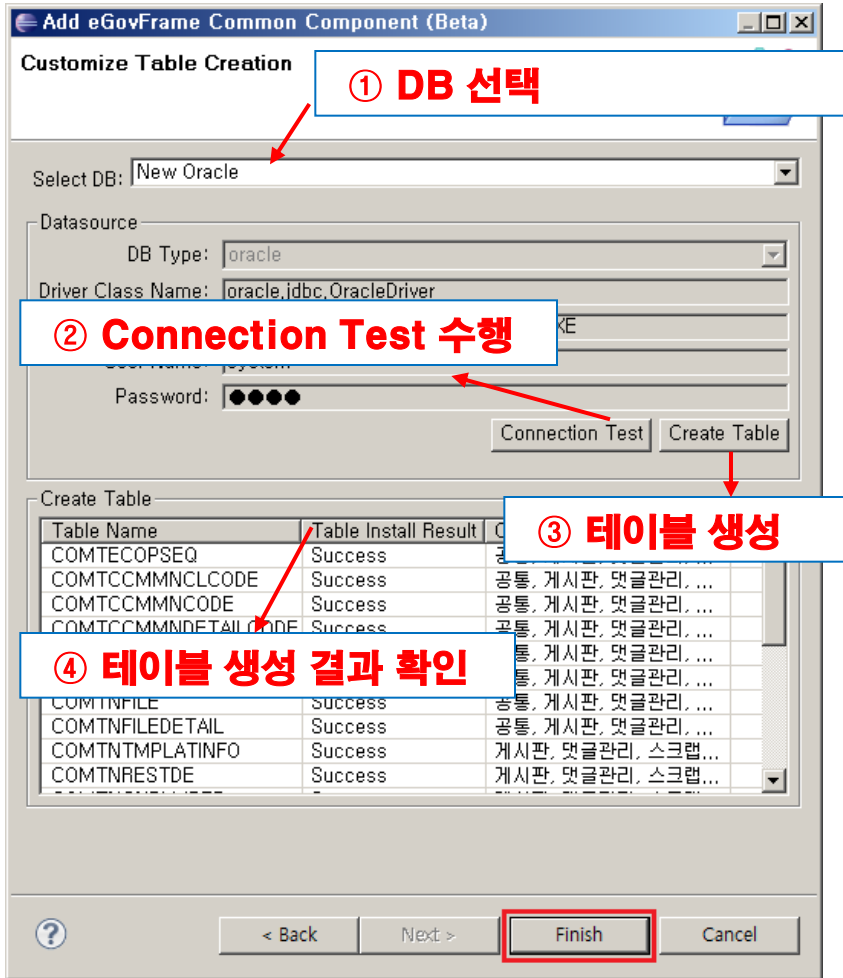
개발자의 편의성을 위하여 전자정부 표준 프레임워크에서 제공하는 **공통컴포넌트 219종**의 설치 마법사를 제공한다.



**사용자지정 선택 시
Next 버튼 활성화**

**기본설치 선택 시
Finish 버튼 활성화**

개발자의 편의성을 위하여 전자정부 표준 프레임워크에서 제공하는 **공통컴포넌트 219종**의 설치 마법사를 제공한다.

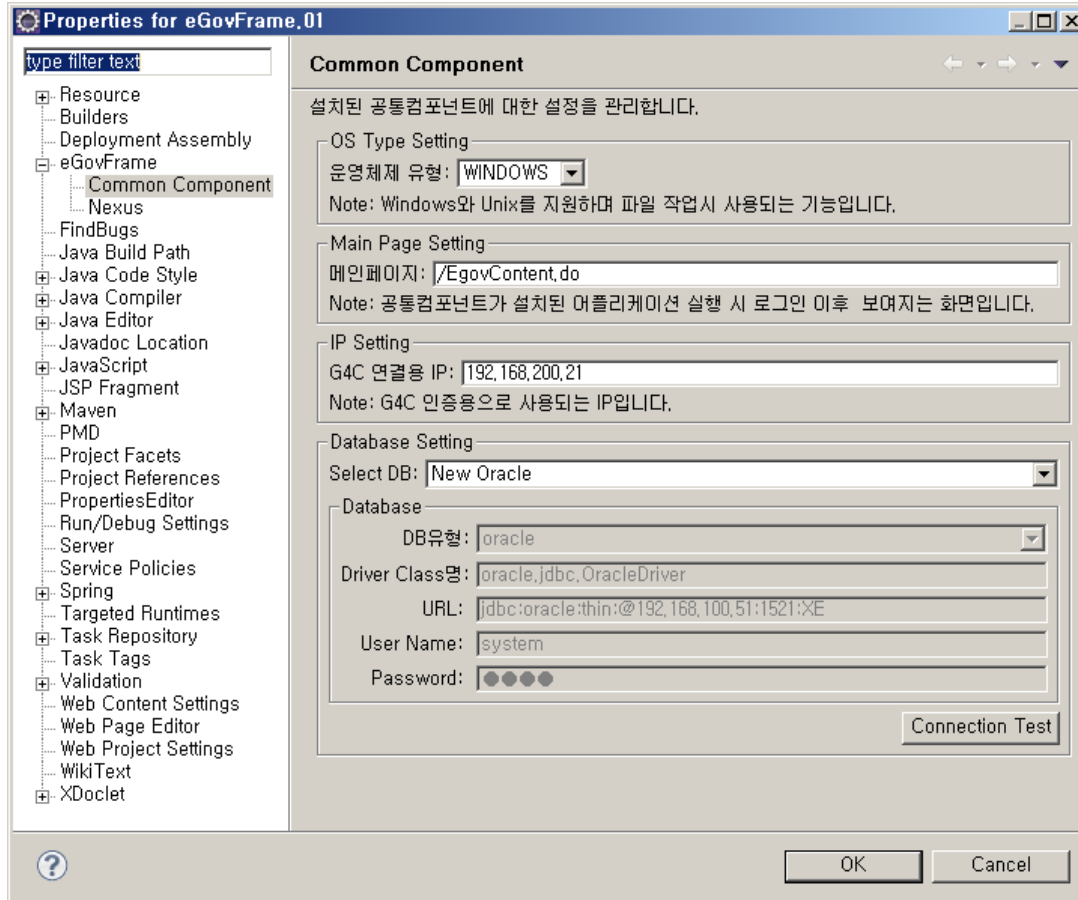


□ Table 생성 절차

- ① **Select DB**에서 사용할 **DB**를 선택한다.
- ② **Connection Test** 를 수행한다.
- ③ **Create Table** 버튼을 눌러 테이블을 생성한다.
- ④ 테이블 생성 결과를 확인한다.

(**'Prepared', 'Success', 'Already Exist', 'Failed'** 의 4개의 결과값으로 결과 확인 가능)

개발자의 편의성을 위하여 전자정부 표준 프레임워크에서 제공하는 공통컴포넌트 219종의 설치 마법사를 제공한다.



□ 운영체제 유형

- 사용자의 운영체제를 선택한다.

□ 메인페이지

- 선택한 공통 컴포넌트 설치 후 실행 시 시작화면을 보여준다.

□ G4C 연결용 IP

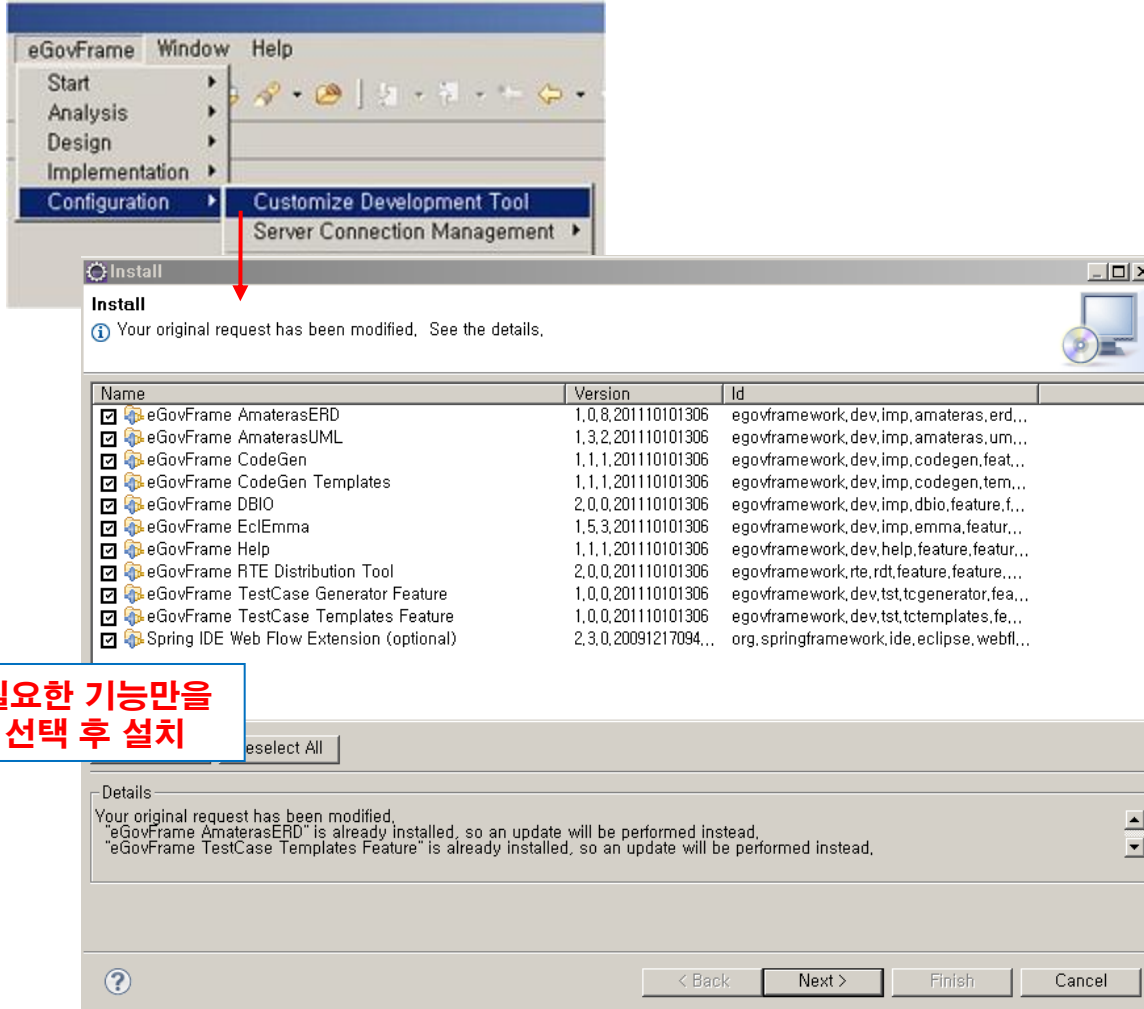
- G4C 인증용 IP를 보여준다.

□ Select DB

- Data Source Explorer에 사용자가 등록한 DB 중 공통 컴포넌트 설치 시 선택한 DB 정보가 선택된다.

※ 설정 변경 시 변경된 내용은 **globals.properties** 에서 확인이 가능하다.

개발환경 구성 시 필요 기능만을 선택하여 개발환경을 선택적으로 구성하는 기능을 제공한다.



□ 제공 기능

- eGovFrame AmaterasERD
- eGovFrame AmaterasUML
- eGovFrame CodeGen
- eGovFrame CodeGen Templates
- eGovFrame EclEmma
- eGovFrame Help
- eGovFrame RTE Distribution Tool
- eGovFrame TestCase Generator Feature
- eGovFrame TestCase Templates Feature
- Spring IDE Web Flow Extension(optional)

오픈소스 중 선정된 UML 작성도구는 AmaterasUML로써 UseCase Diagram, Class Diagram 작성 기능을 제공

□ UseCase Diagram 작성

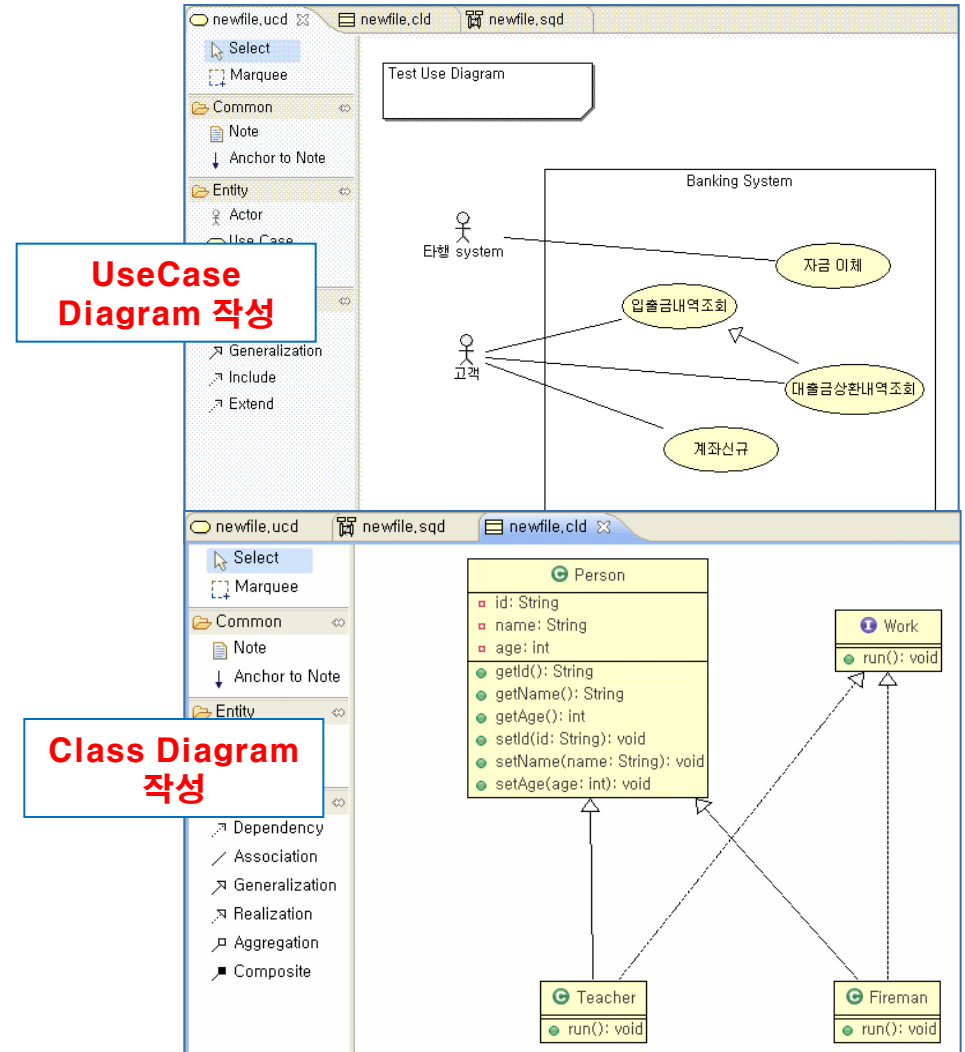
- Use Case Diagram 작성할 수 있는 기능을 제공

□ Class Diagram 작성

- Class Diagram 작성할 수 있는 기능을 제공

□ 제약사항

- UML 1.4 기반으로 설계작업 가능
- UML 2.0 지원하지 않음



오픈소스 중 선정된 ERD 작성도구는 AmaterasERD로서 논리모델과 물리모델 작성을 지원

□ ERD 생성

- Entity 정의와 Entity 간의 관계를 표현할 수 있는 작업창

□ DDL 작성

- 도구를 이용해 작성된 ER Diagram으로 Oracle, PostgreSQL, MySQL, hsqldb 등에 맞게 DDL 스크립트를 생성

□ Reverse Engineering 지원

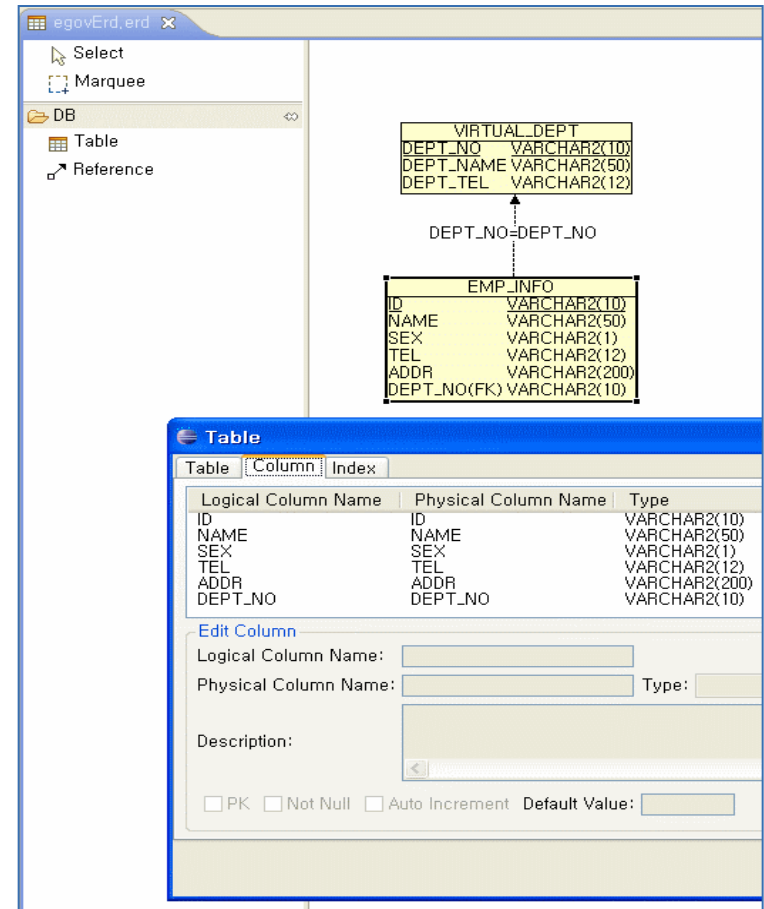
- Oracle, PostgreSQL, MySQL, hsqldb 등의 Database로부터 테이블 Import가 가능

□ 테이블 명세서 Export

- 도구를 이용해 작성된 Table 들의 명세서를 HTML 형식 Export

□ 제약사항

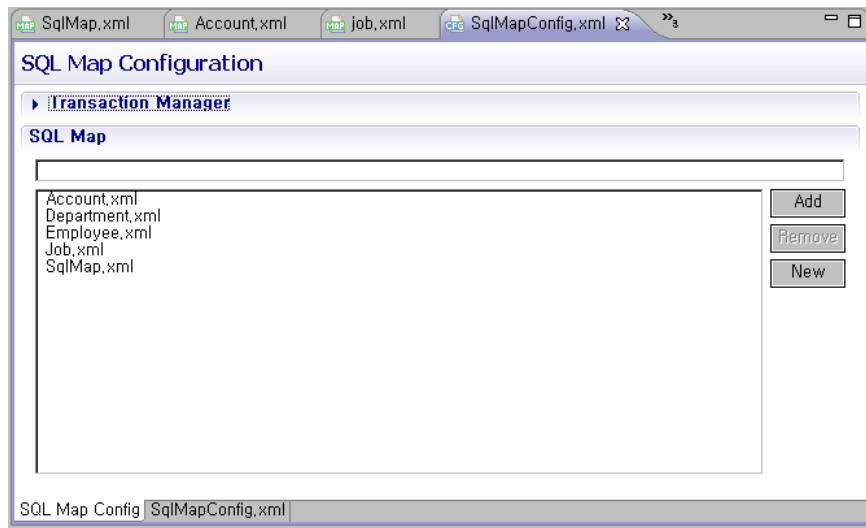
- ERWIN 모델링 도구와 호환 불가



DBIO Editor 는 iBatis SQL Map API 를 통해 DBIO 를 개발하고자 할 때 이를 쉽고 편리하게 작성할 수 있도록 다음과 같은 Editor 기능을 제공한다.

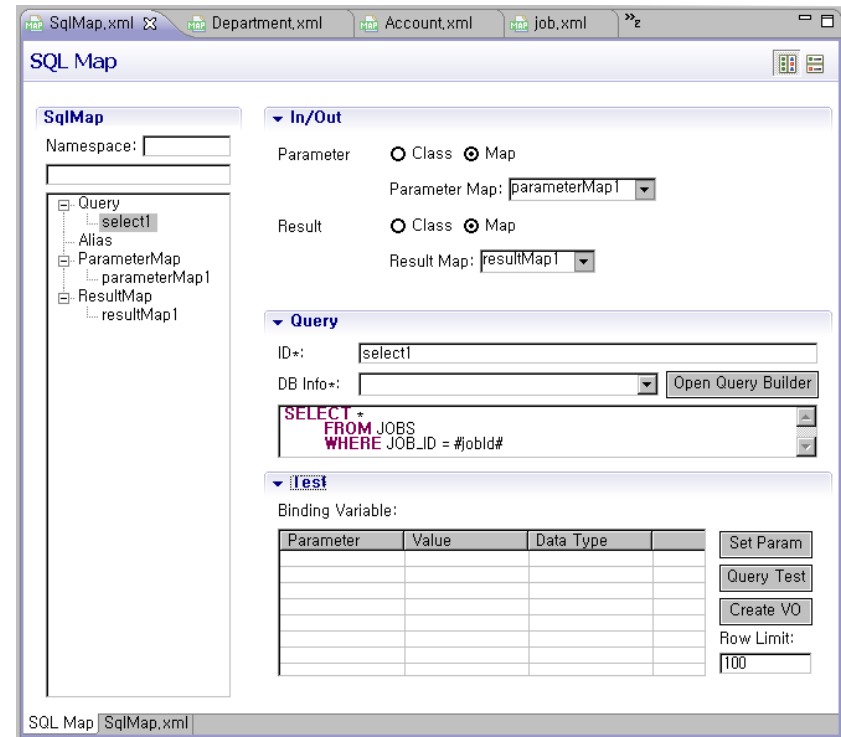
□ SQL Map Config Editor

- SQL Maps 설정 파일을 UI형태로 편집 가능하도록 지원



□ SQL Map Editor

- SQL Maps 파일을 UI형태로 편집 가능하도록 지원



* iBatis SQL Map API 에서는 기본적으로 SQL Map 파일과 SQL Map 설정파일이 요구된다.

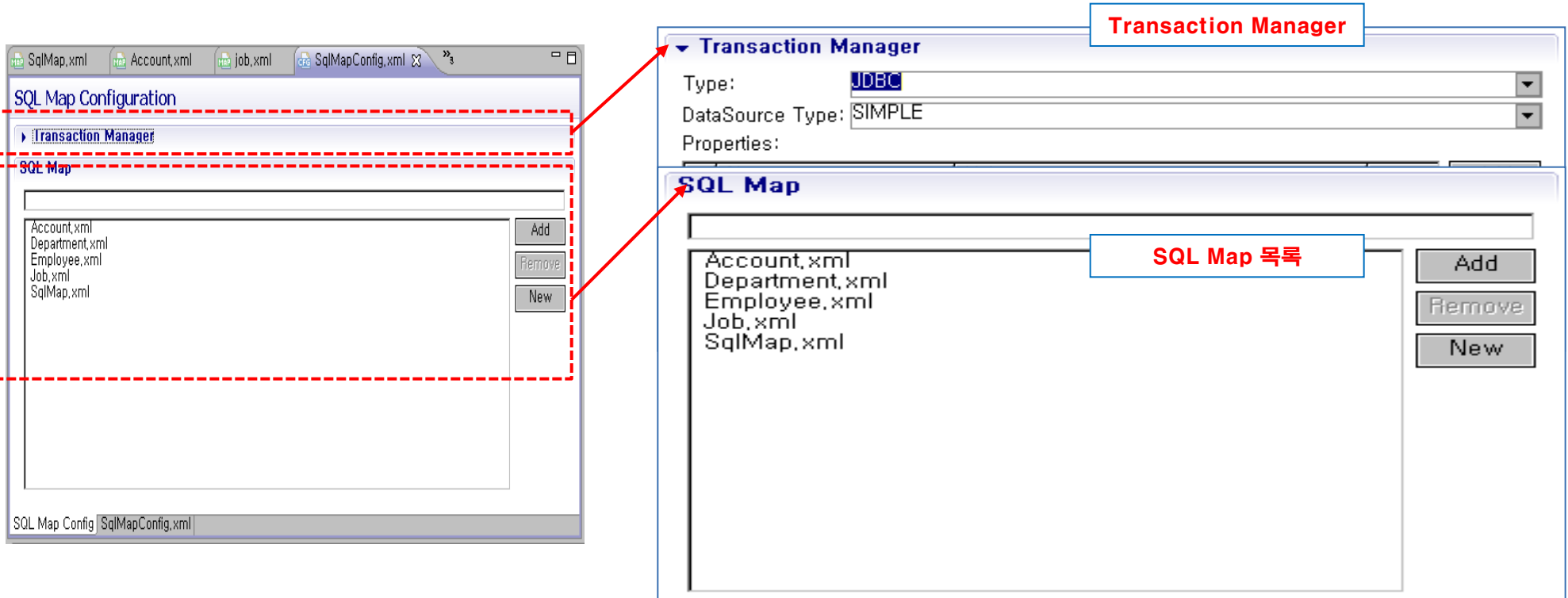
SqlMap Config Editor는 iBATIS SQL Maps 설정 파일을 UI 형태로 편집 가능하도록 지원하는 도구

□ Transaction Manager

- 주요 트랜잭션 Type과 DataSource Type 에 필요한 Properties 항목을 정의한다.

□ SqlMap 목록

- SqlMap 항목을 추가, 삭제, 신규 생성하고 SQLMapConfig 파일 내 등록을 지원한다.



