

# 전자정부 표준프레임워크 개발환경

**eGovFrame**



# Contents

1. \_ 개발환경 개요
2. \_ 구현도구
3. \_ 개인 빌드 (Maven)
4. \_ 테스트 도구
5. \_ 형상관리
6. \_ CI 서버



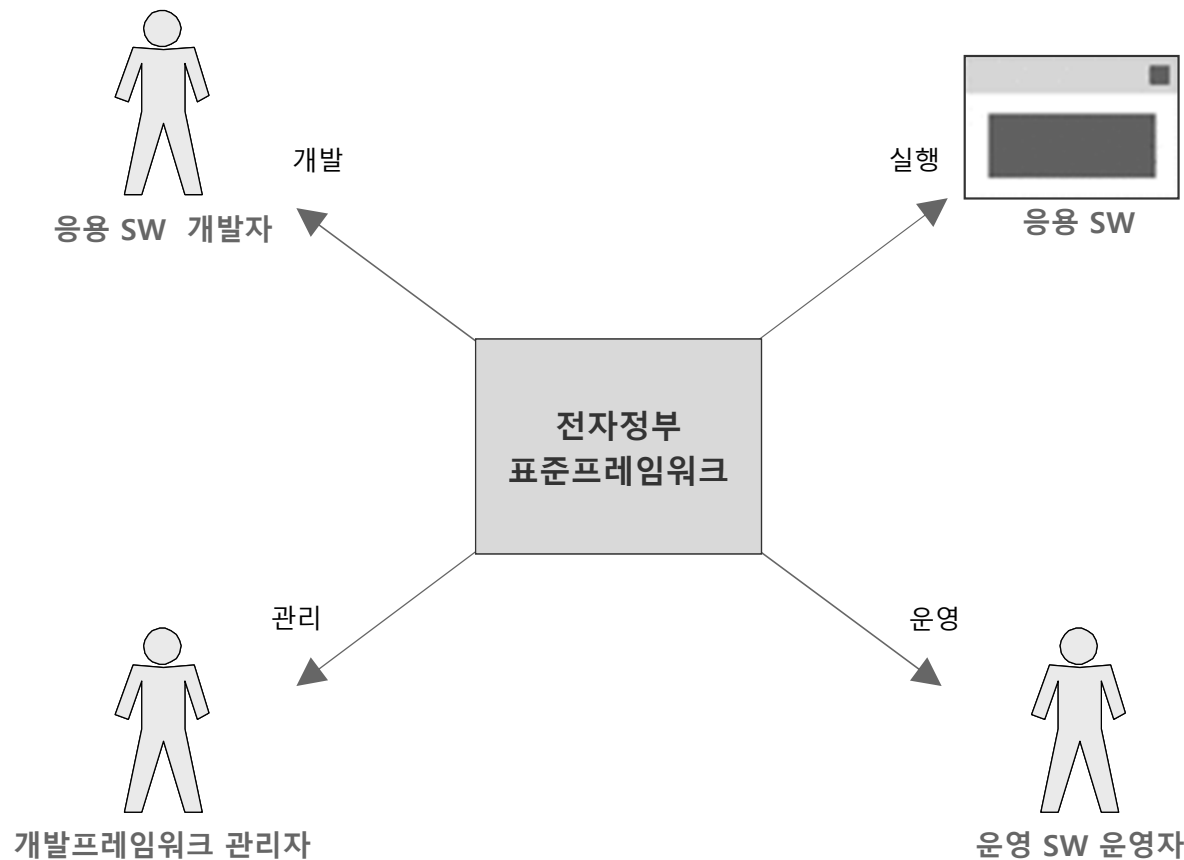
## 1. 개발환경 개요

- 1. 개발환경 개요
- 2. 서비스 도출
- 3. 개발환경 구성요소
- 4. 개발환경 선정 오픈 소스
- 5. 서비스 별 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

## 2. 서비스 도출 (1/2)

## 1. 개발환경 개요

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

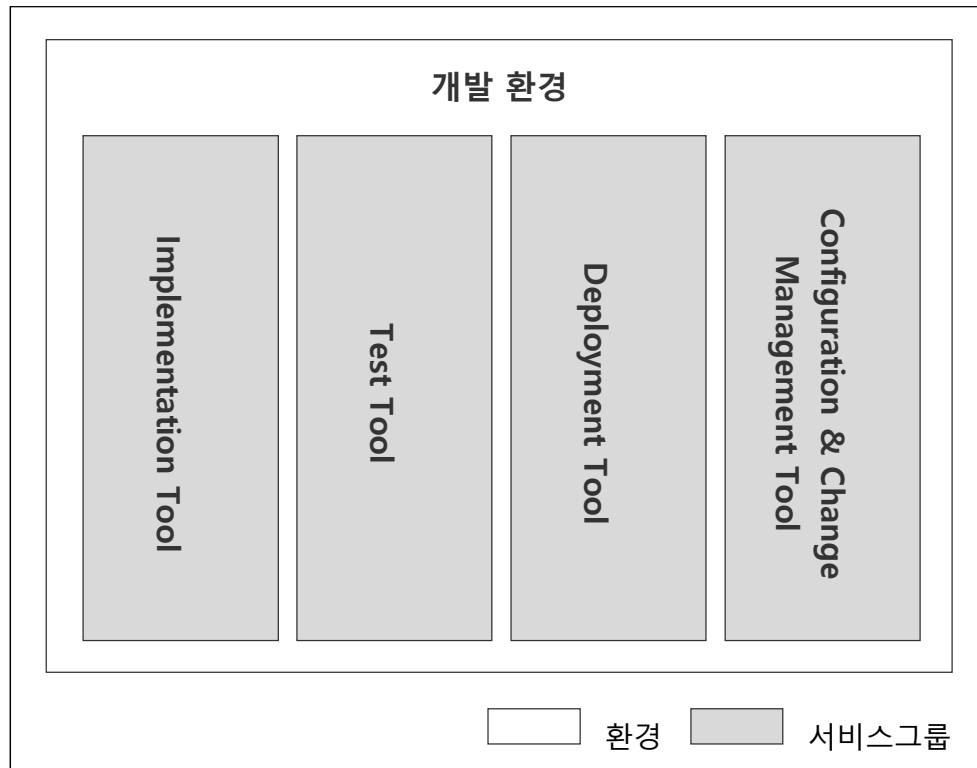
### 개발 환경 서비스 그룹 사례 분석

#### 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 서비스	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, MDSD	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**로 구성됨

개발 환경 서비스 그룹 도출

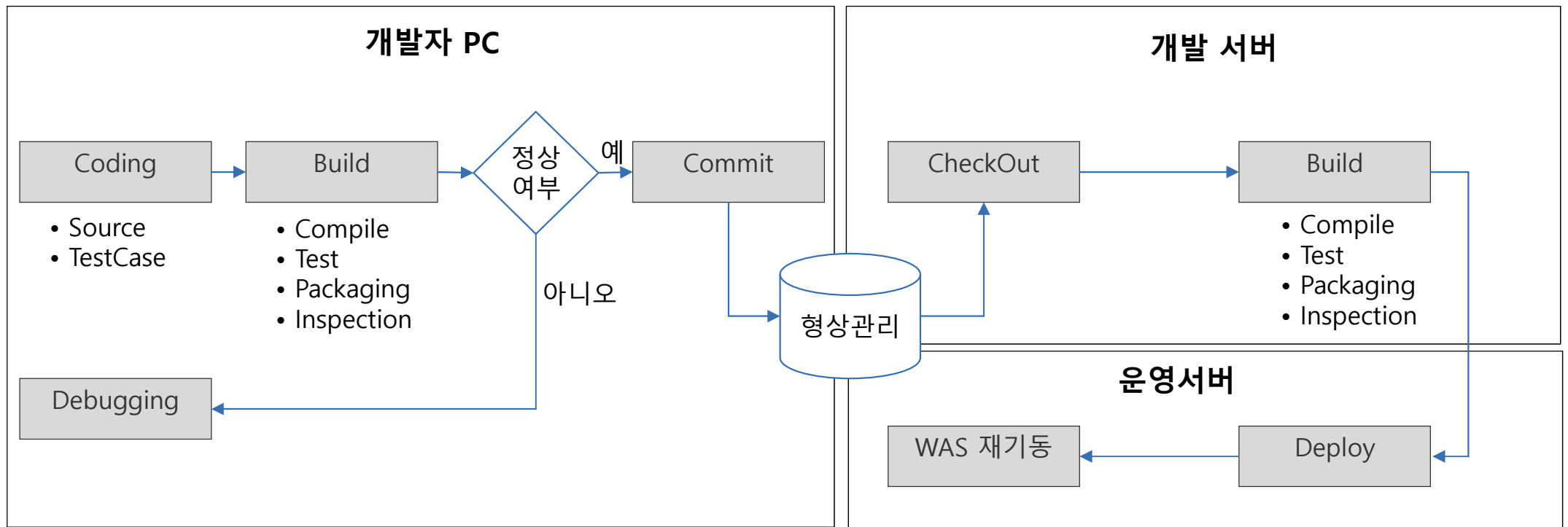
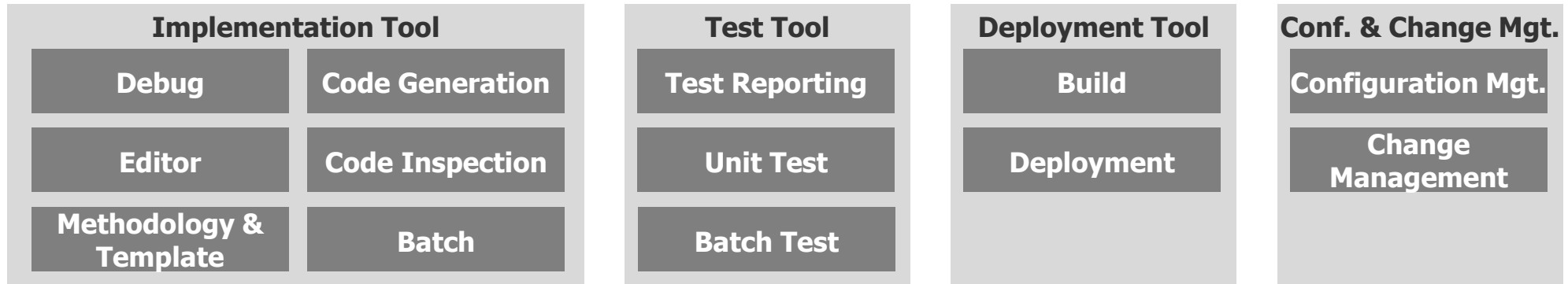


서비스 그룹	설명
Implementation Tool	<ul style="list-style-type: none"> <li>업무 프로그램 구현을 지원하는 도구임</li> </ul>
Test Tool	<ul style="list-style-type: none"> <li>구현된 업무 프로그램의 테스트를 지원하는 도구임</li> </ul>
Deployment Tool	<ul style="list-style-type: none"> <li>구현 완료된 업무 프로그램을 실행 환경에 배포 가능한 형태로 패키징하고, 패키징된 업무 프로그램을 실행 환경에 배포하는 도구임</li> </ul>
Configuration & Change Management Tool	<ul style="list-style-type: none"> <li>형상 및 변경 관리 지원 도구임</li> </ul>

### 3. 개발환경 구성요소

### 1. 개발환경 개요

개발환경은 개발자 PC에서 개발서버, 운영서버로 연결되는 개발 흐름을 지원함



개발환경 오픈소스 SW 1,2차 평가를 통해 선정된 22개의 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			New Batch Template Project	Spring Batch	Apache License 2.0
7			New Batch Job	Spring Batch	Apache License 2.0
8			New Batch Job Launcher	Spring Batch	Apache License 2.0
9		Debug	Local Debug	Eclipse	EPL & EDL
10			Remote Debug	Eclipse	EPL & EDL

개발환경 오픈소스 SW 1,2차 평가를 통해 선정된 22개의 오픈소스 SW는 다음과 같음

### 개발환경 오픈소스 SW 선정결과

No	서비스 그룹	서비스		오픈소스	라이선스
11	테스트도구	Unit Test	TestCase	JUnit	CPL
12			Mock Support	EasyMock	MIT
13			DB Support	DbUnit	LGPL 2.1
14		Test Automation		Ant, Maven	Apache License 2.0
15		Test Reporting		Ant, Maven	Apache License 2.0
16		Test Coverage		EMMA	CPL
17		Batch Job Test		Spring Batch	Apache License 2.0
18	배포도구	Build	Build Tool	Maven	Apache License 2.0
19			CI Server	Hudson/Jenkins	Creative Commons Attribution Share-Alike License,MIT
20			Nexus	Nexus	GPL
21	형상관리도구	Configuration Management		Subversion	Subversion License
22		Change Management		jTrac	Apache License 2.0

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

### 구성요소 별 상세

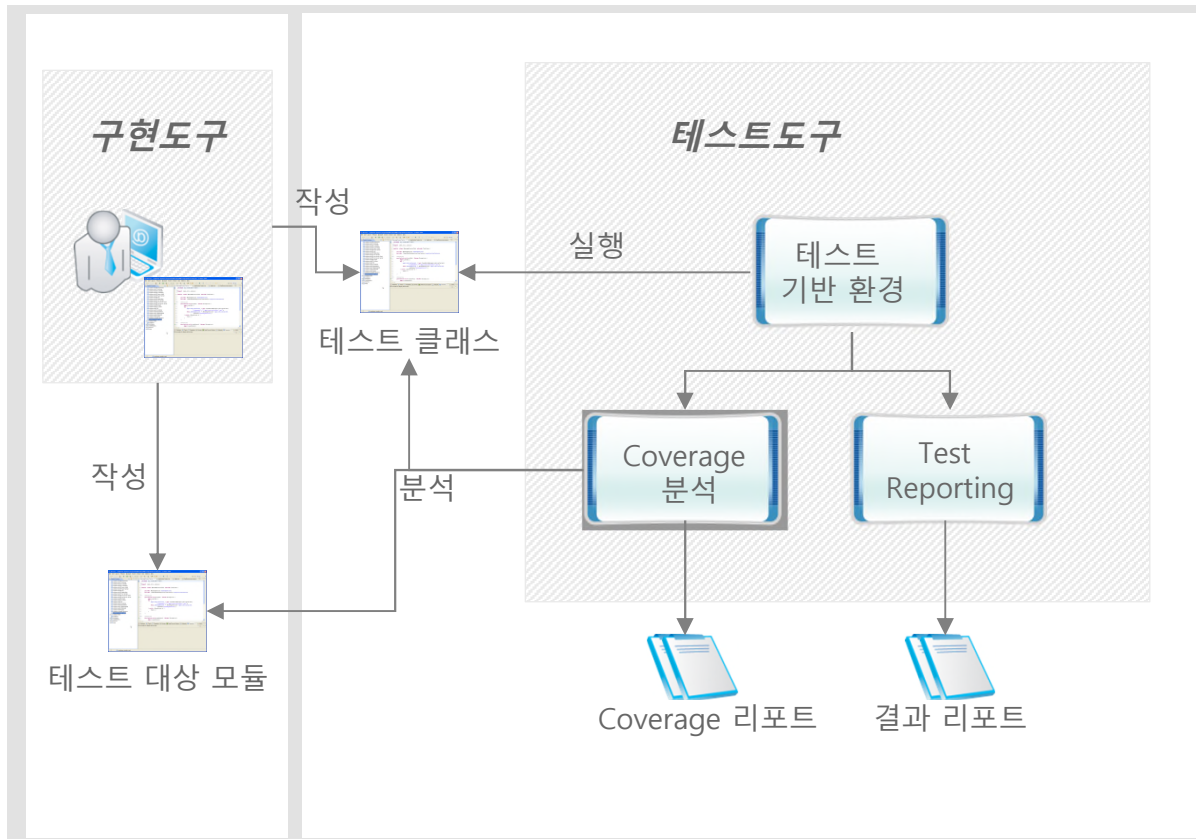
구성요소	설명
에디터	<ul style="list-style-type: none"> <li>• SourceCode Editor로 편리한 소스 코드 작성 환경 제공</li> <li>• UML, ERD 에디터 제공</li> <li>• DBIO 에디터를 통해 SQL작성 및 수행결과 확인 가능</li> <li>• WebFlow Editor를 통해 설정 파일의 유효성 검사 및 Edit</li> </ul>
디버거	<ul style="list-style-type: none"> <li>• 편집모드와 디버깅 모드의 전환을 통한 작업의 편의성 향상</li> <li>• Remote 디버깅 제공</li> </ul>
방법론(개발절차)	<ul style="list-style-type: none"> <li>• 애플리케이션을 개발 가이드 제공</li> </ul>
템플릿 프로젝트	<ul style="list-style-type: none"> <li>• 애플리케이션 개발 시 자동으로 템플릿을 생성 마법사 제공</li> </ul>
공통 컴포넌트	<ul style="list-style-type: none"> <li>• 애플리케이션 개발 시 마법사를 통해 공통컴포넌트를 추가 기능 제공</li> </ul>
맞춤형 개발환경	<ul style="list-style-type: none"> <li>• 프로젝트에서 필요로 하는 기능만을 선택적으로 구성할 수 있는 도구 제공</li> </ul>
배치 템플릿 프로젝트	<ul style="list-style-type: none"> <li>• New eGovFrame Batch Template Project 생성 마법사 실행</li> <li>• 배치 처리 구성 별 예제 제공</li> </ul>
배치 Job 생성	<ul style="list-style-type: none"> <li>• 배치 작업 파일 생성 마법사 실행</li> <li>• 배치 Job / Step / Item 설정으로 XML 파일 생성</li> </ul>
배치 Job Launcher	<ul style="list-style-type: none"> <li>• 배치 작업 실행 파일 생성 마법사 실행</li> <li>• 배치 처리 구성 시 최초에 필요한 Launcher, Repository의 설정 XML 파일 생성</li> </ul>

## 5. 서비스 별 소개 - 테스트 도구

### 1. 개발환경 개요

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

#### 주요 메커니즘

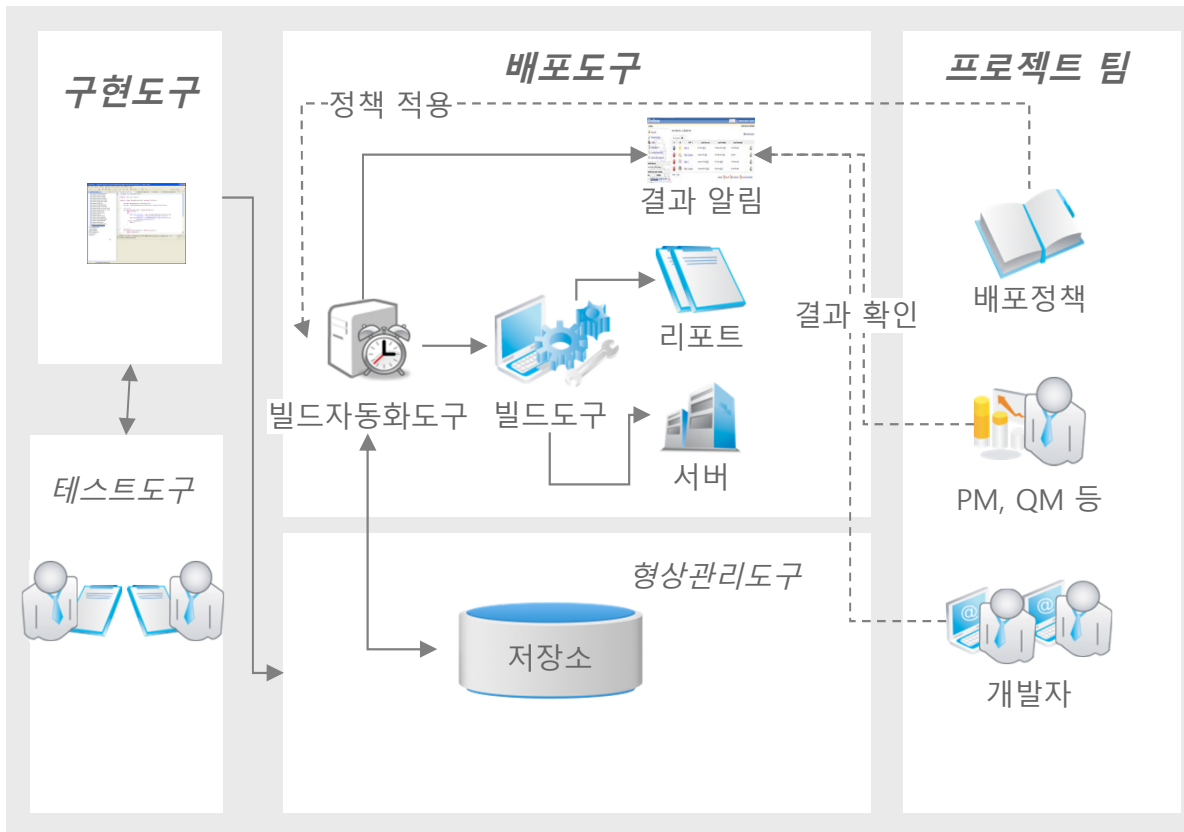


#### 구성요소별 상세

구성요소	설명
<b>테스트 기반환경</b>	<ul style="list-style-type: none"> <li>JUnit 기반의 Unit Test 작성 지원</li> <li>Mock 및 DB Support</li> </ul>
<b>Test Reporting</b>	<ul style="list-style-type: none"> <li>Eclipse 기반의 Green Bar GUI</li> <li>HTML/XML/Excel 형식의 Reporting 지원</li> </ul>
<b>Coverage 분석</b>	<ul style="list-style-type: none"> <li>TestCase가 커버하는 정도를 수치 및 코드 영역을 통해 확인</li> </ul>
<b>Batch Job Test</b>	<ul style="list-style-type: none"> <li>배치 관련 파일을 활용하여 간단히 테스트할 수 있는 테스트 마법사 실행</li> </ul>

Maven 기반의 개발자 빌드와 CI서버를 통한 지속적인 통합을 지향

주요 메커니즘

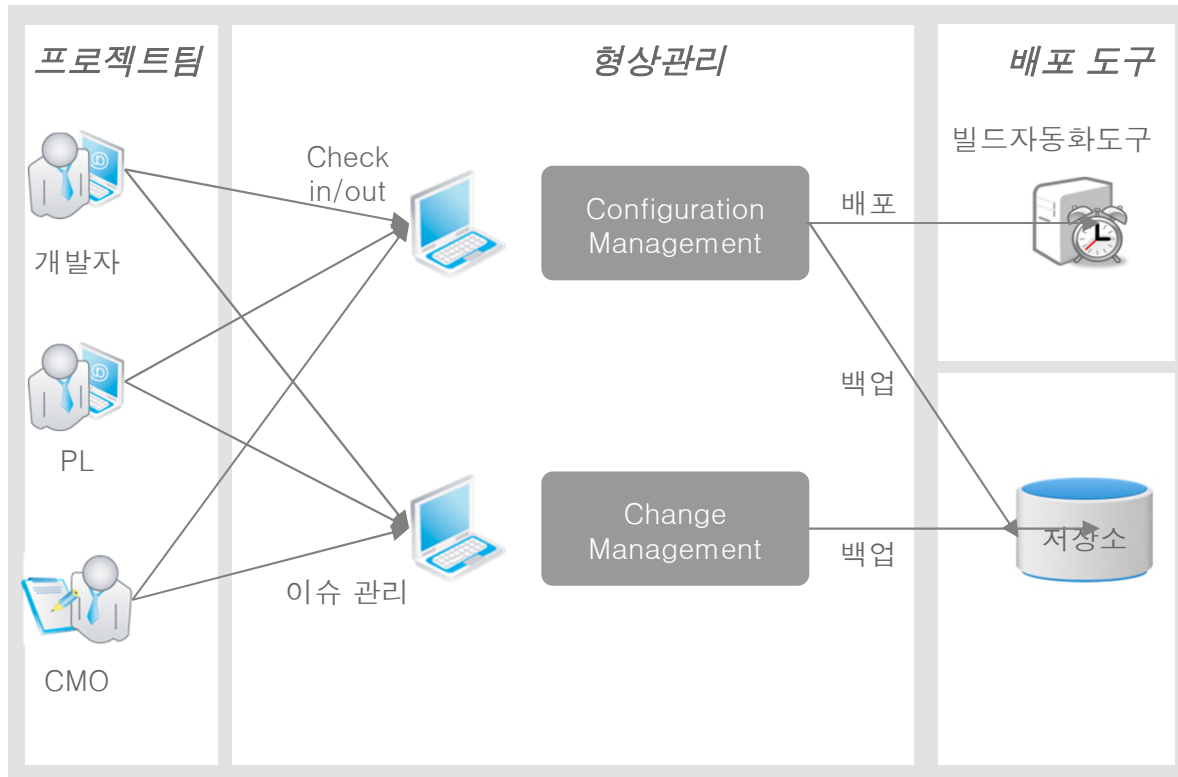


구성요소별 상세

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

형상요소의 식별, 이력관리를 지원하는 형상도구와 JAVA기반의 이슈관리시스템 제공

주요 메커니즘



구성요소별 상세

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

### □ 설치

- eGovFrame-3.10.0.exe 파일을 C:\ 드라이브에서 실행한다.

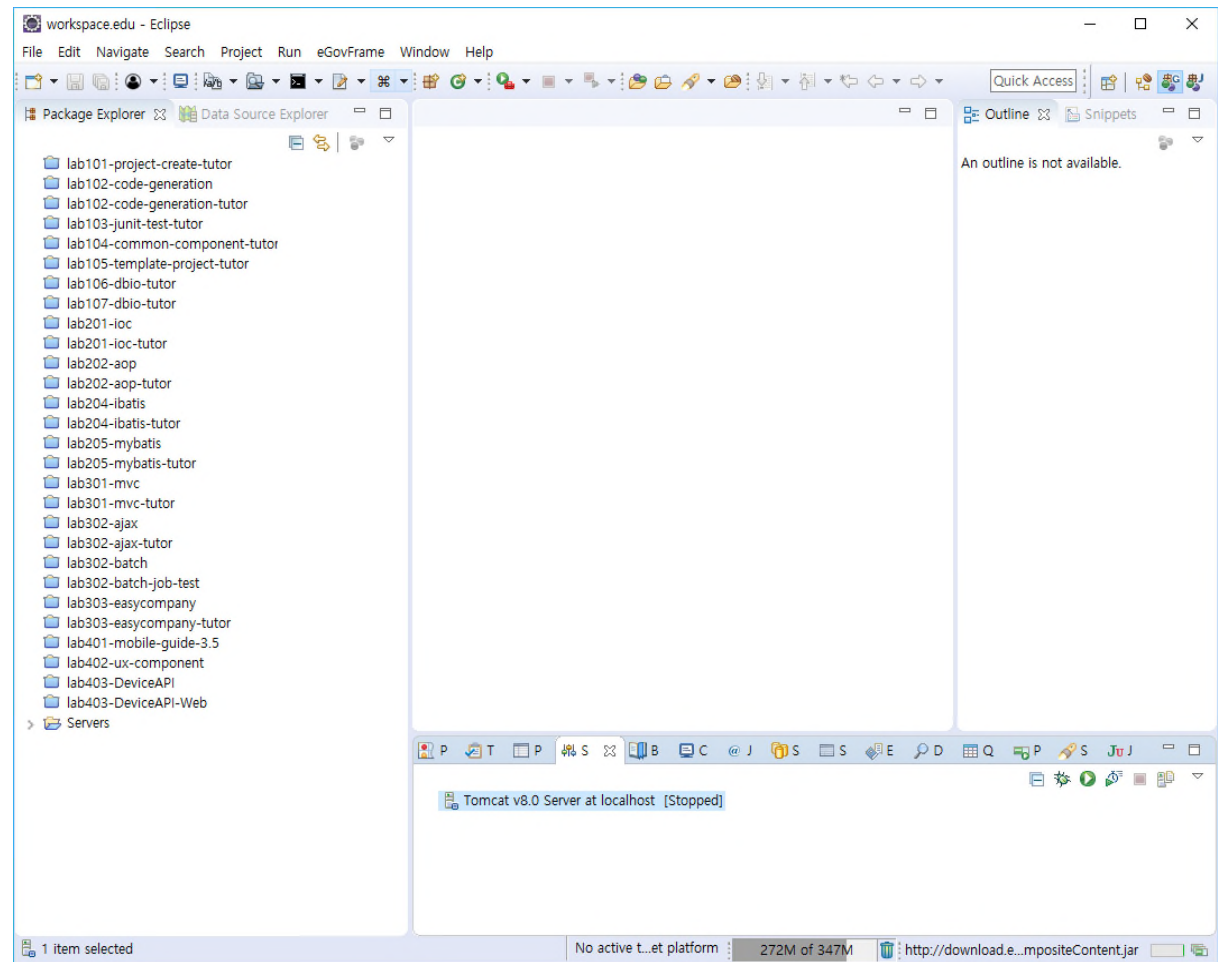
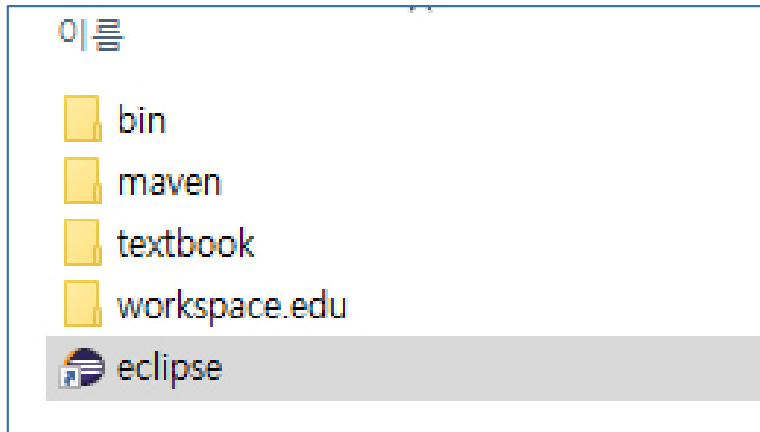
### □ 디렉토리 정보

▼	eGovFrame-3.10.0
▼	bin
>	android-sdk-windows
>	apache-tomcat-8.5.50
>	eclipse
>	jdk8u242-b08
>	mysql-5.6.21
	mysql-5.7.32-winx64
▼	maven
>	repository
>	textbook
>	workspace.edu

디렉토리	설명
bin	실행 파일
android-sdk-windows	윈도우용 안드로이드 SDK 폴더
apache-tomcat-8.5.50	Apache Tomcat
eclipse	개발환경이 포함된 Eclipse IDE for Java EE Developers 2020-06 (4.16.0)
jdk8u242-b08	OpenJDK - hotspot VM(JDK_HOME)
mysql-5.6.21	실습용 DB
mysql-5.7.32-winx64	실습용 DB (64bit)
maven/repository	Local Maven Repository
textbook	교육 교재 파일 (PDF)
workspace.edu	Eclipse Workspace

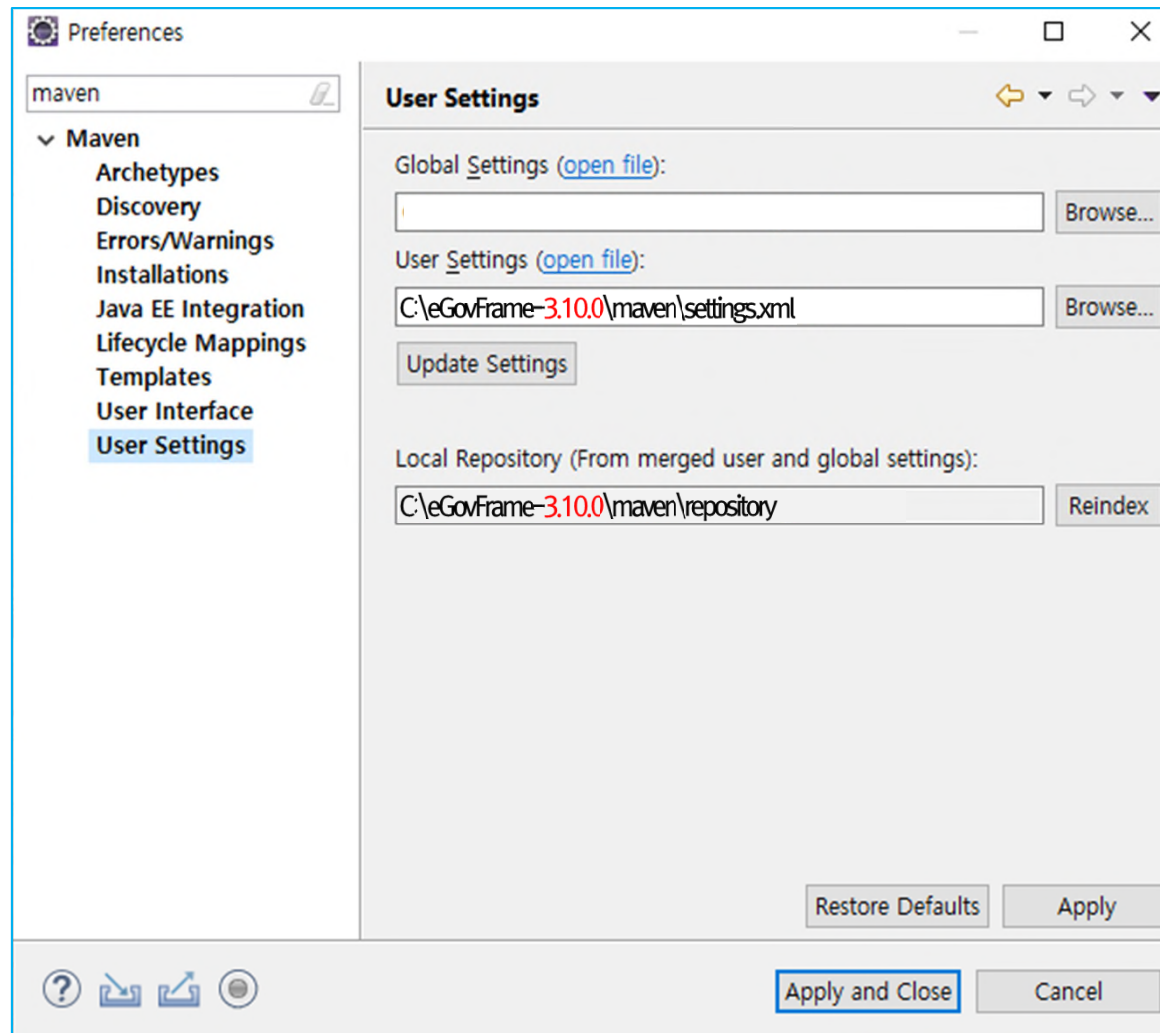
### □ 실행

- “C:\eGovFrame-3.10.0\eclipse” 를 더블 클릭하여 Eclipse를 실행한다.