SqlMap Editor는 iBatis SQL Maps 파일을 UI 형태로 편집 가능하도록 지원하는 도구임

□ 그룹별 구성요소 Tree

- 구성요소 생성, 삭제, 검색, 바로가기 기능

□ 사용자 Layout 조정

- 수평, 수직 레이아웃 선택 기능

□ In/out Section

- Parameter, Result 설정(class, map 선택)

□ Query Section

- SQL 쿼리 작성시 Query Builder 기능 제공

□ Test Section

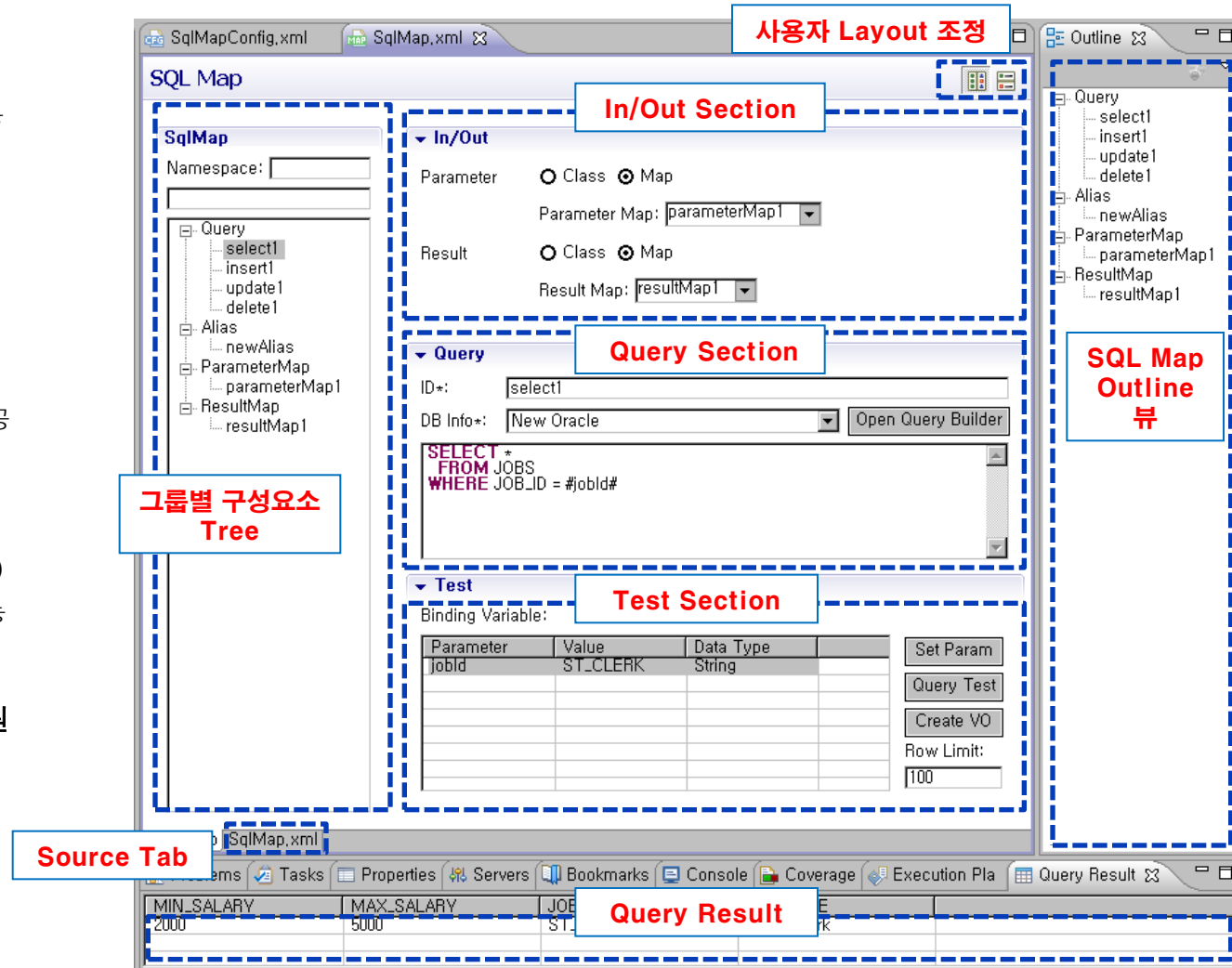
- 쿼리에 따른 Parameter Name 자동 설정
- Row Limit (쿼리 테스트 결과행 길이제한)
- 쿼리 Test 결과에 따른 VO 자동 생성 기능

□ Query Result : 쿼리 수행 결과 확인

□ Source Tab : SqlMap XML 편집 기능 지원

□ SqlMap Outline View

- 구성요소 바로가기 기능 지원



SqlMap Editor에서는 쿼리 작성을 용이하게 하기 위해 Query Builder 기능을 제공함

The screenshot displays the SQL Query Builder interface. On the left, a 'Query' panel shows the ID 'select1' and the database 'New Oracle'. A red dashed box highlights the 'Open Query Builder' button. The main window shows a visual query builder with two tables: 'JOBS' and 'JOB_HISTORY'. The 'JOBS' table has columns JOB_ID, JOB_TITLE, MIN_SALARY, and MAX_SALARY. The 'JOB_HISTORY' table has columns EMPLOYEE_ID, START_DATE, END_DATE, JOB_ID, and DEPARTMENT_ID. A join line connects the JOB_ID columns of both tables. Below the tables are buttons for 'Add Table...' and 'Create Join...'. At the bottom, there is a table for column selection:

Column	Alias	Output	Sort Type	Sort Order
HR.JOBS.JOB_ID		<input checked="" type="checkbox"/>		
HR.JOBS.JOB_TITLE		<input checked="" type="checkbox"/>		
HR.JOB_HISTORY.JOB_ID		<input checked="" type="checkbox"/>		

SqlMap Editor는 Test Section 과 Result View를 통해 쿼리 테스트 기능을 지원함

① DB Info 를 선택

② Query 작성

③ Parameter 값 입력

④ Query Result 확인

MIN_SALARY	MAX_SALARY	JOB_ID	JOB_TITLE
2500	5500	PU_CLERK	Purchasing Clerk
8000	15000	PU_MAN	Purchasing Manager

□ 쿼리 Test 절차

- ① Query Section 에서 DB Info 를 선택한다.
- ② Query Section 에서 Query 를 작성한다.
- ③ Test Section 에서 Parameter 값을 입력한다.
- ④ 결과 값을 확인한다

□ Test Section 버튼 설명

- Set Param : Parameter Name 을 자동 세팅한다.
- Query Test : 쿼리 테스트를 수행한다.
- Create VO : 쿼리 결과에 따른 VO 를 생성한다.
- Row Limit : 쿼리 결과 행 길이 제한 (기본값=100)

SqlMap Editor는 개발자 편의성을 위해 VO(Value Object)를 자동 생성하는 기능을 제공함

test

Binding Variable:

Parameter	Value	Data Type
SearchStr	PU	String

① “Create VO ” 클릭

Set Param
Query Test
Create VO
Row Limit: 100

Query Result

MIN_SALARY	MAX_SALARY	JOB_ID	JOB_TITLE
2500	5500	PU_CLERK	Purchasing Clerk
8000	15000		

③ 쿼리 수행 결과에 따른 VO Class가 생성됨

New Java Class

Java Class

Type name is discouraged. By convention, Java type names usually start with an uppercase letter

Source folder: eGov/src/main/resources

Package: com

Enclosing type:

Name: sampleVo

② VO Class명 입력

Finish

```

package com;

public class sampleVo {

    private String minSalary;
    private String maxSalary;
    private String jobId;
    private String jobTitle;

    public void setMinSalary(String minSalary) {
        this.minSalary = minSalary;
    }

    public String getMinSalary() {
        return minSalary;
    }

    public void setMaxSalary(String maxSalary) {
        this.maxSalary = maxSalary;
    }

    public String getMaxSalary() {
        return maxSalary;
    }

    public void setJobId(String jobId) {
        this.jobId = jobId;
    }
}
    
```

설계 모델을 기반으로 **Skeleton** 소스코드를 생성하는 기능을 제공

□ 절차

1. 구현도구에서 클래스다이어그램(*.cld)를 더블클릭한다.
2. 클래스다이어그램 에디터에서 오른쪽 버튼 클릭하여 메뉴에서 “eGovFrame > Model Based CodeGen” 메뉴를 클릭한다.
3. Output Folder 입력 항목에 소스 디렉토리를 입력하고 “Finish” 버튼을 클릭한다.

The image illustrates the steps to generate code from a model. On the left, a project tree shows the 'model' folder containing 'templatedgm.cld'. A red arrow points from this file to a context menu. The menu path is 'eGovFrame > Model Based CodeGen'. Another red arrow points from this menu item to the 'Model Based Code Gen' dialog box on the right.

The 'Model Based Code Gen' dialog box shows the following configuration:

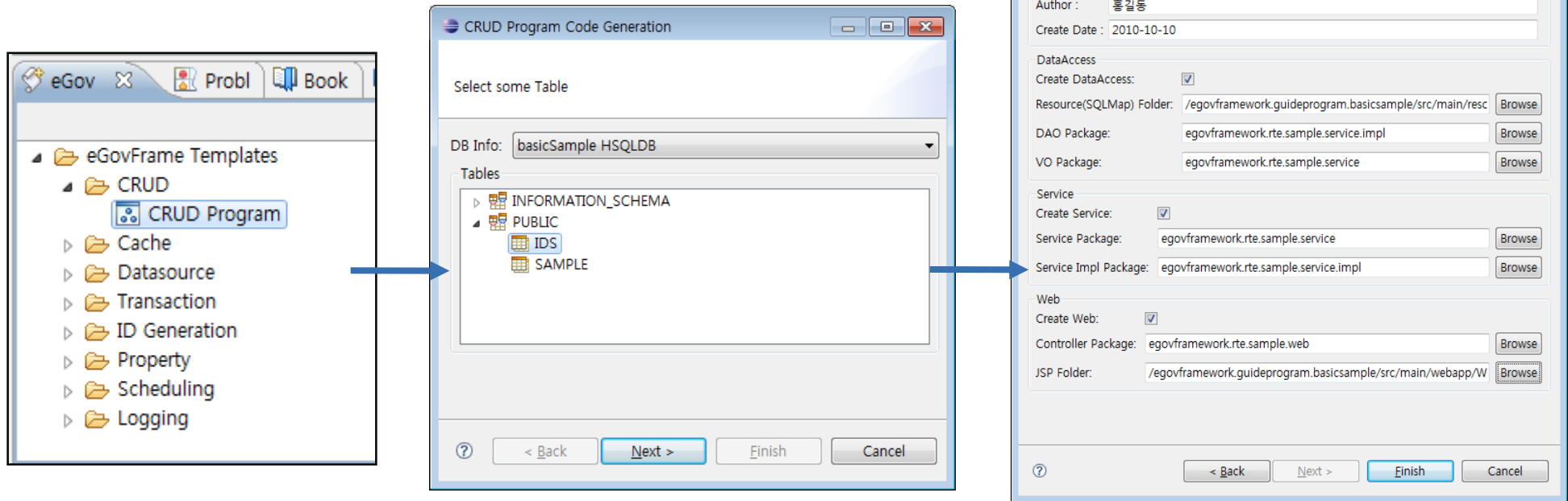
- Output Folder: /egovframework.guideprogram.basic/sample/src/main/java
- Choose Types:

Class	Package	Stereotype	Stereotype Support
<input checked="" type="checkbox"/> Category	codegen	model	not supported
<input checked="" type="checkbox"/> CategoryDAO	codegen	Dao	supported
<input checked="" type="checkbox"/> EgovCategoryController	codegen	Controller	supported
<input checked="" type="checkbox"/> EgovCategoryService	codegen	Service	supported
<input checked="" type="checkbox"/> EgovCategoryServiceImpl	codegen	ServiceImpl	supported
<input checked="" type="checkbox"/> Product	codegen	model	not supported
<input checked="" type="checkbox"/> SearchVO	codegen	Vo	supported
- Buttons: Select All, Deselect All, Finish, Cancel

템플릿 및 Database Table을 기반으로 실행가능한 소스코드를 생성하는 기능을 제공

□ 절차

1. eclipse 개발환경에서 메뉴 > Window > Show View > Other... > eGovFrame Templates 를 선택하고 OK 버튼을 클릭한다.
2. “eGovFrame Templates > CRUD > CRUD Program” 을 더블 클릭한다.
3. CRUD Program 생성대상 테이블을 선택하고 “Finish” 버튼을 클릭한다.
4. 소스 작성자, 생성일자 및 생성될 파일의 디렉토리를 선택하고 Finish 버튼을 클릭하여 생성을 완료한다.



정의된 규칙을 기반으로 소스코드를 검사하여, 오류 및 위험 요인을 식별하여 알려주는 기능 제공

□ 절차

1. eclipse 개발환경의 Package Explorer에서 프로젝트의 컨텍스트 메뉴에서 PMD>Check Code With PMD를 클릭한다.
2. 소스코드 검사가 실행되며 실행 후 PMD Perspective에서 검사 결과를 확인한다.

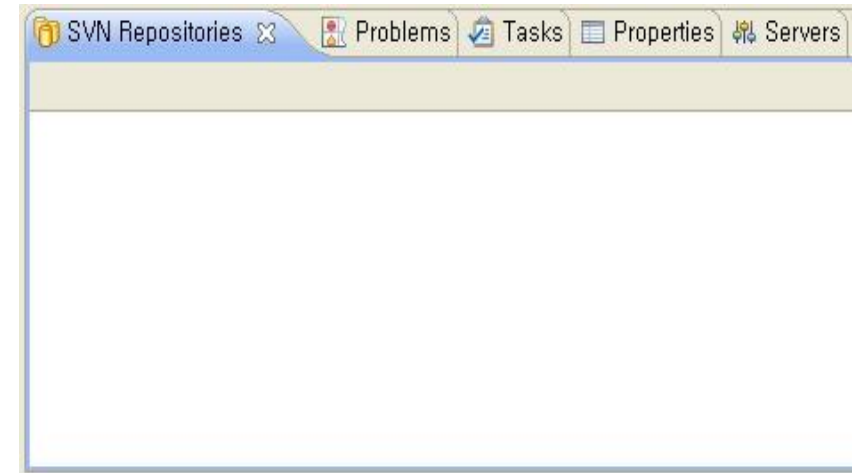
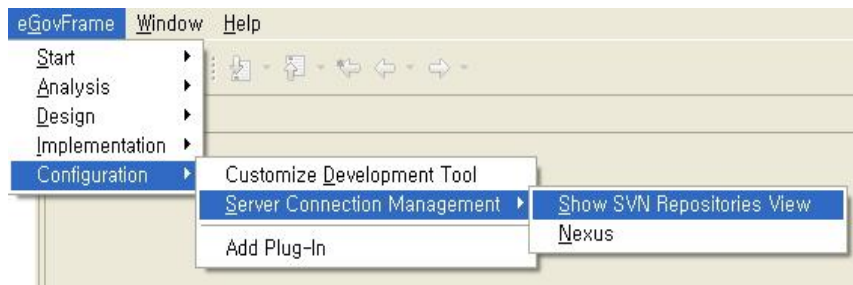
The screenshot shows the Eclipse IDE in the PMD perspective. The Package Explorer on the left shows the project structure with 'SampleDAO.java' selected. The main editor displays the code for 'SampleDAO.java'. The 'Violations Overview' table at the bottom right shows the results of the PMD check.

Element	# Violations	# Violations/L...	# Violati...
egovframework.rte.sample.service.impl	63	863.0 / 1000	3.50
SampleDAO.java	19	1000.0 / 1000	3.17
MethodArgumentCouldBeFinal	(max) 5	263.2 / 1000	0.83
SignatureDeclareThrowsException	10	526.3 / 1000	1.67
ShortVariable	4	210.5 / 1000	0.67
EgovSampleServiceImpl.java	25	714.3 / 1000	4.17

eGovFrame 통합메뉴를 통해 SVN Repositories View를 연동하는 기능을 제공한다.

□ SVN Repositories View

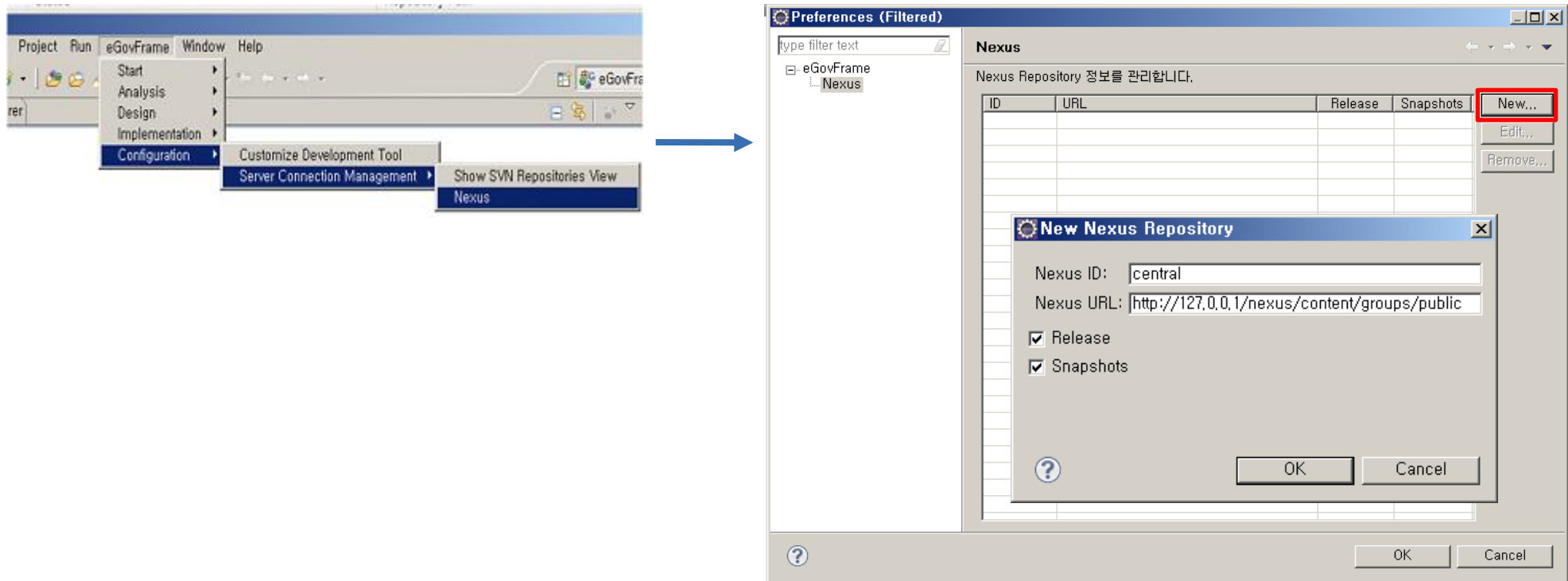
1. eGovFrame 통합메뉴 > Configuration > ServerConnection Management > Show SVN Repositories View 를 클릭한다.
2. 화면 하단에 SVN Repositories라는 View가 열리는 것을 확인한다.



Nexus Repository를 등록, 수정 및 삭제하는 기능을 제공한다.

❑ Nexus(Preferences)

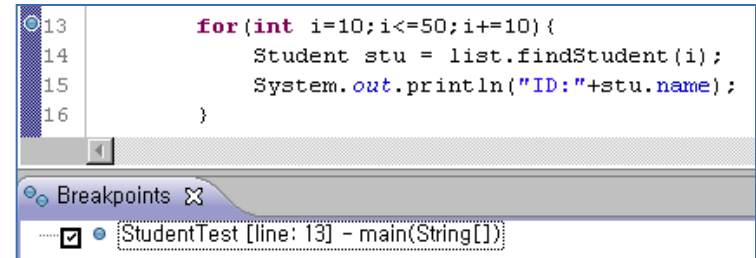
1. eGovFrame 통합메뉴에서 eGovFrame > Configuration > Server Connection Management > Nexus를 선택한다.
2. Preferences 페이지에서 New 버튼을 클릭하여 새로운 Nexus repository를 추가한다.



구현도구에는 로컬 또는 원격으로 실행 중인 프로그램에서 오류를 발견하고 진단할 수 있는 디버거를 제공함

□ 중단점 설정

- 디버깅 중 프로그램의 의심되는 부분을 집중적으로 분석하기 위해 Break point를 설정해 디버깅 포인트를 지정한 부분을 하이라이트



□ 스텝단위 진행

- 프로그램을 한 스텝씩 진행하면서 프로그램의 실행흐름과 내부 상태를 확인

□ 스텝 필터링

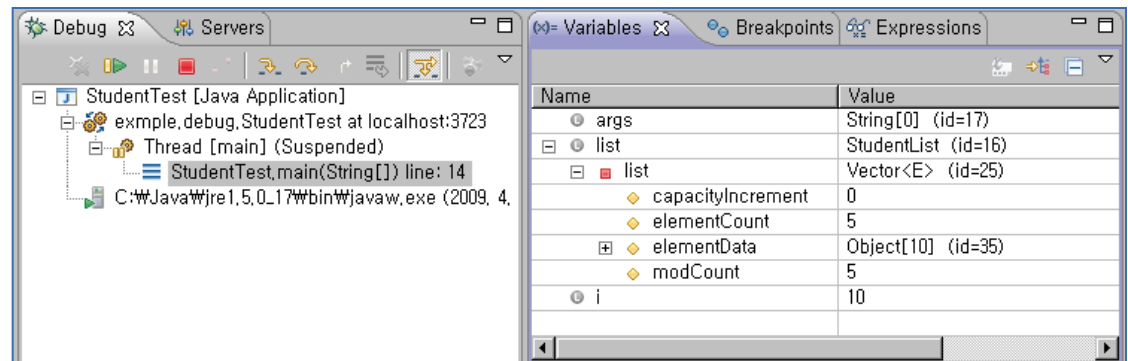
- 필터링 대상 프로그램의 경우 Step Over와 같이 동작하고, 필터링 대상이 아니면 Step Into와 같이 동작한다. Debug View의 Use Step Filters 버튼을 눌러 활성화

□ Evaluating Expressions

- 수행 중인 프로그램을 중지시키지 않고 상태를 확인할 수 있는 기능을 제공

□ Variables View

- Variables View를 통해 메소드에서 사용 중인 변수 값의 상태를 확인할 수 있는 기능 지원



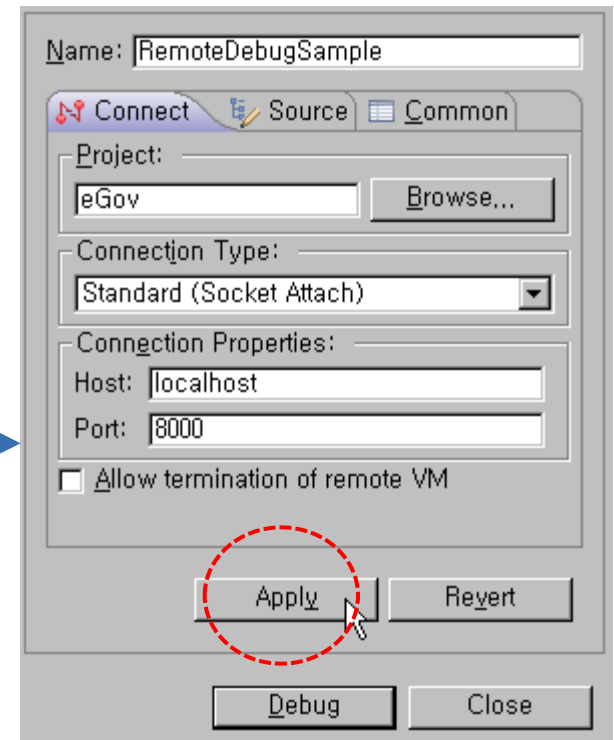
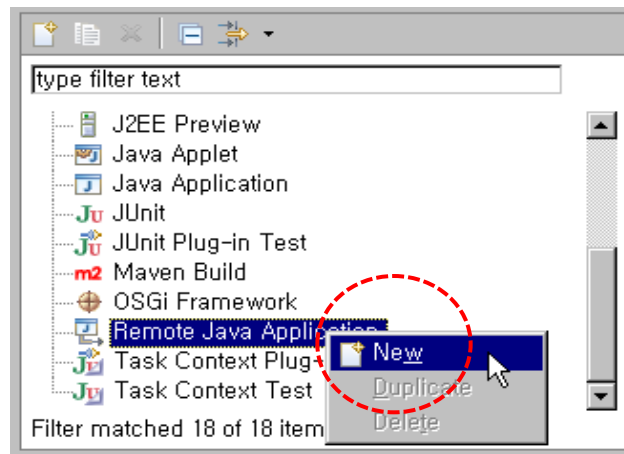
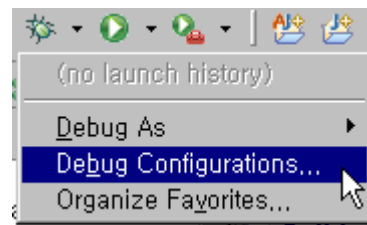
원격 컴퓨터의 WAS(Jboss, JEUS, WebLogic)에서 실행되는 어플리케이션을 디버깅

□ 디버깅 환경 설정

- WAS 환경설정 파일에 “-Xdebug -Xnoagent -Xrunjdw:transport=dt_socket,address=8000,server=y,suspend=n -server” 를 정의 (Jeus 서버 기준)

□ 절차

1. 먼저 위의 환경설정으로 원격지 서버를 가동하고 해당 Java 프로그램을 원격 서버에 Deploy한 것을 전제로 한다.
2. 메뉴>Run > Debug Configurations... 클릭.
3. Remote Java Application 에서 마우스 오른쪽 버튼을 클릭하여 “NEW” 를 실행
4. Connection Properties 를 입력하고 적용(“Apply”)한 뒤 “Debug” 버튼 클릭
5. 웹브라우저에서 해당 프로그램을 실행하면 디버깅이 진행된다.



* 디버그 방법(BreakPoint 등)은 로컬 디버깅과 동일하다.

Eclipse Home

- <http://www.eclipse.org/>

Amateras UML

- http://amateras.sourceforge.jp/cgi-bin/fswiki_en/wiki.cgi?page=AmaterasUML

Amateras ERD

- http://amateras.sourceforge.jp/cgi-bin/fswiki_en/wiki.cgi?page=AmaterasERD

1. **Maven 개요**
2. **Maven 디렉터리 구조**
3. **빌드 Lifecycle**
4. **Maven 플러그인**
5. **의존성 관리 메커니즘**
6. **Maven 저장소**
7. **프로젝트 객체 모델(Project Object Model)**
8. **Maven 이클립스 통합: m2eclipse**
9. **참고자료**

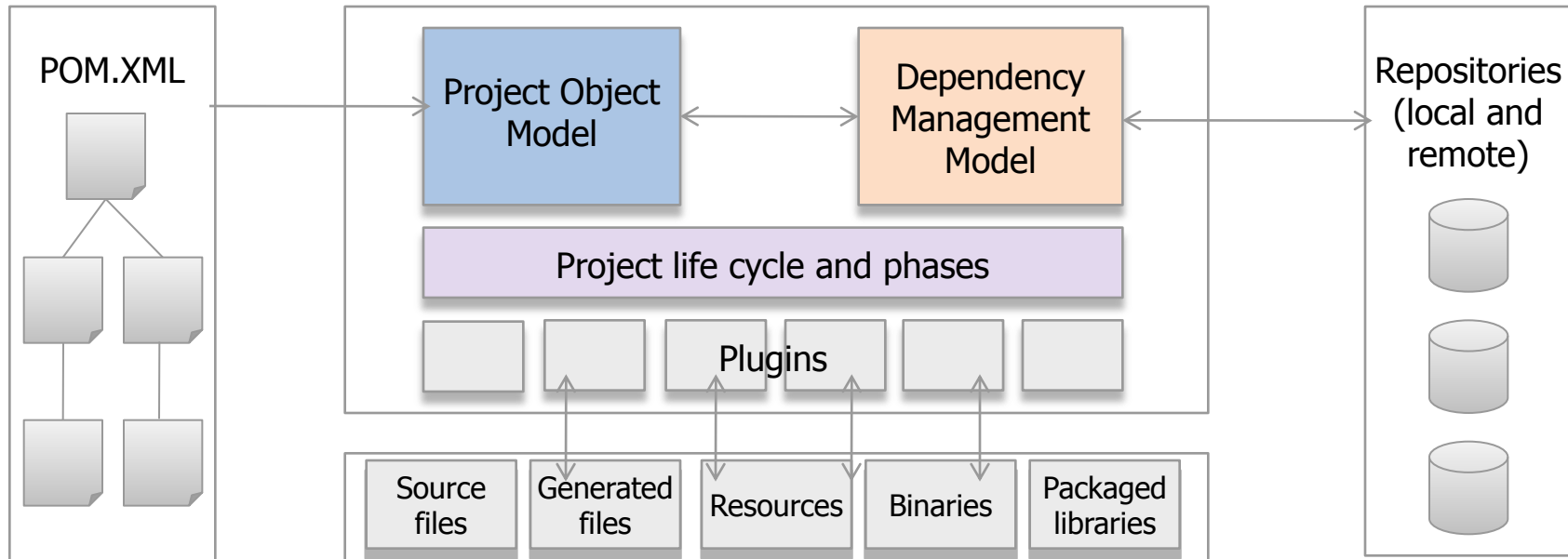
불필요한 설정을 최소화 한다는 개념 아래 Ant와 같은 빌드 기능을 제공할 뿐 아니라 구조화 된 빌드 기능을 통해 **learning curve** 및 재사용 성을 향상 시킴

□Maven 특징

Maven 장점	Maven 단점
<ul style="list-style-type: none">• 뛰어난 의존성 관리 의존성 자동 업데이트 저장소를 통한 라이브러리 일괄 관리•모든 프로젝트에 걸쳐 쉽게 적용 가능한 일관적인 사용법•라이브러리 및 메타 데이터 저장을 위한 지속적으로 확장되고 있는 저장소•쉽게 작성 가능한 플러그 인을 통한 확장성•동시에 다수의 프로젝트 핸들링 할 수 있는 쉬운 설정 기반의 메커니즘•간단한 설정을 통한 배포 관리•Java, C++ 등 다수의 프로그래밍 언어 지원	<ul style="list-style-type: none">• repository 관리의 불편함<ul style="list-style-type: none">- Maven 프로젝트의 급속한 발전으로 central repository가 제공하는 라이브러리들이 급속히 증가하고 있으나 아직 3rd 파티 라이브러리 등 미제공 라이브러리들이 있음• pom.xml 파일 관리<ul style="list-style-type: none">-메이븐 프로젝트 관리에 대한 모든 내용이 pom.xml 파일에 담기게 되므로 길고 장황하게 될 수 있음• 프로젝트에 특화된 복잡한 빌드 기능 제약<ul style="list-style-type: none">- 메이븐 프로젝트 특성상 소프트웨어 빌드에 통용되는 라이프 사이클을 제공하고 있어 세부 항목 또는 특화된 빌드 환경에 대한 지원이 미약함

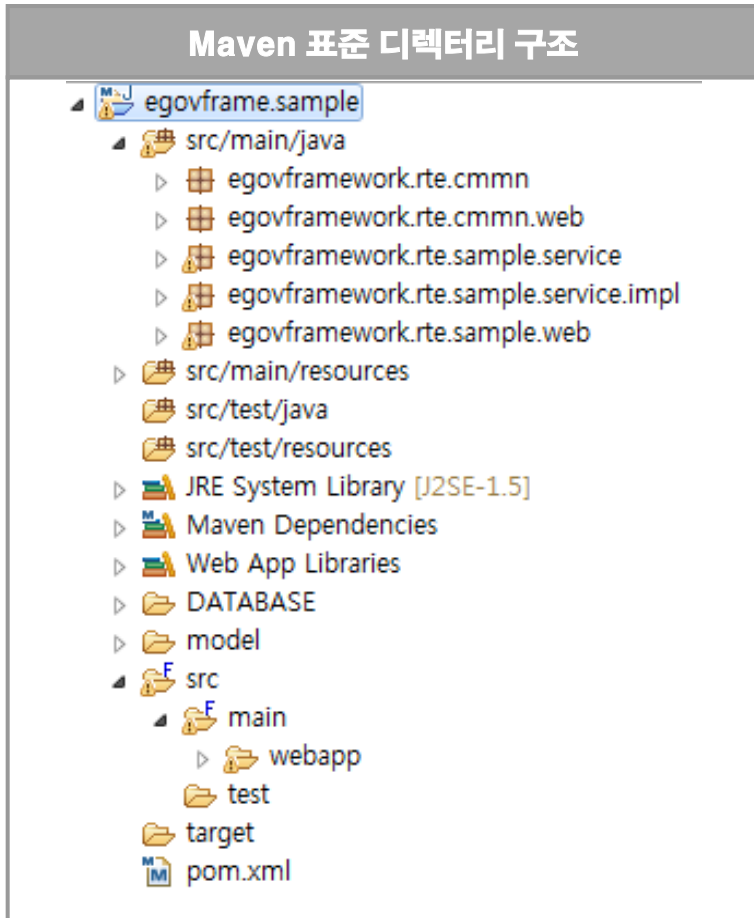
의존성 관리, 라이브러리 관리, 빌드 생명 주기 및 단계 객체 모델을 갖는 프로젝트 관리 도구로 표준화된 빌드 기능 뿐만 아니라 리포팅 및 **documentation** 생성 기능 등을 제공

□Maven 아키텍처



- **프로젝트 객체 모델(POM)**: 메이븐 엔진 내장 + POM.XML 파일에서 선언적으로 제공
- 의존성 관리 모델: 로컬 및 리모트 저장소를 이용하여 관리
- 빌드 생명주기와 각 단계: 잘 정의된 단계들과 빌드 사이클에 따라 플러그인들을 조율

Best practices을 기반으로 정규화된 디렉터리 구조를 제공하고 있으며 모든 소스 파일들은 /src 디렉터리 밑에 빌드 된 output은 /target 디렉터리 밑에 위치함

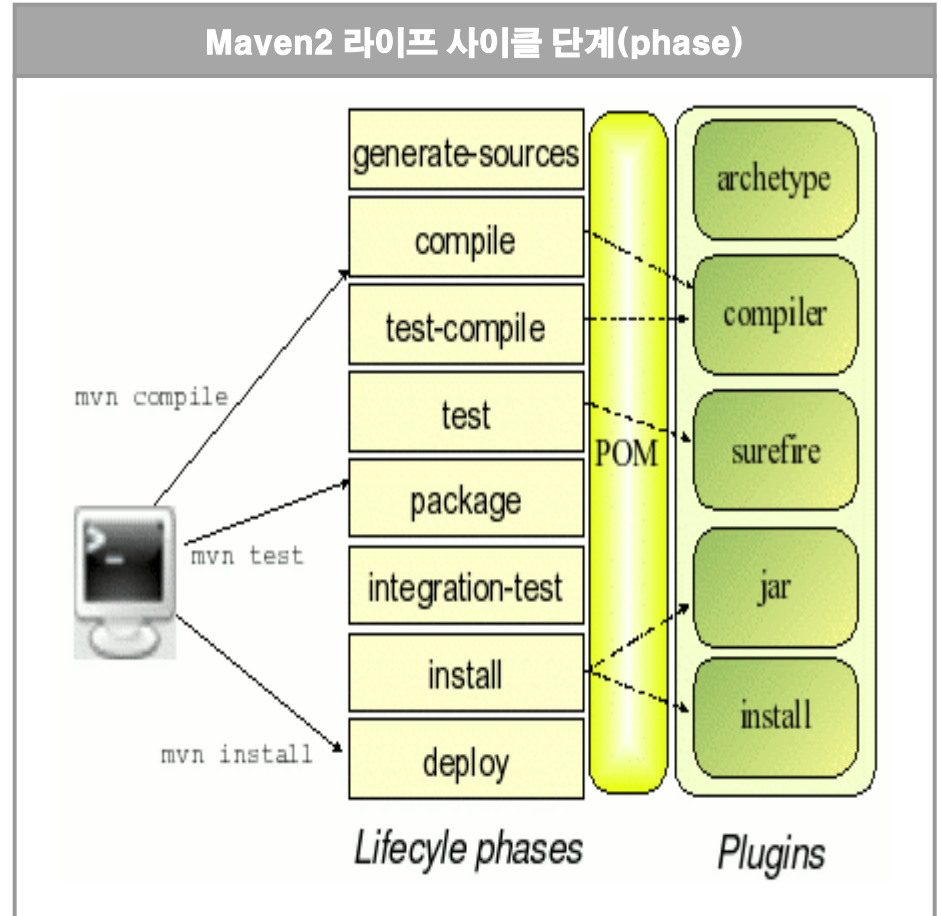


디렉터리/파일	설명
/pom.xml	프로젝트 객체 모델. 해당 프로젝트에 대한 전반적인 정보를 갖는다.
/src/main/java	Java 소스 파일 위치
/src/main/resources	배포할 리소스, XML, properties, ...
/src/main/webapp	웹 어플리케이션 관련 파일 위치(WEB-INF, css 등)
/src/test/java	테스트 케이스 java 소스
/src/test/resources	테스트 케이스 리소스
/target	빌드 된 output이 위치하는 디렉터리

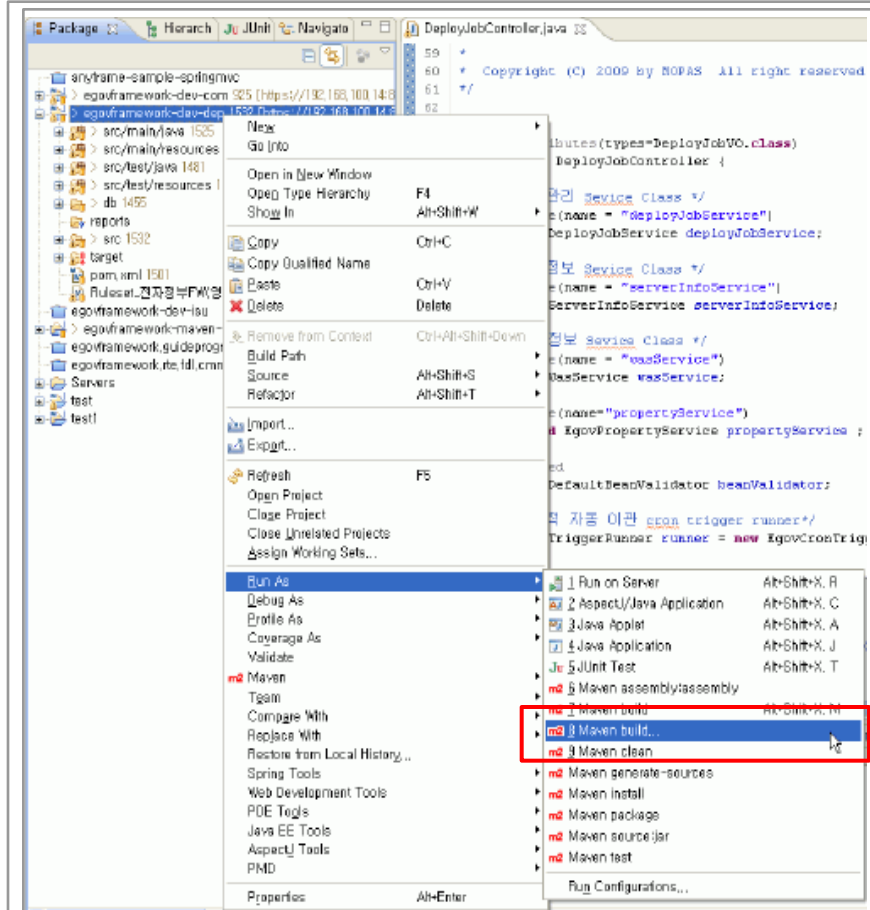
메이븐 빌드는 소프트웨어 프로젝트의 핵심적인 빌드 라이프 사이클 개념을 따르고 있으며 빌드 부터 artifact의 배포까지의 라이프 사이클을 정의하고 있음

빌드 생명 주기 설명	
생명주기 단계	설명
validate	현재 설정과 POM의 내용이 유효한지 확인
generate-sources	코드 생성기가 이 다음의 단계들에서 컴파일 되고 처리할 소스 코드를 생성하기 시작하는 순간
compile	소스 코드를 컴파일 한다. 컴파일 된 클래스들은 타겟 디렉터리 트리 구조에 저장된다.
test	컴파일 된 단위 테스트를 실행하고 그 결과를 표시한다.
package	실행 가능한 바이너리 파일들을 WAR나 JAR같은 배포용 압축 파일로 묶는다.
install	압축 파일을 로컬 메이븐 저장소에 추가한다.
deploy	압축 파일을 원격 메이븐 저장소에 추가한다.

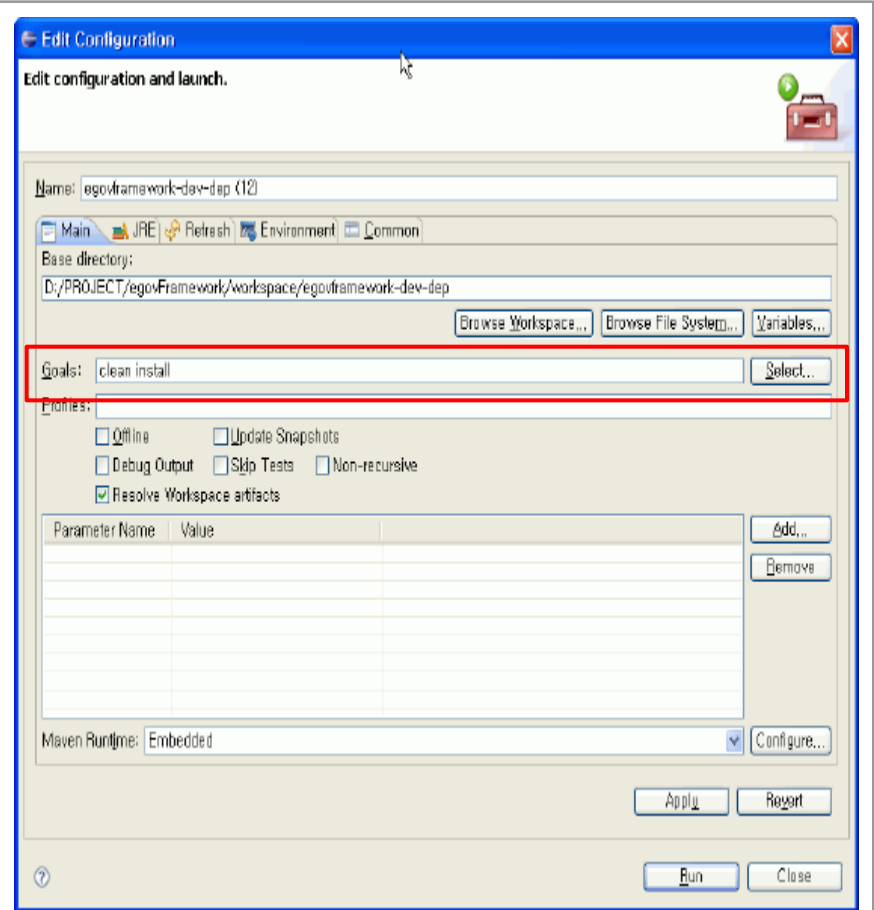
\$mvn compile



이클립스에서 빌드 라이프 사이클에서 정의한 각 단계들 실행하기



메이븐 프로젝트 -> Run As -> Maven build..



Goals: clean install

메이븐은 대부분의 빌드 작업 시에 플러그인을 활용하며 메이븐 엔진이 생명주기 단계들과 플러그인을 바인딩 시켜 빌드 작업을 컨트롤 함

Rating	Plugin	M1	M2	FC	Doc	Rel	Author for m2 version	Comments	Open Issues
☆☆	Maven JPOX Plugin	✓	✓	✗	✗	Alpha			4 issues
☆☆	Maven Clirr Plugin	✓	✗	✗	✗			http://clirr.sourceforge.net/clirr-maven/index.html	
☆☆	Maven JDBC Plug-in	✗	✓	✗	✗		Jesse McConnell	in mojo sandbox pending more features	4 issues
☆☆	Maven Execute Plug-in	✗	✓	✗	✗		Jesse McConnell	replaced by the exec-maven-plugin in mojo	
☆☆	Maven SableCC Plug-in	✗	✓	✗	✗		Jesse McConnell		4 issues
☆☆	Maven AxiomTools Plugin	✓	✓	✗	✗		Jesse McConnell	contains wsdl2java and java2wsdl goals	4 issues
☆☆	Tomcat Plugin	✓	✓	✗	✗		Mark Hobson		
☆☆	XDoclet Plugin	✓	✓	✗	✗		Kenney Westerhof		
☆☆	Maven XML Beans Plugin	✓	✓	✓	✓		David Jencks, Brett Porter and Kris Bravo		4 issues
☆☆	Maven OSGi Plugin	✓	✓	✗	✗		Timothy Bennett (Apache Felix)	M1: http://mavenosgiplugin.berlios.de	
☆☆	Maven Cactus Plugin	✓	✗	✗	✗		Kenney Westerhof		
☆☆	Maven Cargo Plugin	✓	✓	✓	✓		Vincent Massol & Scott Ryan		24 issues
☆☆	Maven Emma Plugin	✓	✗	✗	✗				
☆☆	Maven Commons Attributes	✓	✓	✗	✗				

Maven Plugins

- 플러그인 플랫폼으로써의 메이븐 - 확장 용이
- 자주 사용하는 플러그인들을 모아 패키징 한 상태로 배포
- 대부분의 개발 작업 시에 추가 플러그인 불필요

단계	Plugin	설명
compile	maven-compiler-plugin	소스 코드 컴파일
test	maven-surefire-plugin	단위 테스트 실행
package	maven-jar-plugin	컴파일 된 바이너리 파일들로부터 jar 패키지 생성
emma	maven-emma-plugin	Code coverage report 생성

가용한 Maven 플러그인 매트릭스

□ pom.xml에서의 메이븐 플러그인 선언 예제

```
<project>
  <modelVersion>4.0.0</modelVersion>
  <groupId>egovframework.dev</groupId>
  <artifactId>my-app</artifactId>
  <version>1.0</version>
  .....
  <pluginManagement>
    <plugins>
      <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-compiler-plugin</artifactId>
        <configuration>
          <source>${compileSource}</source>
          <target>${compileSource}</target>
          <encoding>${encoding}</encoding>
          <testFailureIgnore>>true</testFailureIgnore>
        </configuration>
      </plugin>
      <!-- Egovframework JUnit Excel Reporting -->
      <plugin>
        <groupId>egovframework.dev</groupId>
        <artifactId>egovtest-maven-plugin</artifactId>
        <version>1.0.0-SNAPSHOT</version>
      </plugin>
      <!-- EMMA -->
      <plugin>
        <groupId>org.codehaus.mojo</groupId>
        <artifactId>emma-maven-plugin</artifactId>
        <version>1.0-alpha-1</version>
      </plugin>
      <!-- PMD manven plugin -->
      <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-pmd-plugin</artifactId>
        <version>2.4</version>
      </plugin>
    </plugins>
  </pluginManagement>
</project>
```


POM에서 선언적인 **dependency** 설정으로 빌드 및 배포 시에 필요한 라이브러리들을 관리하고 로컬 및 원격 저장소에서 선언된 라이브러리들을 다운로드 받아 사용함

의존성 설정

```
<project>
.....
<dependencies>
  <dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <version>3.8.1</version>
    <scope>test</scope>
  </dependency>
</dependencies>
</project>
```

의존성 scope

- **compile**: 기본값으로 모든 클래스 패스에서 사용 가능
- **provided**: 컴파일과 유사하나 패키지에는 포함되지 않는다. 컨테이너나 JDK에서 제공. 예) Servlet API for web apps
- **runtime**: 컴파일러가 아닌 런타임 시에만 사용됨
예) JDBC drivers
- **test**: 테스트 단계에서만 유용함 예)Junit
- **system**: provided와 유사하나 개발자가 직접 JAR 파일을 제공해야 함.
저장소에서 지정한 dependency를 찾지 않는다.

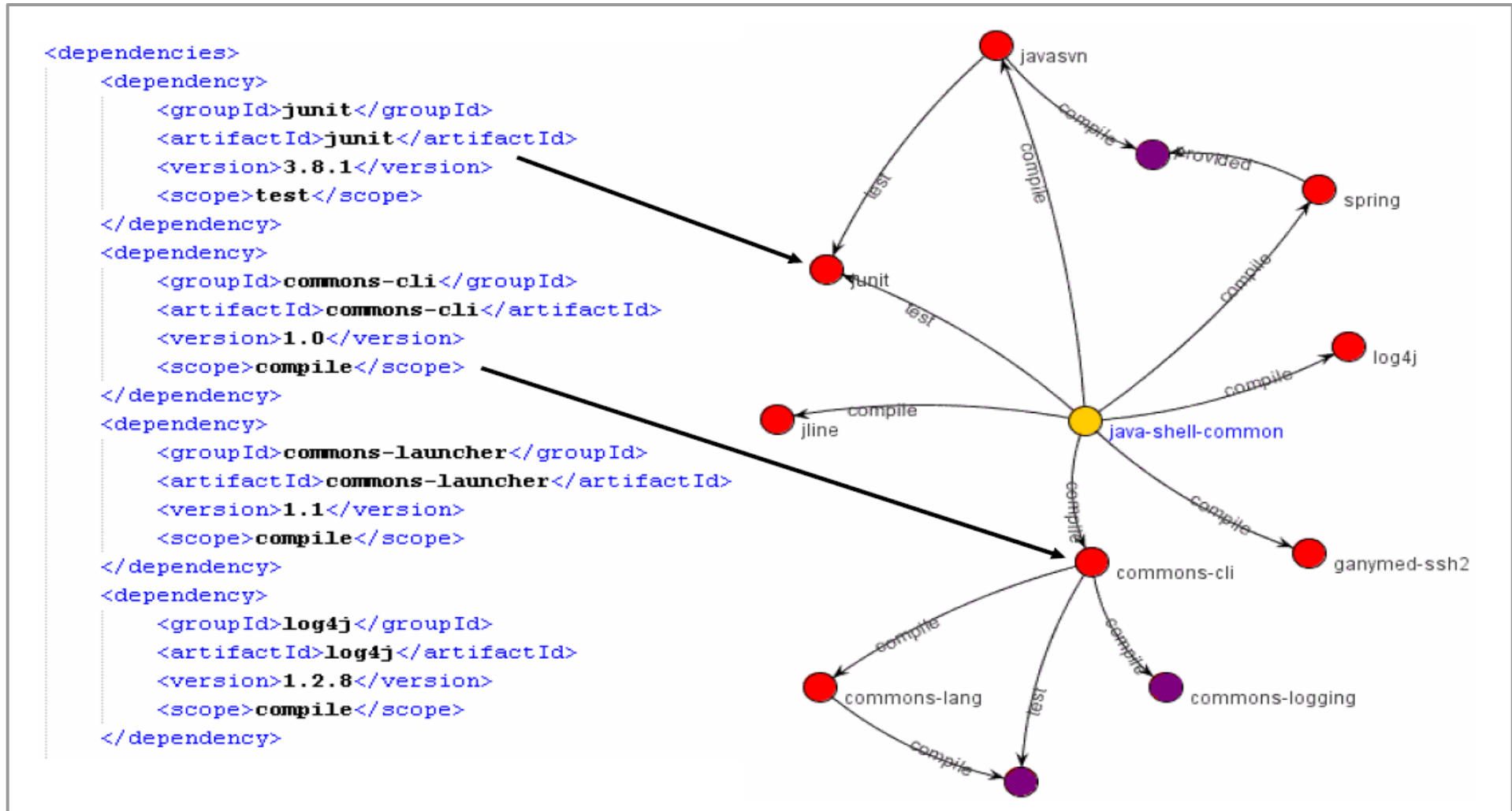
의존성 분석 순서

1. 로컬 저장소에서 의존성 확인
2. 원격 저장소 리스트에서 의존성 확인
3. 1과 2가 실패하면 의존성 에러 보고

의존성 선언 항목

1. **<groupId>**: 부분적인 프로젝트나 조직에서의 라이브러리 집합을 식별하기 위해 제공
2. **<artifactId>**: 프로젝트의 실제 이름으로 groupId와 합쳐져 프로젝트 식별에 쓰임
3. **<version>**: 선언한 의존성 artifact의 버전으로 프로젝트 내에서 사용하는 artifact의 일관성을 추구할 수 있음

□ pom.xml 파일에서의 의존성 설정 및 의존성 분석



artifact들의 저장소로 로컬 및 리모트 repository로 구성되며 프로젝트는 pom.xml에서 선언한 dependency들을 저장소로부터 불러와서 사용함

메이븐 저장소 구조

The diagram illustrates the Maven storage structure. On the left, three developers (Developer 1, 2, and 3) are shown on an Intranet. They are connected to a Public maven repository at ibiblio on the Internet. A file explorer shows the local system structure with folders like .m2, repository, ant, and ant-launcher.

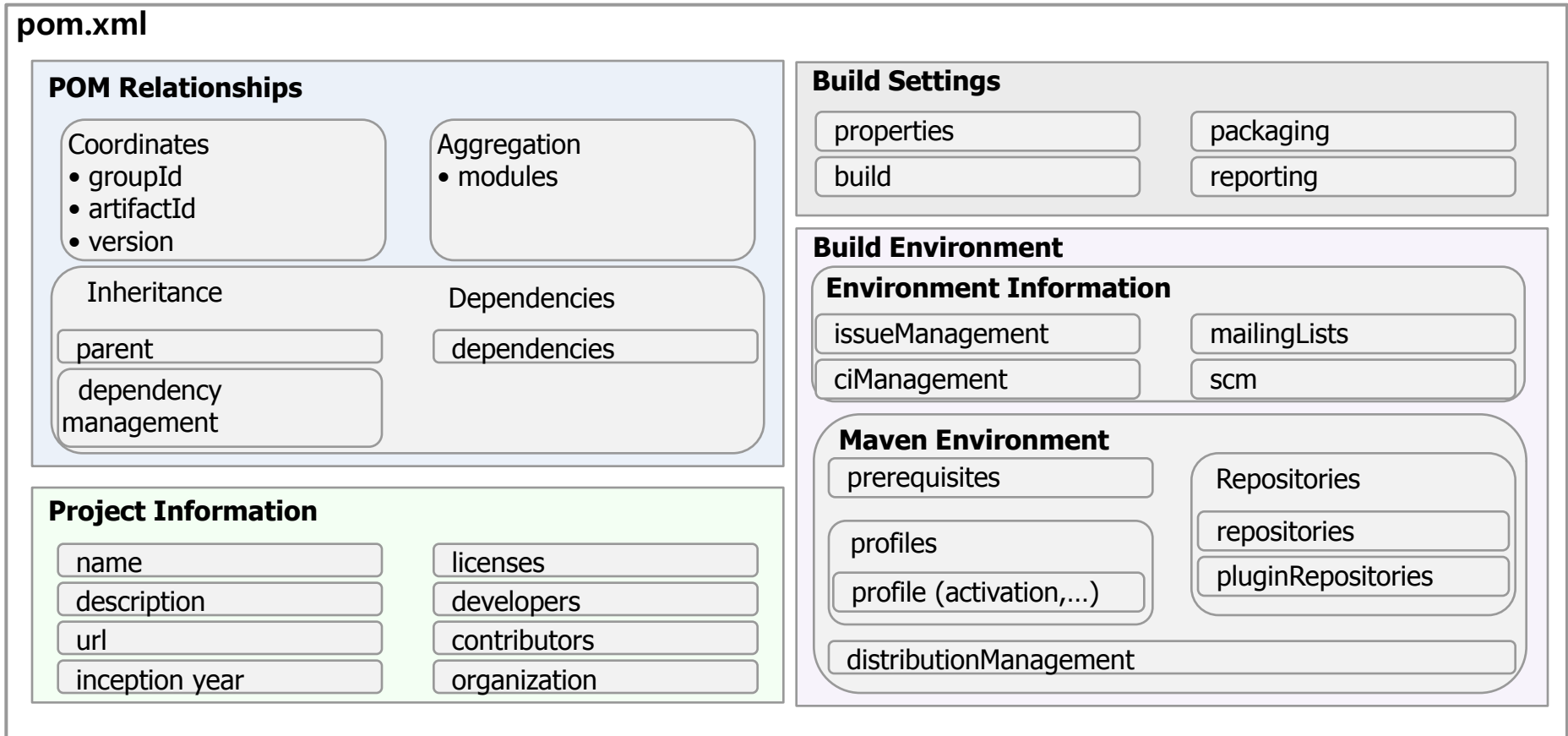
- 로컬 repository는 artifact들을 로컬 파일 시스템에 cache하는 기능으로 리모트 repository에서 다운로드 받아 artifact들을 저장하고 관리한다.
- 리모트 repository는 주 로 HTTP 서버로 3rd 파티에서 제공하는 artifact들을 제외 한 거의 모든 artifact들을 제공한다.

Nexus 연동

The diagram illustrates the Nexus integration. On the left, three developers (Developer 1, 2, and 3) are shown on an Intranet. They are connected to a Maven repository on the Intranet and a Public maven repository at ibiblio on the Internet. A screenshot of the Sonatype Nexus web interface is shown.