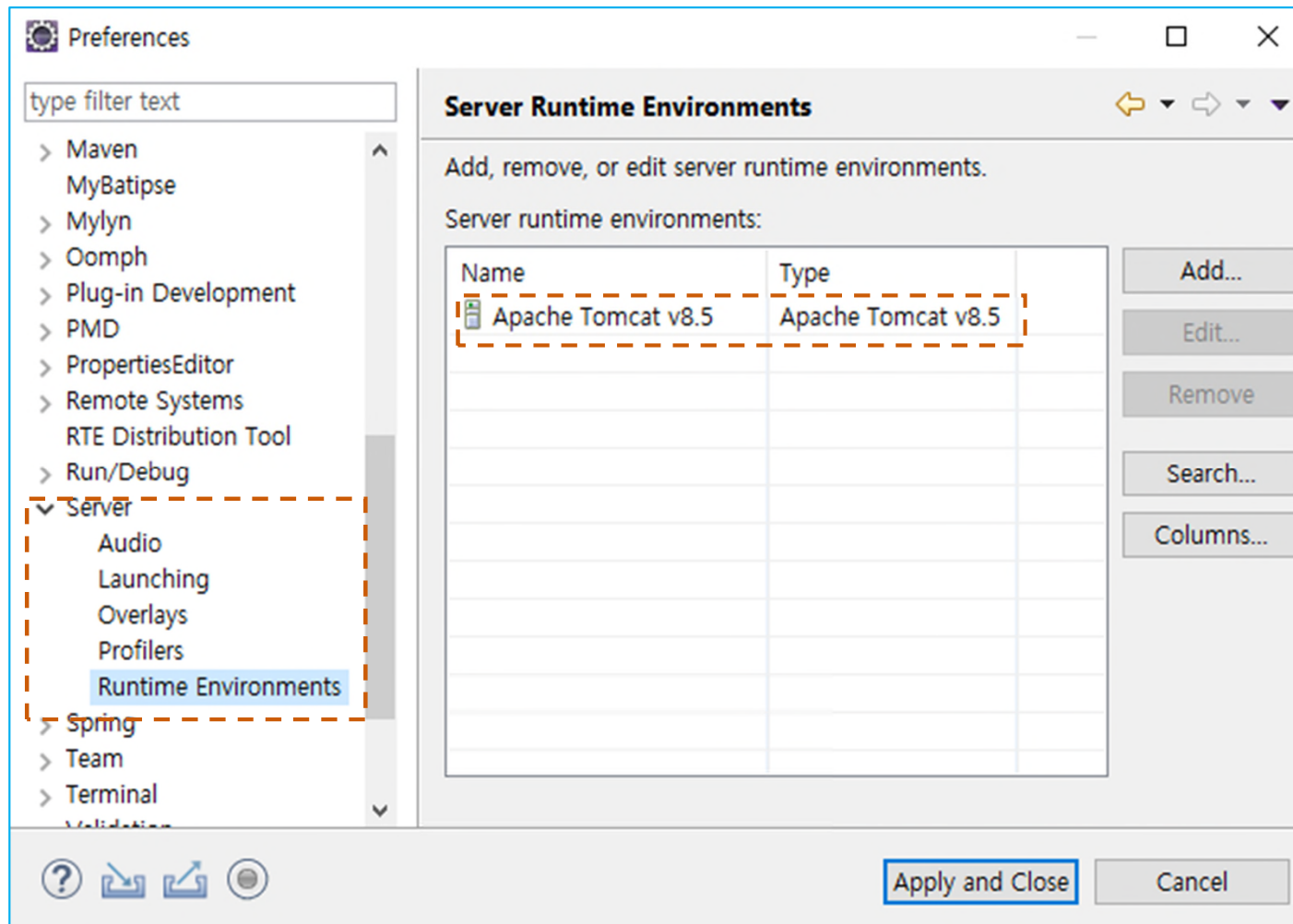
### □ 설정

- Maven Installations 설정 확인



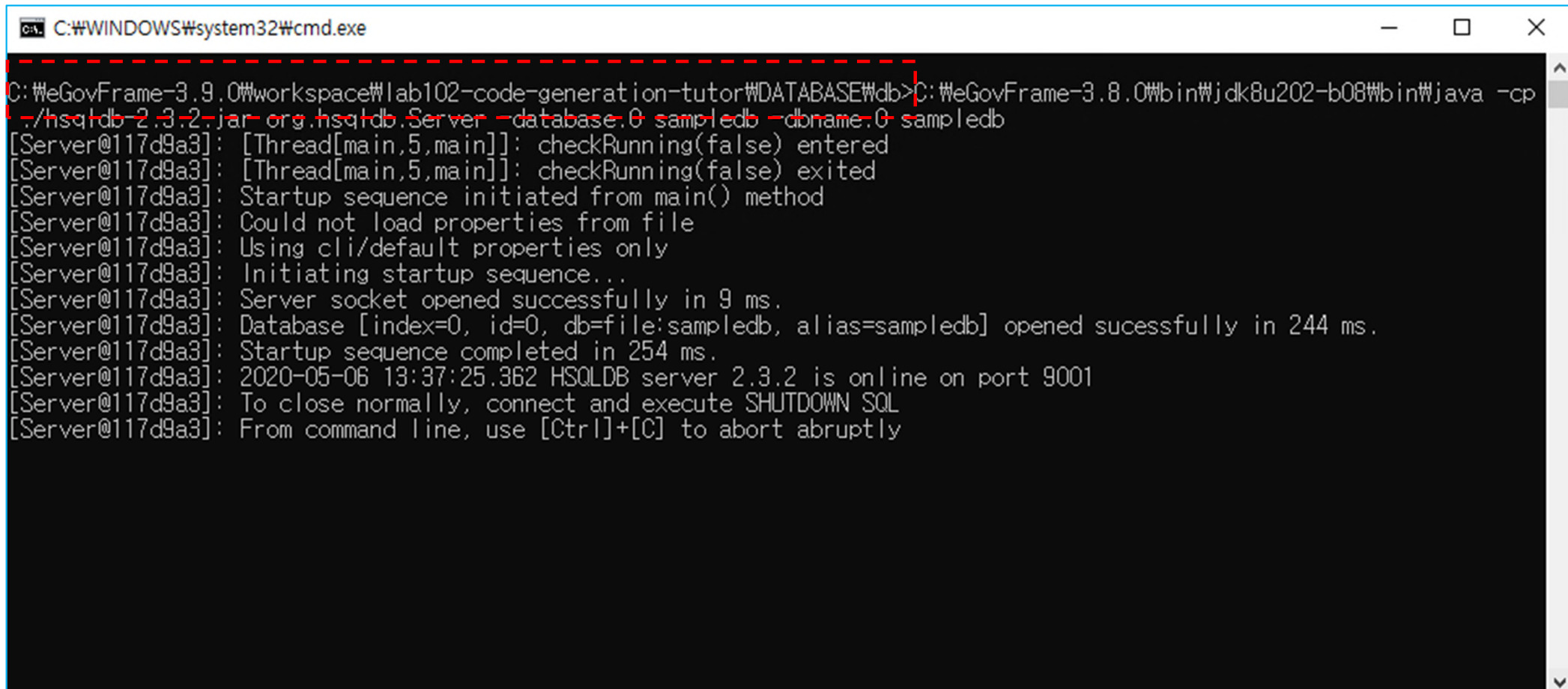
### □ 설정

- Apache Tomcat 설정



### □ 설정

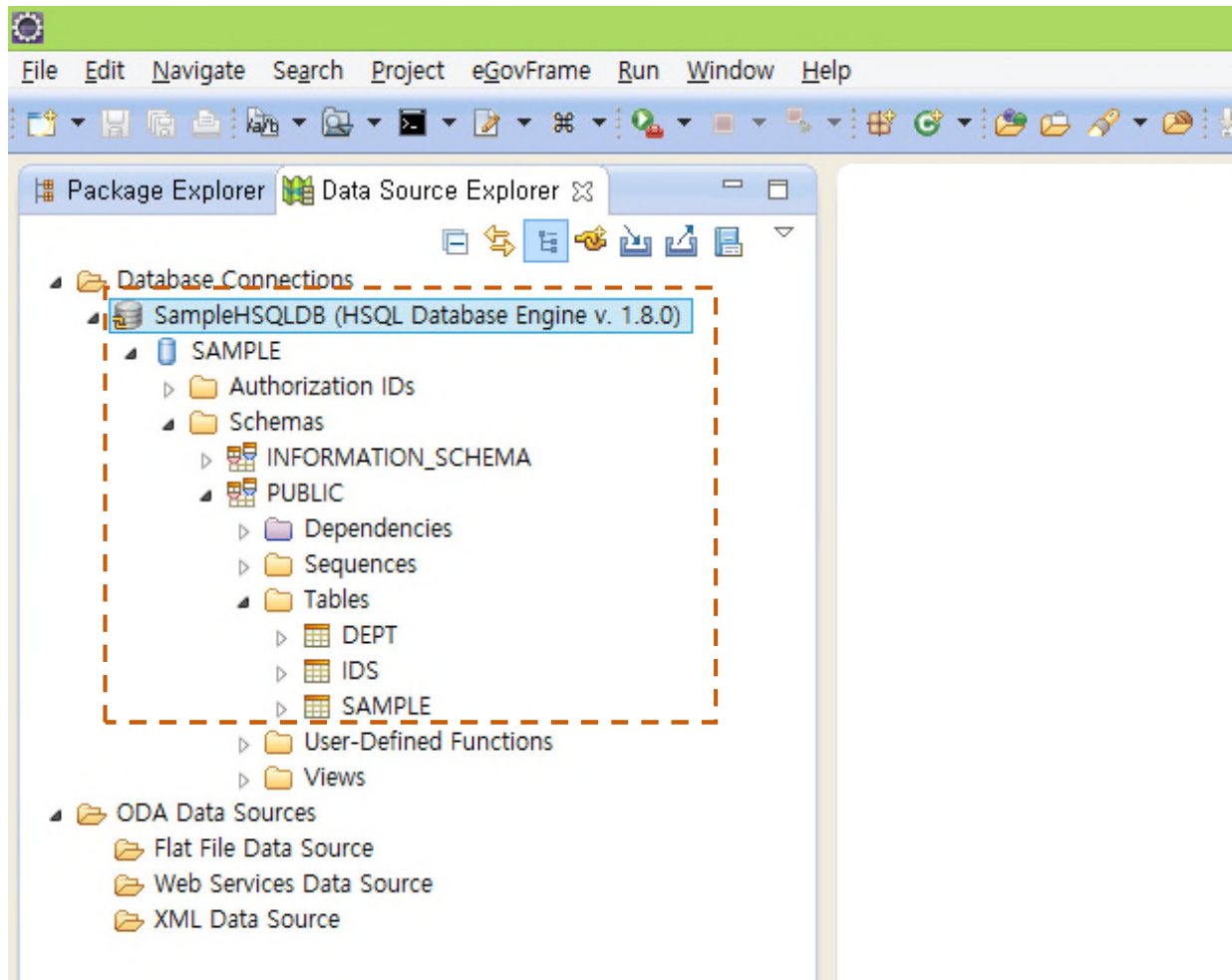
- 대상 프로젝트 DB 구동을 확인한다. ( example project name: lab102-code-generation)
- C:\egovFrame-3.10.0\workspace.edu\lab102-code-generation\DATABASE\db\runHsqlDB.cmd”를 더블클릭 하여 Sample용 DB를 띄운다.



```
C:\WINDOWS\system32\cmd.exe
C:\egovFrame-3.9.0\workspace\lab102-code-generation-tutor\DATABASE\db>C:\egovFrame-3.8.0\bin\jdk8u202-b08\bin\java -cp
./hsqldb-2.3.2-jar-orig\hsqldb.Server -database:0 sampled -dbname:0 sampled
[Server@117d9a3]: [Thread[main,5,main]]: checkRunning(false) entered
[Server@117d9a3]: [Thread[main,5,main]]: checkRunning(false) exited
[Server@117d9a3]: Startup sequence initiated from main() method
[Server@117d9a3]: Could not load properties from file
[Server@117d9a3]: Using cli/default properties only
[Server@117d9a3]: Initiating startup sequence...
[Server@117d9a3]: Server socket opened successfully in 9 ms.
[Server@117d9a3]: Database [index=0, id=0, db=file:sampled, alias=sampled] opened successfully in 244 ms.
[Server@117d9a3]: Startup sequence completed in 254 ms.
[Server@117d9a3]: 2020-05-06 13:37:25.362 HSQLDB server 2.3.2 is online on port 9001
[Server@117d9a3]: To close normally, connect and execute SHUTDOWN SQL
[Server@117d9a3]: From command line, use [Ctrl]+[C] to abort abruptly
```

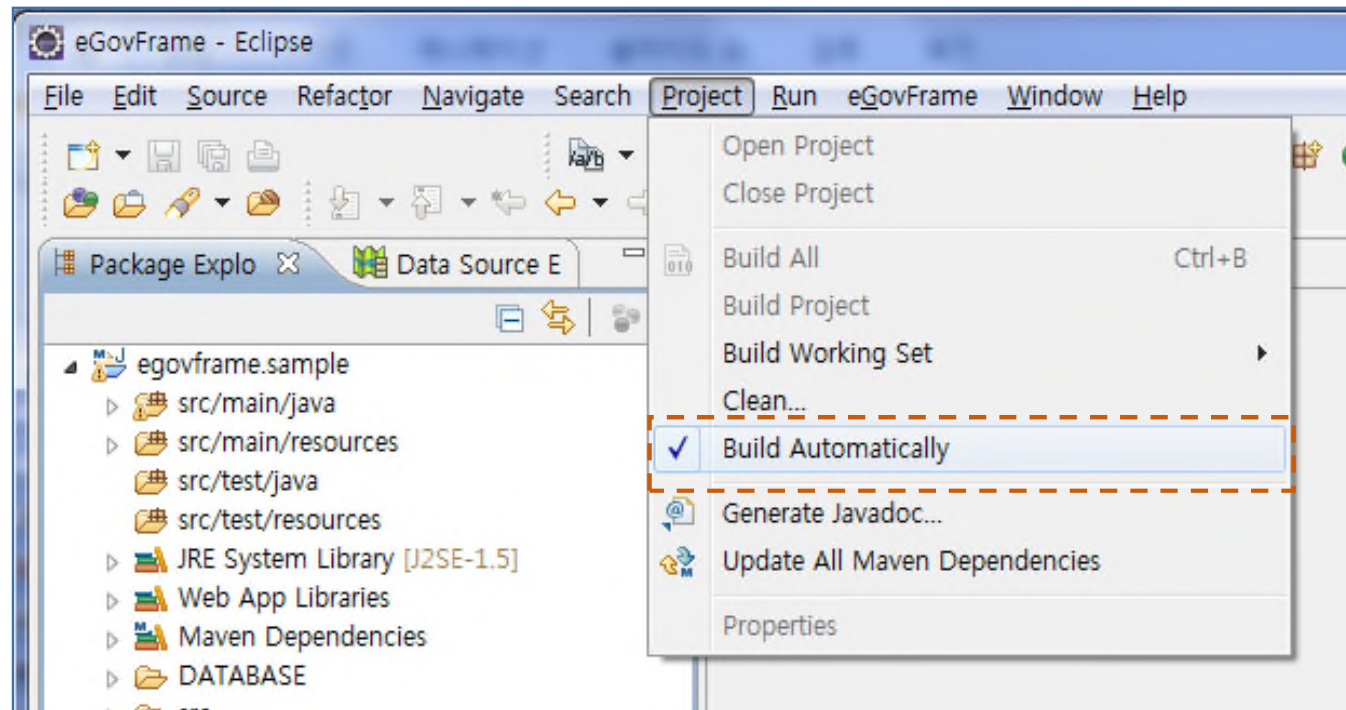
### □ 설정

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



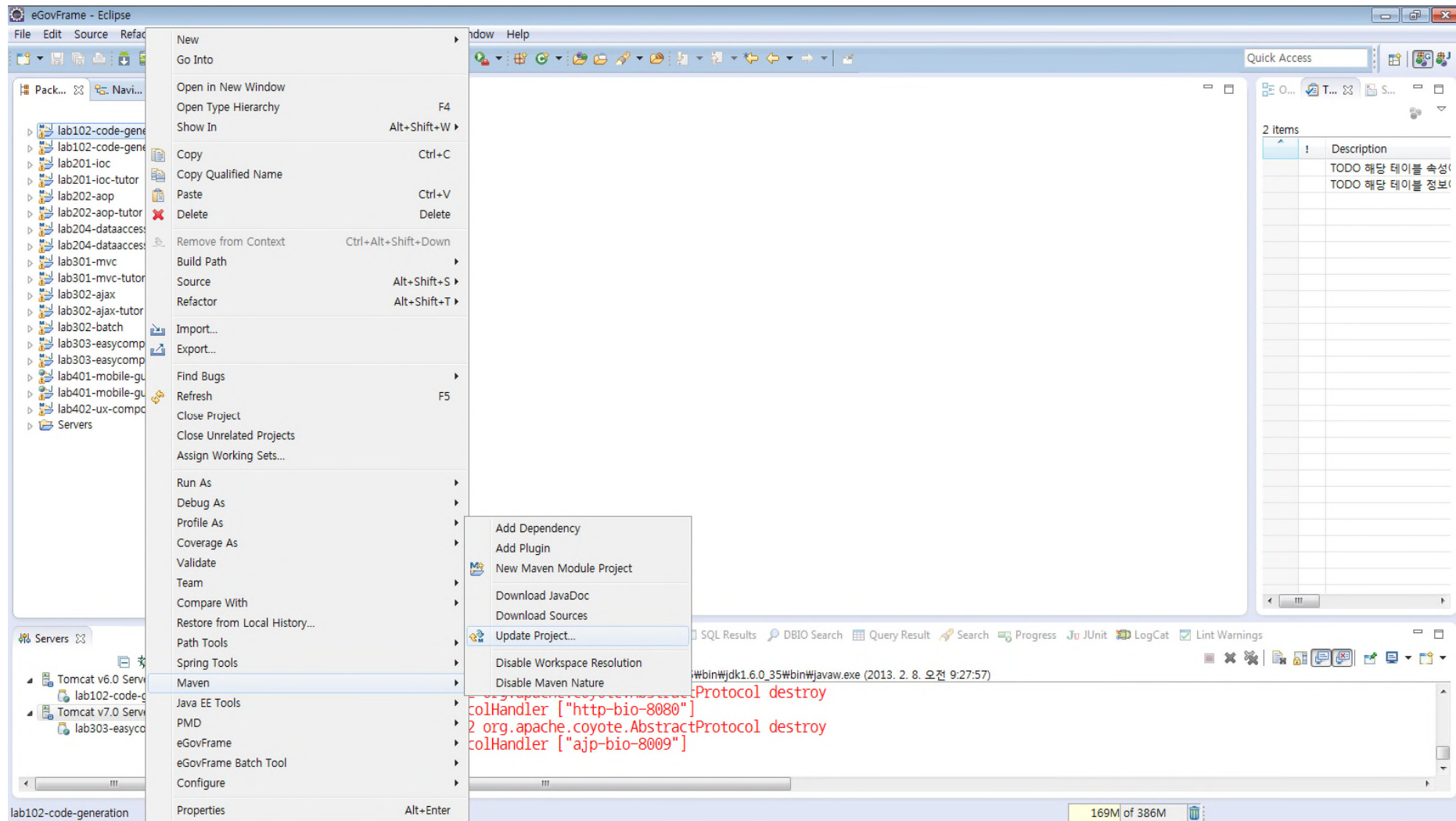
### □ 설정

- Eclipse에서 “Build Automatically”가 선택되어 있다.



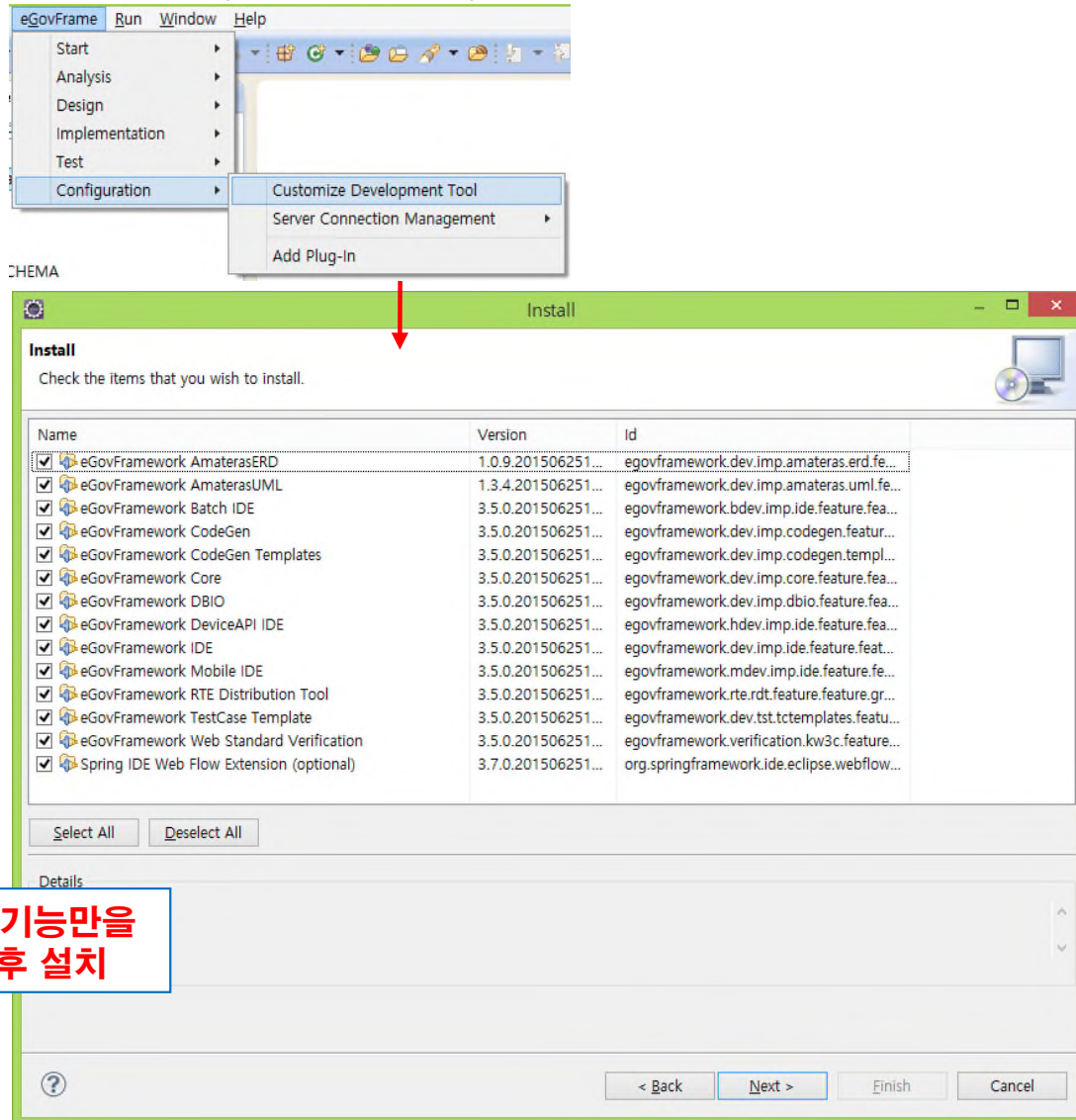
### □ 설정

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



### □ 개발자 개발환경 선택적 구성

- 개발환경 구성 시 필요 기능만을 선택하여 개발환경을 선택적으로 구성하는 기능을 제공



### □ 제공 기능

- eGovFramework AmaterasERD
- eGovFramework AmaterasUML
- eGovFramework Batch IDE
- eGovFramework CodeGen
- eGovFramework CodeGen Templates
- eGovFramework DBIO
- eGovFramework DeviceAPI IDE
- eGovFramework Mobile IDE
- eGovFramework RTE Distribution Tool
- eGovFramework TestCase Templates
- eGovFramework Web Standard Verification
- Spring IDE Web Flow Extension(optional)

# 7. 서버 개발환경 구성

# 1.개발환경 개요

## □ 다운로드

- eGovFrame 포털(<http://www.egovframe.go.kr/>)에 접속하여 "다운로드 > 개발환경"에서 제공하는 서버용 개발환경(Windows, Unix계열) 설치파일을 다운로드 한다.

표준프레임워크 포털 eGovFrame

인기검색어 Syslogao maven Mybatis

표준프레임워크 소개 개발 가이드 다운로드 개발자 교육 기술지원 호환성확인 알림마당

다운로드 > 개발환경 다운로드

### 개발환경 다운로드

전체	3.8.0	3.7.0	3.6.0	3.5.1	3.5.0	3.2.0
3.8.0	개발자용 개발환경 32bit64bit(Implementation Tool) Version 3.8.0	릴리즈 일자 - 2019.02.28				
3.8.0	구현도구(Implementation Tool) Version 3.8.0	릴리즈 일자 - 2019.02.28				
3.8.0	구현도구(Implementation Tool) 소스코드 Version 3.8.0	릴리즈 일자 - 2019.02.28				
3.8.0	서버용 개발환경(Development Tool - Unix, Linux, Win32/64bit) Version 3.8.0	릴리즈 일자 - 2019.02.28				

### 개발환경 다운로드

전체	3.8.0	3.7.0	3.6.0	3.5.1	3.5.0	3.2.0
3.8.0	개발자용 개발환경 32bit64bit(Implementation Tool) Version 3.8.0	릴리즈 일자 - 2019.02.28				
3.8.0	구현도구(Implementation Tool) Version 3.8.0	릴리즈 일자 - 2019.02.28				
3.8.0	구현도구(Implementation Tool) 소스코드 Version 3.8.0	릴리즈 일자 - 2019.02.28				
3.8.0	서버용 개발환경(Development Tool - Unix, Linux, Win32/64bit) Version 3.8.0	릴리즈 일자 - 2019.02.28				

작성자 관리자 작성일 2019-02-28 조회 355

첨부파일 eGovCr-3.8.0\_64bit.exe[495,455,499 byte] eGovCr-3.8.0\_32bit.exe[432,893,627 byte] eGovCr-3.8.0.tar.gz[249,560,002 byte]

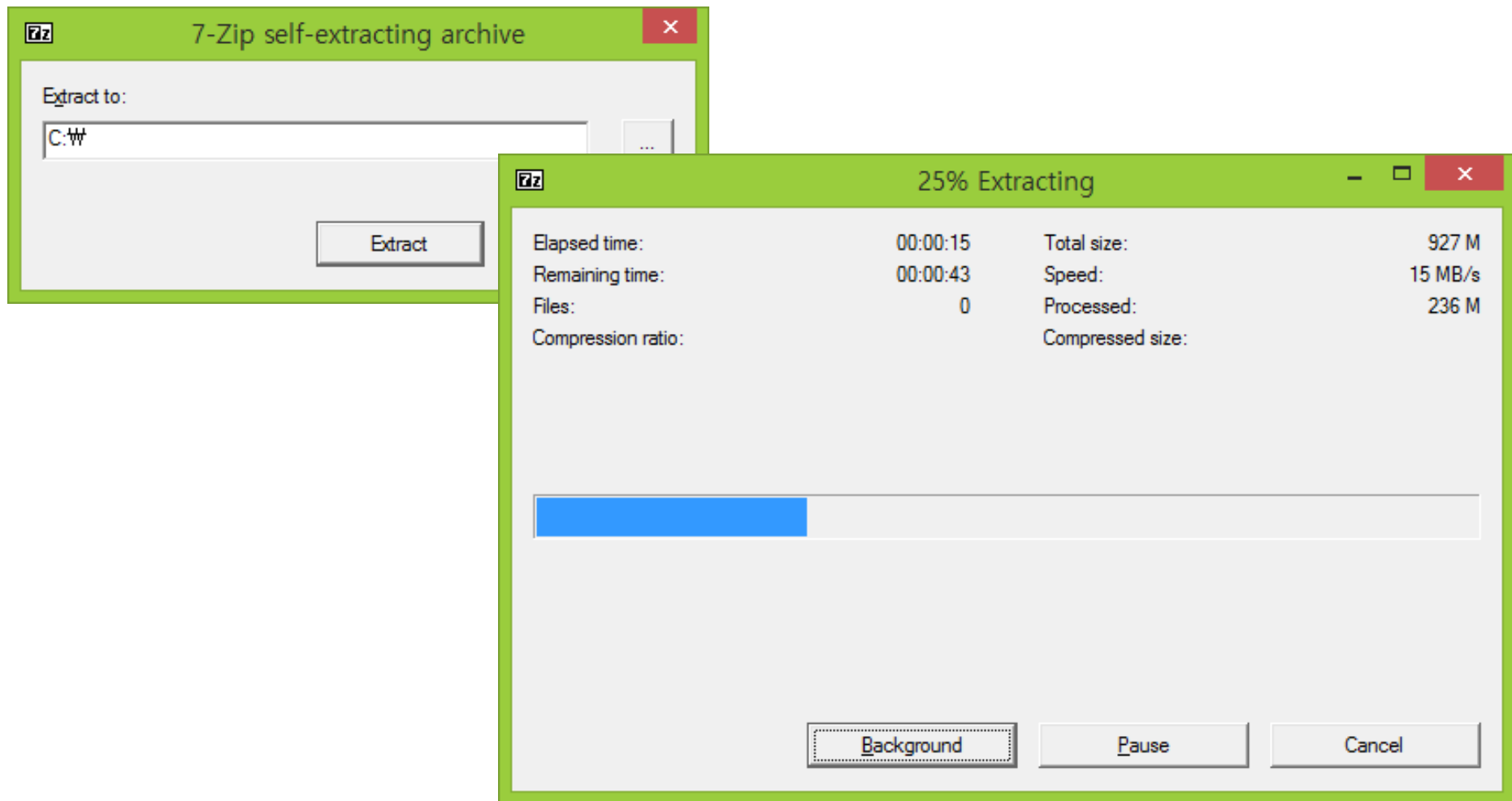
[구성]  
전자정부 서버용 개발환경(Server Development Tool)은 전자정부 표준 프레임워크 기반 어플리케이션 개발 시 통합 빌드 환경 구성에 필요한 소프트웨어로 구성되어 있습니다.  
전자정부 서버용 개발환경은 오픈 소스 솔루션인 NEXUS, SVN을 이용하여 라이브러리 및 소스에 대한 관리를 독립적으로 처리하며 JENKINS CI 서버를 이용해 주기적인 통합 빌드를 자동으로 수행합니다.

- 주소 확인  
\* <http://localhost:8080/jenkins>  
\* 사용포트 변경시(eGovCr-3.8.0/bin/apache-tomcat-jenkins/conf/server.xml) 파일수정  
\* <http://localhost:9090/nexus>  
\* 사용포트 변경시(eGovCr-3.8.0/bin/apache-tomcat-nexus/conf/server.xml) 파일수정

[OS별 설치버전 개요]  
윈도우 32/64bit용  
- 설치 파일 : eGovCr-3.8.0\_64bit.exe, eGovCr-3.8.0\_32bit.exe  
- 주용 설치 컴포넌트  
Apache-ant:1.9.5  
Apache-maven:3.3.3  
Apache-tomcat:8.0.47  
Jenkins 2.138.3  
Nexus 2.11.3-01  
cvs-1.11-22  
jdk1.8.0\_162

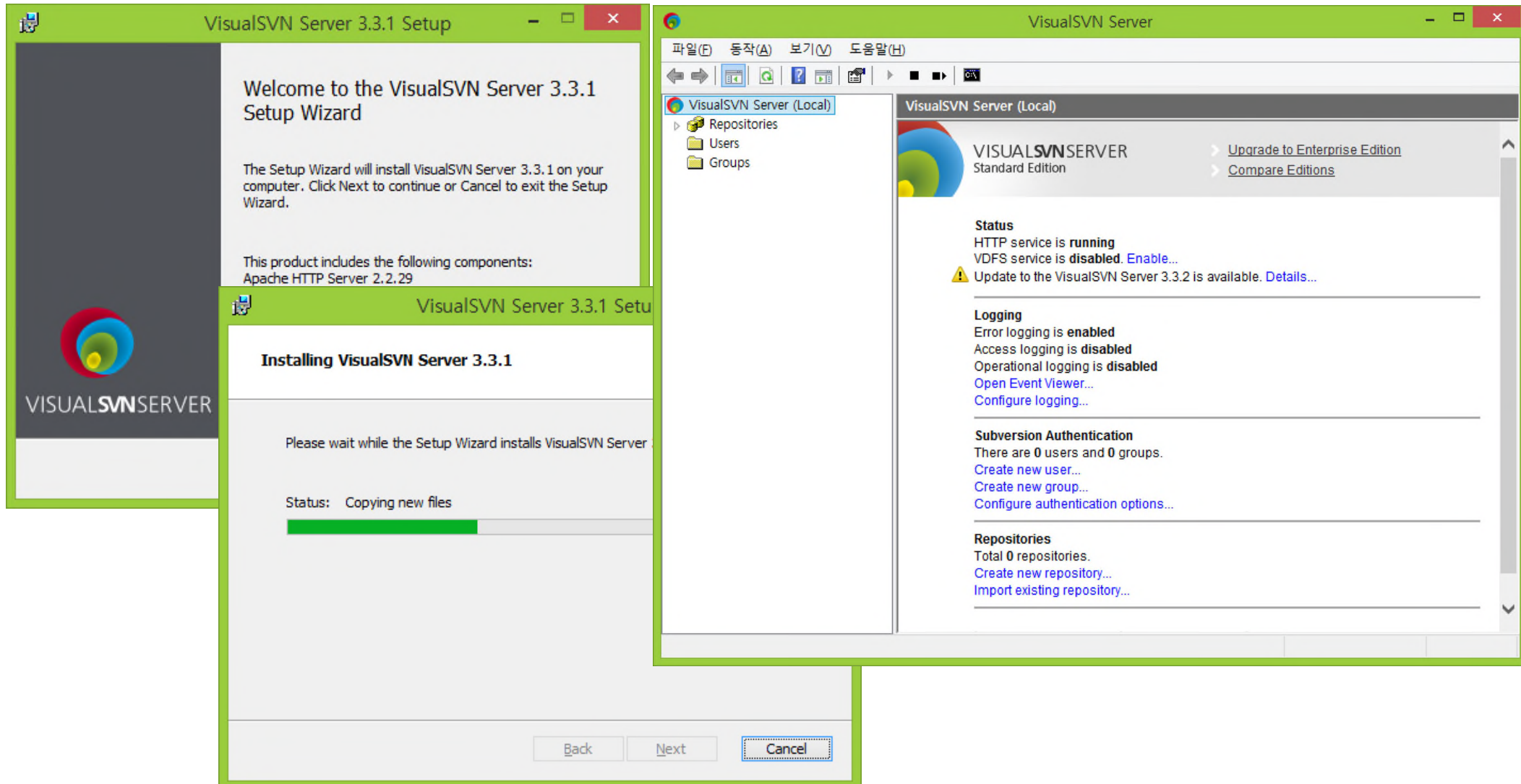
#### □ 설치

- eGovCI-3.9.0\_XXbit.exe 파일을 실행하여 압축을 해제한다.



### □ 설치

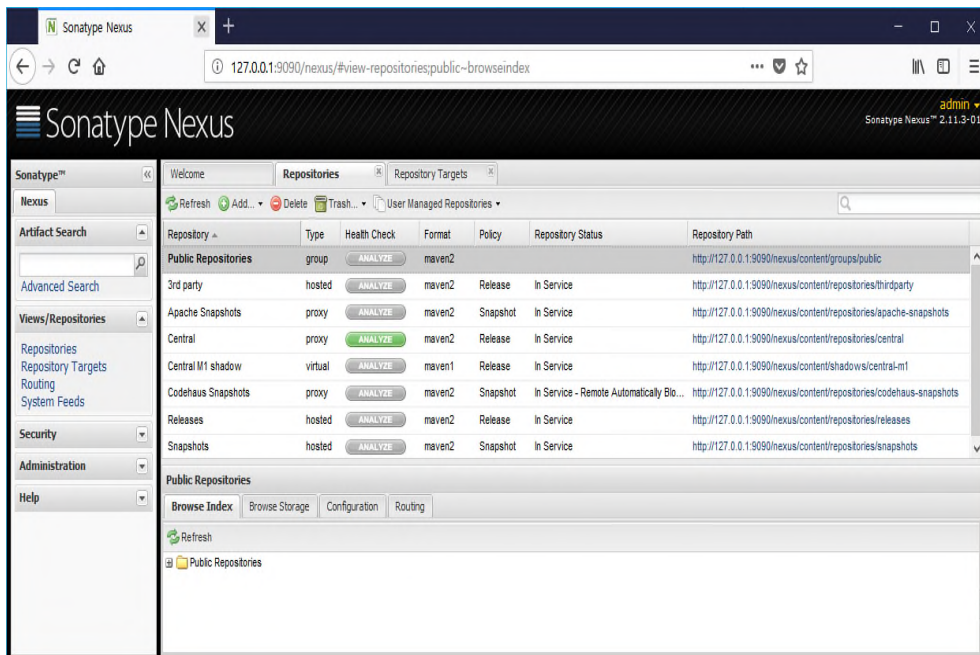
- <eGovCI\_home>\bin\VisualSVN-Server-XXX.msi 파일을 실행하여 VisualSVN을 설치한다.



### □ 실행

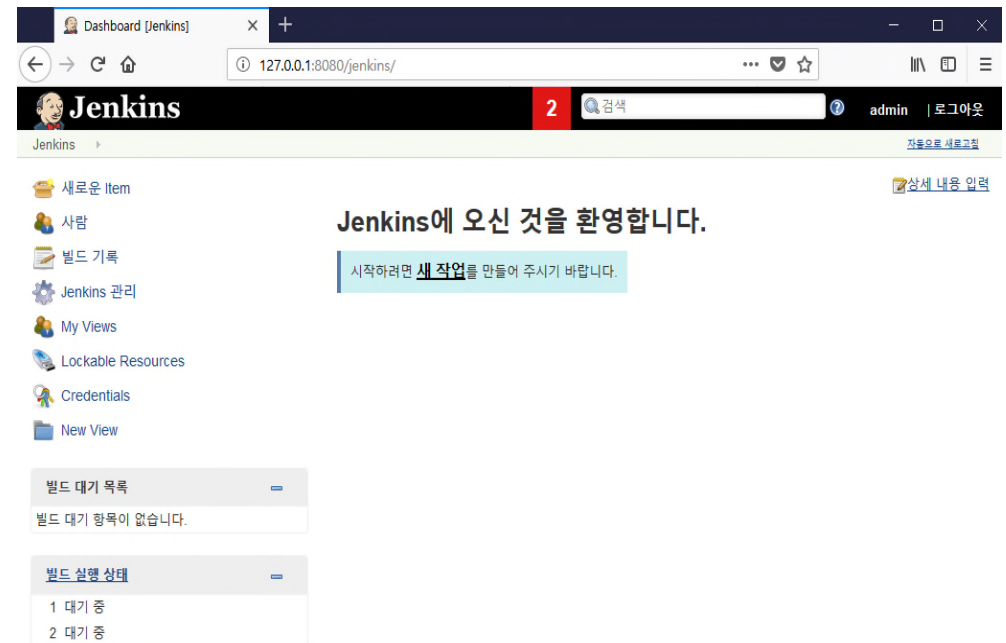
- Nexus 서버 구동 : <eGovCI\_home>\start-nexus.bat 파일을 실행하여 진행한다.
- Jenkins 서버 구동 : <eGovCI\_home>\start-jenkins.bat 파일을 실행하여 진행한다.
- 브라우저 주소 입력창에 Nexus서버와 Jenkins 서버의 URL을 입력하여 설치 및 실행 여부를 확인할 수 있다.

<http://localhost:8081/nexus/>



[그림 1] Nexus가 실행된 모습

<http://localhost:8080/jenkins/>



[그림 2] Jenkins이 실행된 모습

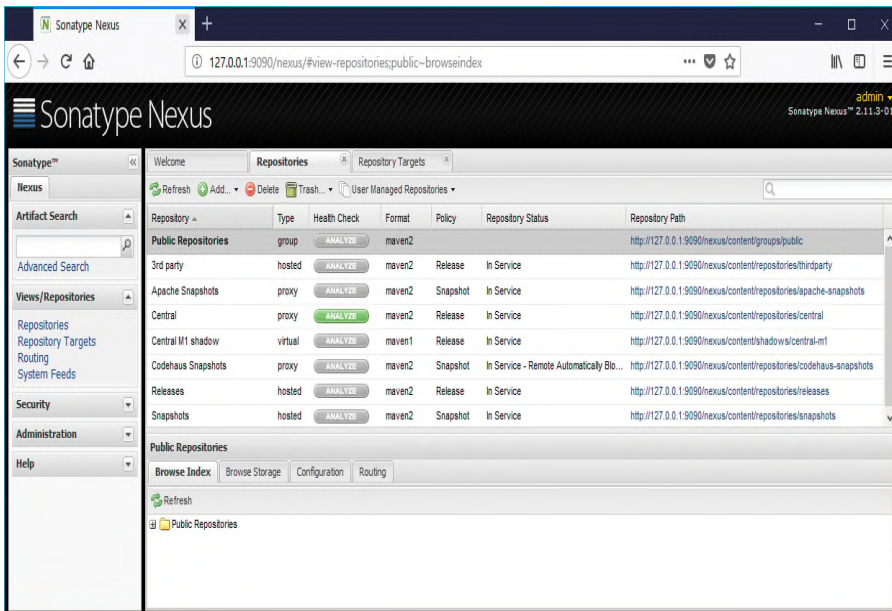
### □ 설치 및 실행

- 서버 개발환경을 설치할 폴더에 eGovCI-3.9.0.tar.gz 파일의 압축을 해제한다.
- JDK, 서버 IP와 포트를 설정한다.
- 설치완료 후 정상적으로 설치가 되었는지 확인하고 서버를 실행한다.

Nexus 서버 구동 : <eGovCI\_home>\start-nexus.sh 파일을 실행하여 진행한다.

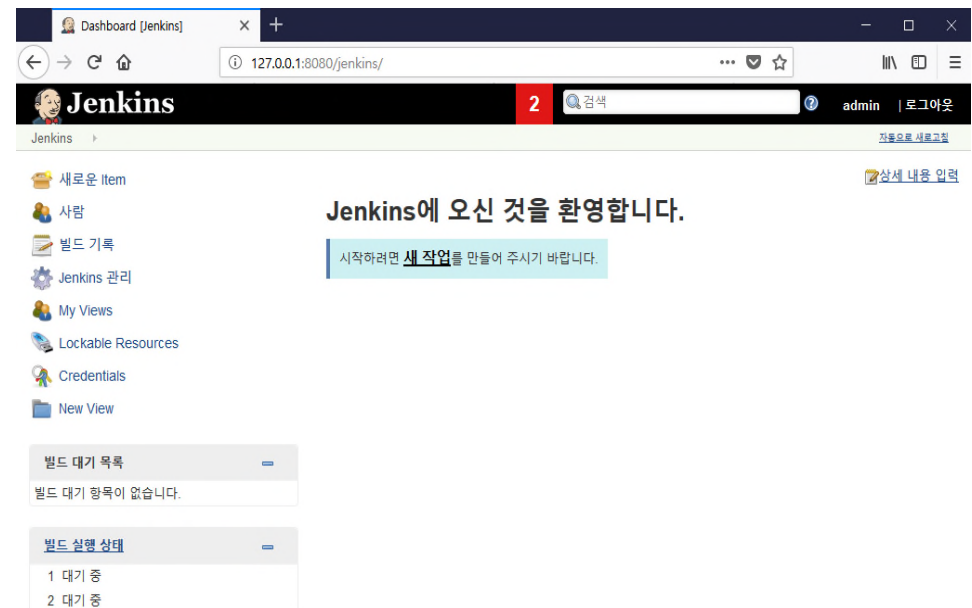
Jenkins 서버 구동 : <eGovCI\_home>\start-jenkins.sh 파일을 실행하여 진행한다.

http://localhost:8081/nexus/



[그림 3] Nexus가 실행된 모습

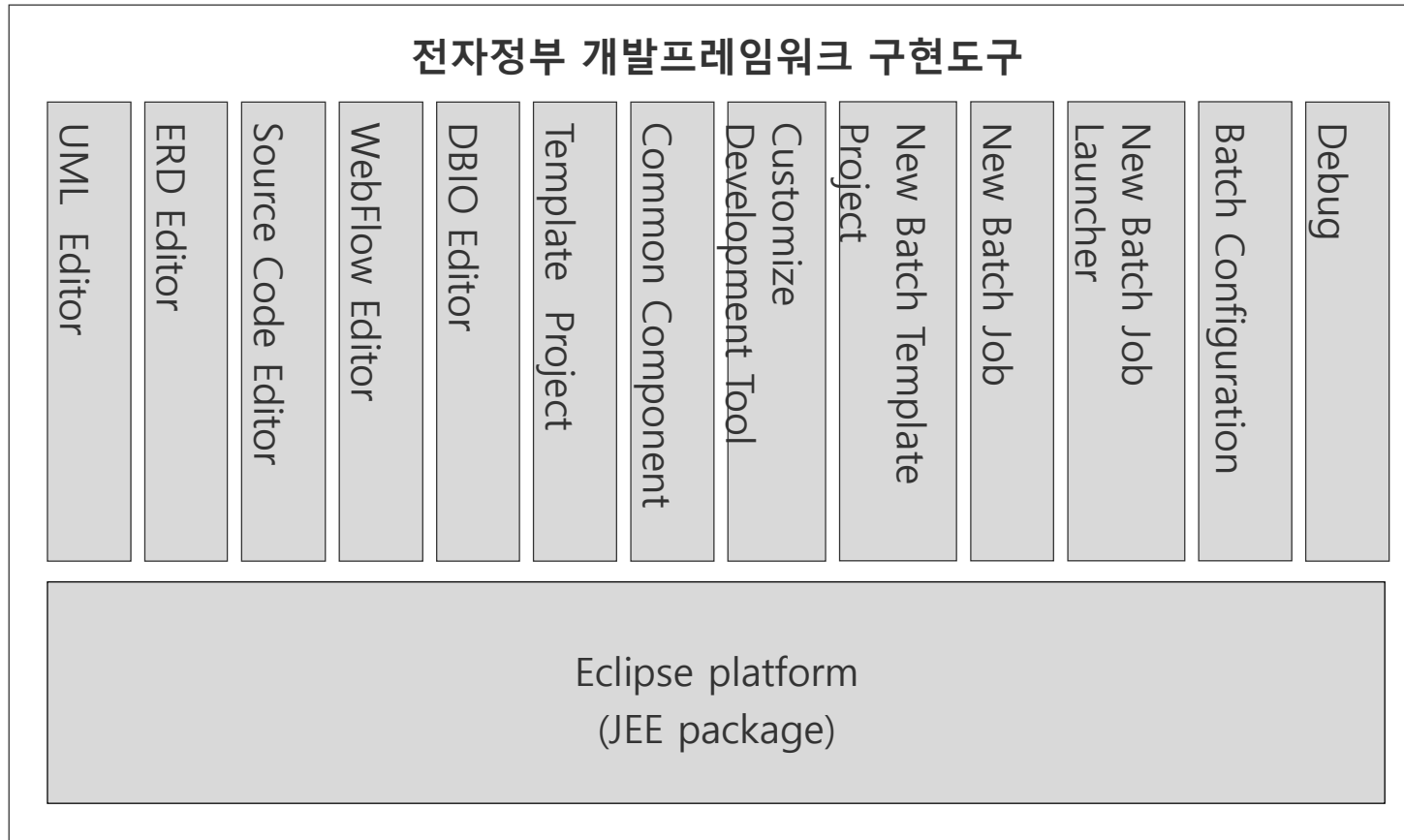
http://localhost:8080/jenkins/



[그림 4] Jenkins이 실행된 모습

1. 개요
2. eGovFrame IDE
3. UML 작성
4. ERD 작성
5. DBIO 작성
6. Code Generation
7. eGovFrame Java Project
8. Common Component
9. Batch Job
10. Batch Job Launcher
11. Batch Configuration
12. eGovFrame Template Project
13. Debug
14. Code Inspection
15. Server Connection Management
16. 참고자료

구현도구는 UML, ERD, Source Code, WebFlow, DBIO Editor와 Template Project, Common Component, Customize Development Tool, 배치 Template 및 Job 생성 그리고 Debug로 구성



# 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, 배치 Template 및 Job 생성 플러그인으로 구성

□ 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 )

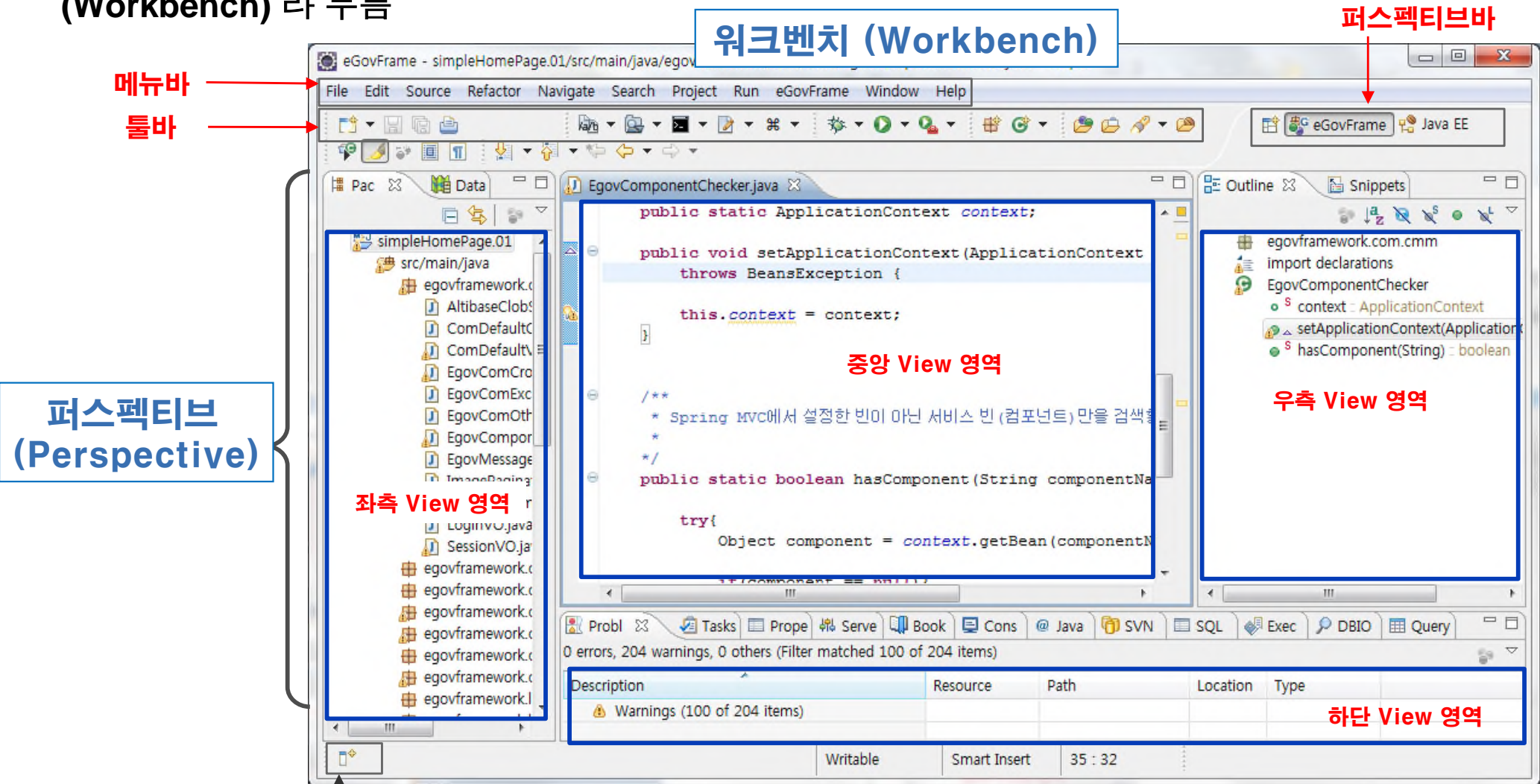
	Java	Java EE	C/C++	C/C++ Linux	RCP/RAP	Modeling	Reporting	Parallel	Scout	Testers	Javascript	Classic
RCP/Platform	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
CVS	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
EGit			✓	✓	✓	✓						
EMF	✓	✓				✓	✓					
GEF	✓	✓				✓	✓					
JDT	✓	✓			✓	✓	✓		✓			✓
Mylyn	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
Web Tools		✓					✓				✓	
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

# 1. 개요 – Eclipse 기본화면구성

## 2.구현 도구

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

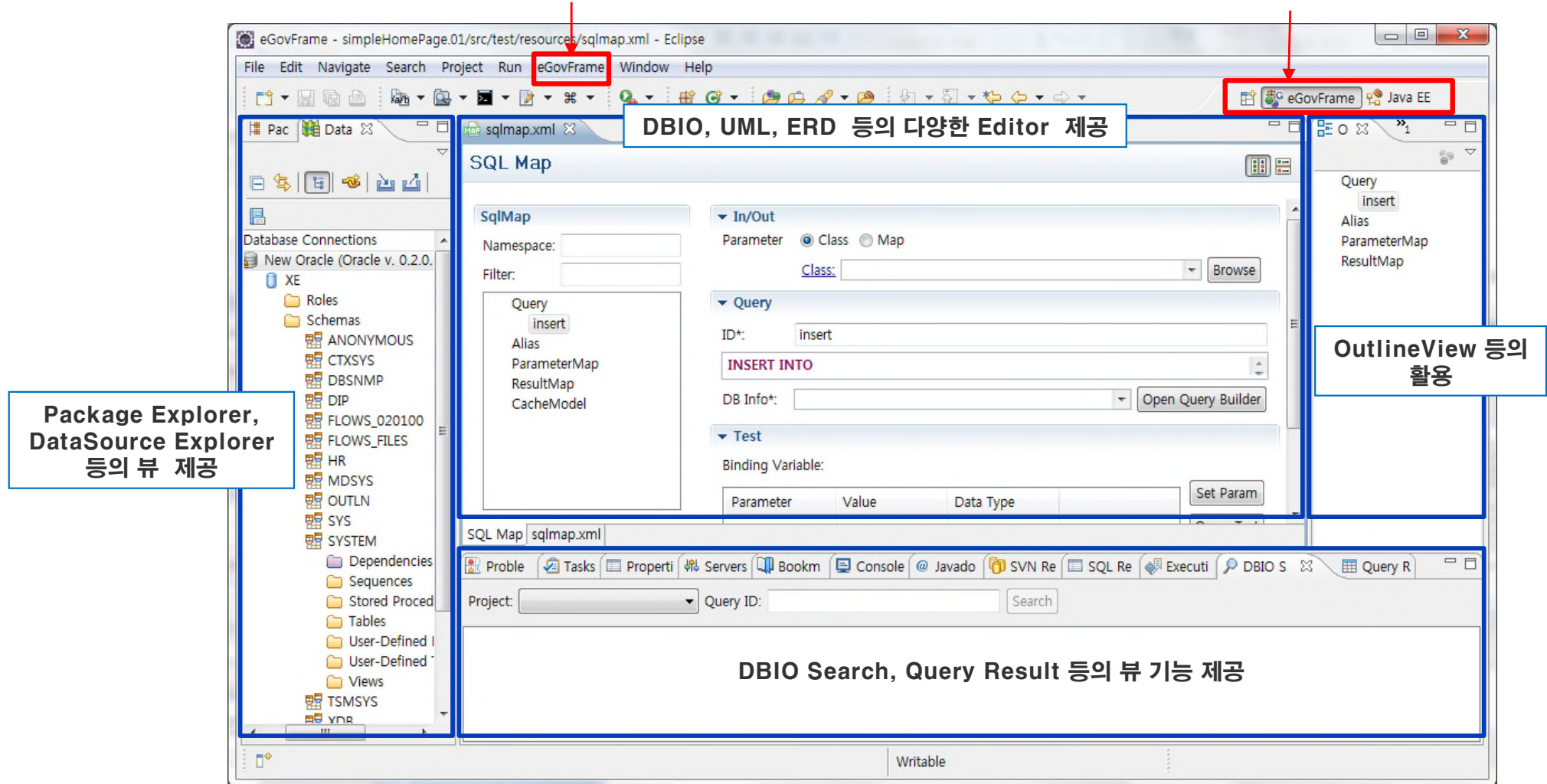


Fast 뷰 : 잘 사용하지 않는 화면들을 최소화 시켜놓고 필요한 시점에만 잠깐 활성화 시킬 때 사용된다.