- Nexus는 메이븐 repository 관리툴로 리모트 repository의 단점을 보완하고 maven 프로젝트의 사용 편의성을 높이기 위해 사용된다. 각 개발자들은 리모트 repository가 아닌 Nexus에서 dependency를 다운 받아 사용함으로써 프로젝트 내의 artifact 버전 등의 일관성을 유지하고 3rd 파티 artifact 등 라이브리리를 효과적으로 공유할 수 있다.

프로젝트의 구조와 내용을 설명하고 있으며 pom.xml 파일에 프로젝트 관리 및 빌드에 필요한 환경 설정, 의존성 관리 등의 정보들을 기술함



- 프로젝트의 세부 메타 데이터 정보를 포함
 - 버전 및 설정 관리, 빌드 환경, 라이브러리 저장소 및 의존성

프로젝트의 pom.xml 파일에 빌드 정보들을 기술 함

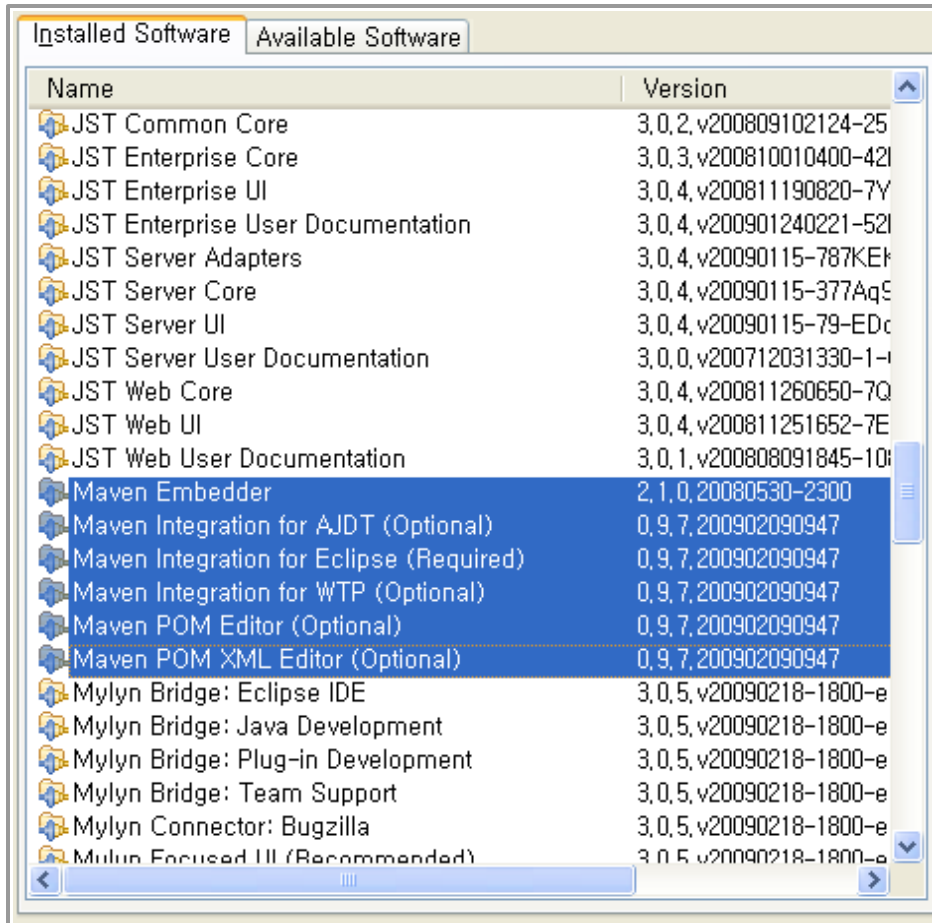
```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0http://maven.apache.org/maven-v4_0_0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>egovframework.dev.com</groupId>
  <artifactId>egovframework-dev-com</artifactId>
  <version>1.0</version>
  <packaging>war</packaging>
  <name>egovframework-dev-com Maven Webapp</name>
  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>4.4</version>
      <scope>test</scope>
    </dependency>
  </dependencies>
  <plugins>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-compiler-plugin</artifactId>
      <configuration>
        <source>1.5</source>
        <target>1.5</target>
      </configuration>
    </plugin>
  </plugins>
</project>
```

Artifact 정보

의존성 관리

플러그인 선언

m2eclipse는 Eclipse IDE에서 Maven을 사용하기 위한 플러그인으로 Maven 프로젝트 생성 뿐 아니라 Maven 빌드와 WTP(Web Tools Project)의 통합 등 다양한 기능을 제공함



m2eclipse 특징

- 메이븐 프로젝트 생성 및 불러오기
- 의존성 관리 및 의존성 통합
- dependency 자동 다운로드 및 업데이트
- 리모트 repository 탐색 기능 제공
- POM 파일 관리 화면 제공 및 dependency list에 대한 자동 업데이트
- 다양한 SCM 저장소로부터 메이븐 프로젝트 check out
- 이클립스에서 메이븐 멀티 모듈 프로젝트 생성 기능 제공
- Web Tools Project (WTP)와의 연동
- aspectj Development Tools (AJDT)과의 연동
- Subversion 플러그인과의 연동

□ 배포된 eGovFramework Archetype을 이용한 메이븐 프로젝트 생성 예

Maven 프로젝트 생성

eGovFramework에서는 메이븐 프로젝트 생성을 위해 두 가지 방식을 제공하고 있음

1. 구현 도구에서 제공하는 **Perspective**를 이용한 Maven 프로젝트 생성
2. Maven **archetype**을 이용한 프로젝트 생성
: 'Archetype is a Maven project templating toolkit.' - 프로젝트에 특화된 pom.xml 및 resource들을 포함한 Maven 프로젝트 생성

Maven Archetype 종류

1. 표준 Archetype
 - maven-archetype-j2ee-simple
 - maven-archetype-quickstart
 - maven-archetype-portlet
2. eGovFramework Archetype
 - egovframework-maven-webapp

* Maven 제공 Archetype 종류:

<http://repo1.maven.org/maven2/org/apache/maven/archetypes/>

Maven 프로젝트의 archetype 선택

New Maven project

Select project name and location

Create a simple project (skip archetype selection)

Use default Workspace location

Location: D:\PROJECT\egovFramework\workspace



New Maven project

Select an Archetype

Catalog: Local C:\java\m2\archetype-catalog.xml

Filter:

Group Id	Artifact Id	Version
egovframework.maven.archetypes	egovframework-maven-webapp	1.0

configure.. 버튼 클릭으로 제공된 archetype을 선택

□ eGovFramework Archetype을 이용한 Maven 프로젝트

The screenshot displays the Eclipse IDE interface. On the left, the Package Explorer shows a project named 'SampleProject' with a red box highlighting its internal structure. The structure includes:

- src/main/java: egovframework, dev, temp, SampleProject
- src/main/resources
- src/test/java
- src/test/resources
- src:
 - main:
 - webapp:
 - META-INF
 - WEB-INF:
 - lib
 - web.xml
 - index.jsp
 - test
 - target
 - pom.xml

On the right, the pom.xml file is open, showing the following dependencies (lines 79-105):

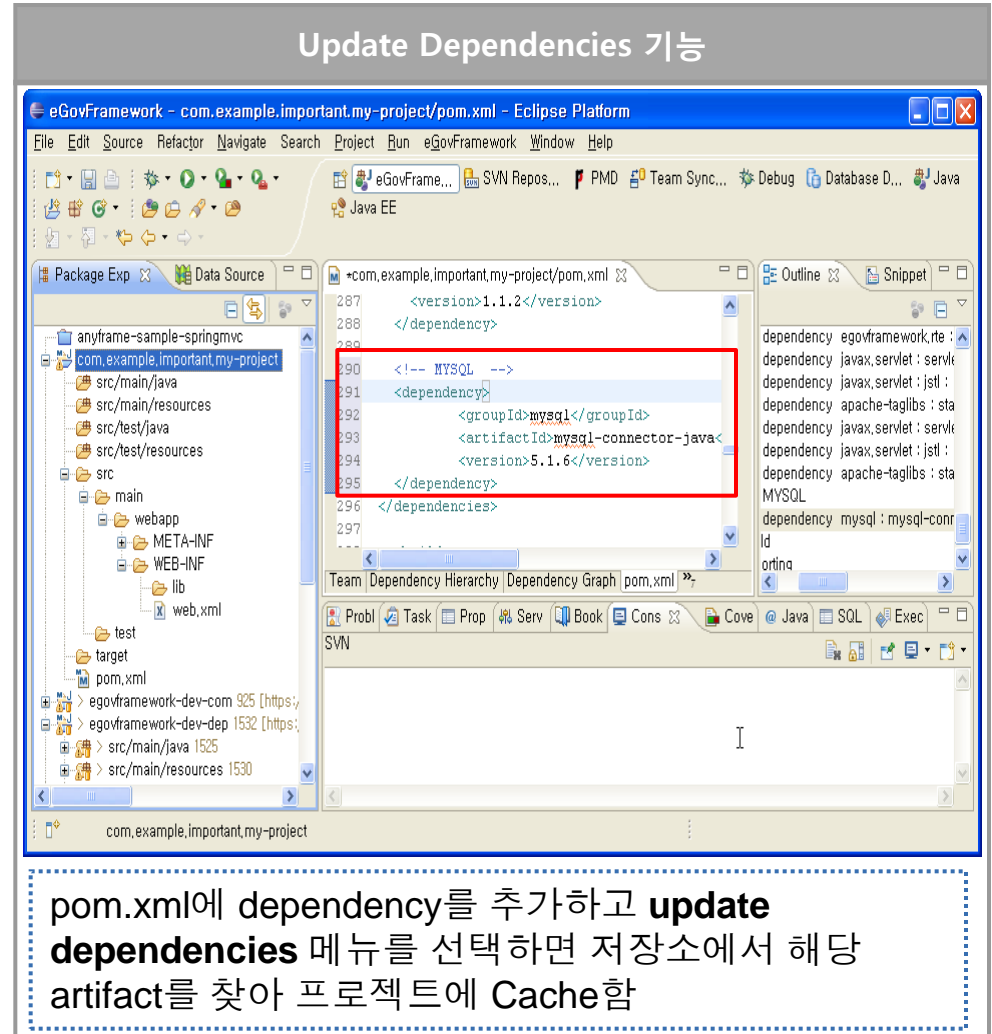
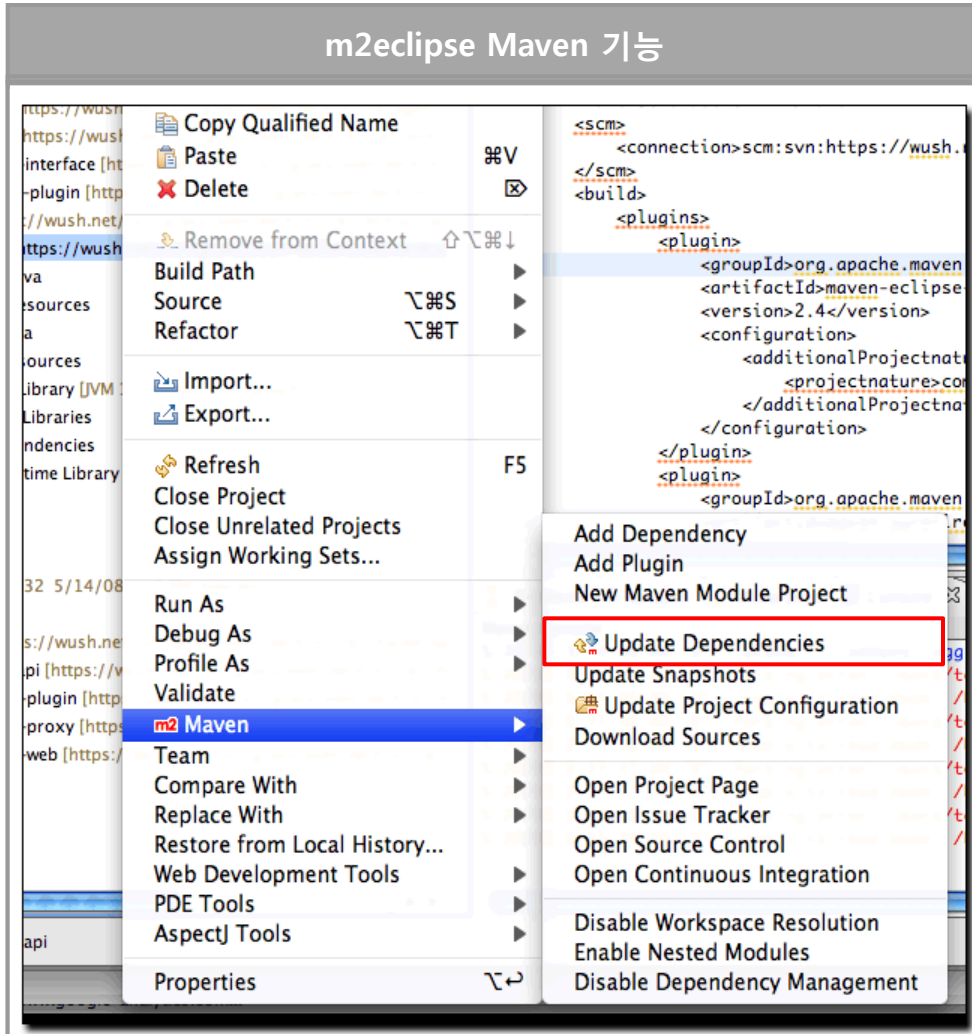
```

79 <dependencies>
80 <!-- Spring Framework -->
81 <dependency>
82 <groupId>org.springframework</groupId>
83 <artifactId>spring-aop</artifactId>
84 <version>${spring.maven.artifact.version}</version>
85 </dependency>
86 <dependency>
87 <groupId>org.springframework</groupId>
88 <artifactId>spring-beans</artifactId>
89 <version>${spring.maven.artifact.version}</version>
90 </dependency>
91 <dependency>
92 <groupId>org.springframework</groupId>
93 <artifactId>spring-context</artifactId>
94 <version>${spring.maven.artifact.version}</version>
95 </dependency>
96 <dependency>
97 <groupId>org.springframework</groupId>
98 <artifactId>spring-context-support</artifactId>
99 <version>${spring.maven.artifact.version}</version>
100 </dependency>
101 <dependency>
102 <groupId>org.springframework</groupId>
103 <artifactId>spring-core</artifactId>
104 <version>${spring.maven.artifact.version}</version>
105 </dependency>

```

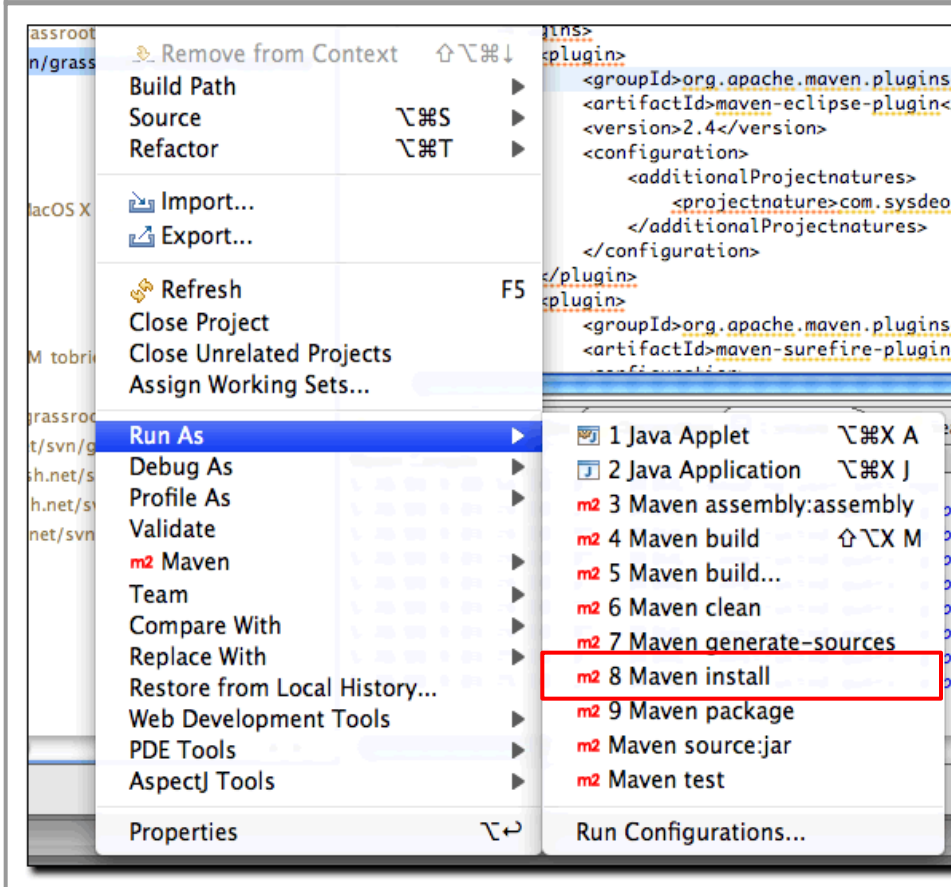
- egovframework-webapp Archetype은 Spring 2.5 기반의 dependency들을 기본으로 제공함

□ m2eclipse에서 제공하는 메뉴에서 메이븐 설정 및 연동 기능을 호출할 수 있음

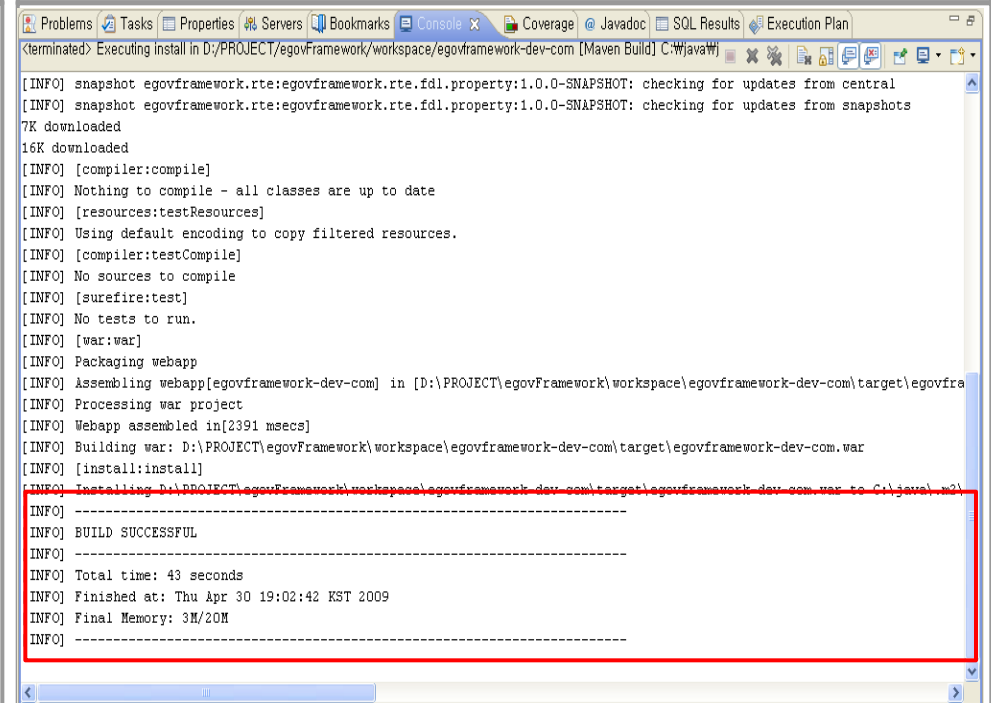


m2eclipse는 이클립스의 Run As 메뉴에 Maven 기본 생명주기 단계를 추가하여 빌드 편의성을 제공함

m2eclipse Maven 빌드 기능



Maven install 기능



Maven install 메뉴를 선택하여 생명 주기의 각 단계를 실행하고 프로젝트를 패키징하여 로컬 repository에 저장함

9. 참고 자료

Apache Maven

– <http://maven.apache.org/>

Maven Central Repository

– <http://mvnrepository.com/>

M2eclipse

– <http://m2eclipse.codehaus.org/>

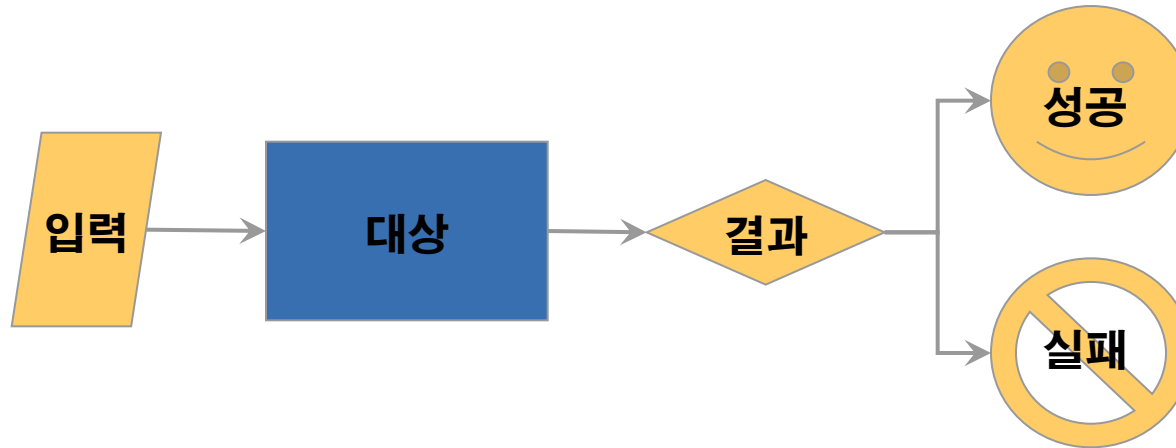
Nexus

– <http://www.sonatype.com/books/nexus-book/reference/>

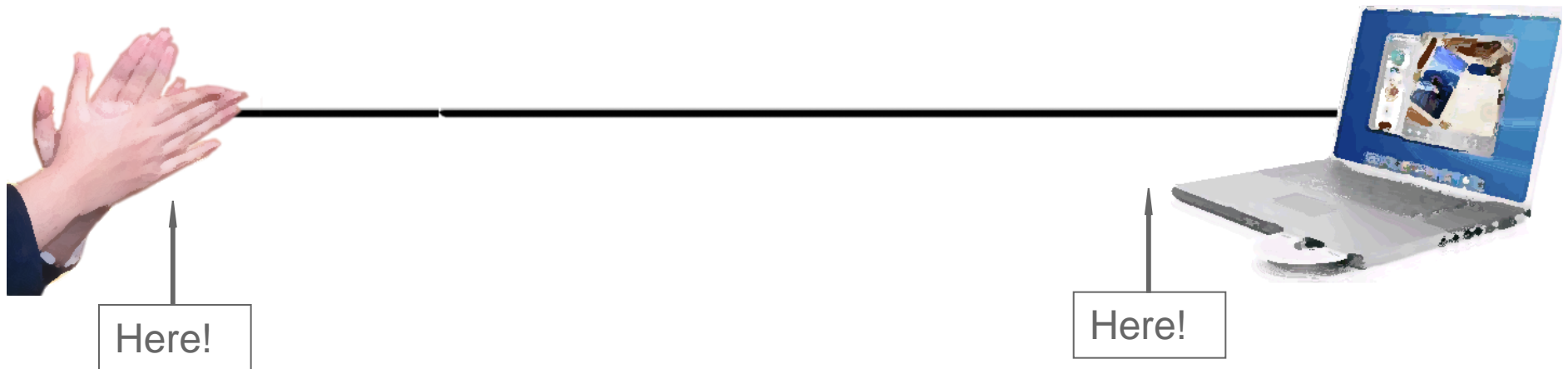
1. 테스트
2. 개요
3. 오픈소스
4. 테스트 종류
5. **Unit Test**
6. **Mock**
7. **DB Test**
8. **Test Automation**
9. **Test Reporting**
10. **Test Coverage**
11. 참고자료

□ 테스트(Test)

- 테스트 대상에 입력값을 넣었을 때 그 결과가 성공 혹은 실패의 결과를 내는 것이다.