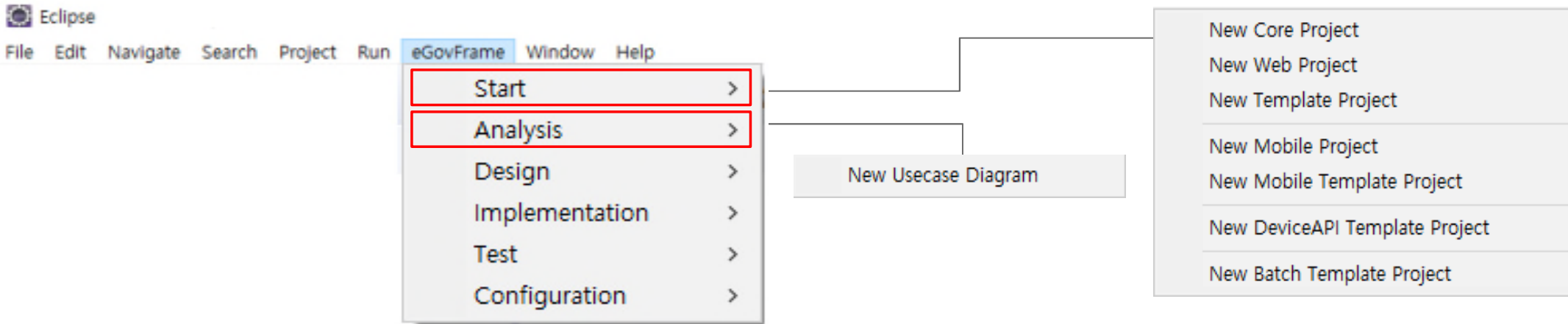
구현도구는 개발자가 개발환경을 쉽고 편리하게 개발할 수 있도록 특성화된 통합메뉴, 퍼스펙티브, 뷰, 에디터 등을 제공

eGovFrame 통합메뉴

eGovFrame 퍼스펙티브 전환



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



\* 세부 메뉴 항목별 설명

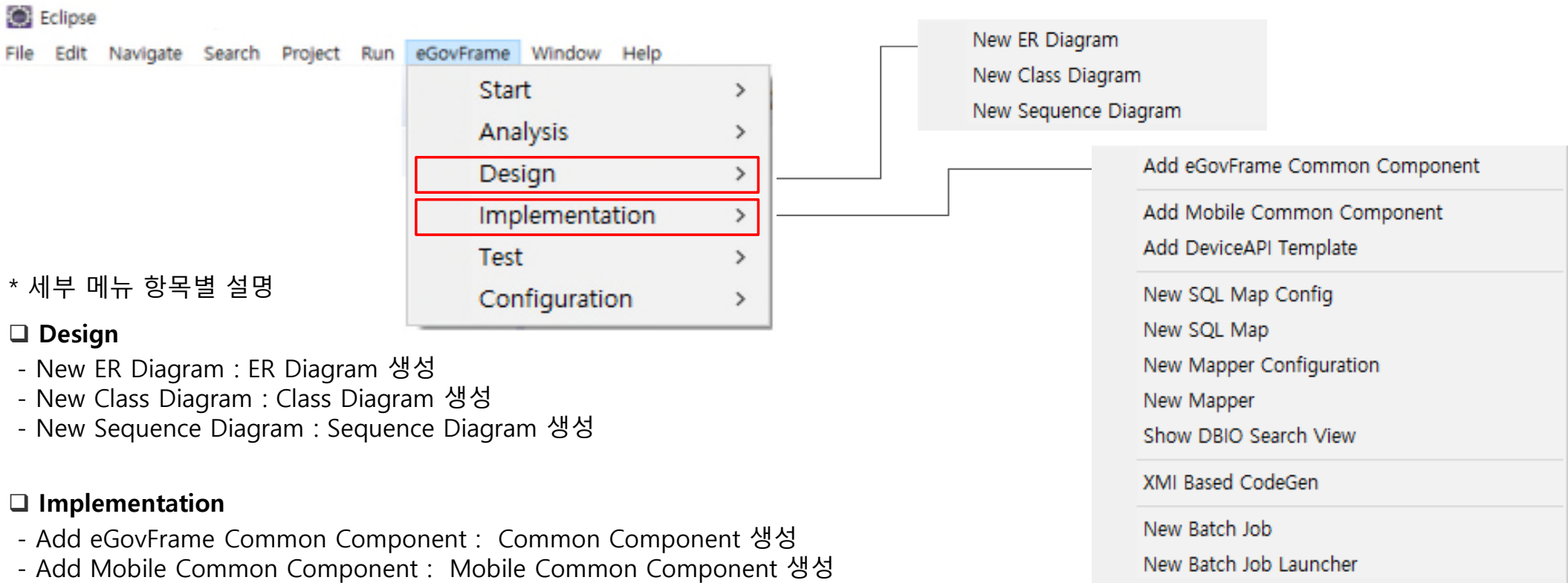
### □ Start

- New Core Project : eGovFrame Core Project 생성
- New Web Project : eGovFrame Web Project 생성
- New Template Project : eGovFrame Template Project 생성
  - Simple Homepage
  - Portal Site
  - Enterprise Business
  - Common All-in-one
- New Mobile Project : eGovFrame Mobile Project 생성
- New Mobile Template Project : eGovFrame Mobile Template Project 생성
  - Mobile Common All-in-one
- New DeviceAPI Template Project : eGovFrame DeviceAPI Template Project 생성
- New Batch Template Project : eGovFrame Batch Template Project 생성

### □ Analysis

- New Usecase Diagram : Usecase Diagram 생성

**eGovFrame Perspective**에서만 활성화되는 메뉴로 **eclipse**내에서 분산되어 있는 플러그인들의 기능 (**eGovFrame**에서 필히 사용되어지는 기능)을 빠르게 접근할 수 있는 통합 메뉴 제공



\* 세부 메뉴 항목별 설명

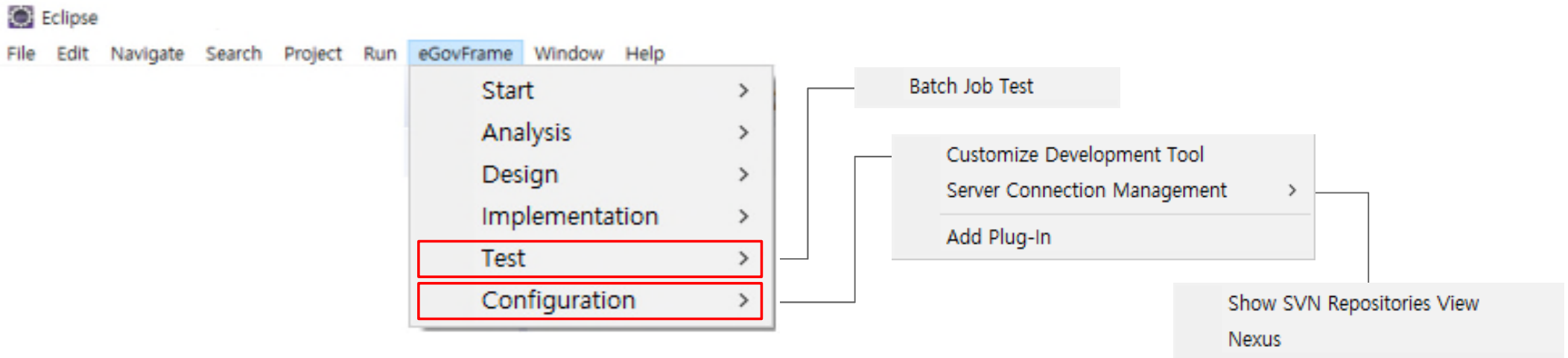
### □ Design

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

### □ Implementation

- Add eGovFrame Common Component : Common Component 생성
- Add Mobile Common Component : Mobile Common Component 생성
- Add DeviceAPI Template : eGovFrame DeviceAPI Template Project 생성
- New SQL Map Config : SQL Map Config 파일의 생성
- New SQL Map : SQL Map 파일의 생성
- New Mapper Configuration : Mapper Configuration 파일 생성
- New Mapper : Mapper 파일 생성
- Show DBIO Search View : DBIO Search View를 나타나게 함
- XML Based CodeGen : UML XML파일 기반으로 소스코드 생성
- New Batch Job : Job/Step/Item 구성해서 배치작업 파일 생성
- New Batch Job Launcher : 배치작업 실행파일 생성

**eGovFrame Perspective**에서만 활성화되는 메뉴로 **eclipse**내에서 분산되어 있는 플러그인들의 기능 (**eGovFrame**에서 필히 사용되어지는 기능)을 빠르게 접근할 수 있는 통합 메뉴 제공



\* 세부 메뉴 항목별 설명

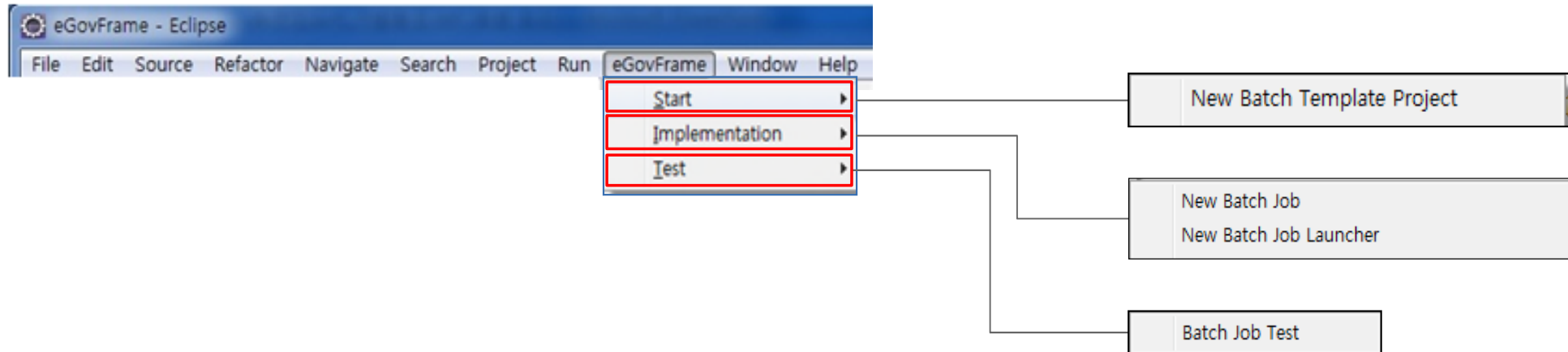
### ❑ Test

- Batch Job Test : Batch Job 테스트

### ❑ 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
- Add Plug-In : 플러그인 설치

**eGovFrame Perspective**에서만 활성화되는 메뉴로 **eclipse**내에서 분산되어 있는 플러그인들의 기능 (**eGovFrame**에서 필히 사용되어지는 기능)을 빠르게 접근할 수 있는 통합 메뉴를 제공



\* 세부 메뉴 항목별 설명

### ❑ Start

- New Batch Template Project : eGovFrame Batch Template Project 생성

### ❑ Implementation

- New Batch Job : Job/ Step/ Item을 구성해서 배치 작업 파일 생성
- New Batch Job Launcher : 배치 작업 실행 파일 생성

### ❑ Test

- Batch Job Test : 구성한 배치를 JUnit을 활용하여 테스트

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

#### □ UseCase Diagram 작성

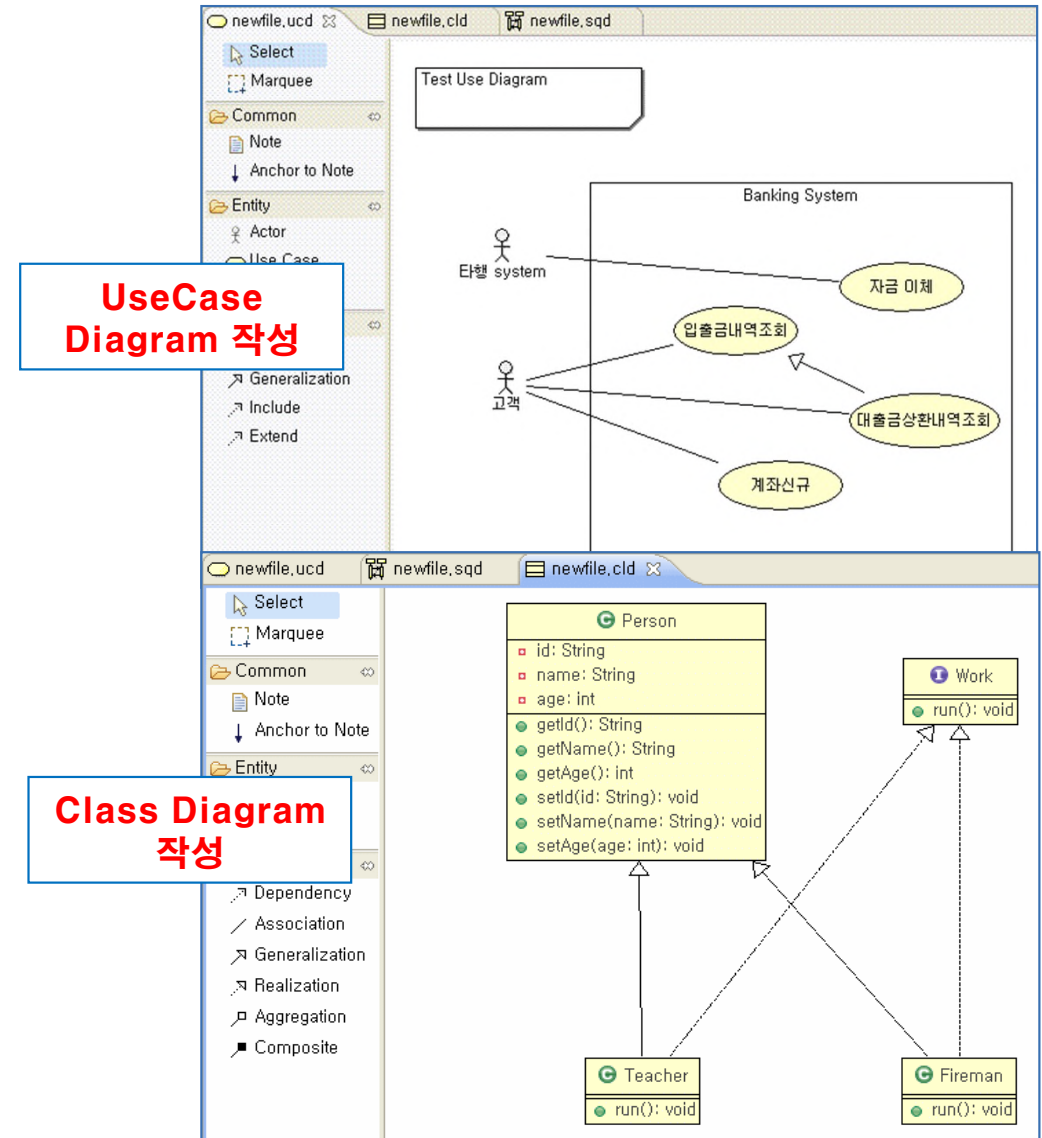
- Use Case Diagram 작성할 수 있는 기능을 제공

#### □ 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

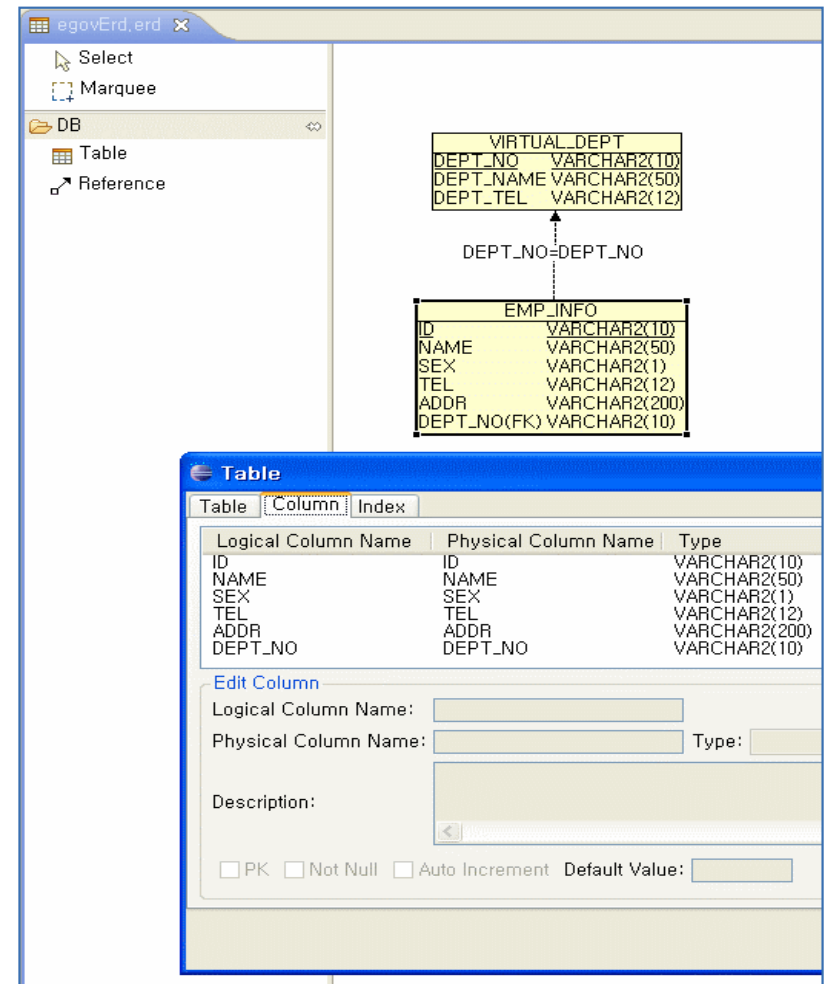
- Oracle, PostgreSQL, MySQL, hsqldb 등의 Database로부터 테이블 Import가 가능

### □ 테이블 명세서 Export

- 도구를 이용해 작성된 Table 들의 명세서를 HTML 형식 Export

### □ 제약사항

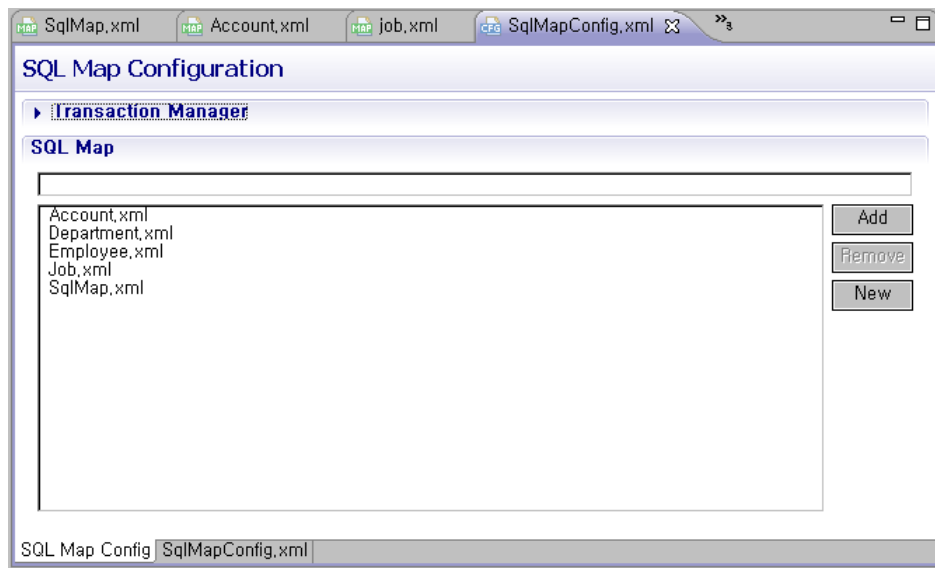
- ERWIN 모델링 도구와 호환 불가



DBIO Editor 는 iBatis SQL Map API 를 통해 DBIO 를 개발하고자 할 때 이를 쉽고 편리하게 작성할 수 있도록 다음과 같은 **Editor** 기능을 제공 (iBatis 전용)

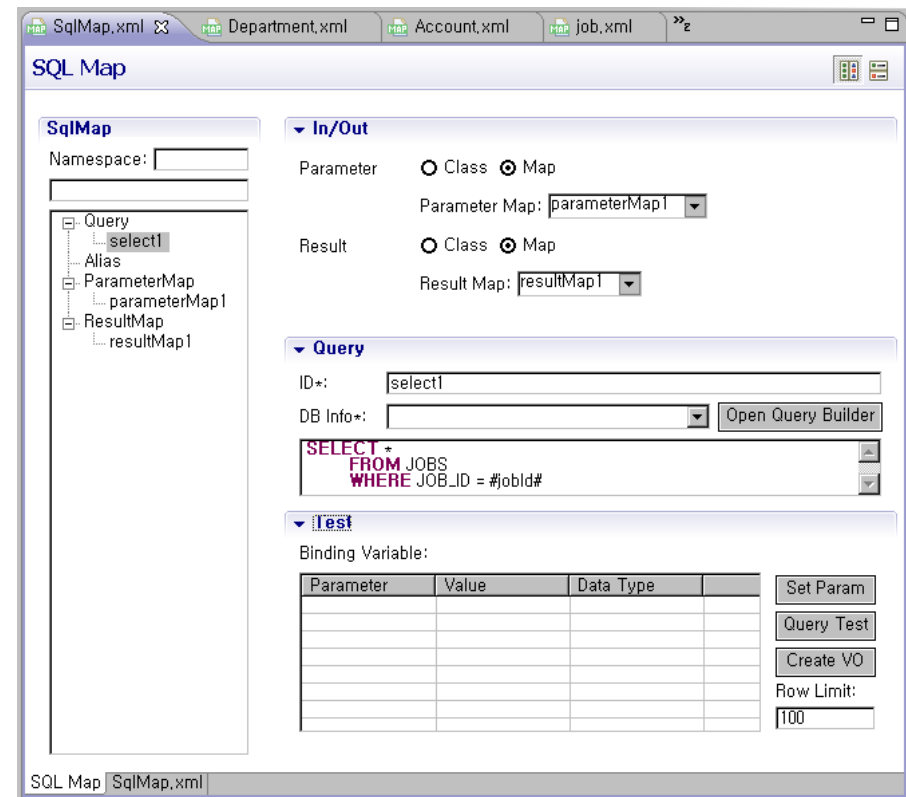
### ❑ SQL Map Config Editor

- 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.

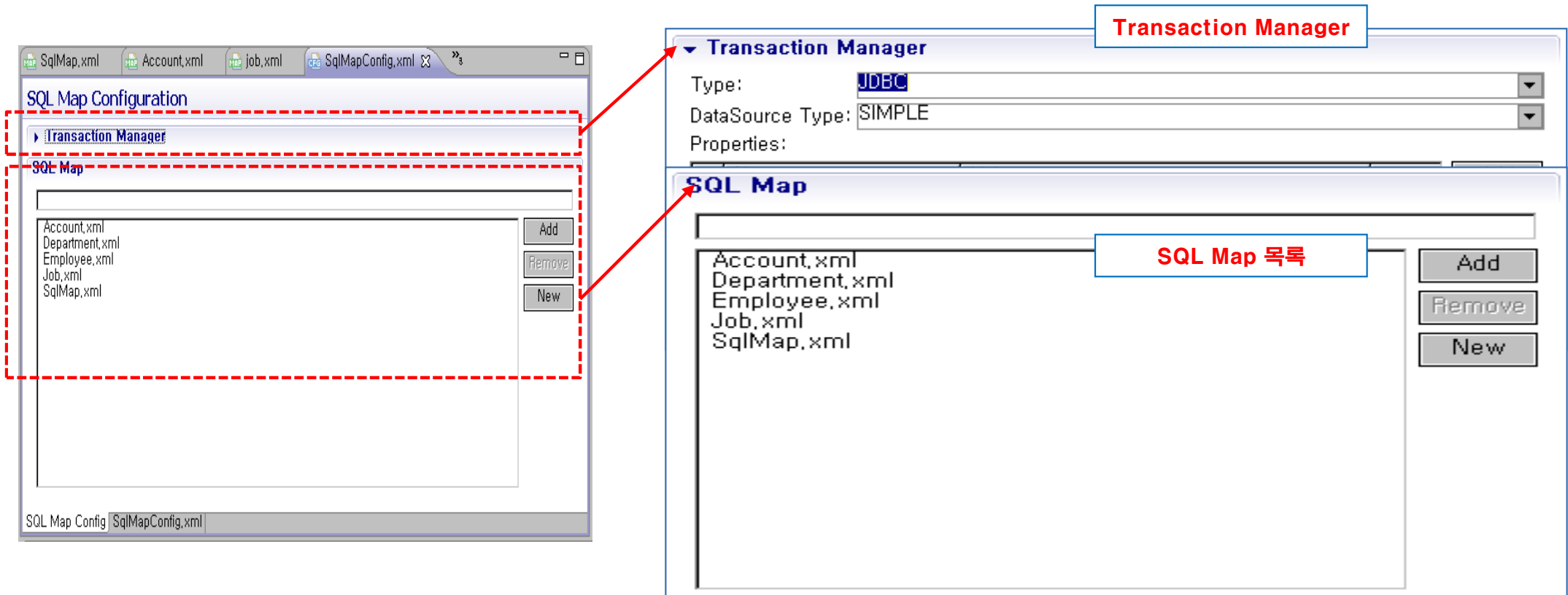
SqlMap Config Editor는 iBATIS SQL Maps 설정 파일을 UI 형태로 편집 가능하도록 지원

### ❑ 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 도구

### ❑ 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

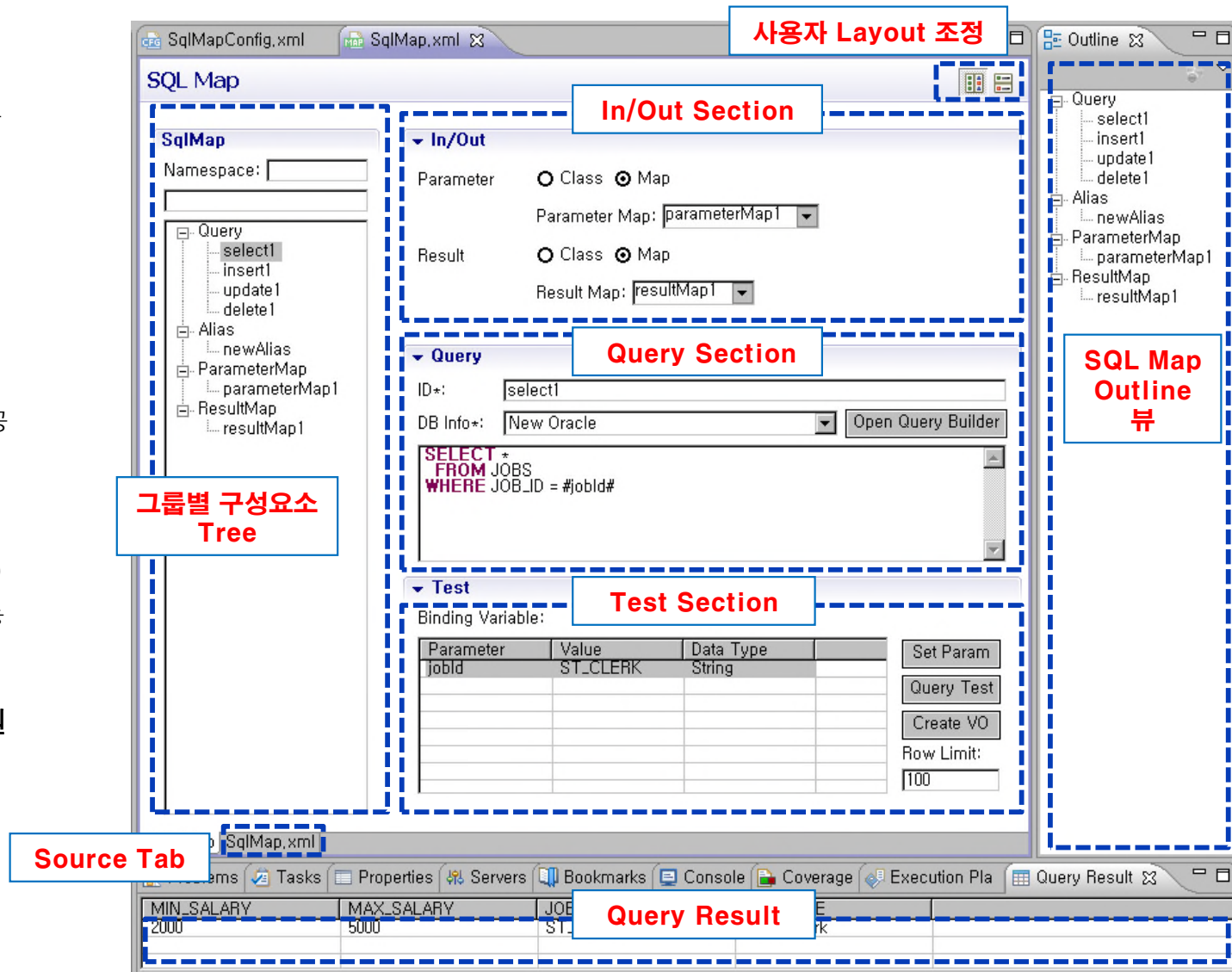
- 쿼리에 따른 Parameter Name 자동 설정
- Row Limit (쿼리 테스트 결과행 길이제한)
- 쿼리 Test 결과에 따른 VO 자동 생성 기능

### ❑ Query Result : 쿼리 수행 결과 확인

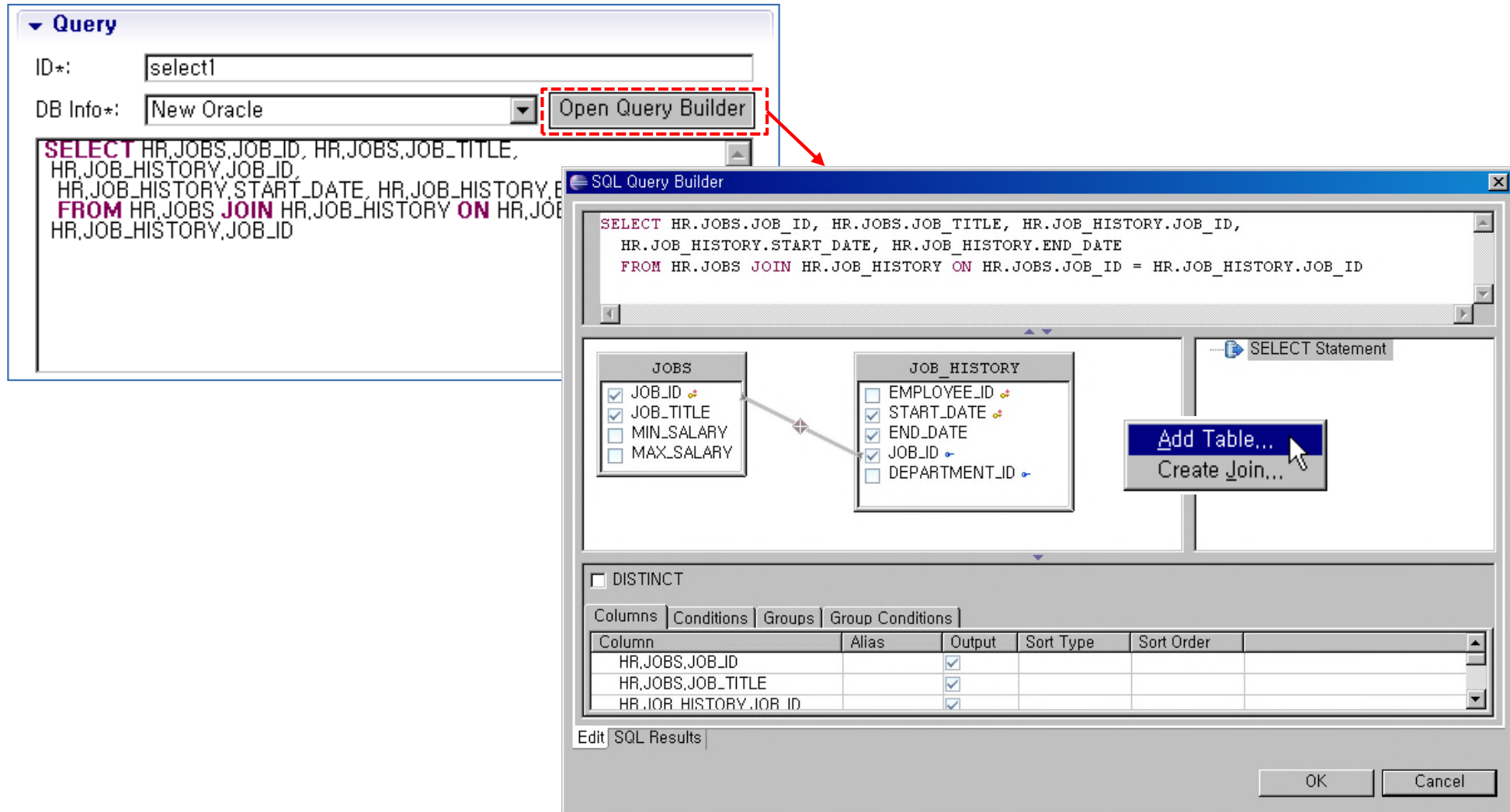
### ❑ Source Tab : SqlMap XML 편집 기능 지원

### ❑ SqlMap Outline View

- 구성요소 바로가기 기능 지원



SqlMap Editor에서는 쿼리 작성을 용이하게 하기 위해 **Query Builder** 기능을 제공함



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**

### ❑ 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)

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

SqlMap Editor는 개발자 편의성을 위해 VO(Value Object)를 자동 생성기능 제공

▼ Test

Binding Variable:

Parameter	Value	Data Type
SearchStr	PU	String

① “Create VO ” 클릭

Set Param

Query Test

Create VO

Row Limit:

100

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 Browse...

Package: com Browse...

Enclosing type: Browse...

Name: sampleVo

② VO Class명 입력

Finish Cancel

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

```

package com;

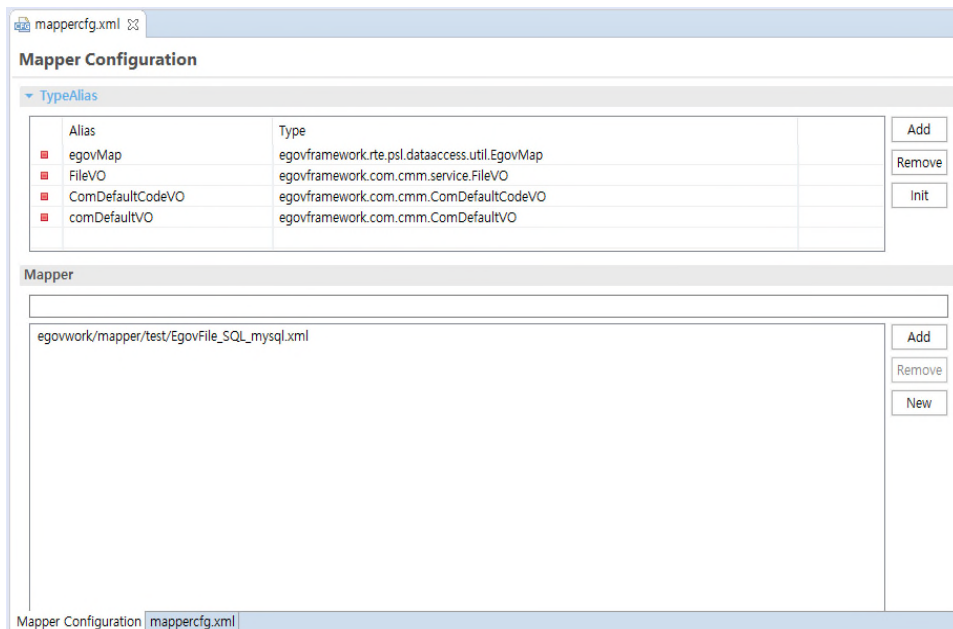
public class sampleVo {

    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;
    }
}
    
```

DBIO Editor 는 MyBatis API 를 통해 DBIO 를 개발하고자 할 때 이를 쉽고 편리하게 작성할 수 있도록 다음과 같은 Editor 기능을 제공 (MyBatis 전용)

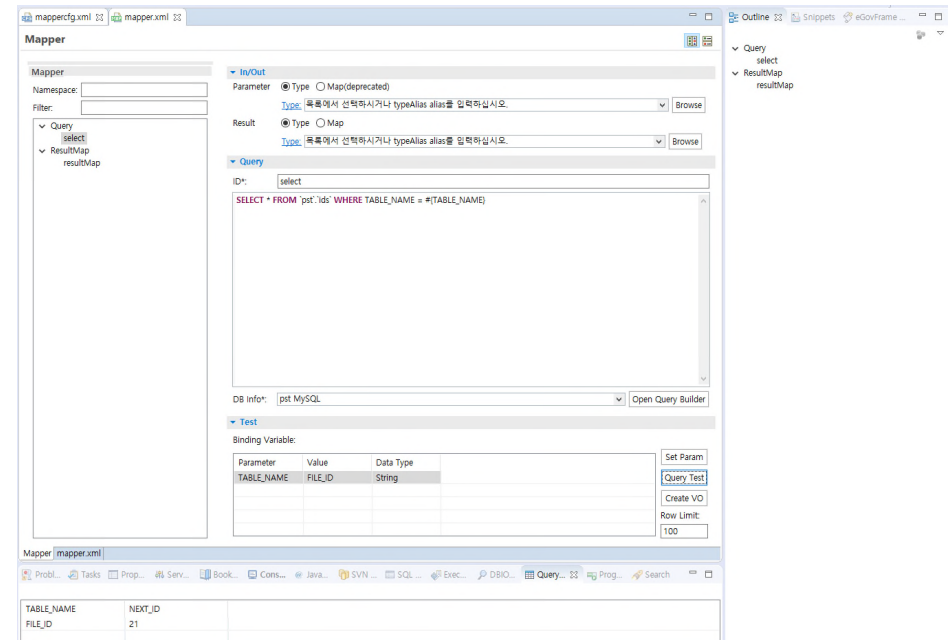
## ❑ Mapper Configuration Editor

- Mapper 설정 파일을 UI형태로 편집 가능하도록 지원



## ❑ Mapper Editor

- Mapper 파일을 UI형태로 편집 가능하도록 지원



\* MyBatis API 에서는 기본적으로 Mapper 파일과 Mapper 설정 파일이 요구된다.

Mapper Configuration Editor는 MyBATIS Mapper 설정 파일을 UI 형태로 편집 가능하도록 지원

### ❑ TypeAliases

- typeAlias alias 명과 type 속성을 정의한다.

### ❑ Mapper 목록

- Mapper 항목을 추가, 삭제, 신규 생성하고 Mapper Configuration 파일 내 등록을 지원한다.

The screenshot displays the Mapper Configuration Editor interface. It is divided into two main sections: 'TypeAlias' and 'Mapper'. The 'TypeAlias' section contains a table with columns 'Alias' and 'Type'. The 'Mapper' section contains a list of mapper files.

**TypeAlias**

Alias	Type
egovMap	egovframework.rte.psl.dataaccess.util.EgovMap
FileVO	egovframework.com.cmm.service.FileVO
ComDefaultCodeVO	egovframework.com.cmm.ComDefaultCodeVO
comDefaultVO	egovframework.com.cmm.ComDefaultVO

**Mapper**

Mapper
egovwork/mapper/test/EgovFile_SQL_mysql.xml

Red dashed boxes highlight the 'TypeAlias' and 'Mapper' sections in the main editor. Red arrows point from these sections to the detailed views shown on the right.

Mapper Editor는 MyBatis Mapper 파일을 UI 형태로 편집 가능하도록 지원하는 도구

### ❑ 그룹별 구성요소 Tree

- 구성요소 생성, 삭제, 검색, 바로가기 기능

### ❑ 사용자 Layout 조정

- 수평, 수직 레이아웃 선택 기능

### ❑ In/out Section

- Parameter, Result 설정(type, map 선택)

### ❑ Query Section

- SQL 쿼리 작성시 Query Builder 기능 제공

### ❑ Test Section

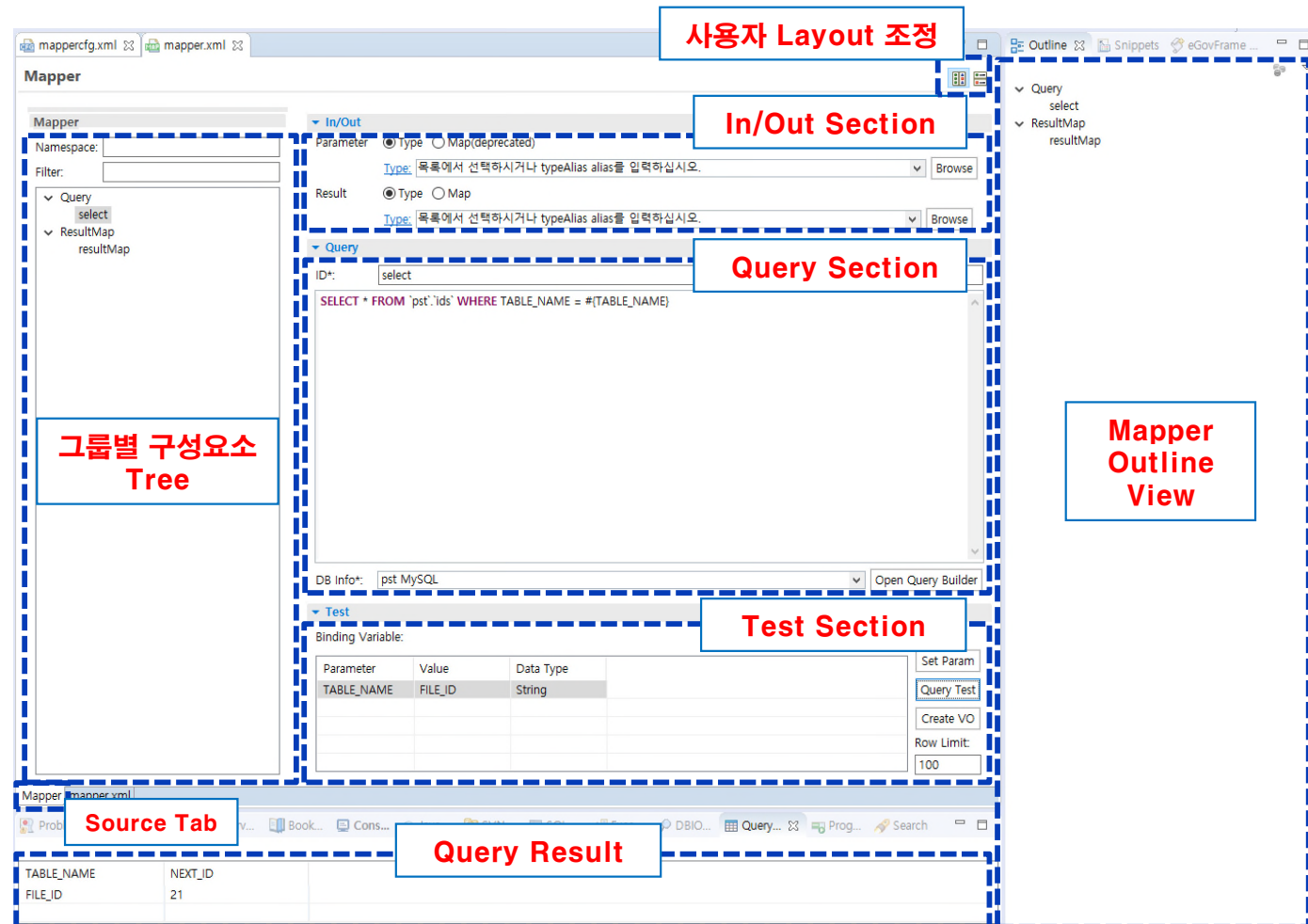
- 쿼리에 따른 Parameter Name 자동 설정
- Row Limit (쿼리 테스트 결과행 길이제한)
- 쿼리 Test 결과에 따른 VO 자동 생성 기능

### ❑ Query Result : 쿼리 수행 결과 확인

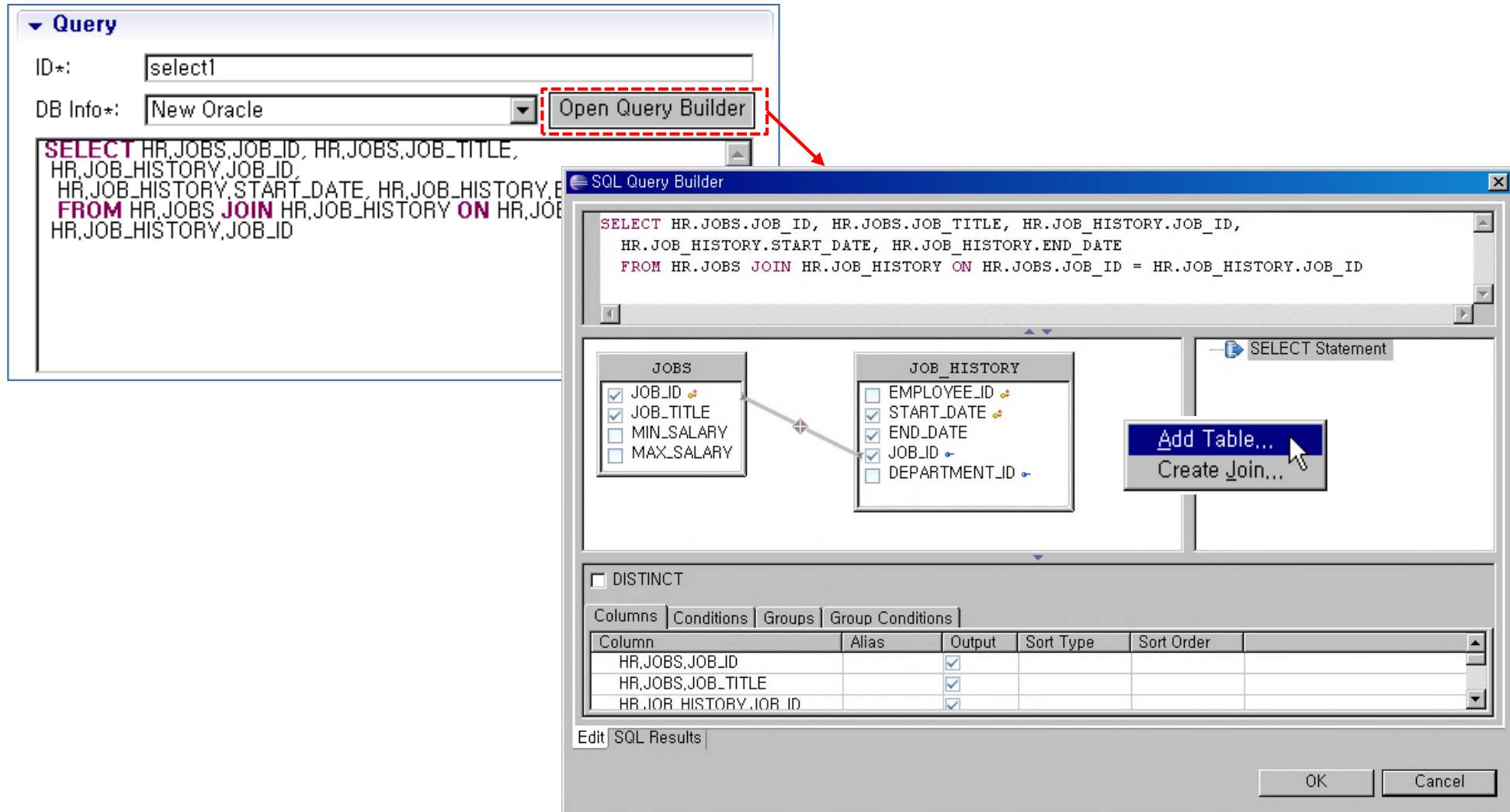
### ❑ Source Tab : Mapper XML 편집 기능 지원

### ❑ Mapper Outline View

- 구성요소 바로가기 기능 지원



Mapper Editor에서는 쿼리 작성을 용이하게 하기 위해 **Query Builder** 기능을 제공함



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

**Query Section**

ID\*: select

**② Query 작성**

```
SELECT *
FROM 'pst'.ids
WHERE TABLE_NAME = #{TABLE_NAME}
```

DB Info\*: pst MySQL **① DB Info 를 선택** Open Query Builder

**Test Section**

Binding Variable:

Parameter	Value	Data Type
TABLE_NAME	FILE_ID	String

**③ Parameter 값 입력**

Buttons: Set Param, Query Test, Create VO, Row Limit: 100

**④ Query Result 확인**

Query Result

TABLE_NAME	NEXT_ID
FILE_ID	21

## □ 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)

Mapper Editor는 개발자 편의성을 위해 VO(Value Object)를 자동 생성기능 제공

▼ Test

Binding Variable:

Parameter	Value	Data Type
SearchStr	PU	String

① “Create VO ” 클릭

Set Param

Query Test

Create VO

Row Limit:

100

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 Browse...

Package: com Browse...

☐ Enclosing type: Browse...