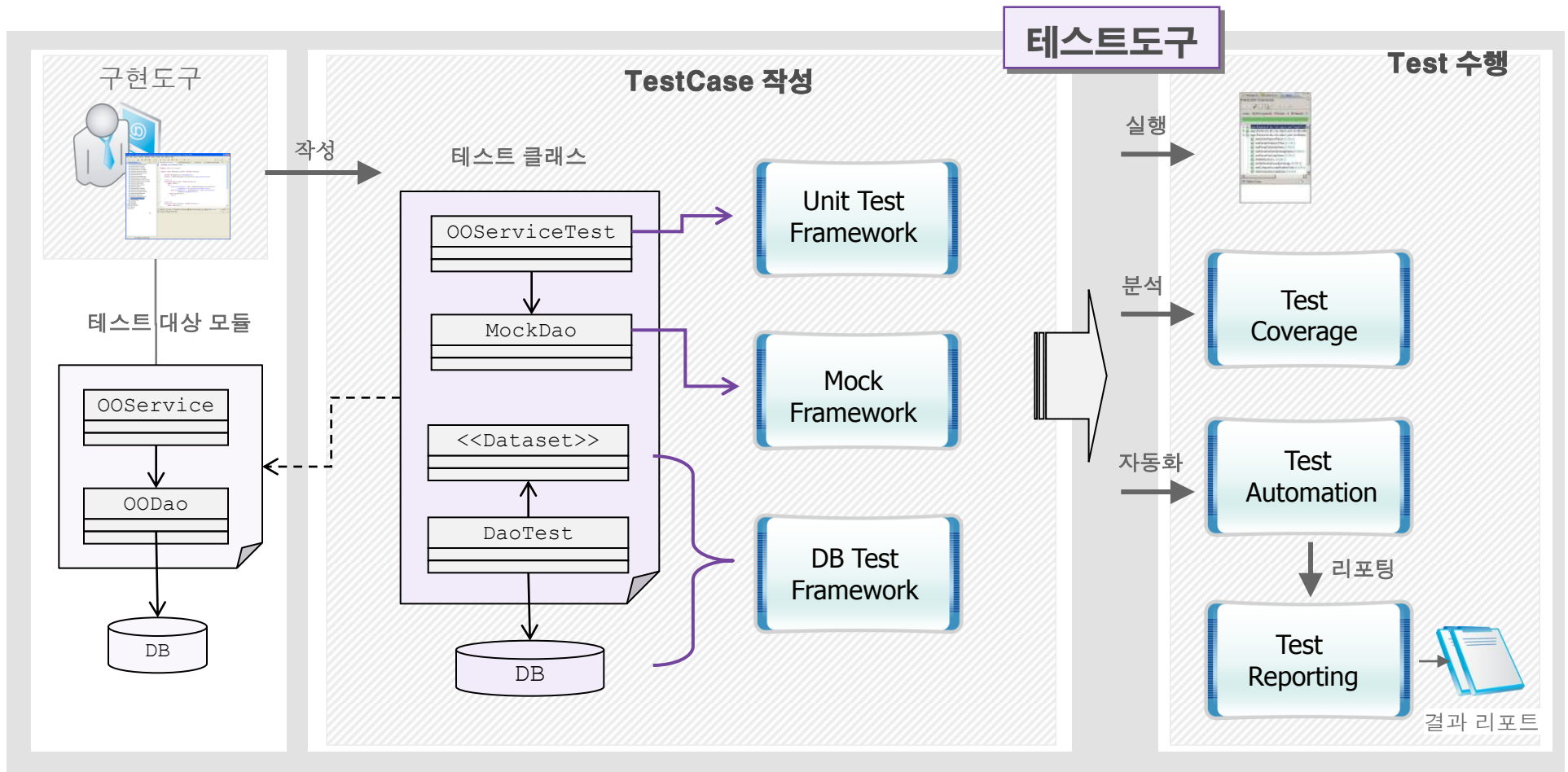


□ 수동 테스트 vs. 자동 테스트

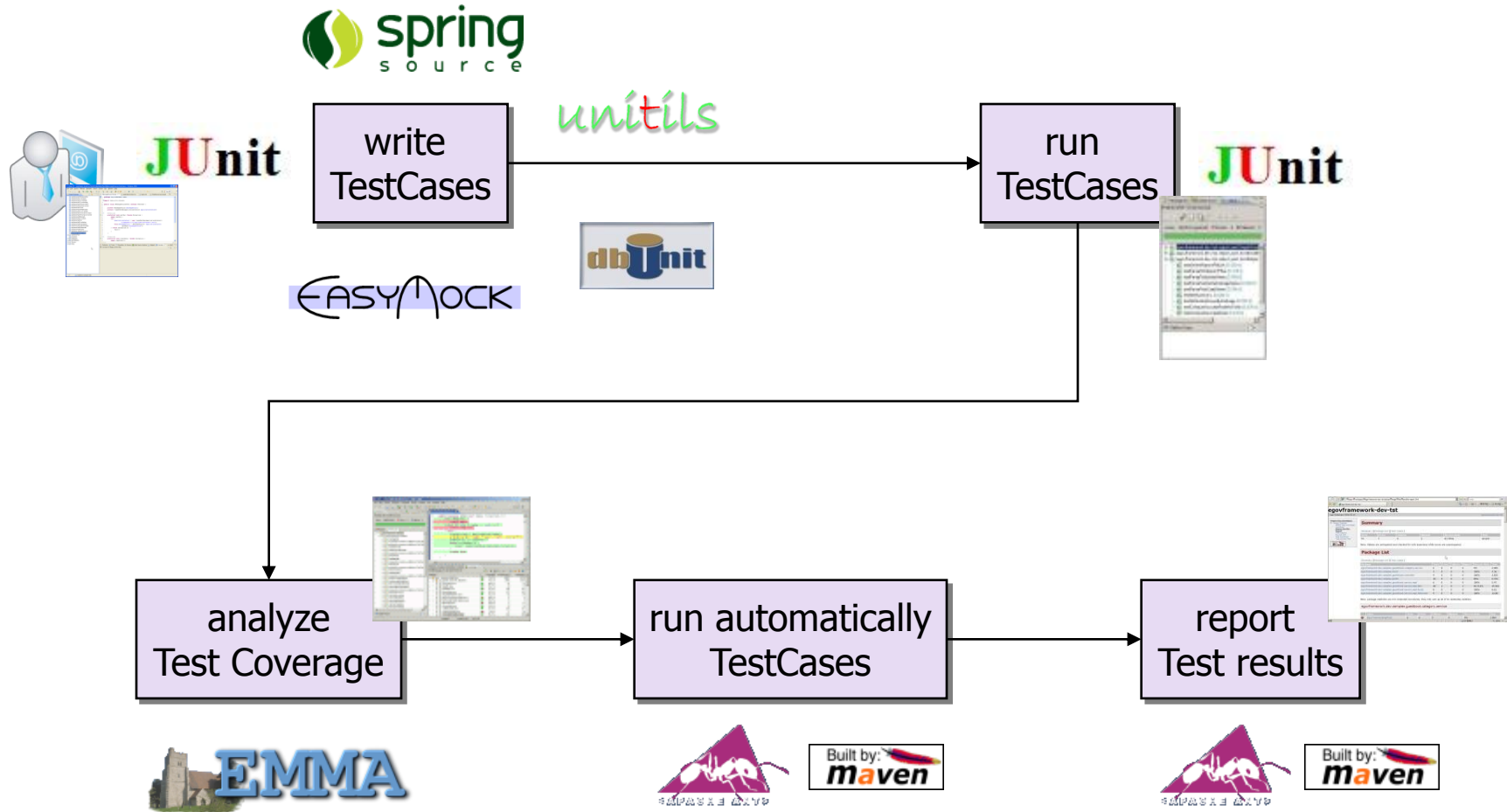


	수동 테스트	자동 테스트
장점	<ul style="list-style-type: none"> □ 쉽다. 간편하다. □ 테스트 불가능한 상황이 별로 없다. 	<ul style="list-style-type: none"> □ 언제든지 같은 테스트를 여러 번 수행 가능 □ 기존 테스트는 새 테스트를 작성하는 발판이 된다. (지식축적) □ 빠르므로 자주 돌려볼 수 있다. □ 개발자가 코드 개선 활동을 수행할 때에도 회귀테스트(Regression Test)를 자주 할 수 있다.
단점	<ul style="list-style-type: none"> □ 휘발성 □ 테스트 항목이 늘어난다면? □ 다른 사람에게 테스트 내용을 설명하기 어려움 □ 재현하기 어려움 	<ul style="list-style-type: none"> □ 코드로 작성해야 한다. <ul style="list-style-type: none"> -기술, 노하우가 필요하다 -코드로 테스트를 작성할 수 없는 상황이 있다. (수동 테스트와 공조) -테스트도 관리 대상이다. □ 모듈화가 잘 되어 있어야 한다. <ul style="list-style-type: none"> -디자인의 개선 필요 -Legacy Code의 어려움

테스트 도구는 Unit Test, Mock, DB Test Framework을 통해 TestCase 작성을 지원하고, Test Automation, Test Coverage, Test Reporting 등의 기능을 제공함



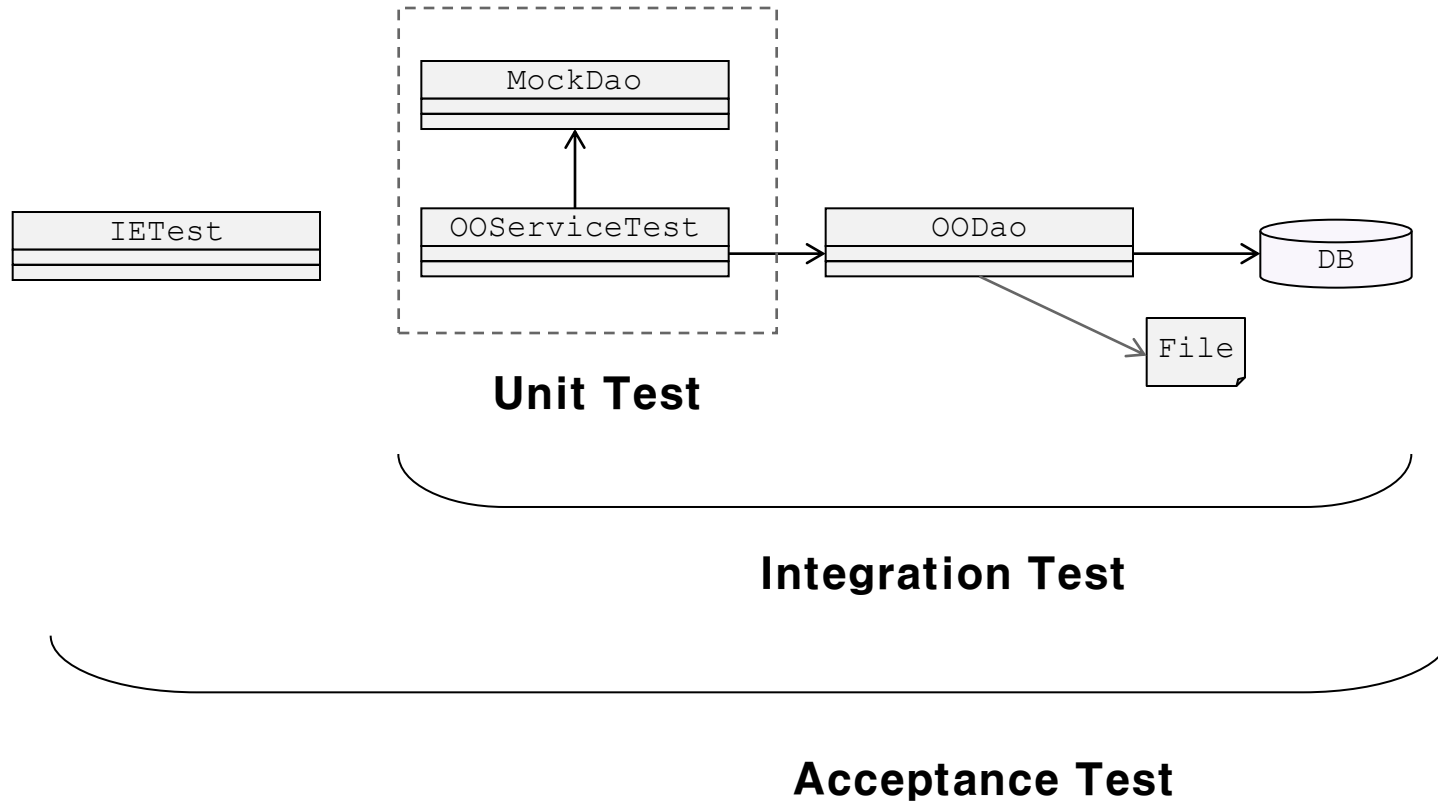
□ 테스트 도구 사용 프로세스



□ 테스트 도구에 채택한 오픈소스

구분	오픈소스	버전	비고
write & run TestCases	JUnit	4.4, 4.3	Test Framework
	EasyMock	2.4	Mock Framework
	Spring test	2.5.6	Spring Test Module
	DBUnit	2.4.2	DB Test Framework
	Unitils	2.2	JUnit, EasyMock, DbUnit, Spring Test, iBatis, Hibernate 등을 유연한 구조로 엮어서 테스트할 수 있는 기능 제공
Test Coverage	EMMA	2.0	Free Java code coverage tool
	EclEmma	1.3.2	EMMA Eclipse plugin
run automatically & report	Ant	1.6.5	Java-based build tool
	Maven	2.0	S/W project management & comprehension tool

테스트도구는 개발자가 코드로 작성하고 도구를 통해 자동화할 수 있는 테스트 중 단위테스트에 해당하며 이후 Integration Test, Acceptance Test 까지 범위를 확장할 수 있는 토대를 마련함



Unit Test란 대상 코드에 대해 테스트하고자 개발자가 작성한 코드로서, 주로 특정 메소드를 실행해서 그 결과가 기대값과 일치하는지 확인하는 형태이며, **Unit Test**는 서로 독립적으로 수행되어야 함.

□ 효과

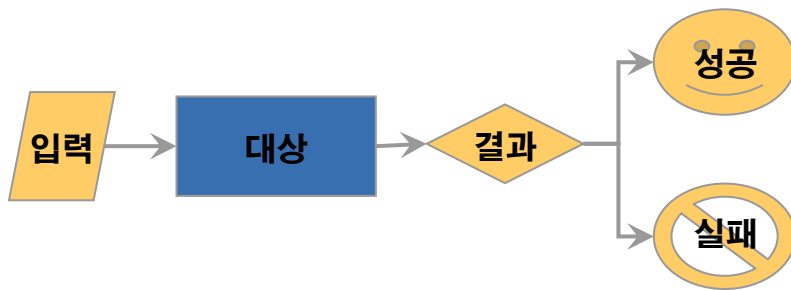
- 작성한 코드의 설계 개선 작업 시, 코드 품질에 대한 확신
- 코드 수정 시 버그를 쉽게 찾을 수 있게 해줌
- 자동화된 회귀 테스트 (Regression Test)를 가능하게 해주는 Source가 됨

□ 작성 범위

- 주요 흐름에 대한 테스트 (the happy path)
- 또 다른 주요 흐름에 대한 테스트 (the main alternative path)
- 경계 조건에 대한 테스트 (null 인자 체크 등)
- Exception 테스트 (Exception 발생하는 조건에 대한 테스트)

□ 구성

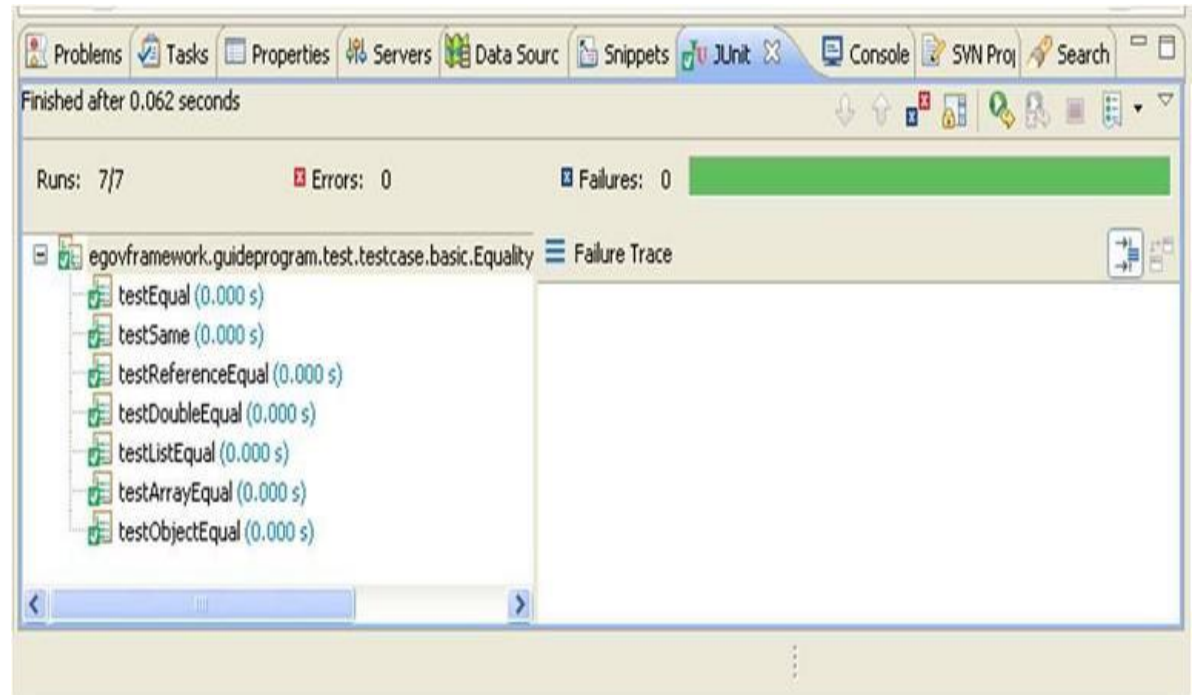
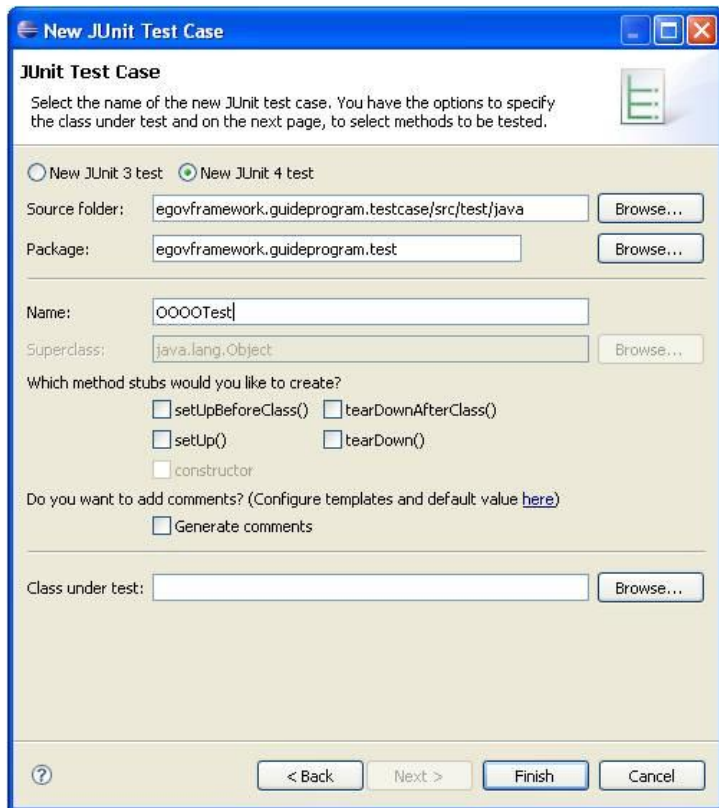
- 테스트 프레임워크를 사용하는 Class
- 공용으로 사용하는 테스트 데이터 (test fixture)
- 테스트 데이터 준비 (Setup of test data)
- 테스트 메소드 (testXXX)
 - (테스트 별 준비)
 - 테스트 대상 메소드 실행
 - assert 문을 이용한 결과 확인 (assertTrue, assertEquals etc.)
- (내부 메소드)



```

1 public class UserAdminTest {
2
3     /* Class under test */
4     private UserAdmin userAdmin;
5
6     /* A simple test user */
7     private User user; // Test Data (Fixture)
8     /* An administrator role */ //
9     private Role adminRole; //
10
11     /**
12      * Initializes the test fixture.
13      */
14     @Before
15     public void setUp() throws Exception {
16         userAdmin = new UserAdmin();
17         user = new User("John", "Doe"); // Test Data Setup
18         adminRole = new Role("Administrator"); //
19     }
20
21     /**
22      * Test for method with ...
23      */
24     @Test
25     public void testAddUser() {
26         user.setAge(18); // Extra Test Setup
27         userAdmin.addUser("jdoe", user, adminRole); // Use
28         Test Data
29
30         User result = userAdmin.getUser("jdoe");
31         assertEquals("John", result.getFirstName());
32         assertEquals("Doe", result.getLastName());
33     }
34 }
  
```

JUnit은 자바 프로그래밍 언어를 위한 Unit Test Framework로, Unit Test 코드를 작성하고 자동화된 테스트를 수행할 수 있는 기능을 제공함



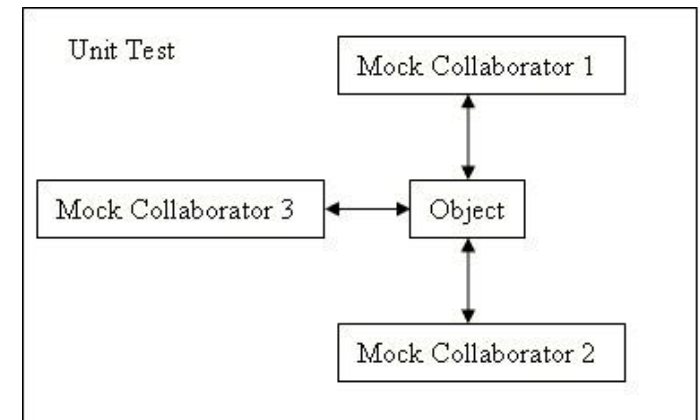
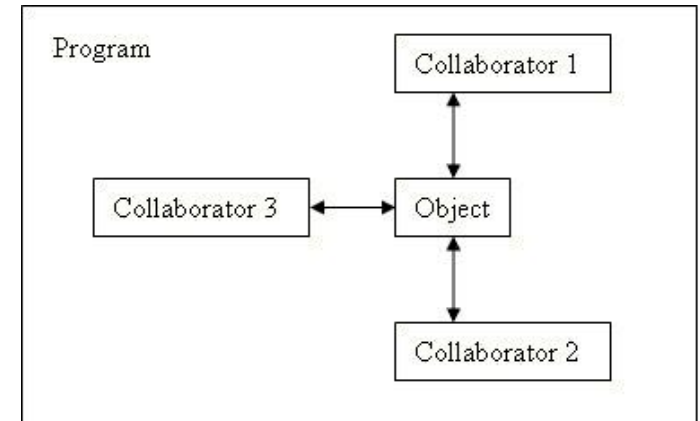
Mock 객체는 Unit Test의 독립성(isolation)을 높여주기 위해 사용하며, 테스트하고자 코드와 관련이 있는 객체(collaborator)를 흉내내어 Unit Test를 수행할 수 있도록 도와주는 객체임

□ Mock 객체를 사용하는 경우

- 진짜 객체를 준비, 설정하기 어렵다
- 진짜 객체가 느리다.
- 진짜 객체가 사용자 인터페이스를 갖거나 그 자체이다.
- 진짜 객체가 아직 없다.

□ Mock 사용 종류

- 이미 구현된 Mock 객체 사용 : Spring test (web, jndi), mockrunner
- EasyMock, JMock, Mockito 등의 Mock 라이브러리 사용
- Mock으로 사용할 가짜 클래스 직접 구현



참고 : <http://www.shinotech.com/thoughts/articles/30-unit-testing-with-mock-objects?start=1>

□ Spring Test web 사용 샘플

```
1 public class SessionMockTest {
2     /** mock object of the HttpServletRequest */
3     private MockHttpServletRequest request;
4
5     /** mock object of the HttpServletResponse */
6     private MockHttpServletResponse response;
7
8     /** mock object of the HttpSession */
9     private MockHttpSession session;
10
11     /** Target Class */
12     private Servlet servlet;
13
14     @Before
15     public void setUp() throws Exception {
16         servlet = new Servlet();
17         request = new MockHttpServletRequest();
18         session = new MockHttpSession();
19     }
20
21     @Test
22     public void testConfirmAdmin() throws Exception {
23         session.setAttribute("userid", "administrator");
24         session.setAttribute("password", "1234");
25
26         request.setSession(session);
27         servlet.confirmAdmin(request, response);
28
29         assertTrue(servlet.isAdmin());
30     }
31 }
32
```

Spring Test의
MockHttpServletRequest/Request
MockHttpSession 등 사용

테스트 대상 클래스

테스트 데이터 설정

실제 session 처럼 사용하여 테스트하고자 하는 메소드
확인

□ Easymock + Unitils 샘플

```

1  @RunWith(UnitilsJUnit4TestClassRunner.class)
2  public class EgovControllerTest {
3
4      @Mock
5      @InjectIntoByType
6      private EgovService mockService;
7
8      @TestedObject
9      private EgovController egovController = new EgovController();
10
11     @Test
12     public void testSelectList() throws Exception {
13         assertNotNull(egovController);
14         expect(mockService.selectList()).andReturn(
15             Arrays.asList(new Message(101), new Message(102)));
16         EasyMockUnitils.replay();
17
18         List<Message> resultList = egovController.selectList();
19         assertNotNull(resultList);
20         assertEquals(2, resultList.size());
21     }
22
23     @Test
24     public void testInsert() throws Exception {
25         mockService.insert(makeVO());
26         EasyMockUnitils.replay();
27
28         SessionStatus status = new SimpleSessionStatus();
29         String view = egovController.add(makeVO(), status);
30     }
31 }

```

Mock으로 생성할 인터페이스에 대한 정의한 뒤, 테스트 대상 클래스에 Mock으로 정의한 클래스를 세팅한다.

리턴값을 갖는 메소드를 Mock 클래스에 정의하고 리턴값을 설정한 뒤, 진짜 객체처럼 사용한다.

리턴하지 않는 메소드를 Mock 객체에 정의 후 진짜 객체처럼 사용한다.

DB Test는 DAO와 DB 모두를 통틀어 Persistence layer를 테스트하는 것을 말하며, 테스트DB와 테스트 데이터를 준비하여 실제로 DB를 이용한 단위테스트를 수행함

□ DB Test Framework

- DbUnit

□ DbUnit 기능

- DB 데이터를 XML 파일 형태로 import/export
- DB 연결, DB 초기화
- DB의 데이터가 기대값과 같은지 확인
- 빌드 도구를 통한 테스트 자동화 기능 제공

□ 효율적인 DB Test를 위해

- DB 구조만 갖고 있는 단위테스트용 DB 준비
- 각 개발자마다 별도의 단위테스트용 DB 공간
- Unitils, Spring Test 활용

□ TestCase 작성

```
1 @RunWith(UnitilsJUnit4TestClassRunner.class)
2 @Transactional(TransactionalMode.COMMIT)
3 @DataSet
4 @SpringApplicationContext( {
5     "/META-INF/persistence/connection/datasource-spring-with-unitils.xml",
6     "/META-INF/spring/context-common.xml",
7     "/META-INF/spring/context-sqlmap.xml" })
8 public class DaoOperationTest_noticeDao {
9     @TestDataSource
10    private DataSource dataSource;
11
12    /** Target Dao */
13    @SpringBean("noticeDao")
14    private NoticeDao noticeDao;
15
16    private NoticeVo noticeVo;
17
18    @Test
19    @ExpectedDataSet("/META-INF/**/AutoVerifyTestResultsTest_ExpectedDataSet.xml")
20    public void testInsert() {
21        assertNotNull(noticeVo);
22        noticeDao.insert(noticeVo);
23        int count = noticeDao.selectCount();
24        assertEquals(4, count);
25    }
26 }
```

3: 클래스 시작할 때 같은 위치에 있는 DaoOperationTest_noticeDao.xml 파일을 읽어 DB를 초기화

5: Datasource 연결 정보 (Unitils를 사용함)

9 : 5 라인에서 설정한 파일에 설정된 dataSource 객체를 설정함. Unitils의 Datasource 정보 사용

13 : Unitils의 Injection of Spring beans - noticeDao 라는 이름으로 정의된 Bean을 Spring Application Context로부터 가져옴

19 : 지정된 위치의 Dataset 파일을 읽어 들여, 메소드 처리 후반에서 결과를 비교함. assert 기능

□ DataSet (AutoInsertionTestDataTest_DataSet.xml)

```
<?xml version="1.0" encoding="UTF-8"?>  
  
<dataset>  
  <NOTICE NOTICE_ID="101"  
    NOTICE_TITLE="101번 공지"  
    NOTICE_FILE_CNT="0"  
    NOTICE_RETRIEVED_CNT="0"  
  />  
  <NOTICE NOTICE_ID="102"  
    NOTICE_TITLE="102번 공지"  
    NOTICE_FILE_CNT="0"  
    NOTICE_RETRIEVED_CNT="0"  
  />  
  <NOTICE NOTICE_ID="103"  
    NOTICE_TITLE="103번 공지"  
    NOTICE_FILE_CNT="0"  
    NOTICE_RETRIEVED_CNT="0"  
  />  
</dataset>
```

□ DataSet 구조

```
<dataset>  
  <TABLE_NAME COLUMN_NAME1="VALUE"  
    COLUMN_NAME2="VALUE"  
  />  
</dataset>
```

□ Expected DataSet (AutoVerifyTestResultsTest_ExpectedDataSet.xml)

```
<?xml version="1.0" encoding="UTF-8"?>  
  
<dataset>  
  <NOTICE NOTICE_ID="201"  
    NOTICE_TITLE="201번 공지"  
    NOTICE_CONTENTS="테스트용으로 자동 입력된 공지사항 201번입니다."  
    NOTICE_LAST_MODIFIER="OracleDataSetTest.class"  
  />  
</dataset>
```

작성된 다수의 TestCase를 자동으로 수행하여 소스 단위의 품질을 높이고자 함. CI 서버와 연계될 때, 주기적인 테스트 수행으로 인해 목표 시스템의 품질을 향상할 수 있음

□ Test Suite

- TestCase 모음

1. Test Suite Class 작성

```
@RunWith(Suite.class)
@SuiteClasses( { HttpRequestMockTest.class, SessionMockTest.class,
FileUploadMockTest.class, EmailMockTest.class, JDBCMockTest.class })
public class MockTestSuite {

}
```

□ Test Suite 설정 방법

- Test Suite Class 작성
- 빌드 도구의 batchtest 묶음

2. batchtest로 묶음

```
<junit . . .>
  <batchtest fork="yes" todir="${testreports.dir}">
    <fileset dir="${testbuild.dir}">
      <include name="**/*Test.class" />
      <exclude name="**/Abstract*Test.class" />
    </fileset>
  </batchtest>
</junit>
```

❑ Ant (build.xml)

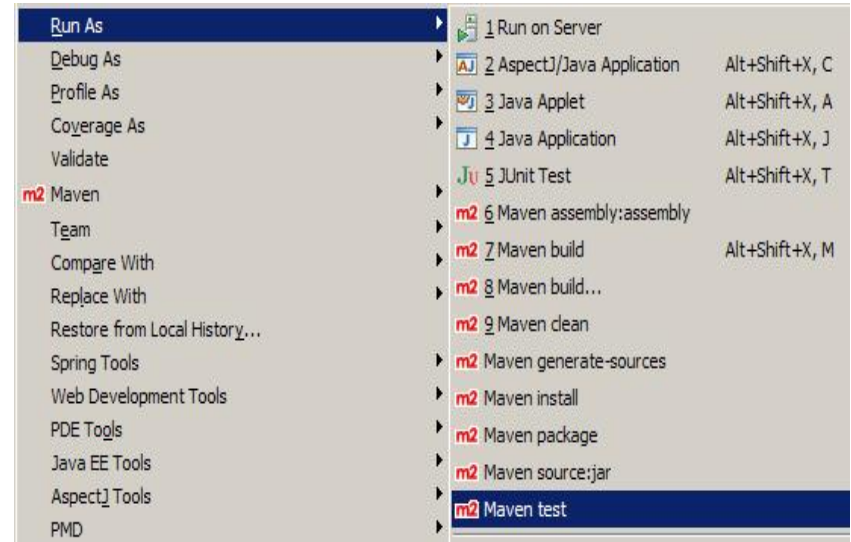
```
<junit forkmode="perBatch" printsummary="true" haltonfailure="yes" haltonerror="yes">
  <classpath refid="master-classpath" />
  <classpath refid="test-classpath" />
  <classpath path="${testbuild.dir}" />

  <formatter type="xml" />

  <batchtest fork="yes" todir="${testreports.dir}">
    <fileset dir="${testbuild.dir}">
      <include name="**/*Test.class" />
      <exclude name="**/Abstract*Test.class" />
    </fileset>
  </batchtest>
</junit>
```

❑ Maven (pom.xml) goal : test

```
<build>
  <plugins>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-surefire-plugin</artifactId>
      <configuration>
        <reportFormat>xml</reportFormat>
        <excludes>
          <exclude>**/Abstract*.java</exclude>
        </excludes>
        <includes>
          <include>**/*Test.java</include>
        </includes>
      </configuration>
    </plugin>
  </plugins>
</build>
```



□ Ant (build.xml)

```

<!-- Junit Test Result Report -->
<target name="junitreport" depends="tests">
  <junitreport todir="${testhtml.dir}">
    <fileset dir="${testreports.dir}">
      <include name="TEST-*.xml" />
    </fileset>
    <report format="frames" todir="${testhtml.dir}" />
  </junitreport>
</target>
  
```

□ Maven (pom.xml)

```

<reporting>
  <plugins>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-surefire-report-plugin</artifactId>
      <version>2.4.2</version>
    </plugin>
  </plugins>
</reporting>
  
```

Unit Test Results

Designed for use with [JUnit](#) and [Ant](#).