Name: sampleVo

② VO Class명 입력

Finish Cancel

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

```

package com;

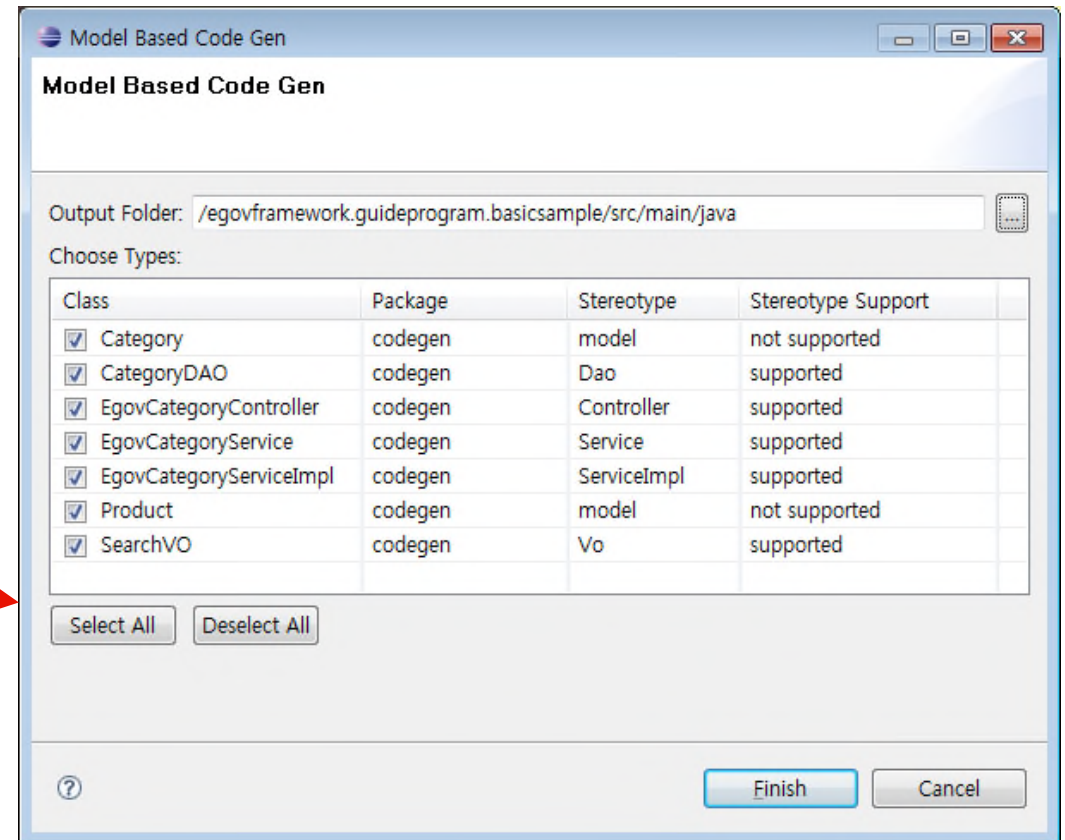
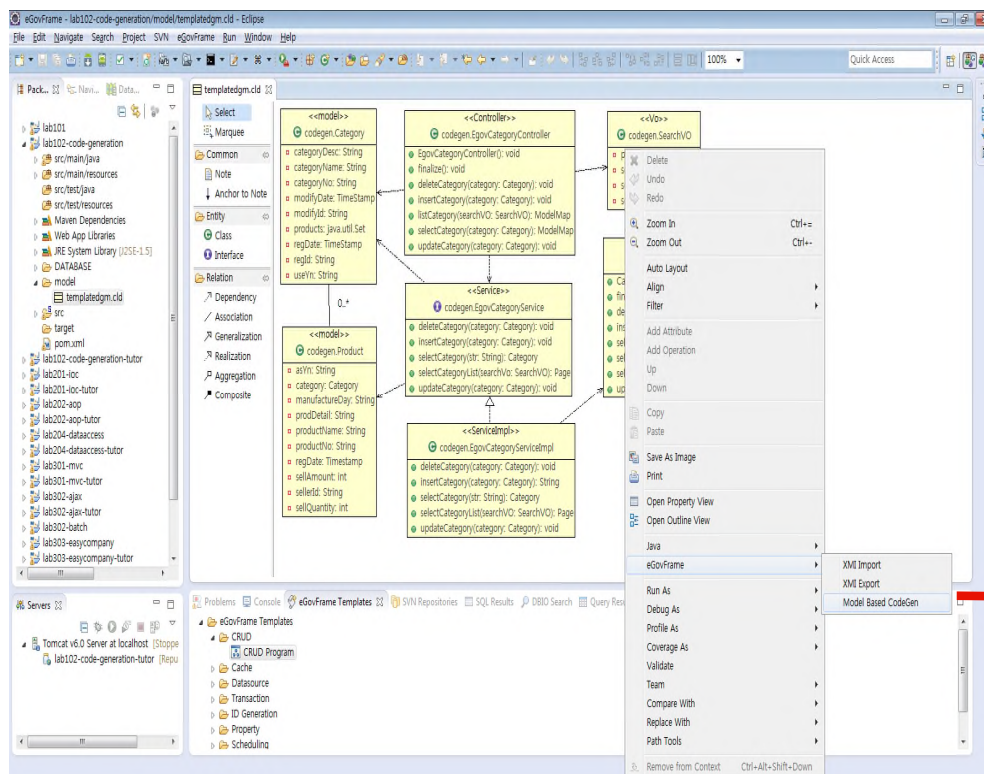
public class sampleVo {

    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 에디터 화면내에서 오른쪽 버튼 클릭하여 메뉴에서 “eGovFrame > Model Based CodeGen” 메뉴를 클릭한다.
3. Output Folder 입력 항목에 소스 디렉토리를 입력하고 “Finish” 버튼을 클릭한다.



## 7. eGovFrame Java Project

## 2.구현 도구

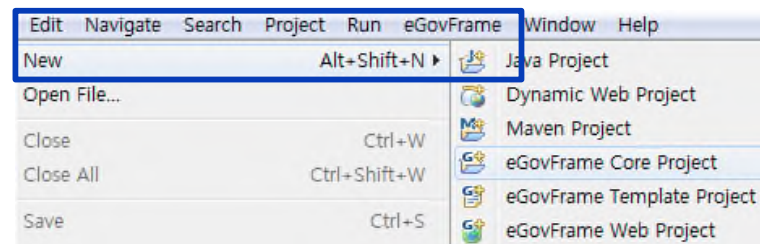
프로젝트 생성 마법사는 일반 혹은 **Web** 기반 프로젝트 자동 생성 기능을 제공

### ❑ eGovFrame Core Project

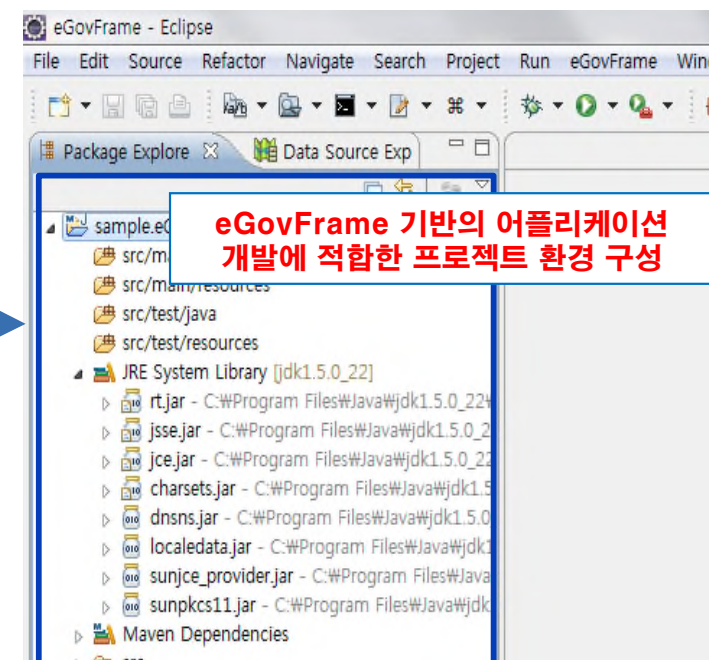
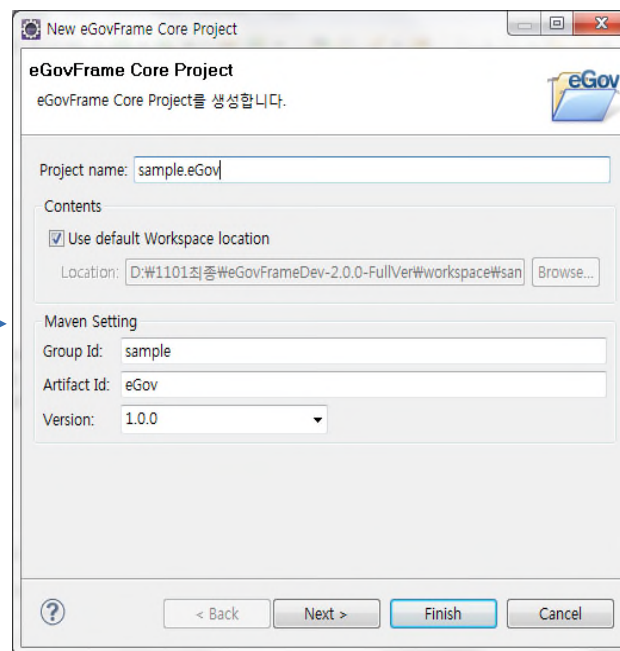
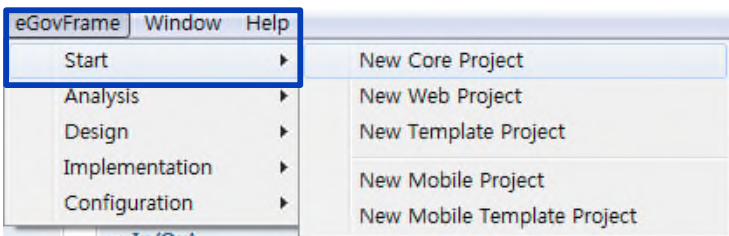
- 일반적인 **JAVA Project** 생성

### ❑ eGovFrame Web Project

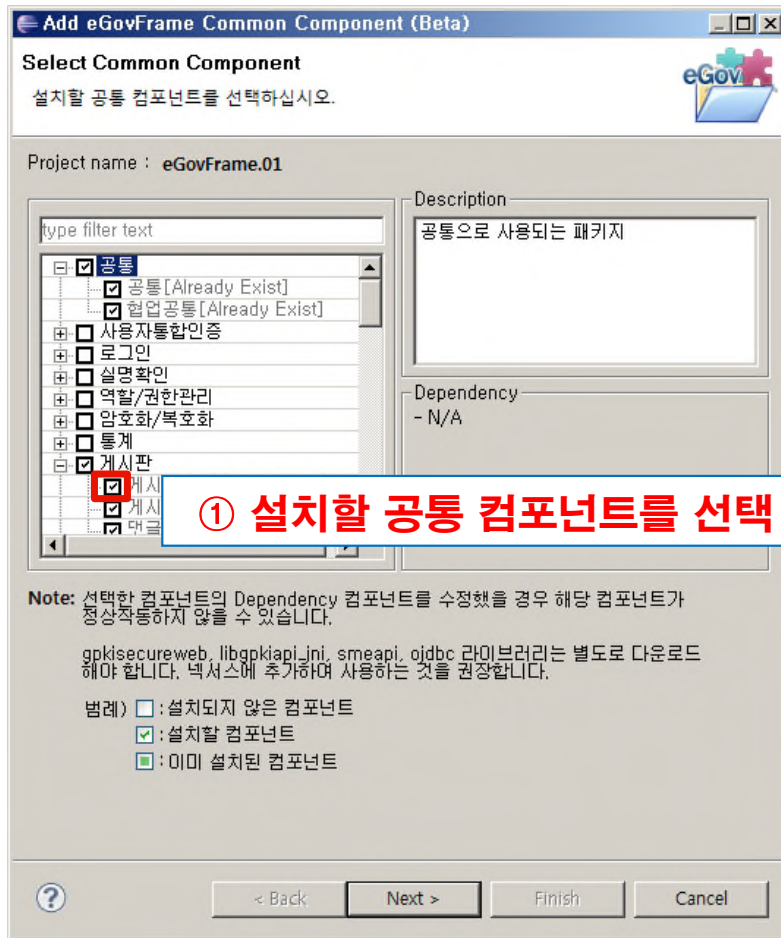
- **Web** 기반 프로젝트 생성



2가지 생성 방식



개발자의 편의성을 위하여 전자정부 표준 프레임워크에서 제공하는 **공통컴포넌트 251종**의 설치 마법사를 제공



**Add eGovFrame Common Component (Beta)**  
Select Common Component  
설치할 공통 컴포넌트를 선택하십시오.

Project name : eGovFrame.01

Description  
공통으로 사용되는 패키지

Dependency  
- N/A

type filter text

- ☒ 공통
  - ☒ 공통 [Already Exist]
  - ☒ 협업공통 [Already Exist]
  - ☐ 사용자통합인증
  - ☐ 로그인
  - ☐ 실명확인
  - ☐ 역할/권한관리
  - ☐ 암호화/복호화
  - ☐ 통계
  - ☐ 게시판
  - ☐ 게시판
  - ☐ 댓글

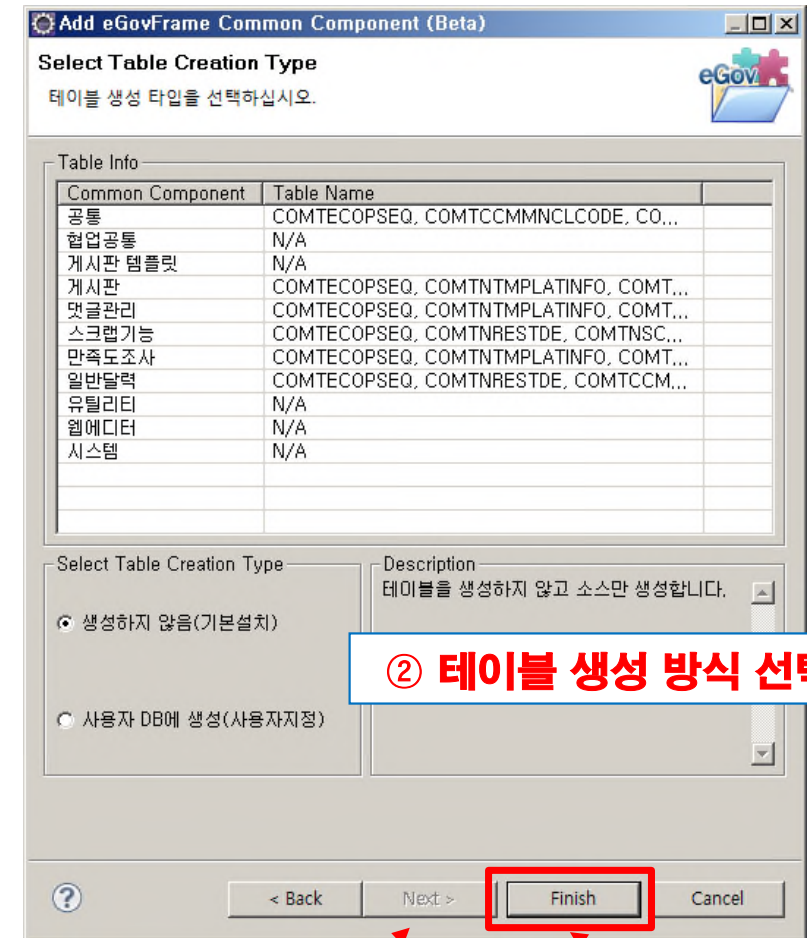
Note: 선택한 컴포넌트의 Dependency 컴포넌트를 수정했을 경우 해당 컴포넌트가 정상작동하지 않을 수 있습니다.

gpkisecureweb, libgpkapi.jni, smeapi, ojdbc 라이브러리는 별도로 다운로드 해야 합니다. 넥서스에 추가하여 사용하는 것을 권장합니다.

범례) ☐ : 설치되지 않은 컴포넌트  
☒ : 설치할 컴포넌트  
☐ : 이미 설치된 컴포넌트

< Back Next > Finish Cancel

① 설치할 공통 컴포넌트를 선택



**Add eGovFrame Common Component (Beta)**  
Select Table Creation Type  
테이블 생성 타입을 선택하십시오.

Table Info

Common Component	Table Name
공통	COMTECOPSEQ, COMTCCMMNCLCODE, CO...
협업공통	N/A
게시판 템플릿	N/A
게시판	COMTECOPSEQ, COMTNTMPLATINFO, COMT...
댓글관리	COMTECOPSEQ, COMTNTMPLATINFO, COMT...
스크랩기능	COMTECOPSEQ, COMTNRESTDE, COMTNSC...
만족도조사	COMTECOPSEQ, COMTNTMPLATINFO, COMT...
일반달력	COMTECOPSEQ, COMTNRESTDE, COMTCCM...
유틸리티	N/A
웹에디터	N/A
시스템	N/A

Select Table Creation Type

☒ 생성하지 않음(기본설치)  
☐ 사용자 DB에 생성(사용자지정)

Description  
테이블을 생성하지 않고 소스만 생성합니다.

< Back Next > **Finish** Cancel

② 테이블 생성 방식 선택

사용자지정 선택 시  
Next 버튼 활성화

기본설치 선택 시  
Finish 버튼 활성화

개발자의 편의성을 위하여 전자정부 표준 프레임워크에서 제공하는 **공통컴포넌트 251종**의 설치 마법사를 제공

**① DB 선택**

Select DB: New Oracle

Datasource

DB Type: oracle

Driver Class Name: oracle.jdbc.OracleDriver

Password: ●●●●

**② Connection Test 수행**

Connection Test Create Table

**③ 테이블 생성**

Create Table

Table Name	Table Install Result	Table Description
COMTECOPSEQ	Success	공통, 게시판, 댓글관리, ...
COMTCCMMNCLCODE	Success	공통, 게시판, 댓글관리, ...
COMTCCMMNCODE	Success	공통, 게시판, 댓글관리, ...
COMTCCMMNDETAILCODE	Success	공통, 게시판, 댓글관리, ...
COMTNFILE	Success	공통, 게시판, 댓글관리, ...
COMTNFILEDETAIL	Success	공통, 게시판, 댓글관리, ...
COMTNFILEPLATINFO	Success	공통, 게시판, 댓글관리, ...
COMTNRESTDE	Success	공통, 게시판, 댓글관리, ...

**④ 테이블 생성 결과 확인**

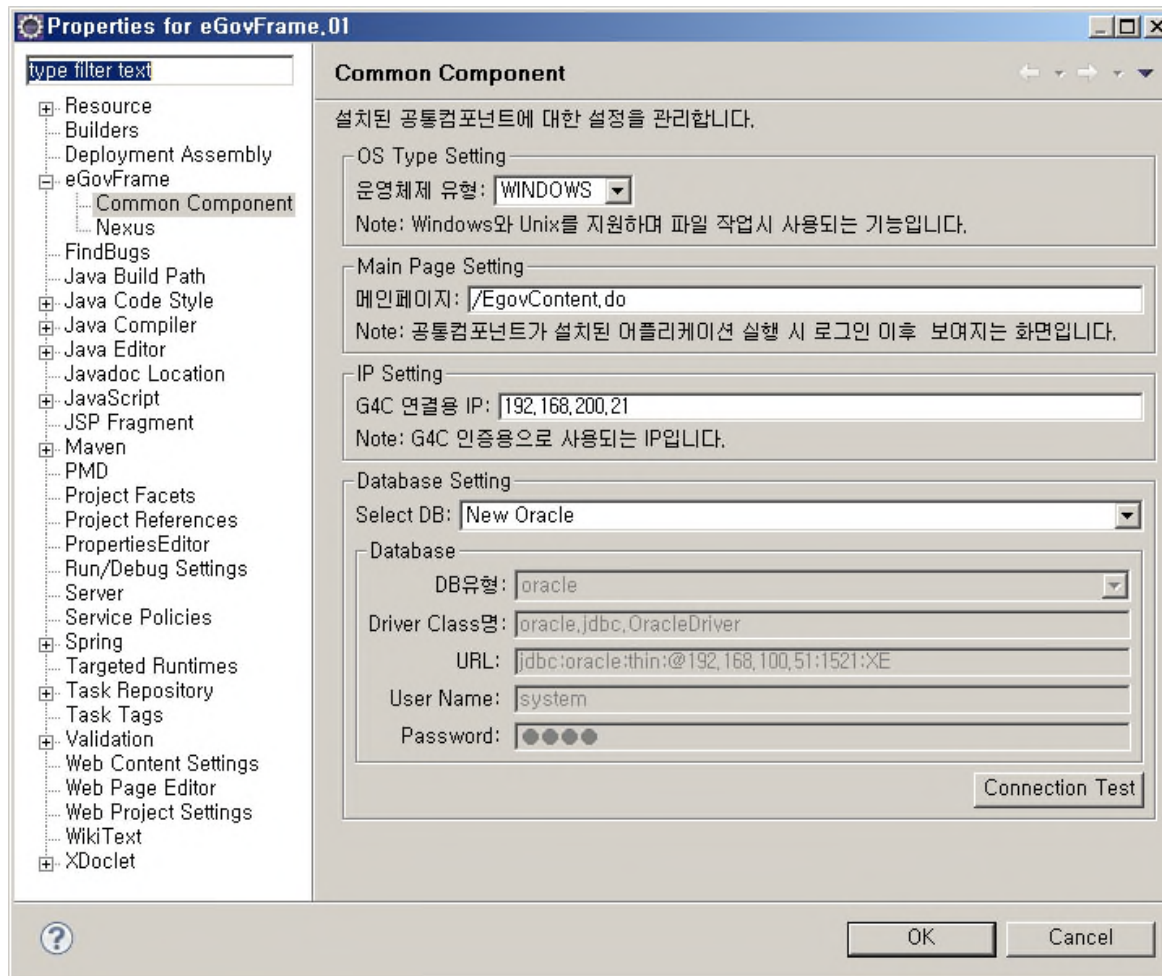
< Back Next > **Finish** Cancel

### □ 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 251종**의 설치 마법사를 제공



### □ 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.

배치 작업 생성 도구를 통해 **Job, Step** 등 일괄처리를 위한 기본 기능부터 **Flow Control, Partitioning** 등 확장 기능까지 설정 값 입력만으로 사용자가 원하는 배치 작업 파일을 생성

The screenshot shows the 'New eGovFrame Batch Job File' dialog box. It is divided into several sections. On the left, there are two tabs: 'testJob' and 'testStep'. The 'testJob' tab is selected. Below the tabs, there are 'Add' and 'Remove' buttons. The main area is titled 'Info' and contains the following fields: 'Job ID' (testJob), 'Step Type' (Normal selected, Partition(File) unselected), 'Step ID' (testStep), and 'Item Configuration'. The 'Item Configuration' section has two rows: 'Reader' (testJob.testStep.fixedFlatFileItemReader) and 'Writer' (testJob.testStep.FormatterFlatFileItemWriter), each with an 'Add' button. Below these is 'Commit-Interval' (2) and an 'Advanced >>' button. At the bottom, there are 'Restore Defaults' and 'Apply' buttons. The bottom-most bar contains a question mark icon, '< Back', 'Next >', 'Finish', and 'Cancel' buttons.

① 배치 Job 정보 생성

② 배치 Step 정보 생성

③ Reader/Writer 정보 생성

④ 배치 작업 파일 생성

### □ 배치 작업 생성 절차

- ① 배치 Job 정보를 생성한다.
- ② Batch Step 정보를 생성한다.
- ③ Reader/Writer 정보를 생성한다.
- ④ 배치 작업 파일을 생성한다.

배치 실행 생성 도구를 통해 Job Launcher, Job Operator 등 배치 실행 정보를 생성하고, 필요 시 사용자가 현재 사용 중인 DB정보를 배치작업에서 활용할 수 있도록 하는 기능을 제공

The screenshot shows the 'New eGovFrame Batch Job Launcher File' dialog box. It is divided into three main sections, each highlighted with a red box and a numbered label:

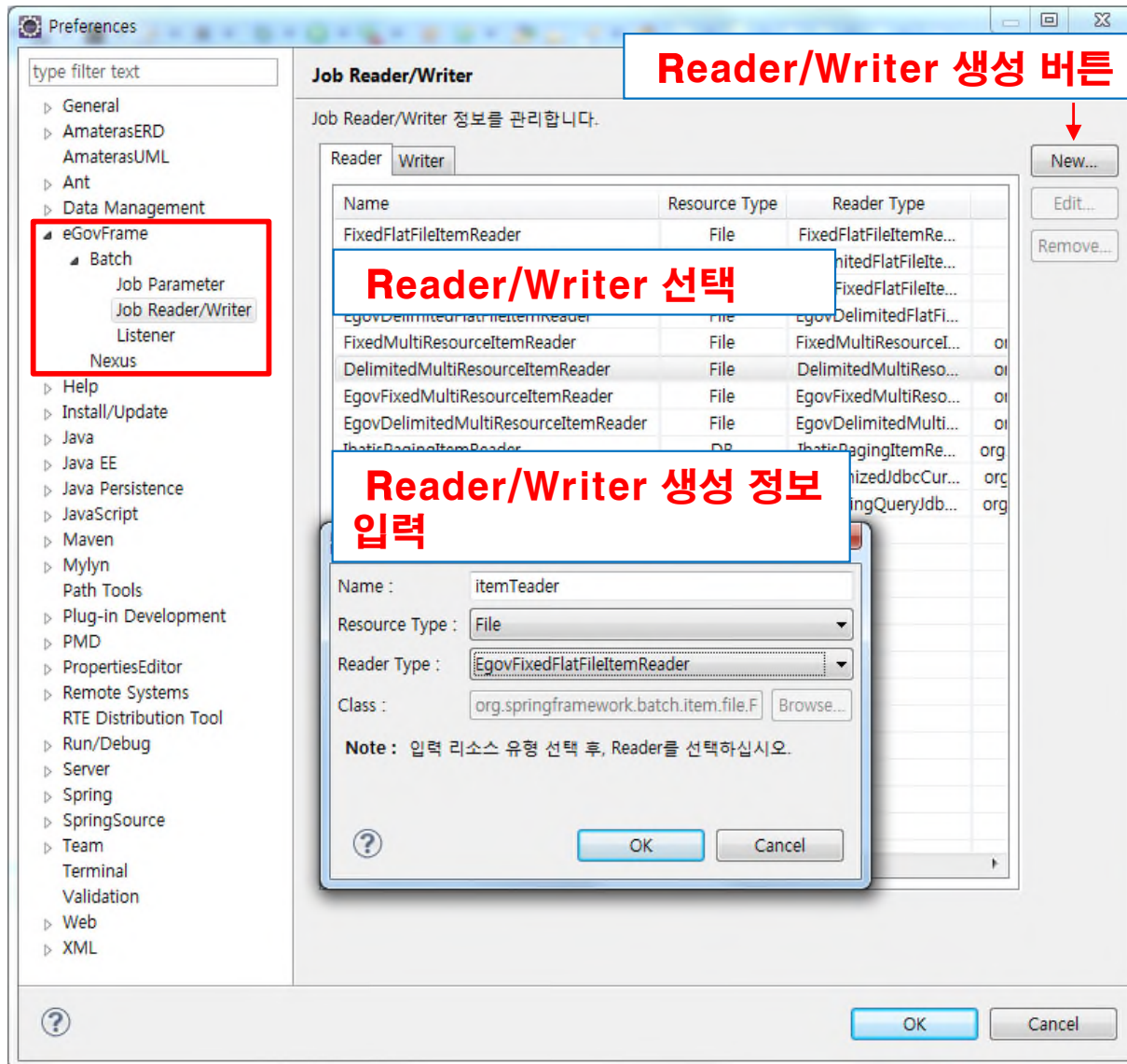
- ① 배치 Job Execution 정보 생성**: This section contains fields for 'Launcher ID' (jobLauncher), 'Execution Type' (Synchronous selected, Asynchronous unselected), 'Operator ID' (jobOperator), 'Explorer ID' (jobExplorer), and 'Register ID' (jobRegistry).
- ② DataSource 정보 생성**: This section is for database configuration. It includes 'Repository Type' (DB(Reference) selected, DB(N) unselected), 'Database' (Datasource Bean ID with a 'Browse...' button), and 'Datasource' (Datasource Name, DB Type, Driver Class Name, URL, User name, and PassWord).
- ③ 배치 실행 파일 생성**: This section is at the bottom and contains the 'Finish' button, which is highlighted with a red box.

At the bottom of the window, there are navigation buttons: '< Back', 'Next >', 'Finish', and 'Cancel'.

### □ 배치 실행 작업 생성 절차

- ① 배치 Job Execution 정보를 생성한다.
- ② DataSource 정보를 생성한다.
- ③ 배치 실행 파일을 생성한다.

배치개발환경 사용시 Job Parameter, Job Reader/Writer, Listener를 직관적으로 등록 및 사용하기 위한 관리 환경을 제공



### □ Job Parameter

- 배치개발환경에서 제공하는 테스트 마법사를 이용할 경우 사용자의 편의성을 위해 배치 구성 파라미터를 추가하는 관리 환경을 제공

### □ Job Reader/Writer

- 배치 수행 시 사용되는 입력 데이터를 등록하여 사용할 수 있는 관리 환경을 제공

- 배치 수행 시 사용되는 출력 데이터를 등록하여 사용할 수 있는 관리 환경을 제공

### □ Listener

- 배치 수행 시 각 Job, Step, Chunk의 Pre 혹은 Post에서 다양한 이벤트를 잡아낼 수 있는 Listeners 관리 환경을 제공

유형별 템플릿 프로젝트를 자동 생성하는 기능을 제공

### □ 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

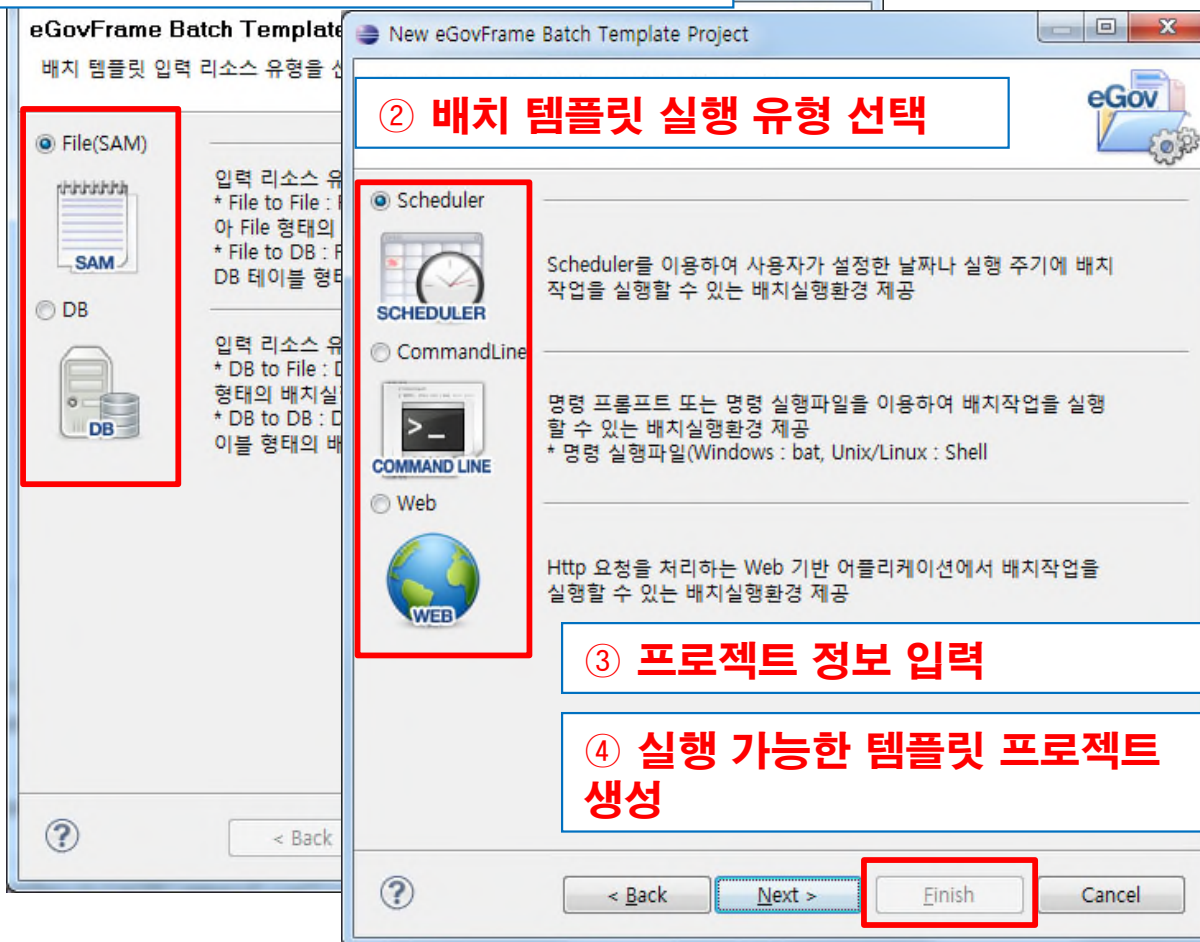
### □ Common All-in-one

- 공통컴포넌트 251종의 모든 기능 제공



배치 템플릿 생성도구를 실행한 후, 입력 리소스 유형과 실행 유형의 선택만으로 총 6종류의 예제 프로젝트를 생성

### ① 배치 템플릿 입력 리소스 유형 선택



### □ 배치 실행 작업 생성 절차

- ① 배치 Job Execution 정보를 생성한다.
- ② DataSource 정보를 생성한다.
- ③ 배치 실행 파일을 생성한다.

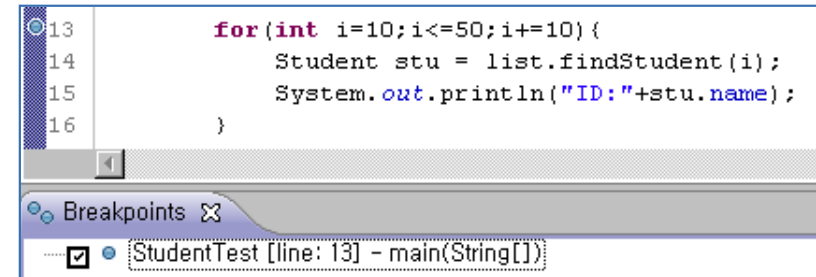
### ③ 프로젝트 정보 입력

### ④ 실행 가능한 템플릿 프로젝트 생성

로컬 또는 원격으로 실행 중인 프로그램에서 오류를 발견하고 진단할 수 있는 디버거 제공

### ❑ 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

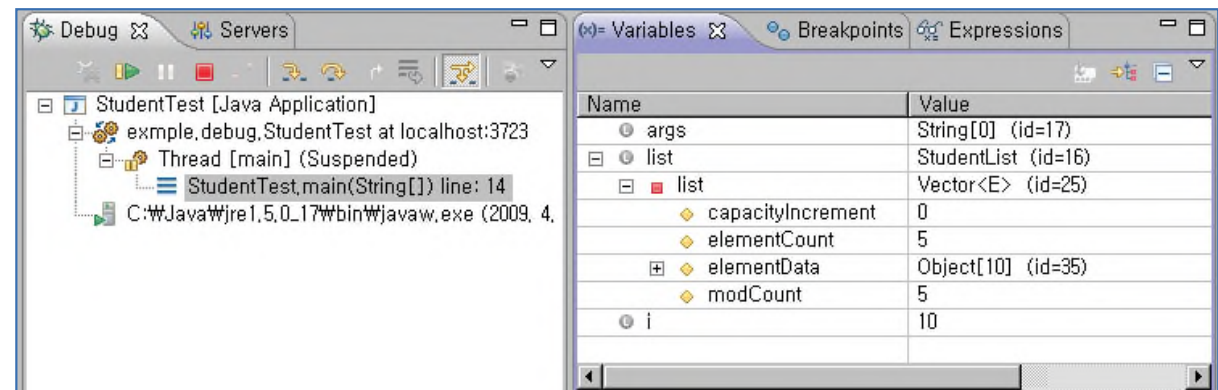
- 필터링 대상 프로그램의 경우 Step Over와 같이 동작하고, 필터링 대상이 아니면 Step Into와 같이 동작한다. Debug View의 Use Step Filters 버튼을 눌러 활성화

### ❑ 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



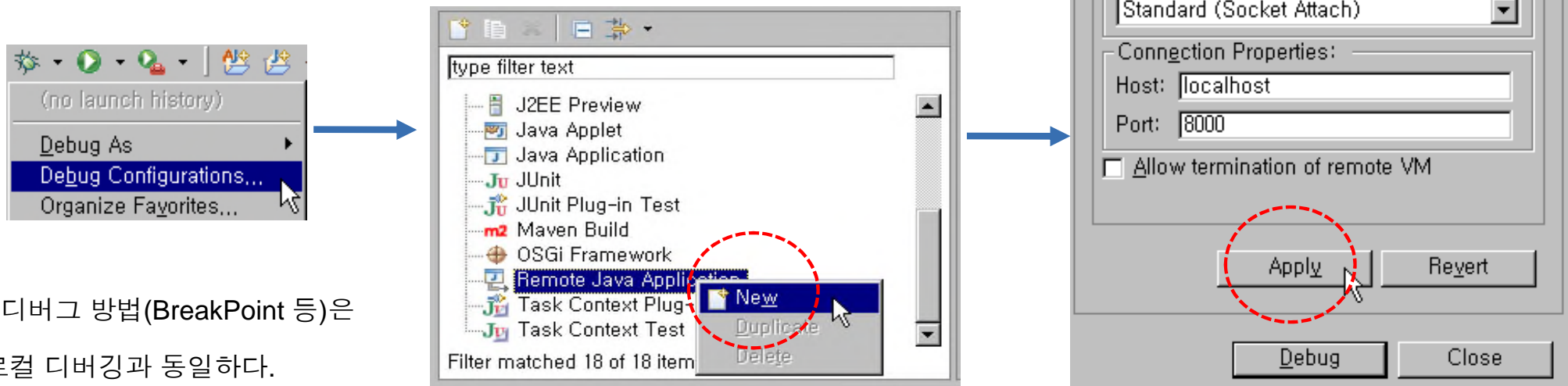
원격 컴퓨터의 **WAS(Jboss, JEUS, WebLogic)**에서 실행되는 어플리케이션을 디버깅

### ❑ 디버깅 환경 설정

- WAS 환경설정 파일에 “-Xdebug -Xnoagent -Xrunjdwp: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.

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. The main editor displays the SampleDAO.java file. The PMD perspective includes a Violations Outline and a Violations Overview table.

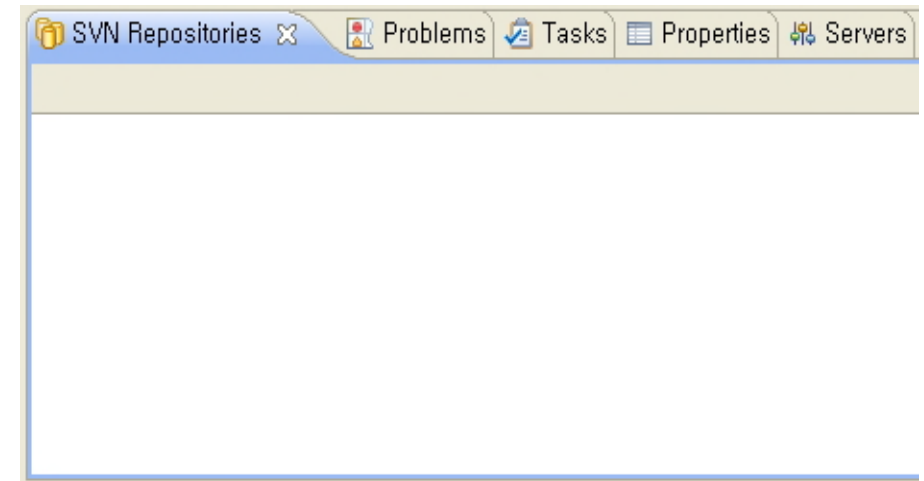
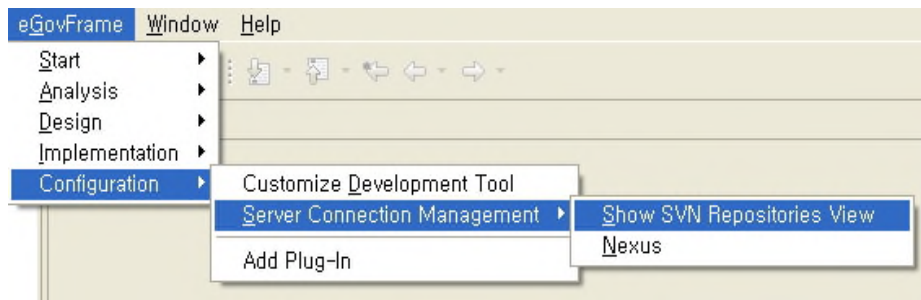
**Violations Overview Table:**

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	714.2 / 1000	1.17

eGovFrame 통합메뉴를 통해 SVN Repositories View를 연동하는 기능을 제공한다.

### ❑ SVN Repositories View

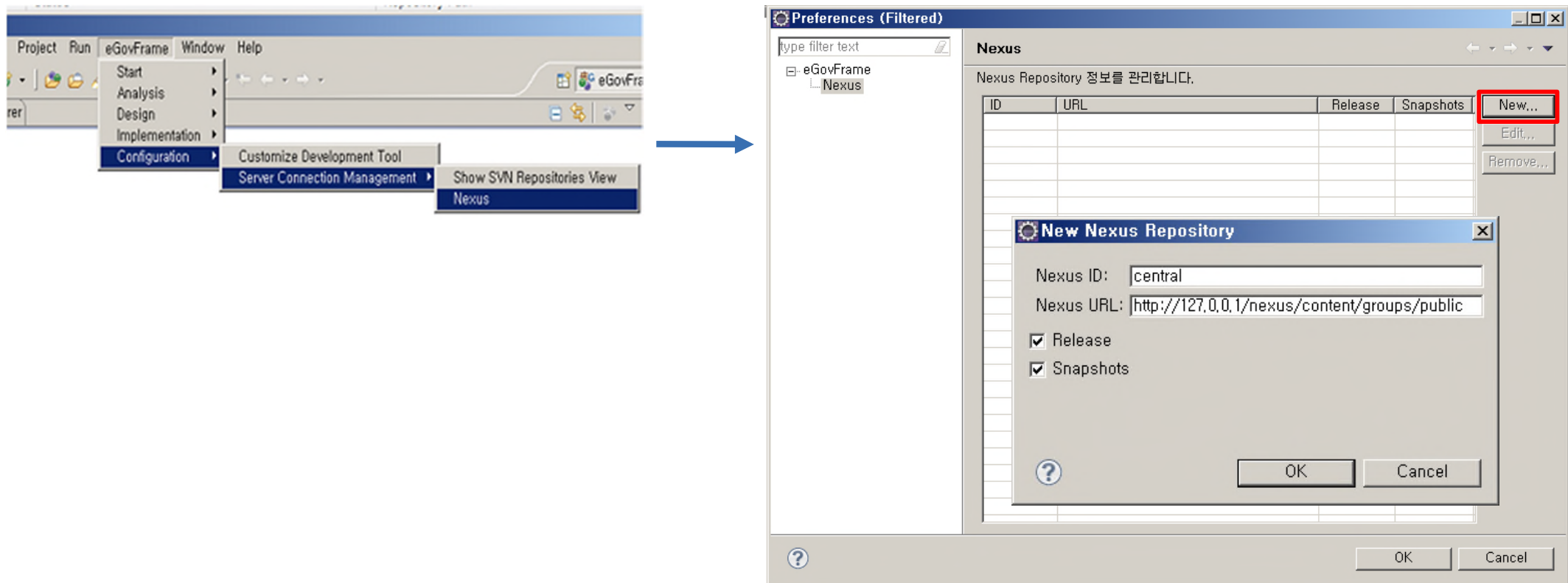
1. eGovFrame 통합메뉴 > Configuration > Server Connection Management > Show SVN Repositories View 를 클릭한다.
2. 화면 하단에 SVN Repositories라는 View가 열리는 것을 확인한다.



Nexus Repository를 등록, 수정 및 삭제하는 기능을 제공

### ❑ Nexus(Preferences)

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

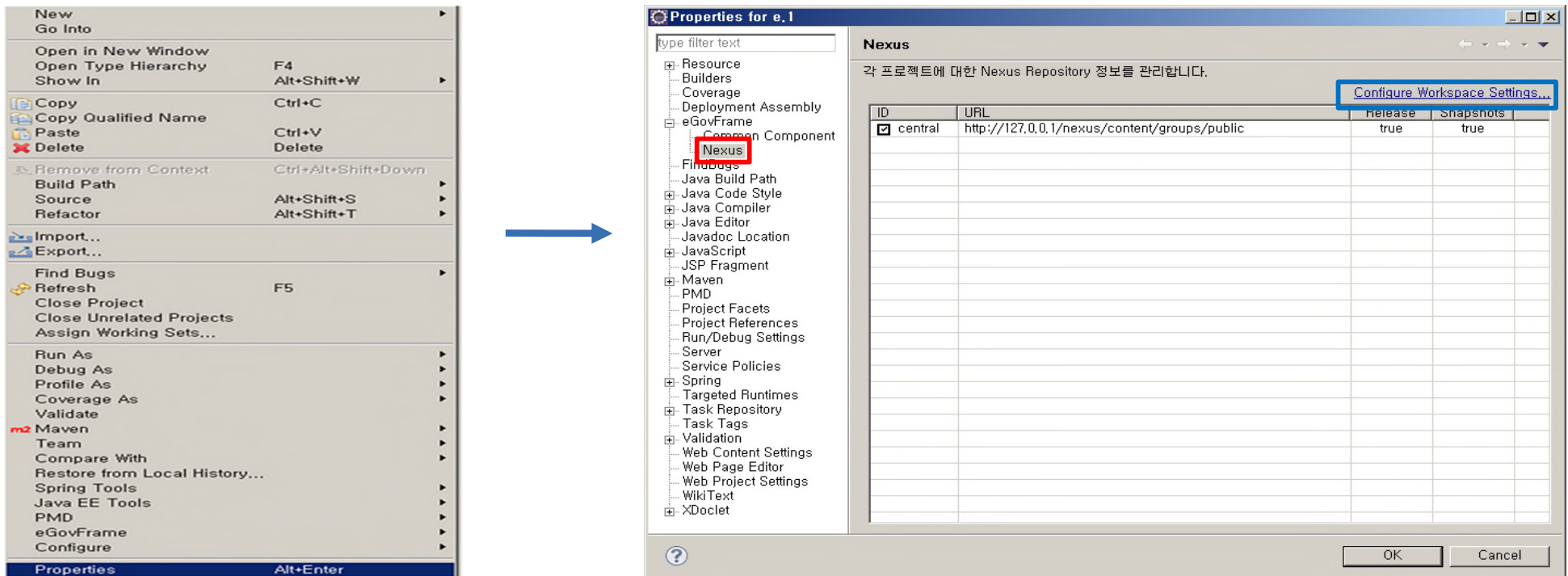


등록된 Nexus Repository 정보를 대상 프로젝트에 적용하는 기능을 제공

### □ Nexus(Properties)

1. 대상이 되는 프로젝트를 선택 후 우클릭하여 Properties를 선택한다.
2. Properties 페이지에서 eGovFrame > Nexus를 선택한다.
3. 기 설정된 Nexus 정보 중 대상 프로젝트에서 사용할 repository를 선택(체크박스) 후 OK버튼을 클릭하여 적용한다.

※ 등록된 Nexus Repository가 없을 경우, Configure Workspace Settings 링크를 클릭하여 나타나는 Preferences 창에서 Nexus Repository를 추가할 수 있다.



### ☐ Eclipse Home

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

### ☐ Amateras UML

- [http://amateras.sourceforge.jp/cgi-bin/fswiki\\_en/wiki.cgi?page=AmaterasUML](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](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

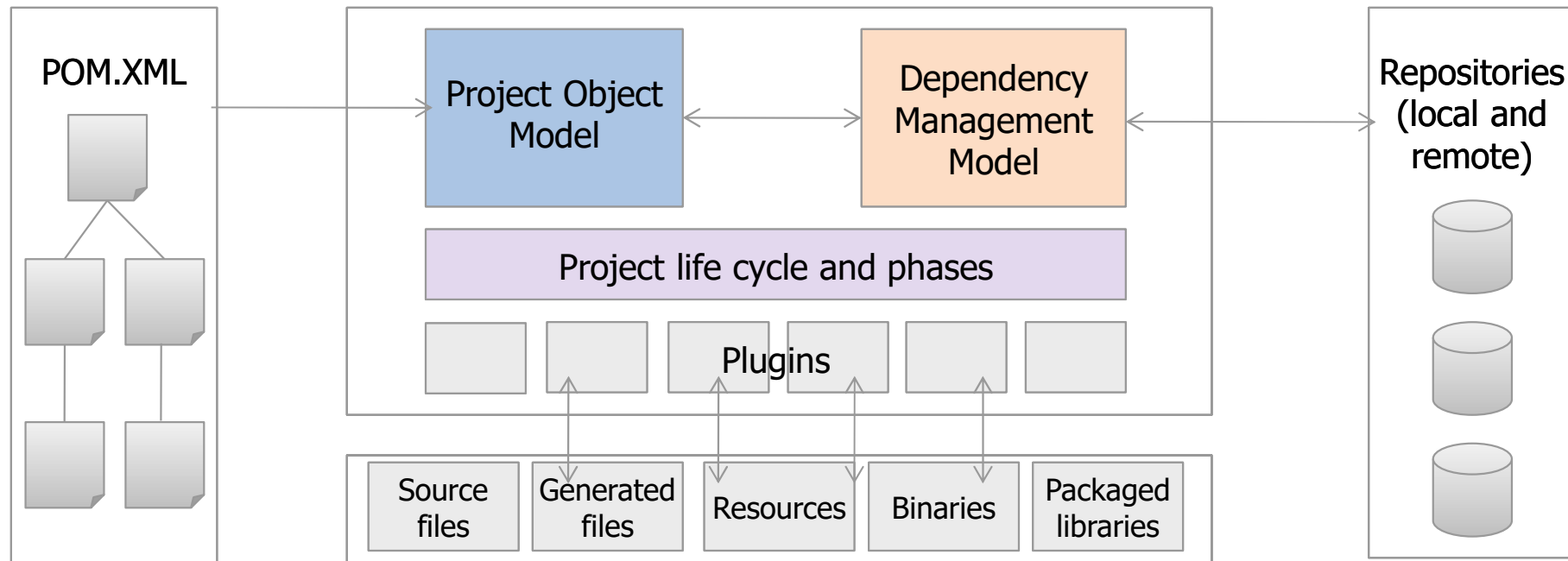
불필요한 설정을 최소화 한다는 개념 아래 **Ant**와 같은 빌드 기능을 제공할 뿐 아니라 구조화 된 빌드 기능을 통해 **learning curve** 및 재사용 성을 향상 시킴

## □Maven 특징

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

의존성 관리, 라이브러리 관리, 빌드 생명 주기 및 단계 객체 모델을 갖는 [프로젝트 관리 도구](#)로 표준화된 빌드 기능 뿐만 아니라 리포팅 및 **documentation** 생성 기능 등을 제공

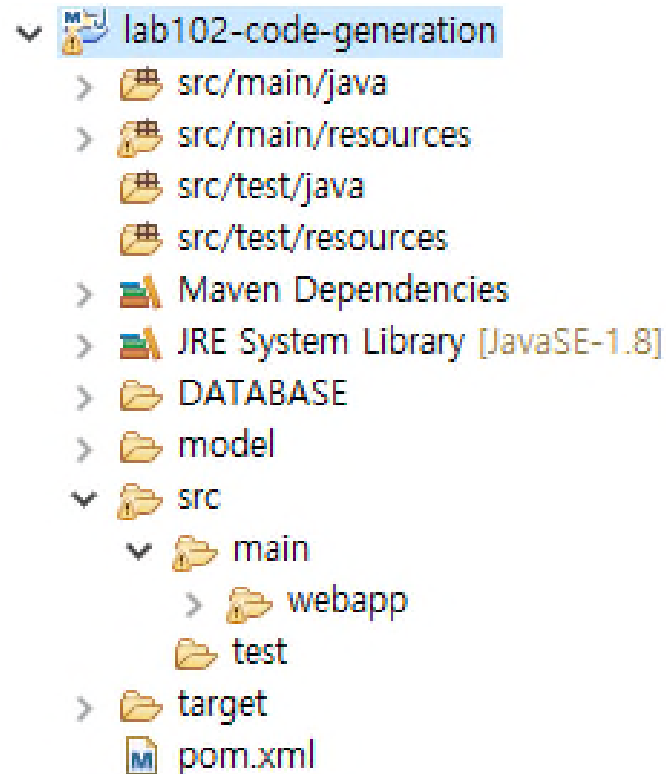
## □Maven 아키텍처



- **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

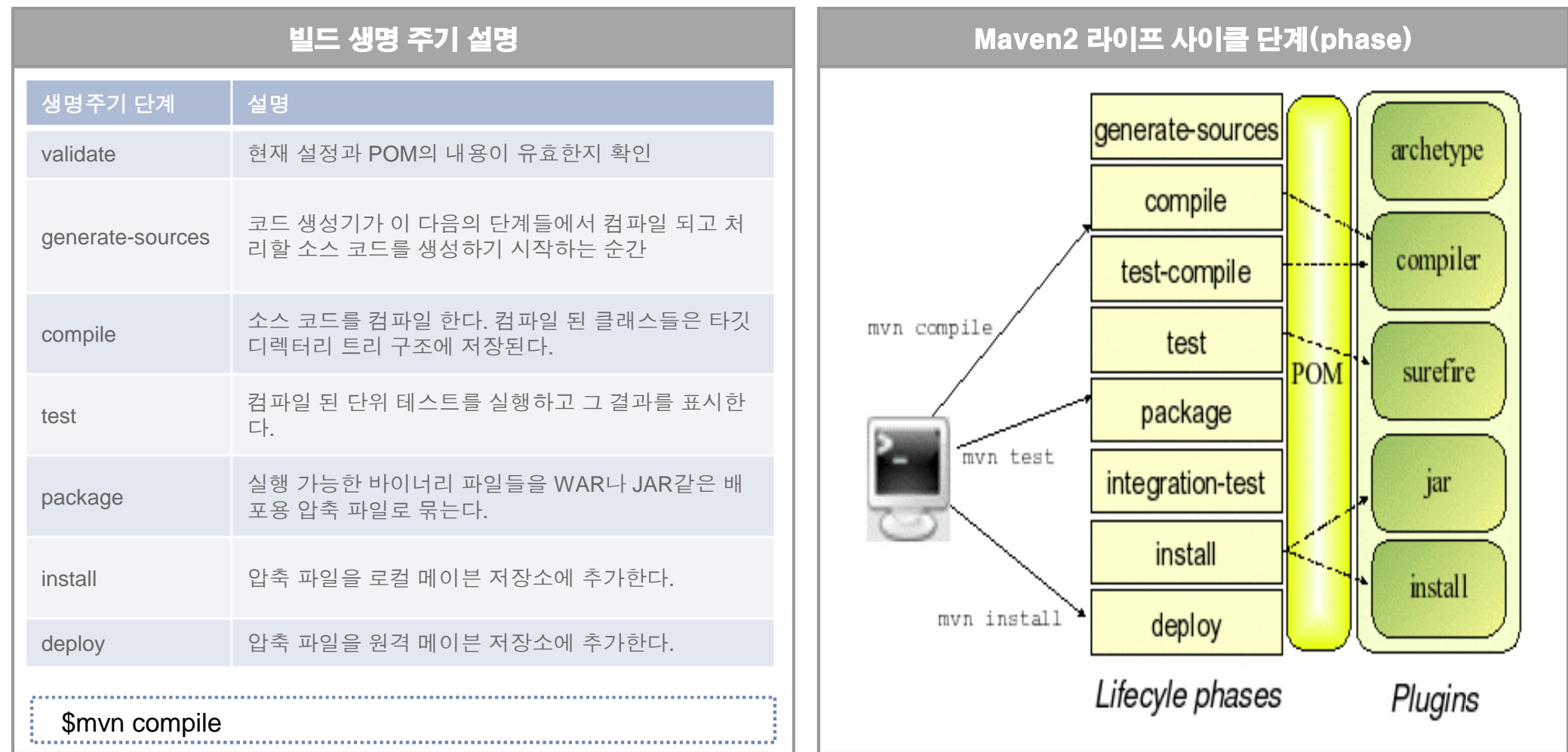
**Best practices**을 기반으로 정규화된 디렉터리 구조를 제공하고 있으며 모든 소스 파일들은 **/src** 디렉터리 밑에 빌드 된 **output**은 **/target** 디렉터리 밑에 위치함

### Maven 표준 디렉터리 구조

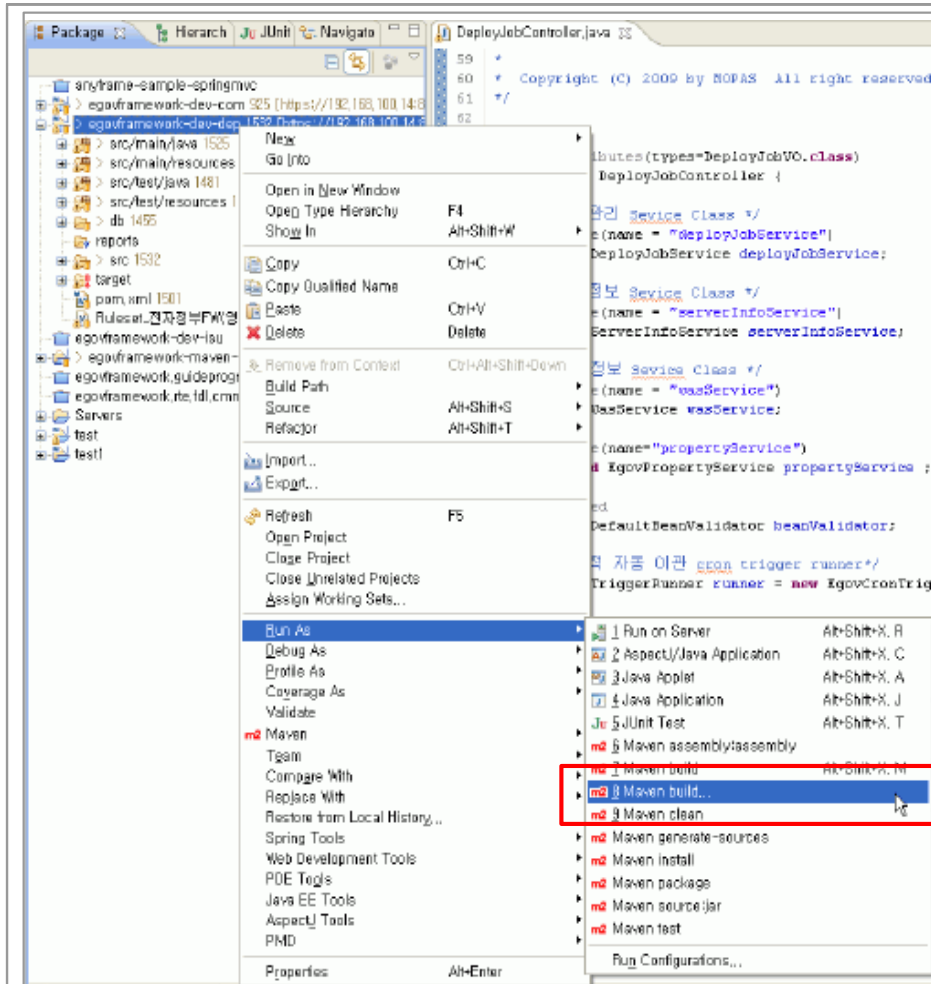


디렉터리/파일	설명
/pom.xml	프로젝트 객체 모델. 해당 프로젝트에 대한 전반적인 정보를 갖는다.
/src/main/java	Java 소스 파일 위치
/src/main/resources	배포할 리소스, XML, properties, ...
/src/main/webapp	웹 어플리케이션 관련 파일 위치(WEB-INF, css 등)
/src/test/java	테스트 케이스 java 소스
/src/test/resources	테스트 케이스 리소스
/target	빌드 된 output이 위치하는 디렉터리

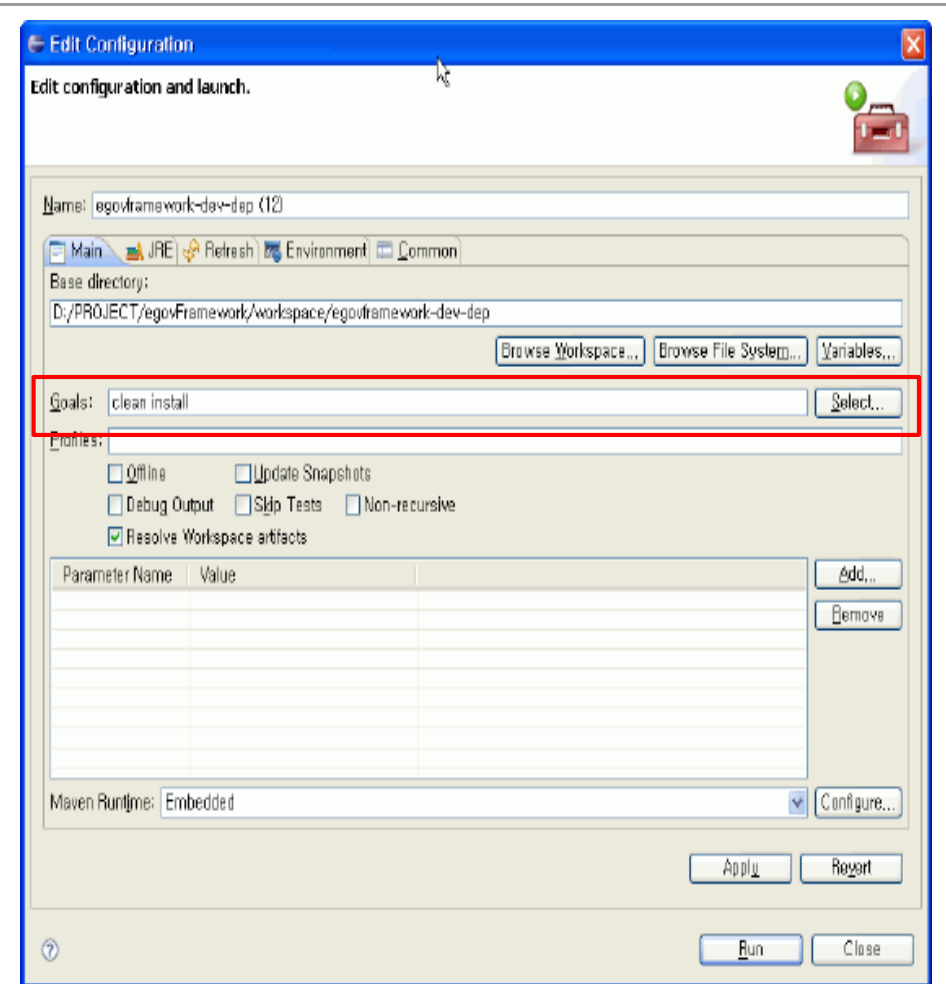
메이븐 빌드는 소프트웨어 프로젝트의 핵심적인 빌드 라이프 사이클 개념을 따르고 있으며 빌드 부터 **artifact**의 배포까지의 라이프 사이클을 정의하고 있음



이클립스에서 빌드 라이프 사이클에서 정의한 각 단계들 실행하기



메이븐 프로젝트 -> Run As -> Maven build..

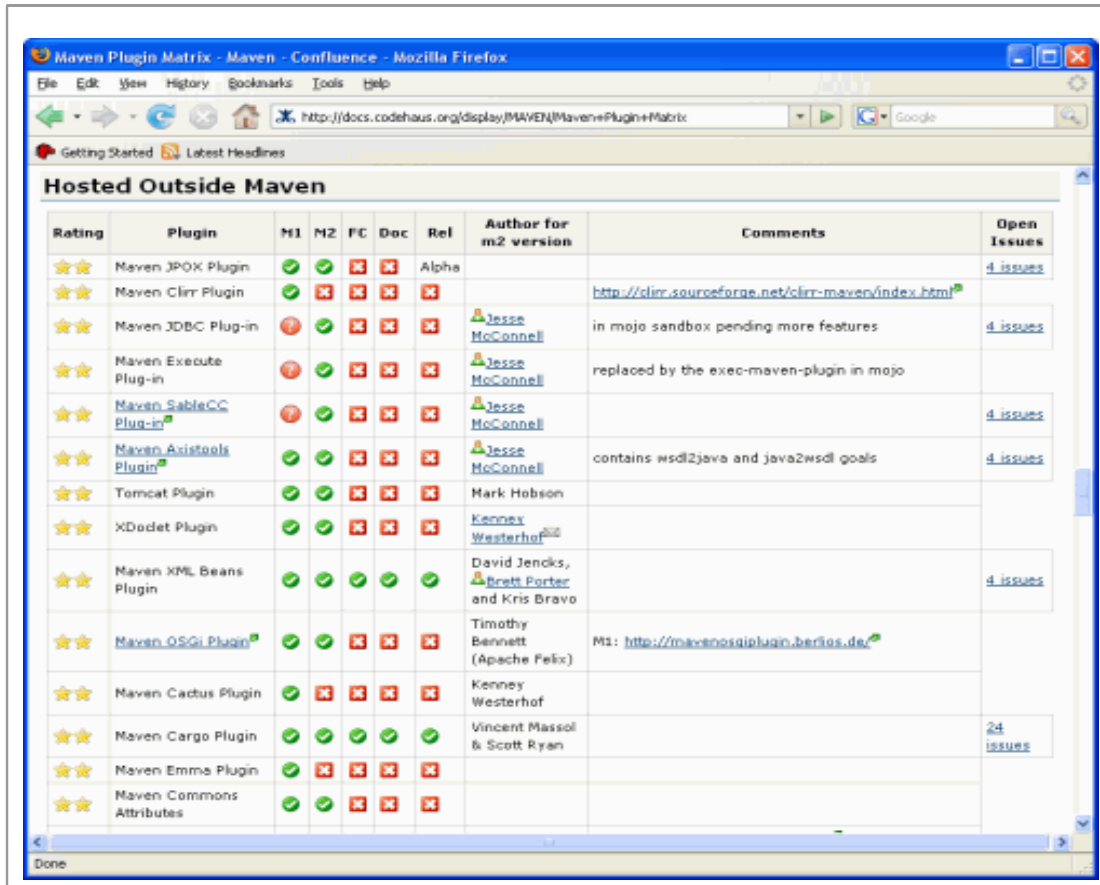


Goals: clean install

## 4. Maven 플러그인 (1/2)

## 3.개인빌드

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	✓	✗	✗	✗	✗		<a href="http://clirr.sourceforge.net/clirr-maven/index.html">http://clirr.sourceforge.net/clirr-maven/index.html</a>	
☆☆	Maven JDBC Plugin	✗	✓	✗	✗	✗	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 Axis2tools Plug-in	✓	✓	✗	✗	✗	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: <a href="http://mavenosgiplugin.berlios.de/">http://mavenosgiplugin.berlios.de/</a>	
☆☆	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>
```

**POM**에서 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.

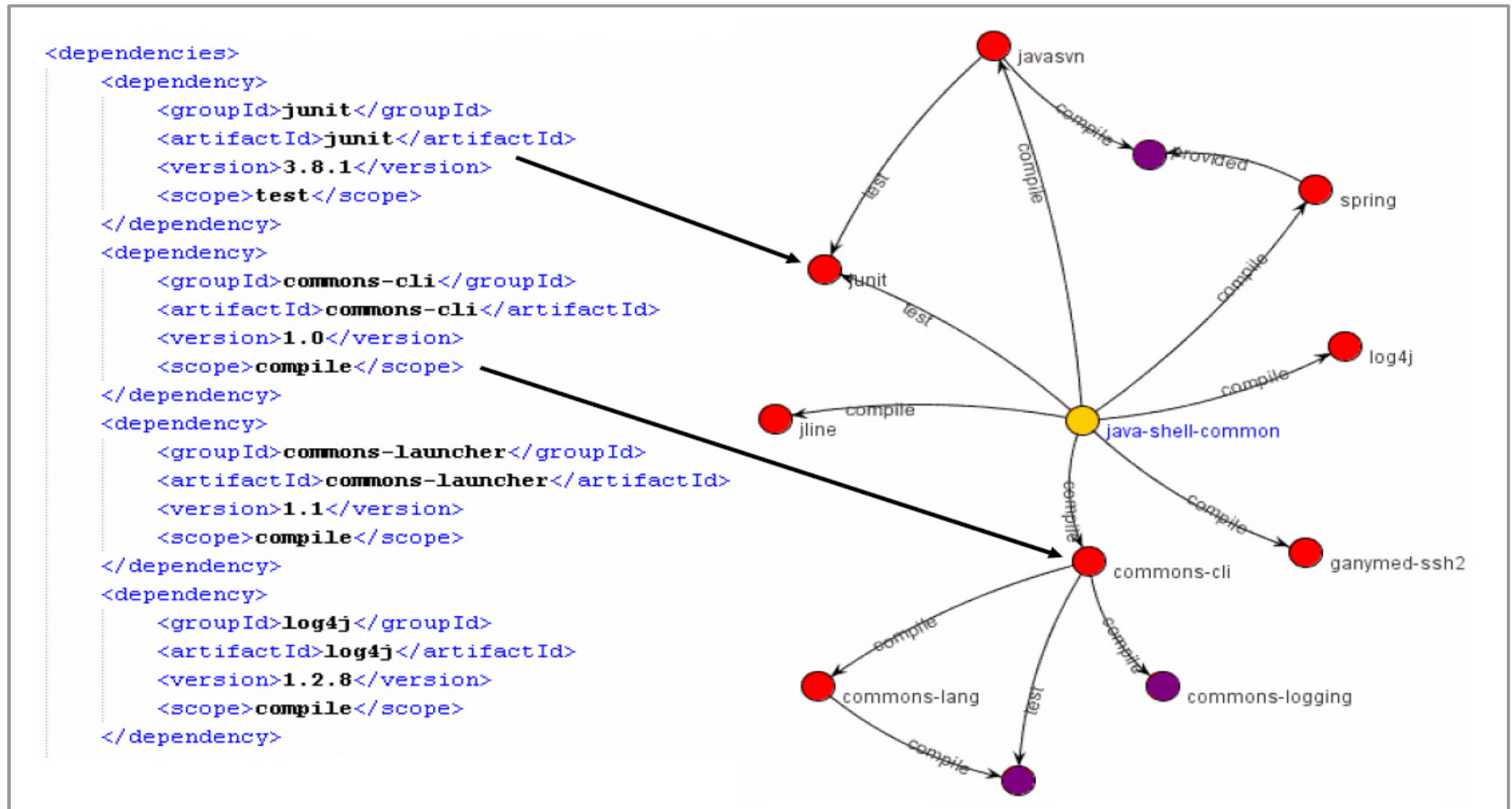
### 의존성 분석 순서

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

### 의존성 선언 항목

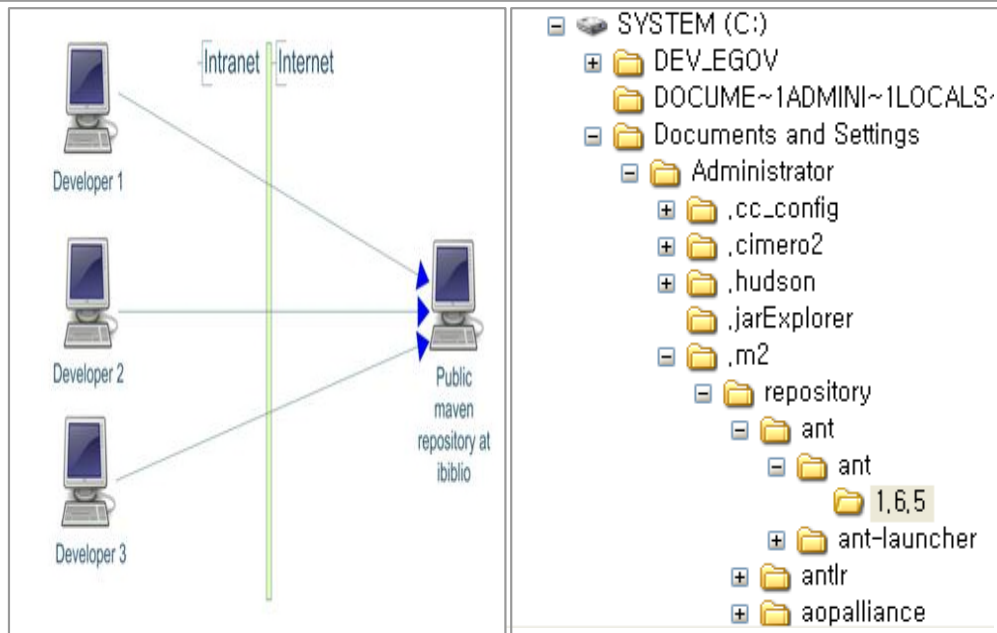
1. **<groupId>**: 부분적인 프로젝트나 조직에서의 라이브러리 집합을 식별하기 위해 제공
2. **<artifactId>**: 프로젝트의 실제 이름으로 groupId와 합쳐져 프로젝트 식별에 쓰임
3. **<version>**: 선언한 의존성 artifact의 버전으로 프로젝트 내에서 사용하는 artifact의 일관성을 추구할 수 있음

## ❑ pom.xml 파일에서의 의존성 설정 및 의존성 분석



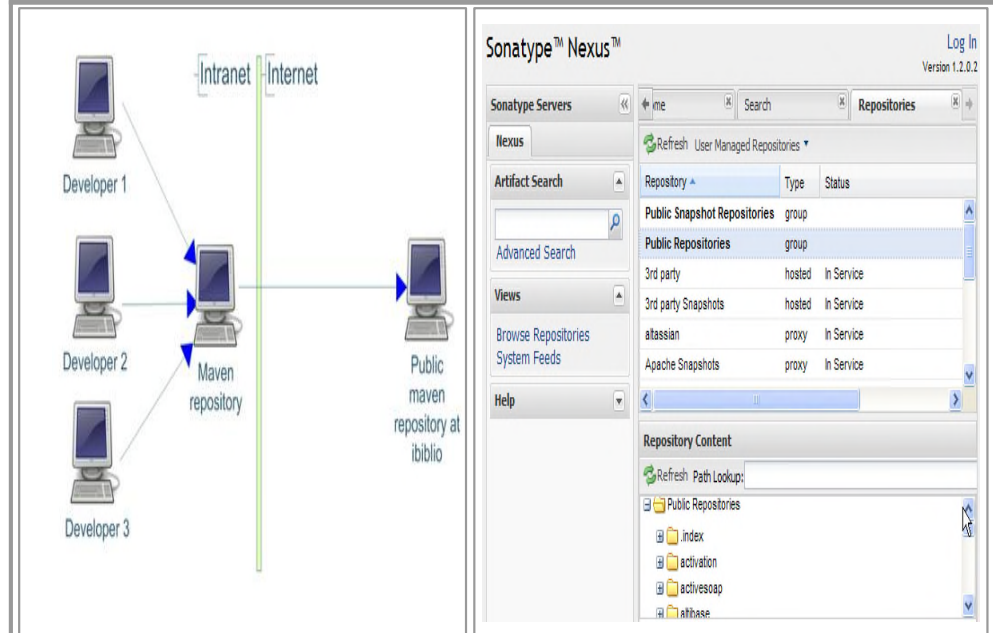
artifact들의 저장소로 로컬 및 리모트 repository로 구성되며 프로젝트는 pom.xml에서 선언한 dependency들을 저장소로부터 불러와서 사용함