Summary

Tests	Failures	Errors	Success rate	Time
60	0	26	56.67%	65.561

Note: failures are anticipated and checked for with assertions while errors are unanticipated.

Packages

Name	Tests	Errors	Failures	Time(s)	Time Stamp	Host
org.springframework.samples.petclinic	1	0	0	0.672	2009-03-11T02:22:19	lgcns-juroh
org.springframework.samples.petclinic.hibernate	10	0	0	8.453	2009-03-11T02:22:23	lgcns-juroh
org.springframework.samples.petclinic.jdbc	10	0	0	5.109	2009-03-11T02:22:35	lgcns-juroh
org.springframework.samples.petclinic.jpa	39	26	0	51.327	2009-03-11T02:22:44	lgcns-juroh

egovframework-dev-tst

Last Published: 2009-03-12

Summary

Tests	Errors	Failures	Skipped	Success Rate	Time
74	7	0	5	83.784%	60.845

Note: failures are anticipated and checked for with assertions while errors are unanticipated.

Package List

Package	Tests	Errors	Failures	Skipped	Success Rate	Time
egovframework.dev.samples.guestbook.category.service	6	6	0	0	0%	2.984
egovframework.dev.samples.mock	3	0	0	0	100%	4.36
egovframework.dev.samples.guestbook.controller	4	0	0	0	100%	2.625
egovframework.dev.samples.junit4	10	0	0	2	80%	0.375
egovframework.dev.samples.guestbook.service.impl	6	0	0	0	100%	5.47
egovframework.dev.samples.guestbook.service.impl.jdbc	28	1	0	3	85.714%	25.561
egovframework.dev.samples.guestbook.service.impl.ibatis	8	0	0	0	100%	6.61
egovframework.dev.samples.guestbook.service.impl.hibernate	9	0	0	0	100%	12.86

Note: package statistics are not computed recursively, they only sum up all of its testsuites numbers.

egovframework.dev.samples.guestbook.category.service

Class	Tests	Errors	Failures	Skipped	Success Rate	Time
EgovTxServiceImpITest	6	6	0	0	0%	2.984

□Ant – Default로 사용한 예

```
<path id="egov.lib">
  <path refid="master-classpath" />
  <path location="${antlib.dir}/egovtest/egovframework-dev-tst-ant.jar" />
</path>

<taskdef resource="egovtest.properties" classpathref="egov.lib" />

<!-- JUnit Excel Report -->
<target name="egovtest" depends="tests">
  <egov-junitreport todir="${testxls.dir}">
    <fileset dir="${testreports.dir}" includes="**/TEST-*.xml" />
  </egov-junitreport>
</target>
```

Excel Reporting Ant Task 설정

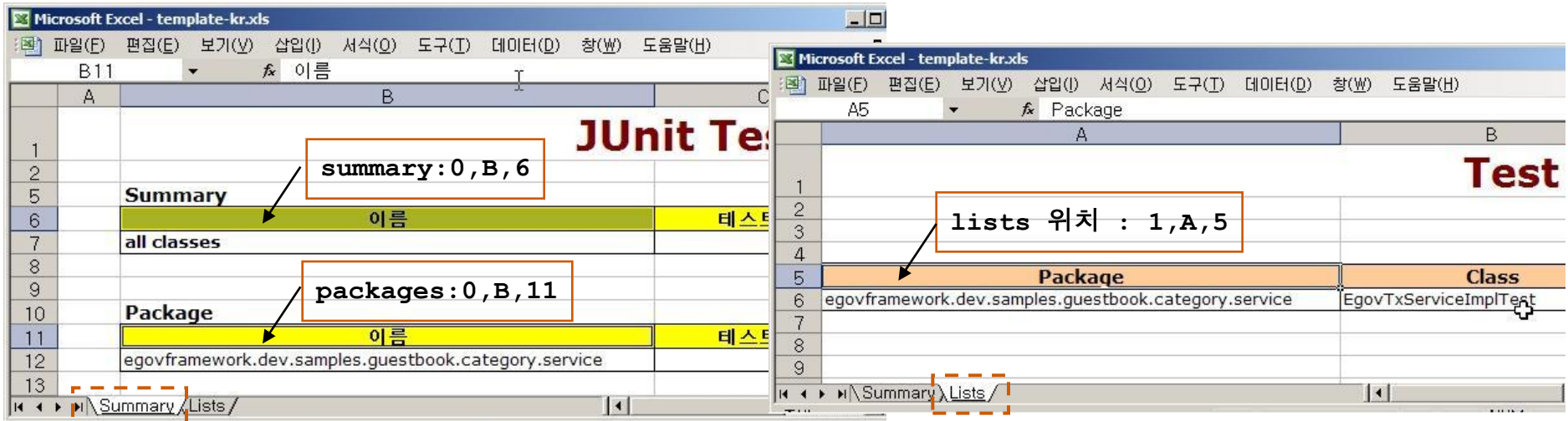
Default 사용예

□Ant – 템플릿 엑셀 파일을 별도로 사용한 예

```
<target name="egovtest-full" depends="tests">
  <egov-junitreport todir="${testxls.dir}"
    outputname="egovtest-junit-full.xls"
    templatepath="${basedir}/build/template-kr.xls"
    summary="0,B,6"
    packages="0,B,11"
    lists="1,A,5">
    <fileset dir="${testreports.dir}"
      includes="**/TEST-*.xml" />
  </egov-junitreport>
</target>
```

□Ant - 템플릿 엑셀 파일을 별도로 사용한 예

property	설명
todir	엑셀 파일 생성 디렉토리
outputname	엑셀 파일명
templatepath	템플릿 엑셀 파일 정보
summary, packages, lists	각 엑셀 테이블의 헤더 위치 정보 (sheet, column, row)
fileset	테스트 결과 XML 파일 위치



□ Maven (goal : egovtest:junit-xls)

```

<build>
  <plugins>
    <!-- Egovframework JUnit Excel Reporting -->
    <plugin>
      <groupId>egovframework.dev</groupId>
      <artifactId>egovtest-maven-plugin</artifactId>
      <version>1.0.0-SNAPSHOT</version>
    </plugin>
  </plugins>
</build>
    
```

JUnit Test Results Summary							
2009-03-27							
Summary							
name	Tests	Errors	Failures	Skipped	Success Rate	Time (s)	
all classes	44	6	0	4	77.27%	36.172	
Package							
name	Tests	Errors	Failures	Skipped	Success Rate	Time (s)	
egovframework.dev.samples.guestbook.category.service	6	6	0	0	0.00%	2.984	
egovframework.dev.samples.junit4	6	0	0	2	66.67%	0.156	
egovframework.dev.samples.guestbook.service.impl	5	0	0	0	100.00%	3.313	
egovframework.dev.samples.guestbook.service.impl.jdbc	18	0	0	2	88.89%	16.859	
egovframework.dev.samples.guestbook.service.impl.hibernate	9	0	0	0	100.00%	12.860	

❑ Unitils Guildelines

– <http://unitils.sourceforge.net/guidelines.html>

❑ Mock Object를 사용해서 쉽게 테스트하기

– <http://www.ibm.com/developerworks/kr/event/screencast/final/01/>

❑ mock object

– <http://www.mockobjects.com/>

❑ Effective Unit Testing with DbUnit

– <http://www.onjava.com/pub/a/onjava/2004/01/21/dbunit.html>

❑ What is a mock objects

– <http://www.shinetech.com/display/www/What+Are+Mock+Objects%3F>

❑ An early look at JUnit4

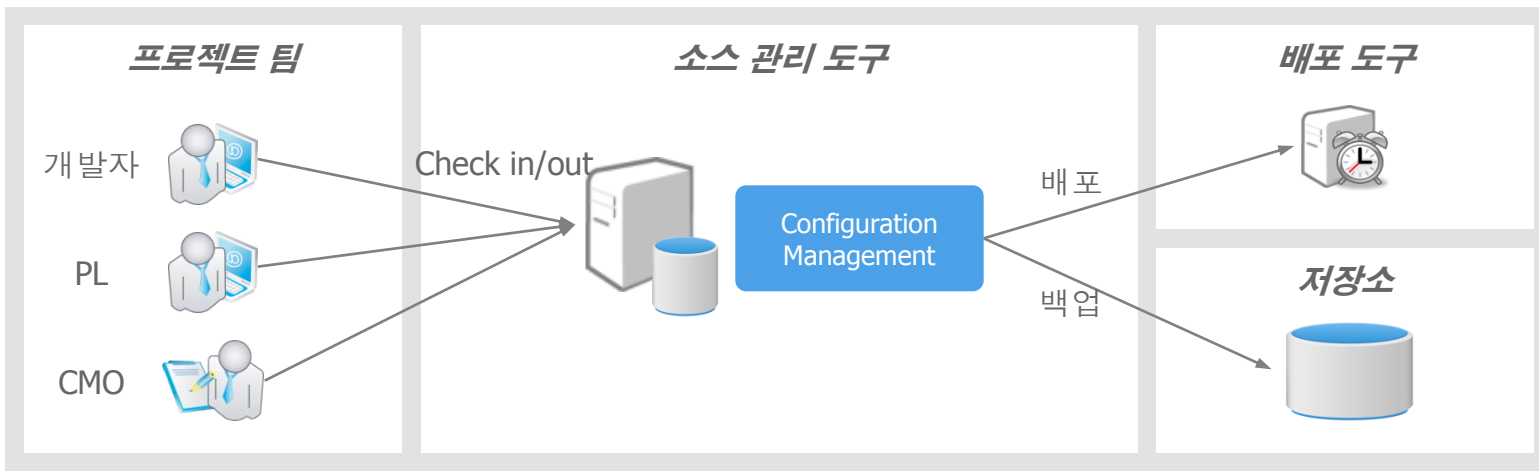
– <http://www.ibm.com/developerworks/java/library/j-junit4.html>

❑ JUnit FAQ

– <http://junit.sourceforge.net/doc/faq/faq.htm>

1. 개요
2. **Subversion** 개요
3. 소스 공유
4. **Checkout**
5. **Update**
6. **Commit**
7. **History** 조회
8. 충돌 해결(merge)
9. 참고자료

- 소스 버전관리 도구는 시스템 형상 요소(소스 및 데이터)를 문서화하고 변경을 소스 버전관리 절차에 따라 관리하고, 효율적으로 처리 하기 위한 통제 환경을 의미
- 소스 버전관리 도구의 특징 및 기능
 - 형상 요소(소스 및 데이터)에 대한 변경 사항을 관리
 - 버전 관리를 수행하고, 배포도구를 통해 변경사항을 배포
 - 일정 주기에 백업 도구를 통해 백업을 수행
- 소스 버전관리 도구의 구성요소 및 관계도



□ Subversion(SVN)의 배경

- CVS의 단점을 개선하고 CVS를 대체할 목적으로 개발되었음
- 오픈소스 기반의 소스 버전관리 도구 (<http://subversion.tigris.org>)

□ CVS 대비 Subversion의 특징점

- CVS와 거의 동일한 사용법. CVS 사용자는 누구나 쉽게 사용할 수 있음
- Commit 단위: CVS는 개별 파일별로 리비전 관리, SVN은 변경이 발생하는 체인지셋(Change Set)별로 리비전 관리
- Atomic Commit: 여러 파일 커밋시, 어느 하나의 파일에서 커밋이 실패할 경우, CVS는 해당 파일 앞의 경우는 커밋되고, 뒷 파일은 그대로 남아있지만, SVN은 모두 이전 상태로 롤백
- 트리별, 파일별 접근 제어 리스트 지원
- 파일에 대한 이름변경/이동, 디렉터리 버전 관리도 지원
- CVS에 비해 빠른 동작 속도(Update/Branching/Tagging Time)

□ SVN 클라이언트 - Eclipse 플러그인

– Subversive

- Polarion 커뮤니티 그룹에서 개발
- Eclipse Ganymede 버전부터 SVN 클라이언트로 공식 채택
- 빠른 update, 안정적인 동작
- 공식 웹 페이지: <http://www.eclipse.org/subversive/>

– Subclipse

- Tigris 커뮤니티 그룹(Subversion 개발 커뮤니티)에서 개발
- Subversion 초창기 폭넓은 사용자 확보
- Subversive 대비 느린 update, 다수의 버그로 불안정
- 공식 웹 페이지: <http://subclipse.tigris.org/>

□ SVN 클라이언트 Eclipse 플러그인 설치/확인

① Eclipse 메뉴, Help > Software Update...

② 플러그인 Update Sites 추가, Install 진행

③ 설치 후 확인 Eclipse 메뉴, Help > About

Subversive SVN 1.5 JavaHL Connector

Update Site	Location
Mylyn for Eclipse 3.4	http://download.eclipse.org/tools/mylyn/update/e3.4
QualityEclipse Plug...	http://download.instantiations.com/out/QualityEclipse/integration/latest/update/3.4
Subversive Site	http://download.eclipse.org/technology/subversive/0.7/update-site/
SVN Connectors Site	http://www.polarion.org/projects/subversive/download/eclipse/2.0/update-site/
Web Tools (WTP) U...	http://download.eclipse.org/webtools/updates/

❑ 저장소(Repository)

- 모든 프로젝트의 프로그램 소스들과 소스에 대한 변경 사항이 저장 됨
- 네트워크를 통해서 여러 사람이 접근
- SVN에서는 Berkeley DB(Default) 또는 파일시스템(Optional)을 저장소로 사용

❑ 저장소 접근

① Eclipse 메뉴,
Window > Open Perspective
> Other... >
SVN Repository Exploring

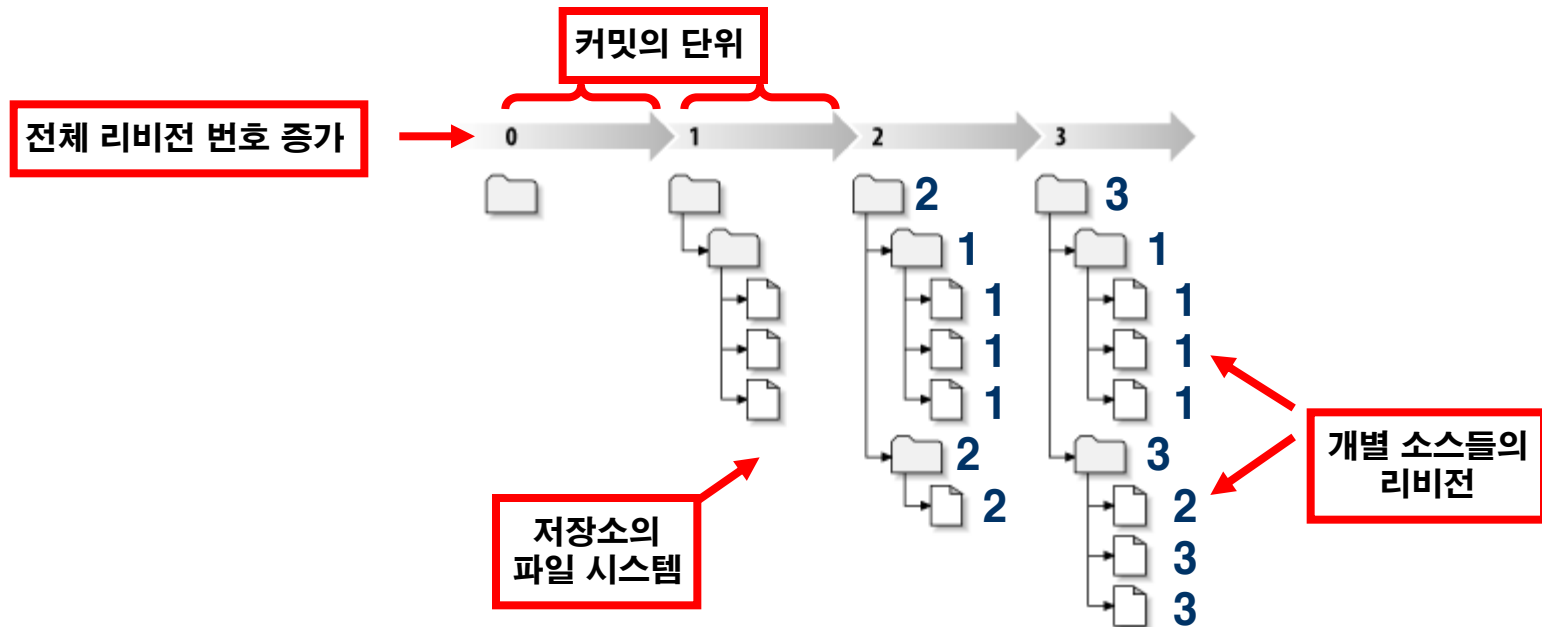
② Eclipse 메뉴,
File > New >
Repository Location

③ 마법사에서,
URL과 Authentication 항목을 입력

□ Revisions

- 소스 파일 등을 수정하여 커밋하게 되면 일정한 규칙에 의해 숫자가 증가
- 저장소에 저장된 각각의 파일 버전이라 할 수 있음
- SVN의 경우 파일 별로 리비전이 매겨지지 않고 변경발생 단위로 전체 리비전이 매겨 짐
- 리비전을 보고 프로젝트 진행 상황을 알 수 있음

□ 저장소를 통해 본 리비전 이해



□ 프로젝트 공유

- 새로운 프로젝트를 저장소에 추가

The screenshot shows the Eclipse IDE interface with the 'Team' menu open and 'Share Project...' selected. A red box highlights this menu path with the text: ① Eclipse 패키지/네비게이션 뷰, 프로젝트 오른쪽 클릭 > Team > Share Project... A red arrow points from this box to the 'Share Project Wizard' dialog. In the wizard, the 'SVN' repository type is selected in the 'Select a repository type:' list, highlighted by a red box with the text: ② Share Project 마법사, SVN 선택. Another red arrow points from this box to the 'Use existing repository location:' section of the wizard. This section contains a table with repository locations, and the 'Finish' button is highlighted by a red box with the text: ③ Share Project 마법사, 저장소 위치 선택 > Finish.

① Eclipse 패키지/네비게이션 뷰,
프로젝트 오른쪽 클릭 >
Team > Share Project...

② Share Project 마법사,
SVN 선택

③ Share Project 마법사,
저장소 위치 선택 > Finish

Label	URL
https://192.168.100.14:8443/svn/egovdev	https://192.168.100.14:8443/svn/egovdev
https://192.168.100.14:8443/svn/freework	https://192.168.100.14:8443/svn/freework

□ 소스 단위 공유

- 공유 중인 프로젝트 내에서 해당 소스를 저장소에 추가

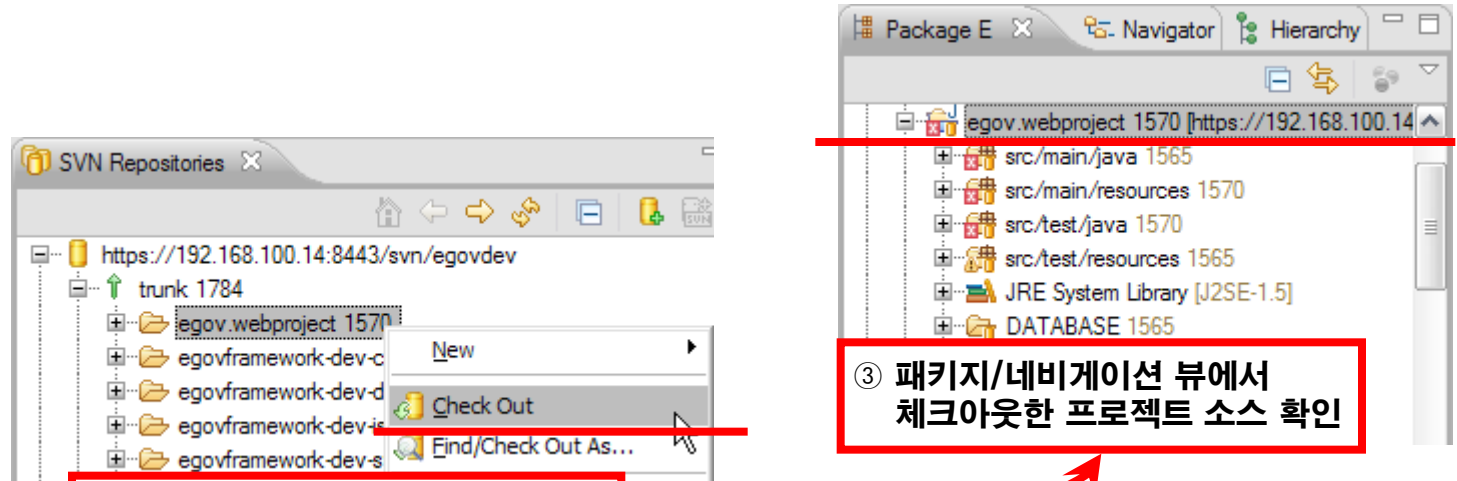
① Eclipse 패키지/네비게이션 뷰, 대상 소스 오른쪽 클릭 > Team > Add to Version Control...

Resource	Content	Properties
<input checked="" type="checkbox"/> egovframework.dev.tst.togenerator/src/egovfra...	New	

② 팝업 창에서 내용 확인 > OK

□ 초기 checkout

- 저장소에서 전체 소스의 최종 리비전을 받아오는 것
- 저장소 인증 정책에 따라 ID, Password 입력여부가 결정



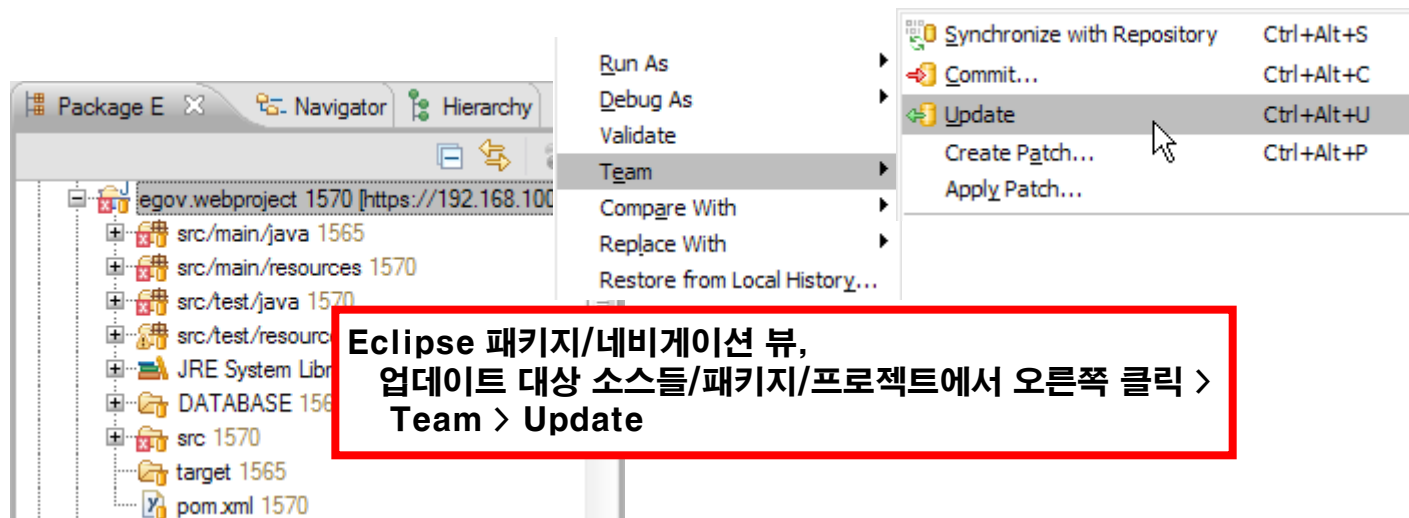
② SVN Repositories 뷰,
저장소를 확장하여 trunk 아래의
프로젝트 선택 > Check Out

③ 패키지/네비게이션 뷰에서
체크아웃한 프로젝트 소스 확인

① Eclipse 메뉴,
Window > Open Perspective
> Other... >
SVN Repository Exploring

□ 저장소에 소스 업데이트

- 체크아웃을 한 이후의 타인에 의한 소스 변경사항을 확인
- 저장소에 있는 소스 중 로컬과 비교하여 변경된 항목의 최신 버전의 소스를 가져옴
- 변경된 항목의 최신 버전은 로컬 본에 통합됨



□ Commit

- 로컬에 체크아웃 한 소스를 수정, 파일 추가, 삭제 등을 한 뒤 저장소에 저장하여 갱신 하는 것
- 커밋을 하면 전체 리비전이 1 증가(CVS의 경우 수정한 각각 파일의 리비전이 증가)

② 팝업 창에서 반영 내용 확인 및 Comment 입력

① Eclipse 패키지/네비게이션 뷰, 커밋 대상 소스들/패키지/프로젝트에서 오른쪽 클릭 > Team > Commit...

Resource	Content	Properties
<input checked="" type="checkbox"/> egovframework.dev.tst.tcgenerator/lib/easysql.jar	Deleted	
<input checked="" type="checkbox"/> egovframework.dev.tst.tcgenerator/.classpath	Modified	
<input checked="" type="checkbox"/> egovframework.dev.tst.tcgenerator/build.properties	Modified	
<input checked="" type="checkbox"/> egovframework.dev.tst.tcgenerator/META-INF/MANIFEST...	Modified	
<input checked="" type="checkbox"/> egovframework.dev.tst.tcgenerator/src/egovframework/de...	Modified	

□ History

- SVN에 의해 관리되는 특정 소스와 관련된 과거 모든 리비전의 내역

□ History 조회

- Eclipse의 기본 카테고리인 'Team'의 'History' 뷰를 통해 조회할 수 있음

① Eclipse 패키지/네비게이션 뷰, 대상 소스 오른쪽 클릭 > Team > Show History

Revision	Date	C...	Author	Comment
*634	09. 4. 29. 오후 5:11	7	suyol_yun	query section maximize added
628	09. 4. 29. 오전 10:25	5	hjee	[no comment]
625	09. 4. 29. 오전 10:12	7	hyungjoe	In/Out 의 Class, Map 항목의 배타적 표시
621	09. 4. 24. 오후 12:38	7	hyungjoe	[no comment]
618	09. 4. 23. 오후 2:30	7	hyungjoe	1.바인딩 변수타입 선택...
610	09. 4. 22. 오전 10:43	1	hjee	[no comment]
608	09. 4. 21. 오후 1:31	7	hyungjoe	[no comment]

리비전 로그 목록

선택된 리비전의 Comment

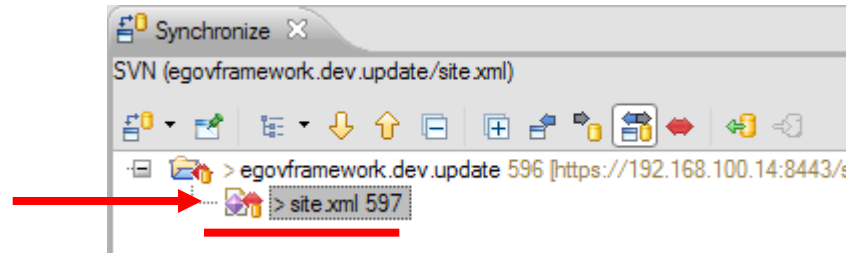
선택된 리비전의 구조와 변경 파일들

② History 뷰에서 대상 소스의 History 목록과 리비전 상세 내역 조회

□ 충돌(Conflict)

- 로컬에 체크아웃 이후 수정한 소스를 저장소에 커밋할 때, 저장소의 리비전이 더 높을 경우 충돌 발생 (타인에 의해 수정되고 커밋된 상태)

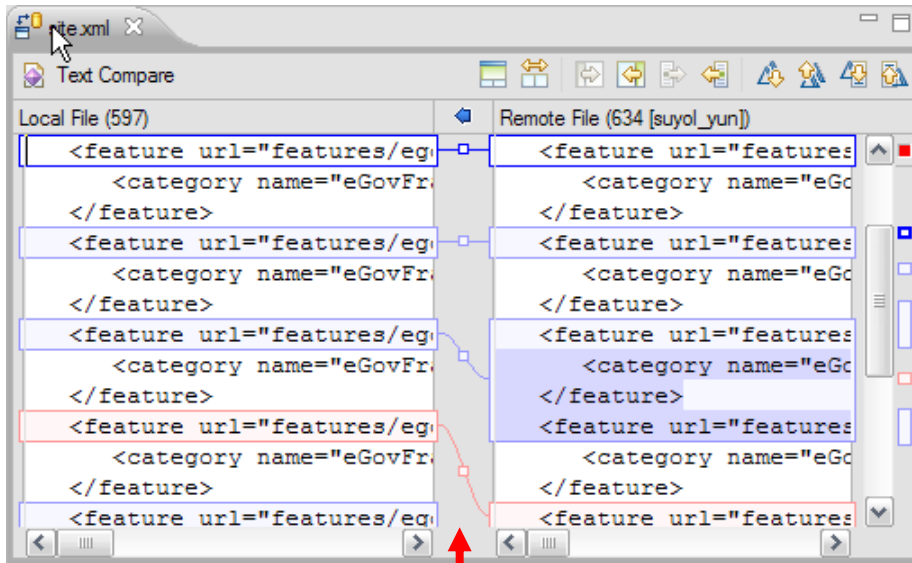
충돌 발생상황 예:
로컬 리비전 = 597 < 저장소 리비전 = 634



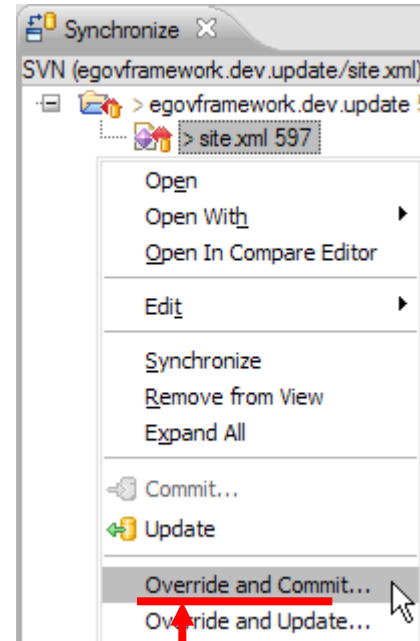
□ 충돌 해결 방안

- postpone: 즉시 반영없이, 소스 코드를 계속 수정. 수정이 완료된 시점에 다시 시도 (충돌 해결 안함)
- diff: 충돌된 상황을 비교. 로컬과 저장소의 소스의 차이를 비교
- edit: 로컬의 파일을 다시 열어 수정
- mine-full: 저장소의 리비전 내용을 무시하고, 로컬의 내용으로 커밋.
- theirs-full: 로컬의 수정 내역을 무시하고, 저장소의 리비전으로 대체하여 업데이트

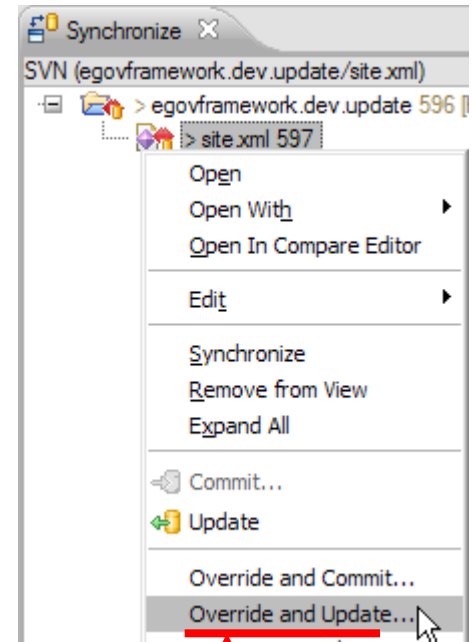
□ 충돌 해결 방안 예



diff/edit:
로컬과 저장소의 소스를
모두 열어 비교/편집



mine-full:
충돌이 발생한 파일에서 오른쪽 클릭 >
Override and Commit... 선택



theirs-full:
충돌이 발생한 파일에서 오른쪽 클릭 >
Override and Update... 선택

9. 참고 자료

❑ Subversion Project Home Page

– <http://subversion.tigris.org/>

❑ CVS Project Home Page

– <http://www.nongnu.org/cvs/>

❑ Subversive Eclipse Plug-in

– <http://www.eclipse.org/subversive/>

❑ Subversive SVN Connectors Eclipse Plug-in

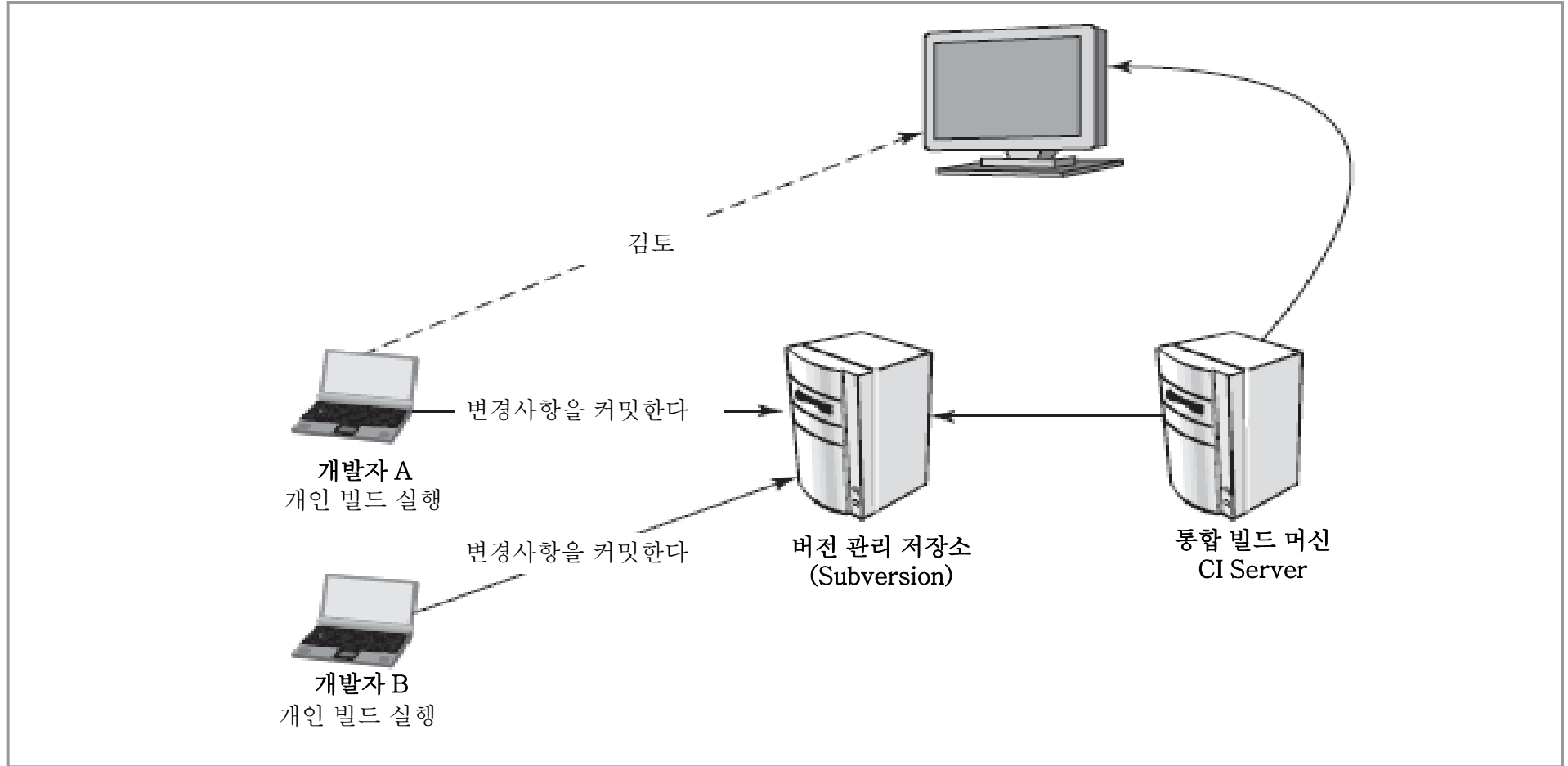
– <http://www.polarion.com/products/svn/subversive.php?src=eclipseproject>

❑ Subclipse Eclipse Plug-in

– <http://subclipse.tigris.org/>

1. CI 정의
2. CI 효과와 특징
3. CI 구성 요소
4. Hudson 개요
5. Hudson 특징 및 주요 기능
6. Hudson 대시보드
7. Hudson 프로젝트 메뉴
8. Hudson – JUnit 테스트 연동
9. Hudson – 코드 인스펙션 도구 연동
10. 참고자료

여러 명으로 구성된 팀이 작업한 것을 자주-각 팀원이 하루에 한 번 이상, 매일 여러 번-통합하는 것을 가리키는 소프트웨어 개발 Practice임



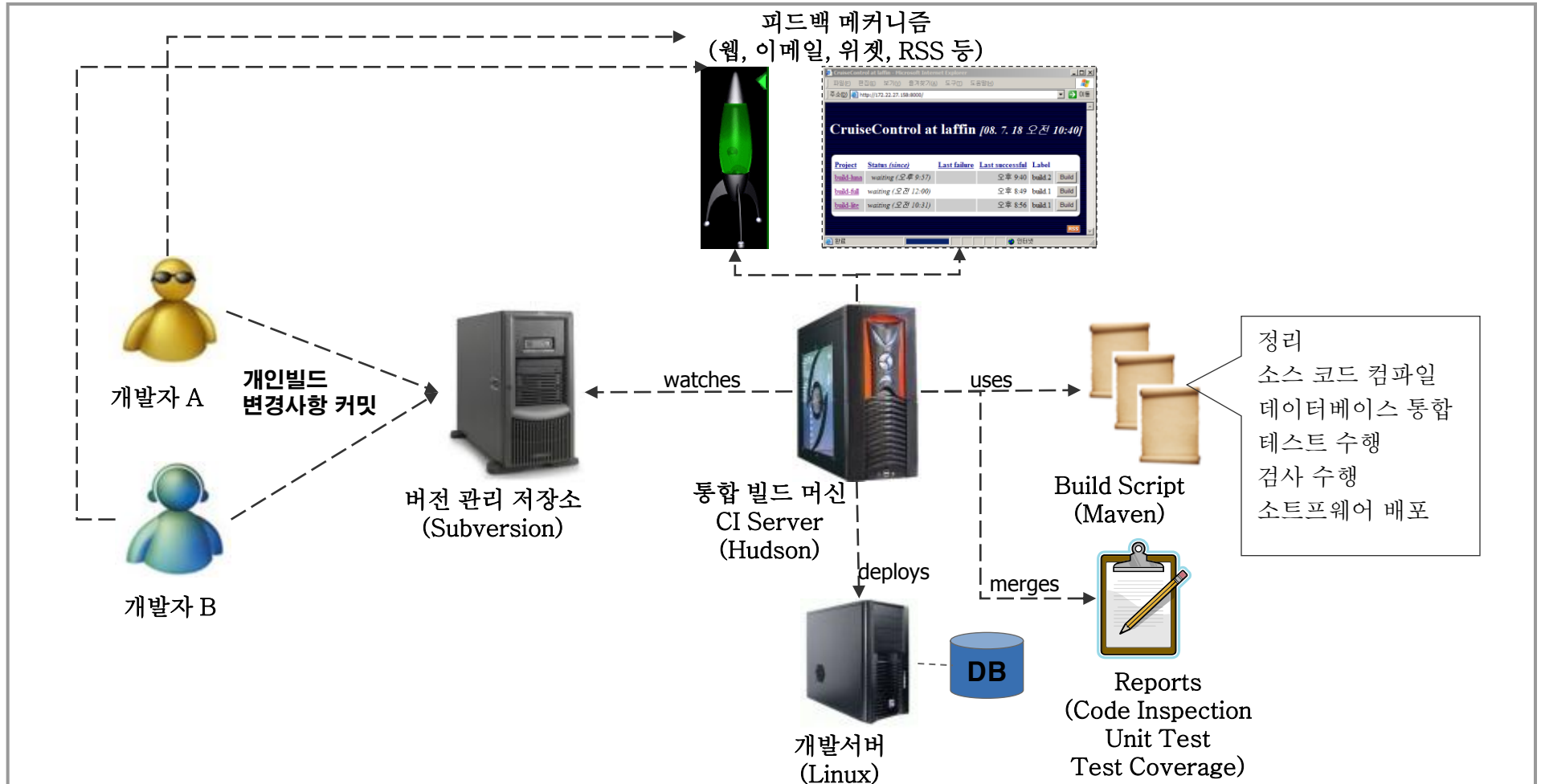
□ 효과

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

□ 특징

- 소스 코드 일관성 유지 : 소스 관리 시스템이 필요하다.
- 자동 빌드 : 빌드는 CI 툴에 의해 자동으로 이루어져야 한다.
- 자동 테스트 : 소스 검증과 함께 기능/비기능 테스트를 매번 검증함으로써 코드의 품질을 높일 수 있다.
- 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 됨



Hudson은 오픈 소스 CI 서버로 소스 빌드 스크립트를 사용한 자동 빌드 및 빌드 결과를 개발자에게 피드백 하는 메커니즘을 제공함

Hudson search ?

Hudson > DEV 자동 재실행 켜기

새 작업 Hudson 관리 개발자 빌드 기록 Edit View Delete View Project Relationship Check File Fingerprint

소개 내용 입력

S	W	작업 ↓	최근 성공	최근 실패	최근 소요 시간
		egovframework-dev-dep-14	2 days 18 hr (#47)	19 days (#30)	5 min 45 sec
		egovframework-dev-isu-14	2 days 18 hr (#43)	1 mo 9 days (#14)	4 min 31 sec
		egovframework-dev-tst	2 days 15 hr (#31)	—	1 min 25 sec
		egovframework-dev-tst-ant	2 days 15 hr (#30)	—	46 sec
		egovtest-maven-pluqin	2 days 15 hr (#32)	—	1 min 4 sec

아이콘: [S](#) [M](#) [L](#)

Legend for all for failures for just latest builds

• 초기화면: 프로젝트 리스트와 빌드 상태 및 빌드 성공 여부를 보여줌

□ Hudson 특징

- 쉬운 Installation : hudson.war형태로 servlet container에 배포
- 웹 기반 UI를 통한 쉬운 설정 및 즉각적인 피드백
- 플러그인 기반의 확장성 제공
- 깔끔하고 잘 정의된 문서 및 API

□ Hudson 주요 기능

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

Hudson은 빌드 결과 및 테스트 결과 등을 보여 주는 대시보드 기능을 제공함

Build History (trend)

Build #	Timestamp
#44	2009. 4. 28 오후 9:01:14
#43	2009. 4. 27 오후 9:01:44
#42	2009. 4. 26 오후 9:01:04
#41	2009. 4. 25 오후 9:01:05
#40	2009. 4. 24 오후 9:01:04
#39	2009. 4. 23 오후 9:01:04
#38	2009. 4. 22 오후 9:01:49
#37	2009. 4. 21 오후 9:01:36
#36	2009. 4. 20 오후 9:01:34
#35	2009. 4. 19 오후 9:01:30
#34	2009. 4. 18 오후 9:01:30

Test Result Trend

PMD Trend

Hudson은 등록된 프로젝트의 소스 코드 변경 내용 추적을 위한 메뉴와 수동 빌드 메뉴 그리고 프로젝트 빌드 상태를 보여주는 메뉴 등을 제공함

Hudson > DEV > egovframework-dev-dep-14

Project egovfram

배포 도구

- Workspace
- Recent Changes
- Latest Test Result (no failures)

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)

#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
#41	2009. 4. 25 오후 9:01:05

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 강제적으로 빌드
- Build History**
 : 빌드 수행 시기와 빌드 상태 및 성공 여부를 보여줌

Hudson은 JUnit 테스트 리포트를 출력해 주는 기능을 제공하며 테스트의 Progress 누적 그래프를 프로젝트 초기 화면에 출력함

JUnit 테스트 리포트

Test Result : egovframework.dev.dep

0 failures (±0)

6 tests (±0)

All Tests

Class	Duration	Fail	(diff)	Total	(diff)
DeployModuleServiceTest	0.12 sec	0		2	
PropertyServiceTest	32 ms	0		2	
ScheduleServiceTest	77 ms	0		2	

프로젝트의 단위 테스트 결과 및 테스트 소요 시간 등에 관한 리포트를 보여줌

테스트 Progress 누적 그래프

Test Result Trend

[\(just show failures\)](#) [enlarge](#)

프로젝트 빌드에 따른 JUnit 테스트 누적 그래프로 빌드 번호화 테스트 method 개수로 빌드 결과를 보여줌

Hudson은 코드 인스펙션 도구인 PMD 플러그인을 제공하며 연동을 통한 코드 인스펙션 결과 리포트를 제공함

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.

Warnings Trend

* PMD 인스펙션 결과 리포트

All Warnings	New Warnings	Fixed Warnings
1	0	0

Summary

Total	High Priority	Normal Priority	Low Priority
1	1	0	0

Details

Details

File: [AltibaseDialect.java](#), Line: 6, Type: ImportFromSamePackage, Priority: High, Category: PMD_for_Eclipse_3.2.5

RULE#21:Do not need to use import in the same package.

10. 참고 자료

□ CI 정의

- http://en.wikipedia.org/wiki/Continuous_Integration

□ Continuous Integration (Martin Fowler)

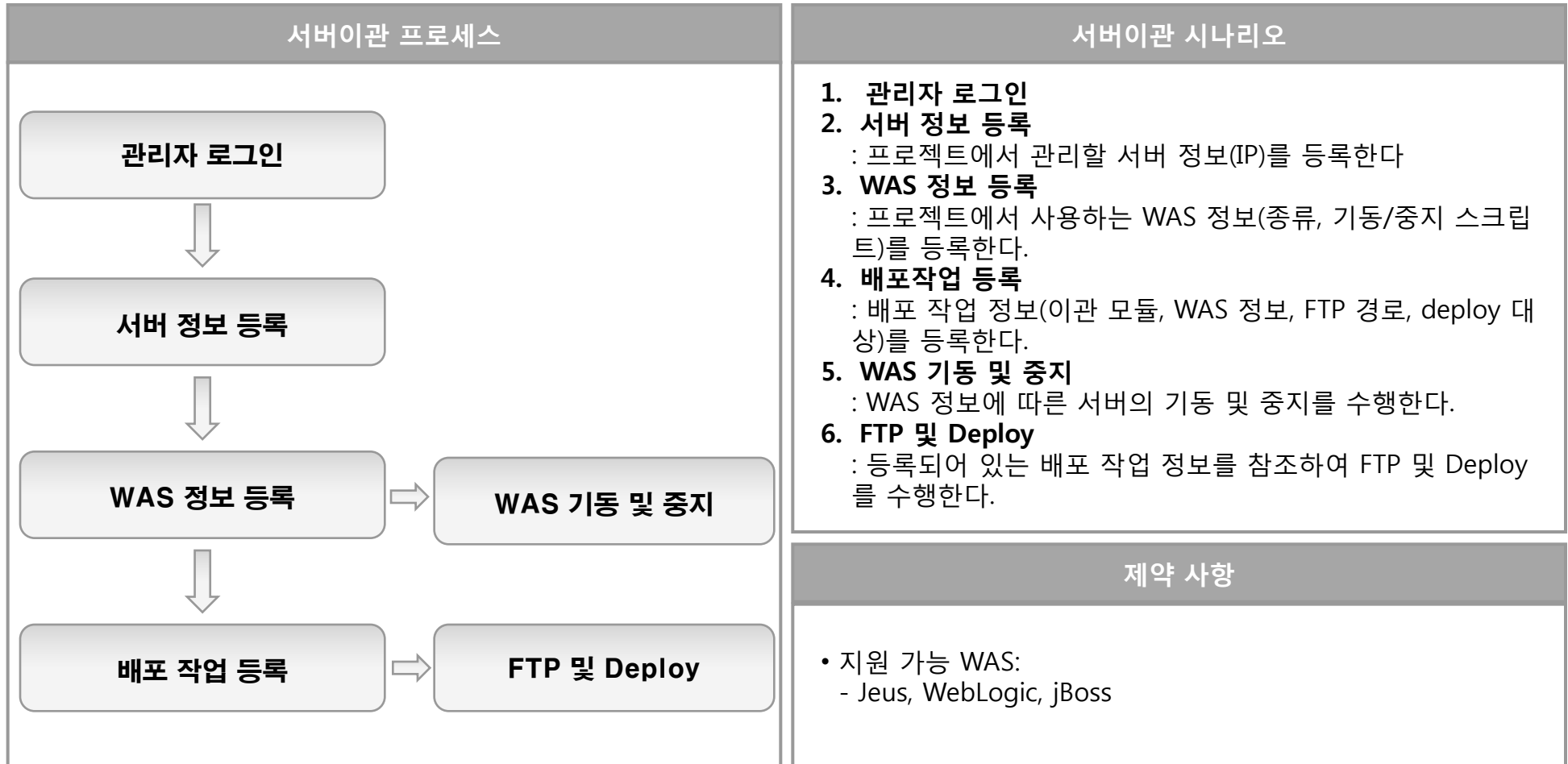
- <http://martinfowler.com/articles/continuousIntegration.html#AutomateDeployment>

□ Hudson Home

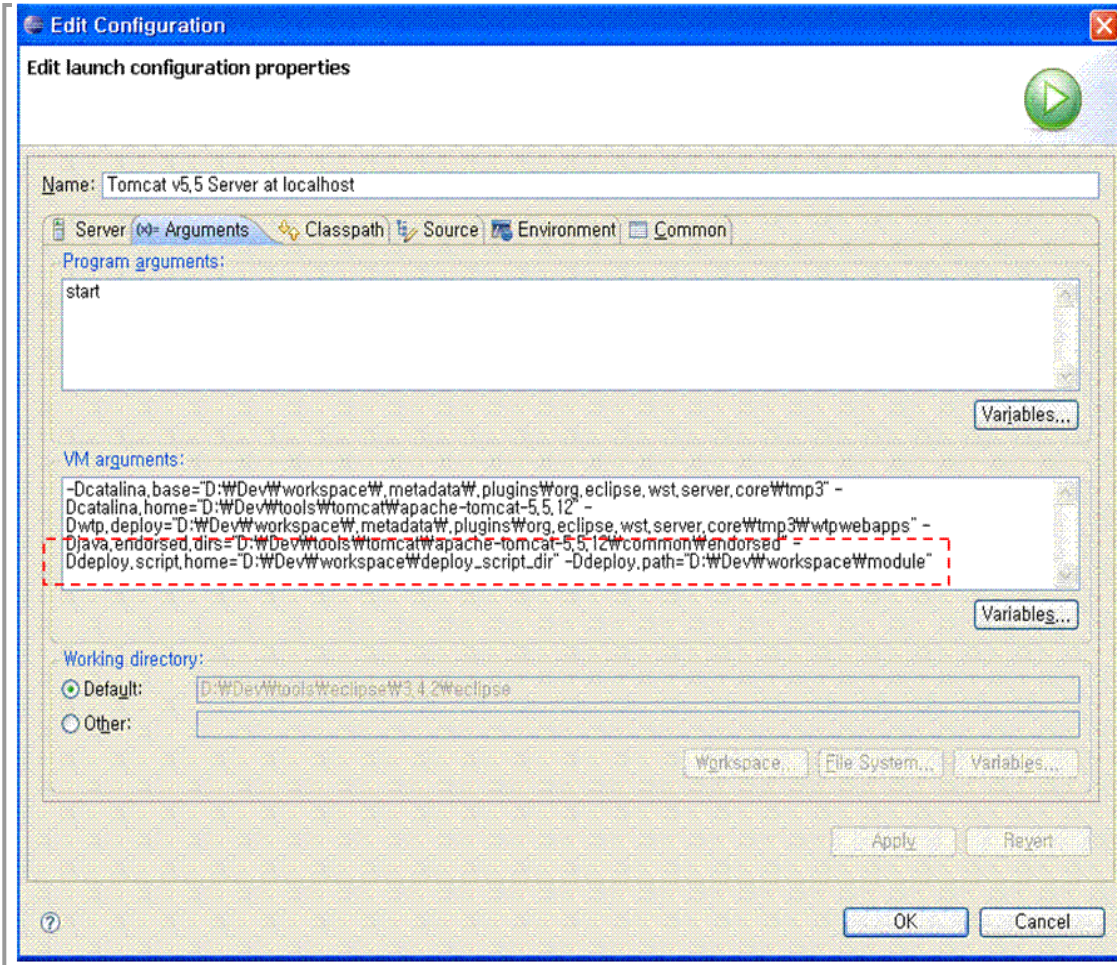
- <https://hudson.dev.java.net/>

1. 개요
2. 환경 설정
3. 주요 기능

개발 완료된 war 파일에 대해 WAS 서버로의 FTP 이관, WAS Deploy 및 해당 WAS 서버의 기동/중지 기능을 제공함



WAS 구동 시 환경 변수로 `deploy.script.home`과 `deploy.path`를 설정하고 제공된 `common.properties` 파일의 내용을 WAS 환경에 맞게 수정



WAS 구동 시 환경 변수

- `Ddeploy.script.home`="배포스크립트 및 로그파일이 저장될 디렉토리"
- `Ddeploy.path`="war 파일이 존재하는 루트 디렉토리"

Hudson 주요 기능

`was.deploy.script.path`="배포스크립트 및 로그파일이 저장될 디렉토리"

- 배포대상 WAS 서버가 weblogic인 경우
 - `weblogic.deploy.jar.path`="weblogic 설치 하위의 weblogic.jar 파일의 위치"
 - `weblogic.adm.id`="weblogic admin id"
 - `weblogic.adm.pwd`="weblogic admin password"

서버 이관 도구는 Admin 관리자 계정 하나만을 두고 있음

행정안전부 egovframework

로그인

Login
아이디 로그인

아이디 admin

비밀번호

로그인

egovframework

Copyright (c) 2009 egovframework. All right reserved

모듈 배포를 위한 대상 서버의 정보를 등록



행정안전부 egovframework

HOME 비밀번호변경 로그아웃

배포관리

작업관리

서버정보등록

배포관리홈 > 작업관리 > 서버정보목록 > 서버정보등록

* 서버이름	개발서버
* 서버IP	127.0.0.1
* 사용자 아이디	test
* 비밀번호	••••

저장 목록 취소

egovframework

Copyright (c) 2009 egovframework. All right reserved.

서버정보 등록 순서

1. 서버이름, 서버 IP, 사용자 아이디, 비밀번호를 입력
2. '저장' 버튼을 클릭
3. 서버정보가 등록된 후 서버 목록 화면으로 이동
4. '목록' 버튼을 클릭하여 서버 목록 화면으로 이동
5. '취소' 버튼을 클릭하여 작성한 내용을 초기화

모듈 배포를 위한 대상 WAS의 정보를 등록

WAS정보등록 배포관리홈 > 작업관리 > WAS목록 > WAS등록

* WAS명 :	개발 Jeus WAS
* 서버명 :	운영WAS
* WAS서버 구분 :	jeus
실행경로 :	/user/dev/jeus/bin
실행유저ID :	jeus
실행유저PWD :	●●●●●●
기동 스크립트 :	startup.sh
중지 스크립트 :	stop.sh
WAS 홈 경로 :	/user/dev/jeus

WAS 정보 등록 순서

1. WAS명, 실행 경로, 실행유저ID, 실행유저 PWD, 기동 스크립트, 중지 스크립트, WAS 홈 경로를 입력
2. 서버 명을 선택한 후 등록했던 서버명중 해당 WAS가 설치된 서버를 선택
3. WAS 서버를 선택 – weblogic, jeus, jboss
4. '저장' 버튼을 클릭
5. WAS 정보가 등록된 후 WAS 목록 화면으로 이동
6. '목록' 버튼을 클릭하여 WAS 목록 화면으로 이동
7. '취소' 버튼을 클릭하여 작성한 내용을 초기화

모듈 배포를 위한 FTP 경로 등 배포작업 정보를 등록

배포작업등록 배포관리홈 > 작업관리 > 배포작업목록 > 배포작업등록

* 작업명 :	<input type="text" value="웹로직 배포작업1"/>
* 대상WAS명 :	<input type="text" value="Product WAS Weblogic"/>
* 대상모듈 :	<input type="text" value="hudson.war"/> <input type="button" value="모듈선택"/>
* FTP 경로 :	<input type="text" value="/user/dev/weblogic/deploy"/>
디플로이 대상 : (jeus:컨테이너, weblogic:server)	<input type="text" value="devServer"/>
Admin URL : (only weblogic, ex) t3://ip:port)	<input type="text" value="t3://127.0.0.1:7011"/>

배포작업 등록 순서

1. 작업명, FTP 경로를 입력
2. 대상 WAS명을 선택
3. '모듈선택' 버튼을 클릭하여 배포 대상 모듈을 선택
4. Deploy 대상을 입력 - jeus인 경우 컨테이너명, weblogic인 경우 server 명을 입력
5. Admin URL을 입력 - weblogic인 경우 입력
6. '저장' 버튼을 클릭
7. 배포작업 정보가 등록된 후 배포작업 목록 화면으로 이동
8. '목록' 버튼을 클릭하여 배포작업 목록 화면으로 이동
9. '취소' 버튼을 클릭하여 작성한 내용을 초기화

모듈 배포를 위한 FTP로 이관하는 기능 제공

배포작업목록 배포관리홈 > 작업관리 > 배포작업목록

No.	<input type="checkbox"/>	작업명	WAS명	대상서버IP	대상모듈	작업		
1	<input type="checkbox"/>	JBOSS-dev-dep	Product WAS jboss	192.168.200.21	hudson.war	FTP	Deploy	자동이관설정
2	<input type="checkbox"/>	productJob1	Product WAS Weblogic	192.168.200.21	hudson.war	FTP	Deploy	자동이관설정
3	<input type="checkbox"/>	productJob2	Product WAS jeus	192.168.200.21	hudson.war	FTP	Deploy	자동이관설정
4	<input type="checkbox"/>	productJob3	Product WAS jboss	192.168.200.21	hudson.war	FTP	Deploy	자동이관설정
5	<input type="checkbox"/>	Weblogic-dev-dep	Product WAS Weblogic	192.168.200.21	egovframework-dev-dep.war	FTP	Deploy	자동이관설정

작업로그 조회 배포관리홈 > 작업관리 > 배포작업목록 >

작업명 : productJob2
작업구분 : ftp

```

[echo] Build logged to
productJob2_ftp_20090430161700.log
ftp-init: duration 0 seconds

was-ftp:
[ftp] Opening FTP connection to 192.168.200.21
[ftp] connected
[ftp] logging in to FTP server
[ftp] login succeeded
[ftp] changing the remote directory
[ftp] sending files
    
```

FTP 순서

1. 배포 작업 목록을 조회
2. 대상 작업의 'FTP' 버튼을 클릭하여 대상 모듈을 해당 WAS로 전송
3. FTP가 진행되는 도중 '작업중입니다' 메시지 출력
4. 전송이 완료되면 FTP 작업로그를 화면에 출력

모듈 배포를 위한 Deploy 기능 제공

배포작업목록 배포관리홈 > 작업관리 > 배포작업목록

No.	작업명	WAS명	대상서버IP	대상모듈	작업
1	JBOSS-dev-dep	Product WAS jboss	192.168.200.21	hudson.war	FTP Deploy 자동이관설정
2	productJob1	Product WAS Weblogic	192.168.200.21	hudson.war	FTP Deploy 자동이관설정
3	productJob2	Product WAS jeus	192.168.200.21	hudson.war	FTP Deploy 자동이관설정
4	productJob3	Product WAS jboss	192.168.200.21	hudson.war	FTP Deploy 자동이관설정
5	Weblogic-dev-dep	Product WAS Weblogic	192.168.200.21	egovframework-dev-dep.war	FTP Deploy 자동이관설정

작업로그 조회 배포관리홈 > 작업관리 > 배포작업목록 >

작업명 : productJob2
작업구분 : deploy

```

deploy-was:
BUILD SUCCESSFUL
Total time: 21 seconds
$
[sleep] sleeping for 20000 milliseconds
[echo] was deploy complete
was-deploy: duration 44 seconds
                    
```

Deploy 작업 순서

1. 배포 작업 목록을 조회
2. 대상 작업의 'Deploy' 버튼을 클릭하여 WAS에 Deploy
3. Deploy가 진행되는 도중 '작업중입니다' 메시지 출력
4. Deploy가 완료 되면 Deploy 작업로그를 화면에 출력

모듈 배포를 위한 WAS 기동 기능을 제공

The screenshot shows the 'WAS 목록' (WAS List) page in the egovframework. It features a search bar for WAS name and server name, and a table with columns for No., WAS name, server name, server IP, and WAS operation. A mouse cursor is pointing at the '기동' (Start) button for the first entry. Below the table are '등록' (Register) and '삭제' (Delete) buttons. At the bottom, there is a terminal window showing the execution of a script to start the WAS container.

No.	WAS명	서버명	서버IP	WAS작업
1	PrdJct WAS Iboss	운영 WAS	192.168.200.21	기동 중지
2	PrdJct WAS jeus	운영 WAS	192.168.200.21	기동 중지
3	PrdJct WAS Weblogic	운영 WAS	192.168.200.21	기동 중지

```

WAS명 : Product WAS jeus
작업구분 : start

was_dev container is down successfully
$
[tnet] ./dev_start.sh
[tnet] JEUS prepend classpath : file:/product/jeus/lib/system/jext.jar
was_dev container is started successfully
$
[sleep] sleeping for 20000 milliseconds
[echo] was start complete
was-start: duration 1 minute 17 seconds
    
```

WAS 기동 순서

1. WAS 목록을 조회
2. 대상 WAS의 '기동'버튼을 클릭하여 WAS를 기동
3. WAS 기동 작업이 진행되는 도중 '작업중입니다' 메시지 출력
4. WAS 기동 작업이 완료 되면 WAS 기동 작업 로그를 화면에 출력

모듈 배포를 위한 WAS 중지 기능을 제공

The screenshot shows the 'WAS 목록' (WAS List) page in the egovframework. It features a search bar for WAS name and server name, and a table with columns for No., checkbox, WAS name, server name, server IP, and WAS operation (start/stop buttons). Below the table, there are '등록' (Register) and '삭제' (Delete) buttons. A terminal window at the bottom shows the execution of the stop command for 'Product WAS jeus'.

No.	<input type="checkbox"/>	WAS명	서버명	서버IP	WAS작업
1	<input type="checkbox"/>	Product WAS Iboss	운영WAS	192.168.200.21	기동 [중지]
2	<input type="checkbox"/>	Product WAS jeus	운영WAS	192.168.200.21	기동 [중지]
3	<input type="checkbox"/>	Product WAS Weblogic	운영WAS	192.168.200.21	기동 [중지]

```

WAS명 : Product WAS jeus
작업구분 : stop

[telnet] cd /user/dev/jeus
[telnet] $
[telnet] ./dev_stop.sh
[telnet] JEUS prepend classpath : file:/product/jeus/lib/system/jext.jar
was_dev container is down successfully
$
  [sleep] sleeping for 20000 milliseconds
  [echo] was stop complete
was-stop: duration 24 seconds
    
```

WAS 중지 순서

1. WAS 목록을 조회
2. 대상 WAS의 '중지'버튼을 클릭하여 WAS를 중지
3. WAS 중지 작업이 진행되는 도중 '작업 중입니다' 메시지 출력
4. WAS 중지 작업이 완료 되면 WAS 중지 작업 로그를 화면에 출력

서버이관 작업과 관련된 로그 조회 기능을 제공

배포관리
작업관리

작업로그목록 배포관리 > 식별관리 > 작업로그목록

로그구분: 전체 검색기간: 2009년 04월 30일 ~ 2009년 04월 30일 🔍 검색

No.	작업명	작업로그
1	productJob2	productJob2_deploy_200904301E2855.log
2	productJob2	productJob2_deploy_200904301E2533.log
3	productJob2	productJob2_deploy_200904301E1700.log
4	productJob2	productJob2_deploy_200904301E1428.log
5	JBOSS dev dep	JBOSS dev dep_ftp_200904301C0752.log

작업로그 조회 배포관리 > 작업관리 > 배포작업목록 > 배포작업

작업명 : JBOSS-dev-dep
로그구분 : ftp
로그성형일시 :

```

[echo] Build logged to JBOSS-dev-dep_ftp_20090430160752.log
ftp-init: duration 0 seconds

was-ftp:
[ftp] Opening FTP connection to 192.168.200.21
[ftp] connected
[ftp] logging in to FTP server
[ftp] login succeeded
[ftp] changing the remote directory
[ftp] disconnecting
was-ftp: duration 1 seconds
    
```

작업로그 순서

1. 검색조건(로그 구분 및 검색기간)을 입력하고 '검색' 버튼을 클릭
2. 검색조건에 해당하는 작업로그 정보를 출력
3. 상세조회 할 '작업로그'를 클릭하면 해당 작업의 작업로그 조회 화면으로 이동하여 상세 정보 출력
4. '목록' 버튼을 클릭하면 작업로그목록 화면으로 이동

1. 개요
2. Feature
3. Process
4. 주요기능
5. 참고자료

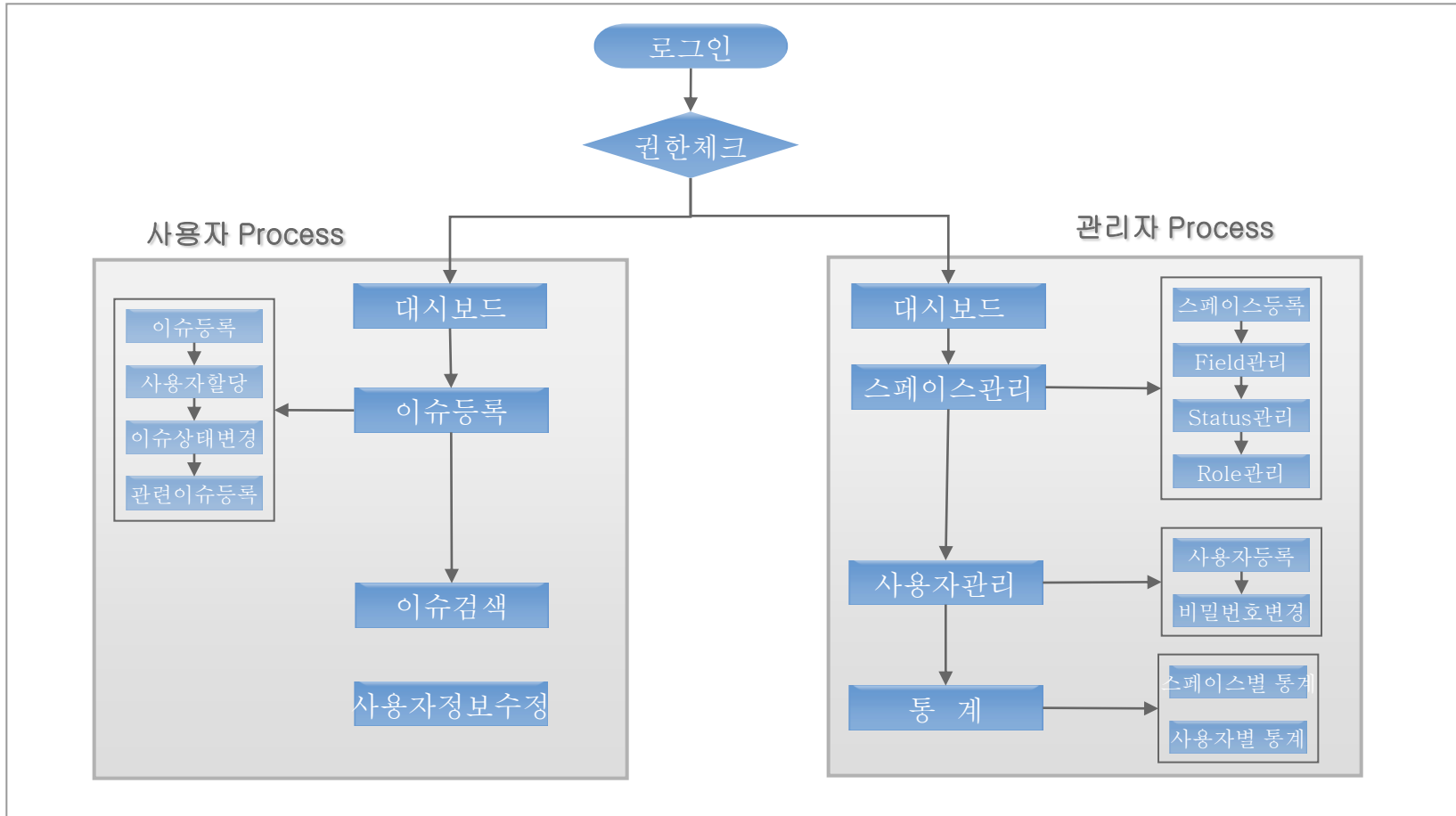
프로젝트(또는 복수의 프로젝트)에서 개발자간의 커뮤니케이션을 돕고 체계적인 이슈(버그)를 관리하는 시스템을 의미함

스페이스	이슈관리		상태	보고한 이슈	할당된 이슈	전체이슈
	이슈생성	이슈검색				
배포관리	이슈등록	이슈검색	OPEN	12	4	1944
			New	1	0	1
			Newww	1	0	1
			CLOSED	0	0	0
			Sub Total	14	4	1946
Browser호환성	이슈등록	이슈검색	OPEN	12	9	12
			Delay	1	0	1
			Wait	0	0	0
			CLOSED	0	0	0
			Sub Total	13	9	13
이슈관리	이슈등록	이슈검색	OPEN	3	3	2040
			CLOSED	0	0	0
			Sub Total	3	3	2040
Total				30	16	3999

이슈를 체계적으로 관리하기 위해 스페이스관리, 이슈관리, 통계, 대시보드등 다양한 기능을 제공함

Feature	상세 내용
Easy to Install	□ 별도의 Customizing이 필요 없이 제공된 war의 디플로이를 통한 간단한 설치메커니즘 제공
Eclipse Integraton	□ 이클립스의 형상연계 Plug-in과의 연계를 통해 커밋 시 이슈와 연계할 수 있고, 이슈를 조회할 수 있다.
E-Mail Integration	□ 사용자등록, 이슈등록 및 상태수정시에 할당된 담당자에게 해당 내용을 E-Mail로 전송할 수 있다.
Cross Referencing of Items	□ 등록된 Item간의 상호 참조를 할 수 있으며, 이를 통해 연관된 이슈와의 통합이 가능하다.
Custom Fields / Status	□ 스페이스의 효율적인 관리를 위해 스페이스의 Field와 Item의 Status를 정의할 수 있다.
Detailed History View	□ 등록된 Item의 등록, 담당자, 상태, 과거 변경내역을 조회할 수 있다.
Statistics	□ 등록된 스페이스와 사용자별 통계를 통해 전체 이슈와 사용자의 진행상황을 파악할 수 있다.
Search	□ 다양한 검색조건(삭제된 이슈검색 포함)을 제공하여 빠른 시간에 원하는 Item을 검색할 수 있다.
Dashboard	□ 해당 사용자별로 자신에게 등록된 스페이스, 등록된 이슈, 할당된 이슈 등을 파악할 수 있다.
Administration Management	□ 사용자관리, 스페이스 관리 , 설정관리, 통계, 이슈관리등 강력한 관리기능을 제공한다.

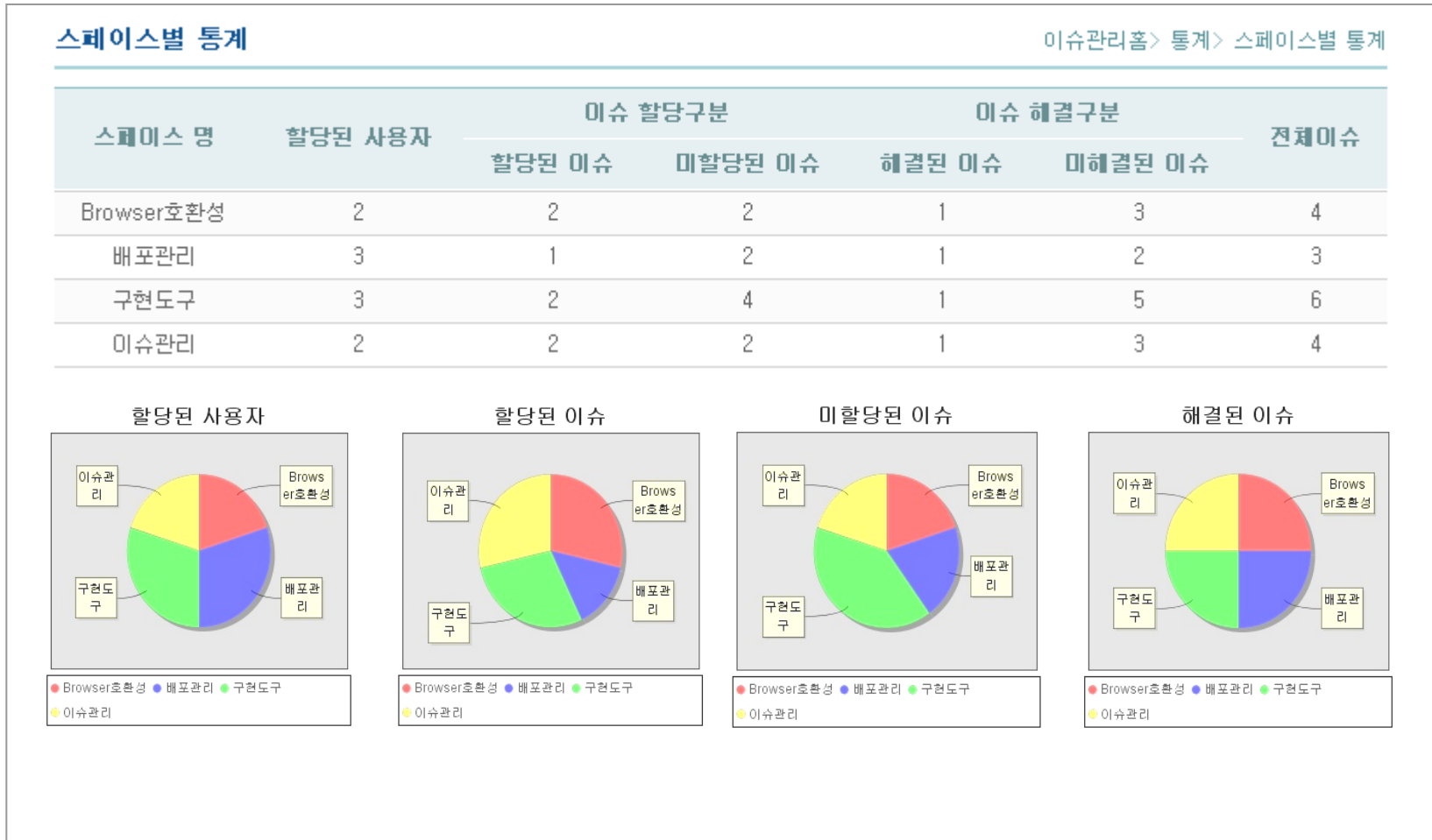
이슈 관리는 관리자기능인 스페이스관리(필드, 상태)와 사용자관리등을 통해 이슈 관리환경을 구성하고, 일반사용자는 설정된 환경에서 이슈를 관리하는 프로세스로 구성



해당 사용자별로 자신에게 등록된 스페이스, 이슈 상태별 등록된 이슈 및 할당된 이슈들을 체계적으로 파악할 수 있다.

대시보드		이슈관리홈 > 대시보드				
스페이스	이슈관리		상태	보고한 이슈	할당된 이슈	전체이슈
	이슈생성	이슈검색				
배포관리	[이슈등록]	[이슈검색]	OPEN	12	4	1944
			New	1	0	1
			Neww	1	0	1
			CLOSED	0	0	0
			Sub Total	14	4	1946
Browser호환성	[이슈등록]	[이슈검색]	OPEN	12	9	12
			Delay	1	0	1
			Wait	0	0	0
			CLOSED	0	0	0
			Sub Total	13	9	13
이슈관리	[이슈등록]	[이슈검색]	OPEN	3	3	2040
			CLOSED	0	0	0
			Sub Total	3	3	2040
Total			30	16	3999	

스페이스 또는 사용자별로 카테고리화 하고 이를 시각화하여, 이슈관리시스템을 다양한 시각에서 분석하고 제어할 수 있는 통계자료를 제공한다.



사용자를 등록하고 정보를 수정하는 기본적인 사용자관리기능과, 사용자에게 스페이스와 롤을 등록하고 삭제하는 롤관리기능을 제공한다.

사용자목록 이슈관리홈 > 사용자관리 > 사용자 목록

글수 6 이름

<input type="checkbox"/>	No.	사용자 이름	사용자 아이디	E-Mail	핸드폰	롤관리	수정
<input type="checkbox"/>	1	admin	admin	brick@lgcns.com	0112223333	<input type="button" value="롤관리"/>	<input type="button" value="수정"/>
<input type="checkbox"/>	2	민병석	bsmin	bsmin@lgcns.com	0101111111	<input type="button" value="롤관리"/>	<input type="button" value="수정"/>
<input type="checkbox"/>	3	송영인	brick	bricksong@naver.com	0101111111	<input type="button" value="롤관리"/>	<input type="button" value="수정"/>

사용자롤등록 이슈관리홈 > 사용자롤관리 > 사용자롤등록

사용자 : [민병석]

스페이스	사용자 롤	삭제
Browser호환성	DEFAULT	<input type="button" value="삭제"/>

* 할당하려는 스페이스와 롤을 선택하세요.

스페이스	<input type="text" value="스페이스를 선택하세요"/>
사용자롤	<input type="text" value="롤을 선택하세요"/>

스페이스를 등록하고 정보를 수정하는 스페이스 관리기능과, 스페이스에 롤을 부여한 사용자를 등록하고 삭제하는 사용자관리기능을 제공한다.

스페이스목록 이슈관리홈 > 스페이스관리 > 스페이스 목록

<input type="checkbox"/>	No.	스페이스 명	스페이스 키	Description	사용자할당관리	수정
<input type="checkbox"/>	1	Browser호환성	Cross	설 명	사용자할당	수정
<input type="checkbox"/>	2	배포관리	Deploy	배포 이슈관리	사용자할당	수정
<input type="checkbox"/>	3	구현도구	Implement	구현이슈관리	사용자할당	수정
<input type="checkbox"/>	4	이슈관리	Issue	이슈관리 스페이스임	사용자할당	수정

[스페이스등록](#) [스페이스삭제](#)

스페이스사용자등록 이슈관리홈 > 사용자관리 > 스페이스사용자등록

스페이스 : [이슈관리]

사용자 이름	사용자 ID	롤	삭제
admin	admin	DEFAULT	삭제
미승룡	leesyong	DEFAULT	삭제

* 등록하려는 사용자와 사용자의 롤을 선택하세요.

사용자	사용자를 선택하세요 ▼
사용자롤	롤을 선택하세요 ▼

[사용자등록](#) [목록](#)

이슈 등록, 메일/SMS발송, 수정, 삭제등의 이슈관리기능과 등록된 이슈의 처리결과에 따른 상태를 변경하고 이를 이력(History)으로 관리하는 기능을 제공한다.

이슈상세조회
이슈관리홈 > 이슈관리 > 이슈상세조회

*** 이슈정보**

이슈 ID	Implement-2	관련 이슈	<input type="button" value="추가"/> -----
상태	CLOSED	등록일시	2009-04-24 14:57:33.0
등록자	admin	할당된 담당자	admin
이슈제목	구현도구 Issue2		
상세내용	구현도구 Issue2		

*** 이력**

No.	등록자	상태	할당된 담당자	내용	첨부파일	등록일시
1	admin	OPEN	admin			2009-04-24 14:57:33.0
2	admin	CLOSED	admin	완료		2009-04-24 15:01:58.0

*** 상태변경**

* 상태	<input type="button" value="선택하세요"/>	* 담당자 할당	<input type="button" value="선택하세요"/>
* 내용	<input style="width: 100%;" type="text"/>		
메일 수신자	<input type="checkbox"/> admin <input type="checkbox"/> 이승룡 <input type="checkbox"/> 전만성		
SMS 수신자	<input type="checkbox"/> admin <input type="checkbox"/> 이승룡 <input type="checkbox"/> 전만성		
첨부파일	<input type="button" value="파일 선택"/> 선택된 파일 없음		

□ jTrac Home

- <http://www.jtrac.info/>