### 메이븐 저장소 구조



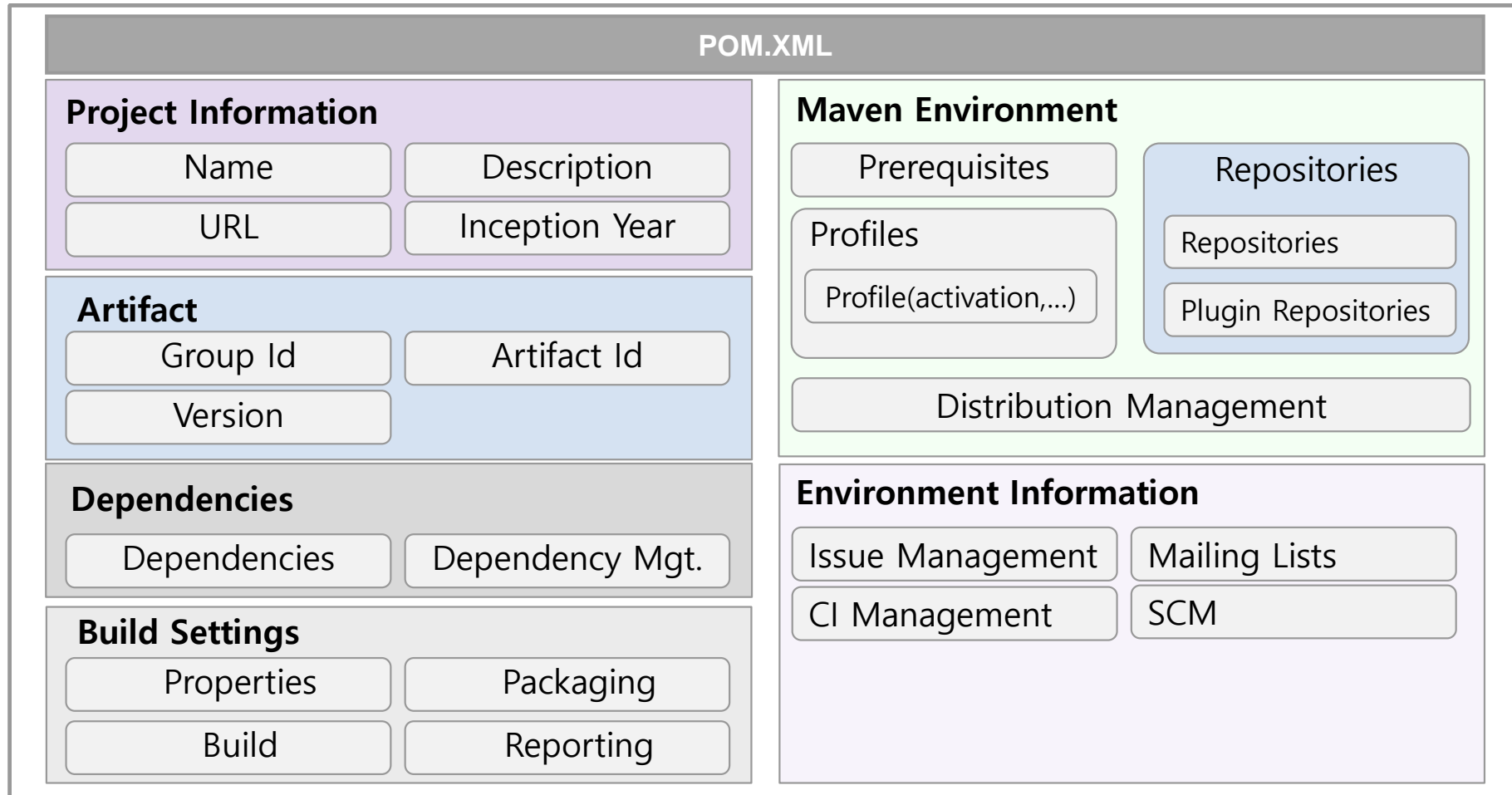
- **로컬 repository**는 artifact들을 로컬 파일 시스템에 cache하는 기능으로 리모트 repository에서 다운로드 받아 artifact들을 저장하고 관리한다.
- **리모트 repository**는 주 로 HTTP 서버로 3rd 파티에서 제공하는 artifact들을 제외 한 거의 모든 artifact들을 제공한다.

### Nexus 연동



- **Nexus**는 메이븐 repository 관리툴로 리모트 repository의 단점을 보완하고 maven 프로젝트의 사용 편의성을 높이기 위해 사용된다. 각 개발자들은 리모트 repository가 아닌 Nexus에서 dependency을 다운 받아 사용함으로써 프로젝트 내의 artifact 버전 등의 일관성을 유지하고 3rd 파티 artifact 등 라이브러리를 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

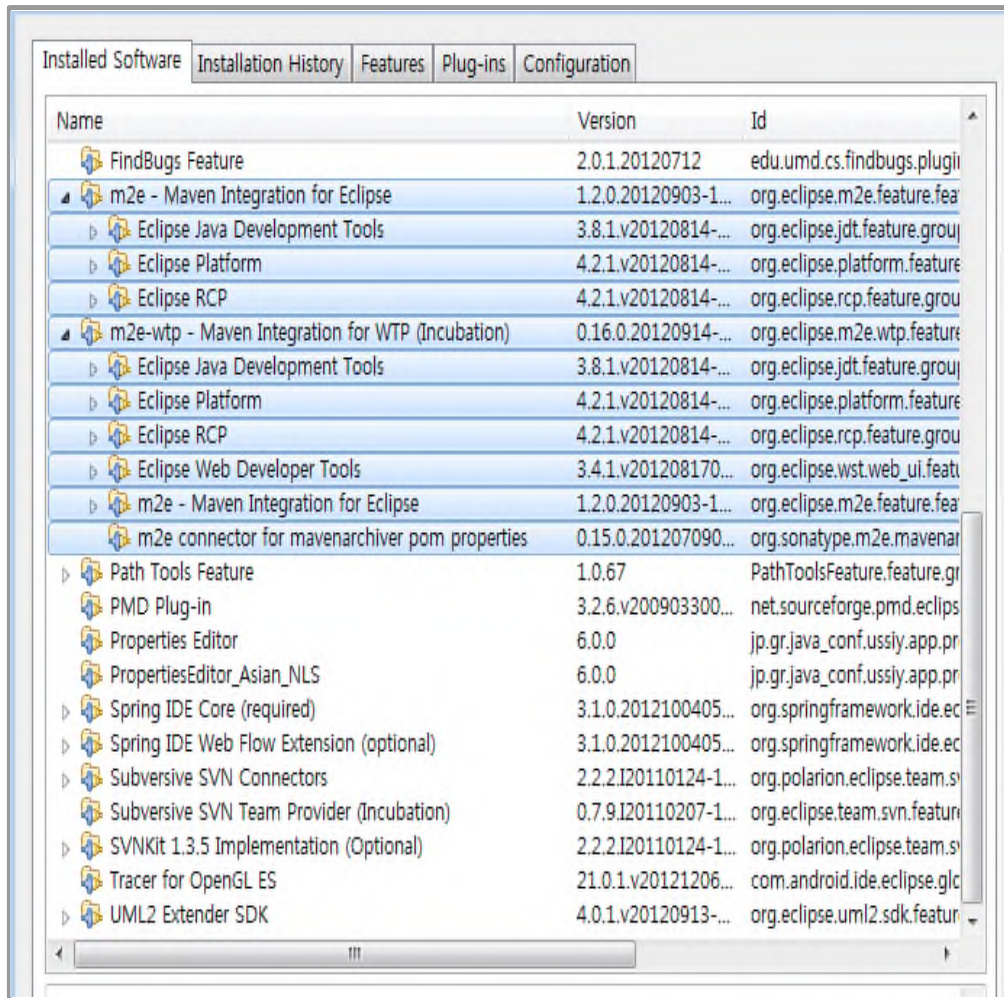
```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0http://maven.apache.org/maven-v4_0_0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>egovframework.dev.com</groupId>
  <artifactId>egovframework-dev-com</artifactId>
  <version>1.0</version>
  <packaging>war</packaging>
  <name>egovframework-dev-com Maven Webapp</name>
  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>4.12</version>
      <scope>test</scope>
    </dependency>
  </dependencies>
  <plugins>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-compiler-plugin</artifactId>
      <configuration>
        <source>1.8</source>
        <target>1.8</target>
        <encoding>UTF-8</encoding>
      </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

Browse...

##### New Maven project

Select an Archetype

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

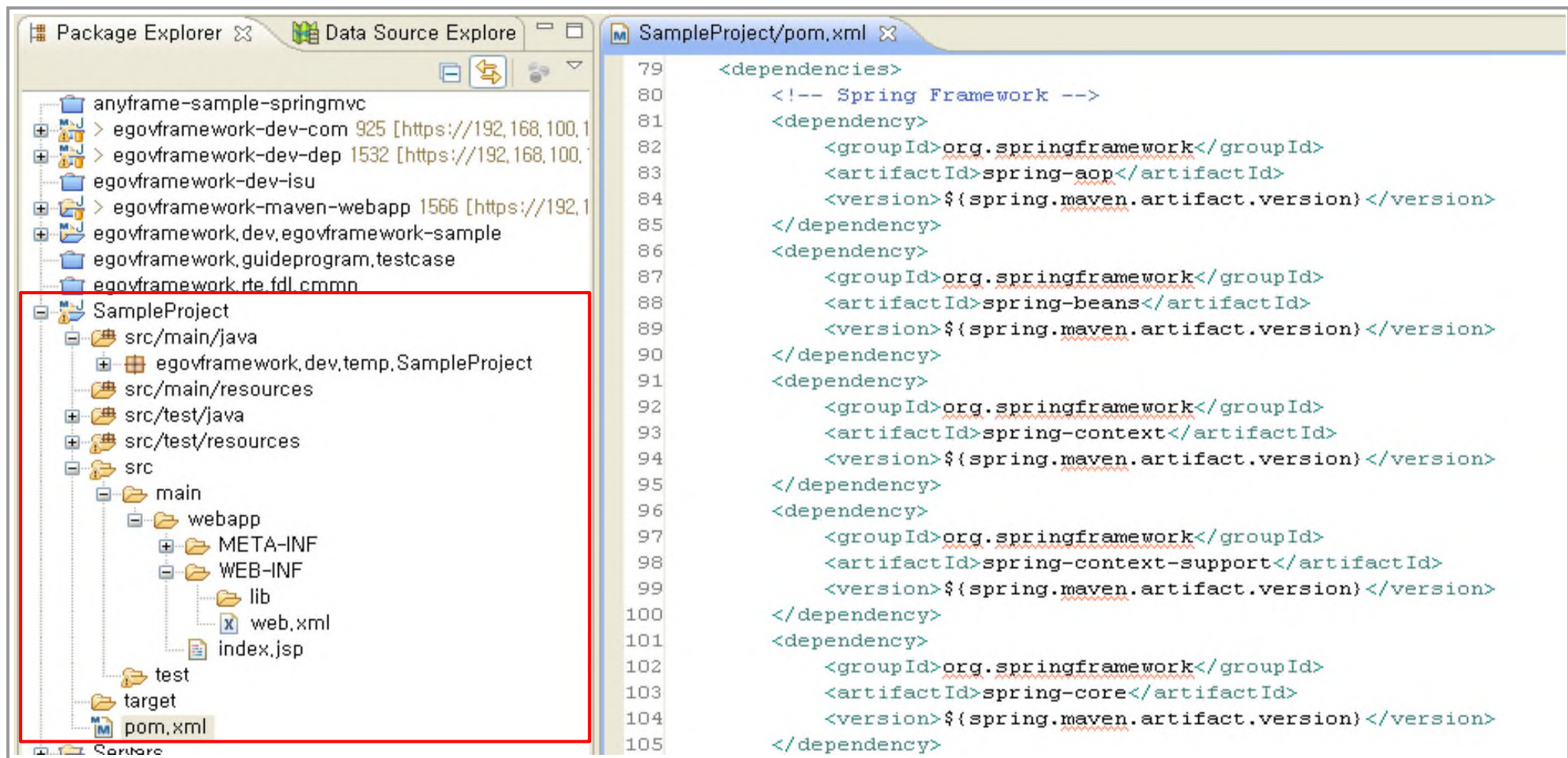
Configure...

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 a 'src' directory with 'main' and 'test' subdirectories, each containing 'java' and 'resources' folders. The 'main' directory also has a 'webapp' subdirectory with 'META-INF', 'WEB-INF', and 'lib' folders, along with 'web.xml' and 'index.jsp' files. The 'pom.xml' file is located at the root of the project. On the right, the 'SampleProject/pom.xml' file is open, showing a list of dependencies for Spring Framework components, including 'spring-aop', 'spring-beans', 'spring-context', 'spring-context-support', and 'spring-core'. The dependencies are defined with their respective group IDs, artifact IDs, and versions, using a placeholder for the version.

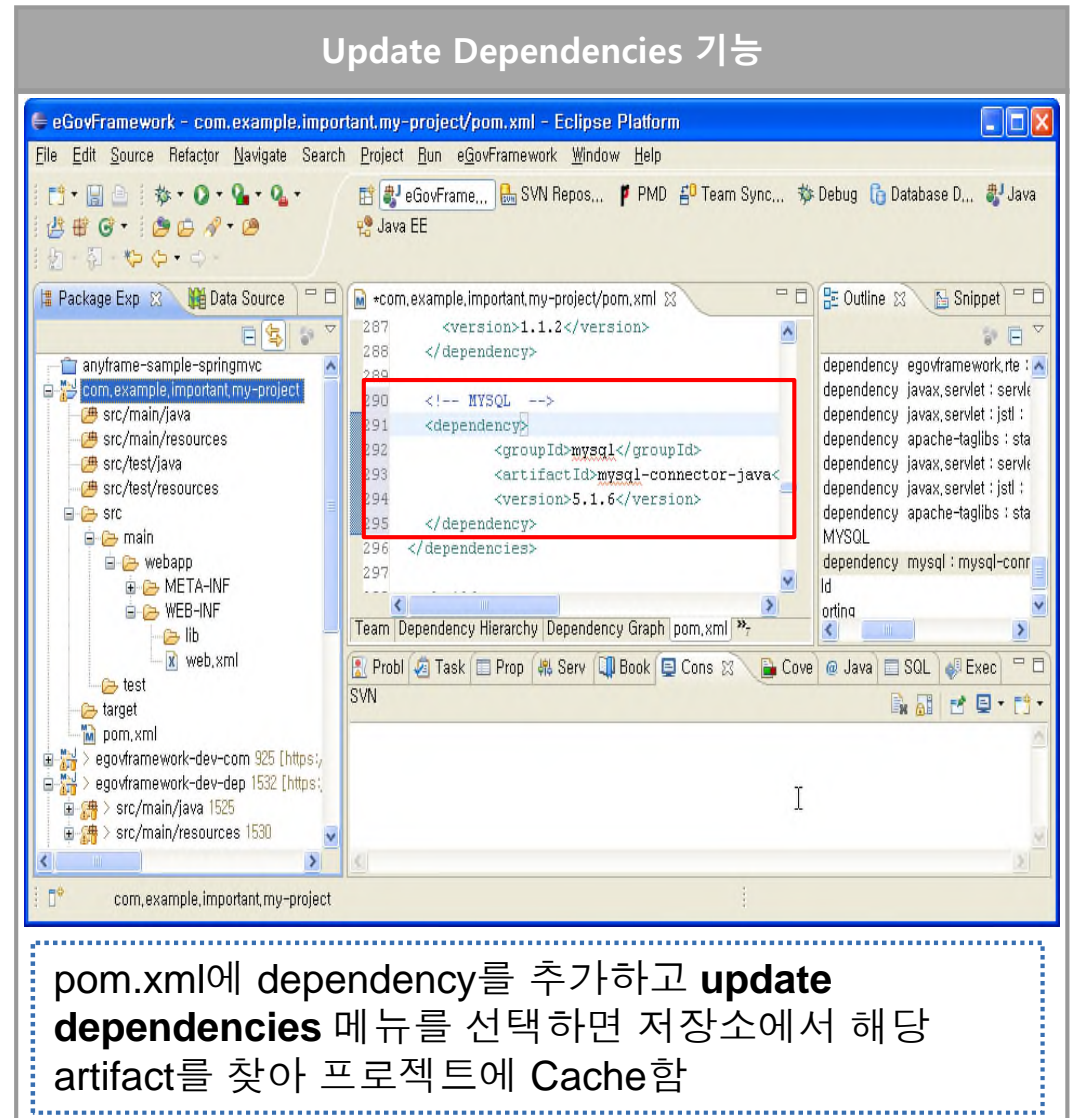
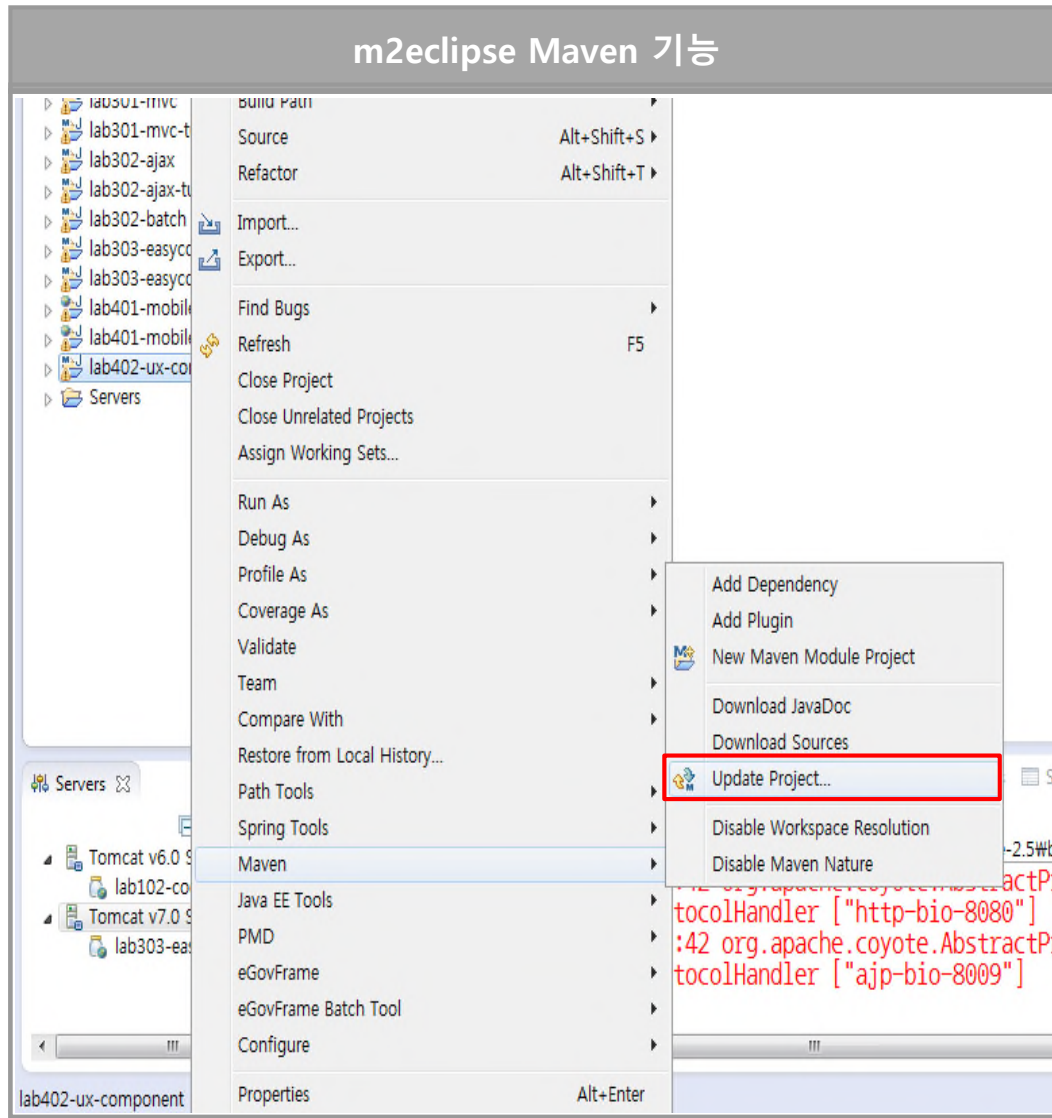
```

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>

```

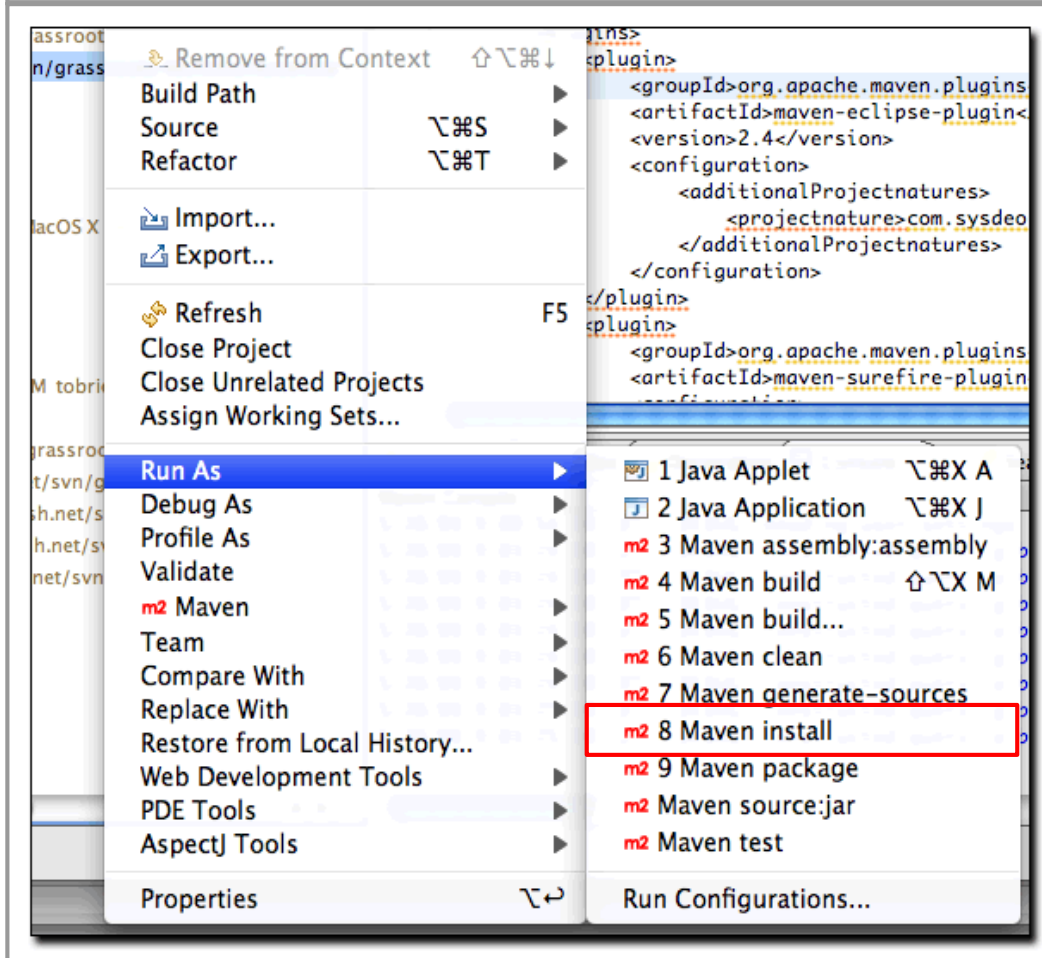
- Spring 기반의 dependency들을 기본으로 제공함

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

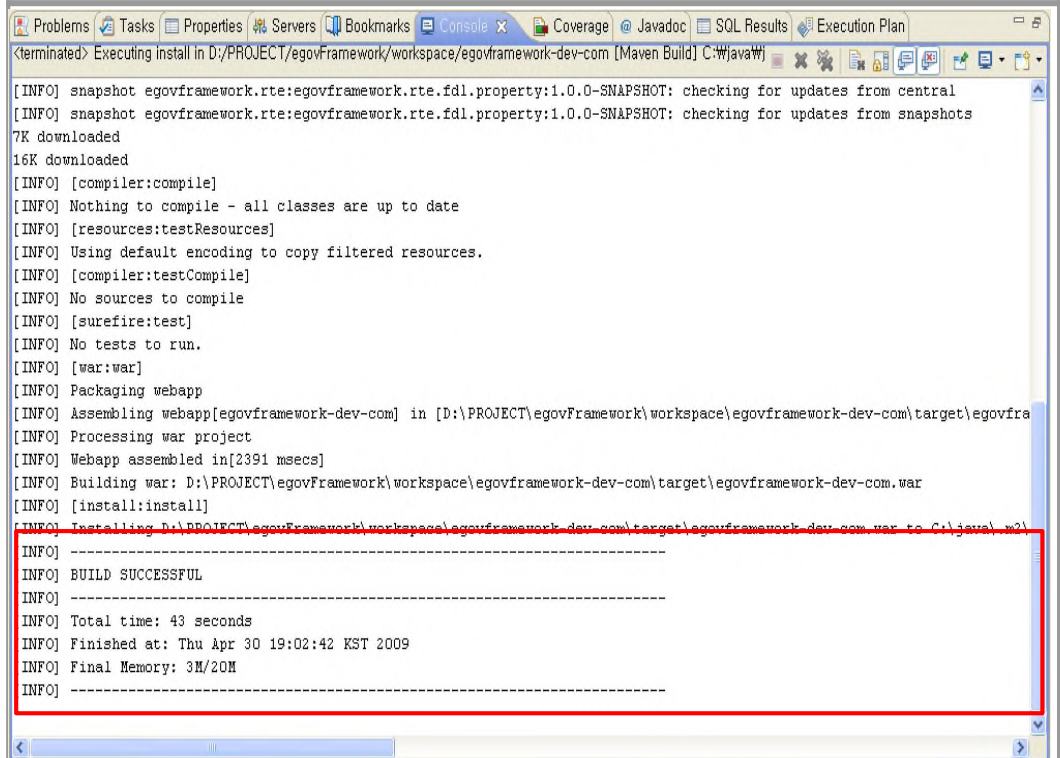


m2eclipse는 이클립스의 Run As 메뉴에 Maven 기본 생명주기 단계를 추가하여 빌드 편의성을 제공함

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/>

#### 4. 테스트 도구

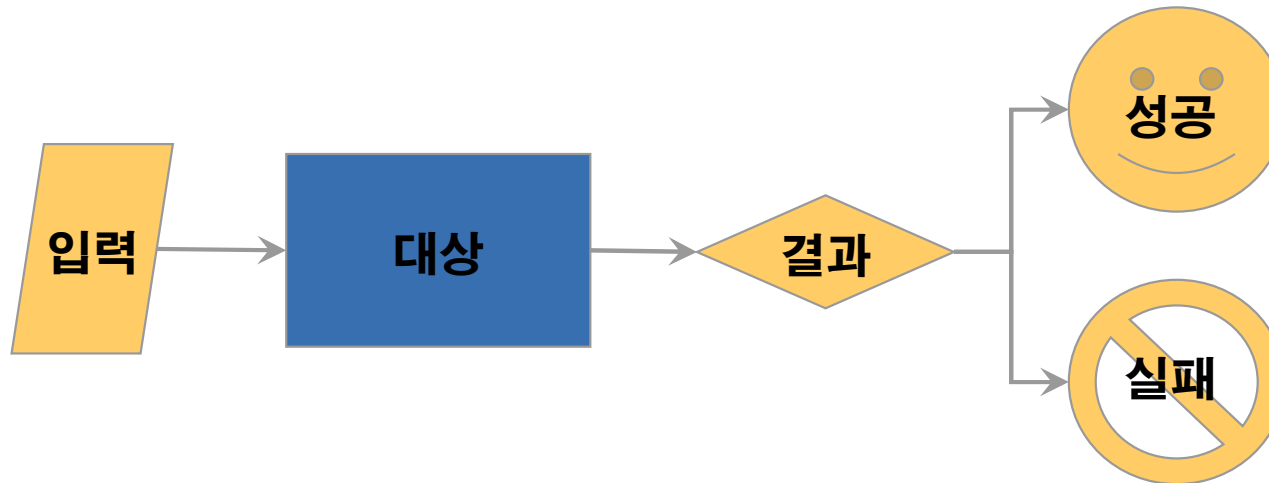


1. 테스트
2. 개요
3. Unit Test
4. Mock

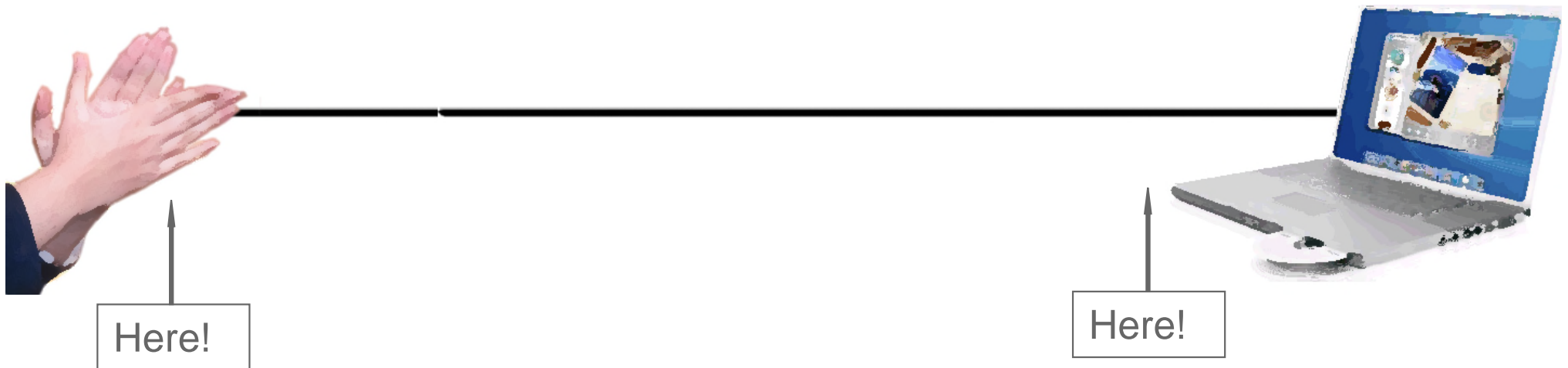
5. Batch Job Test
6. DB Test
7. Test Automation
8. Test Reporting

### □ 테스트(Test)

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

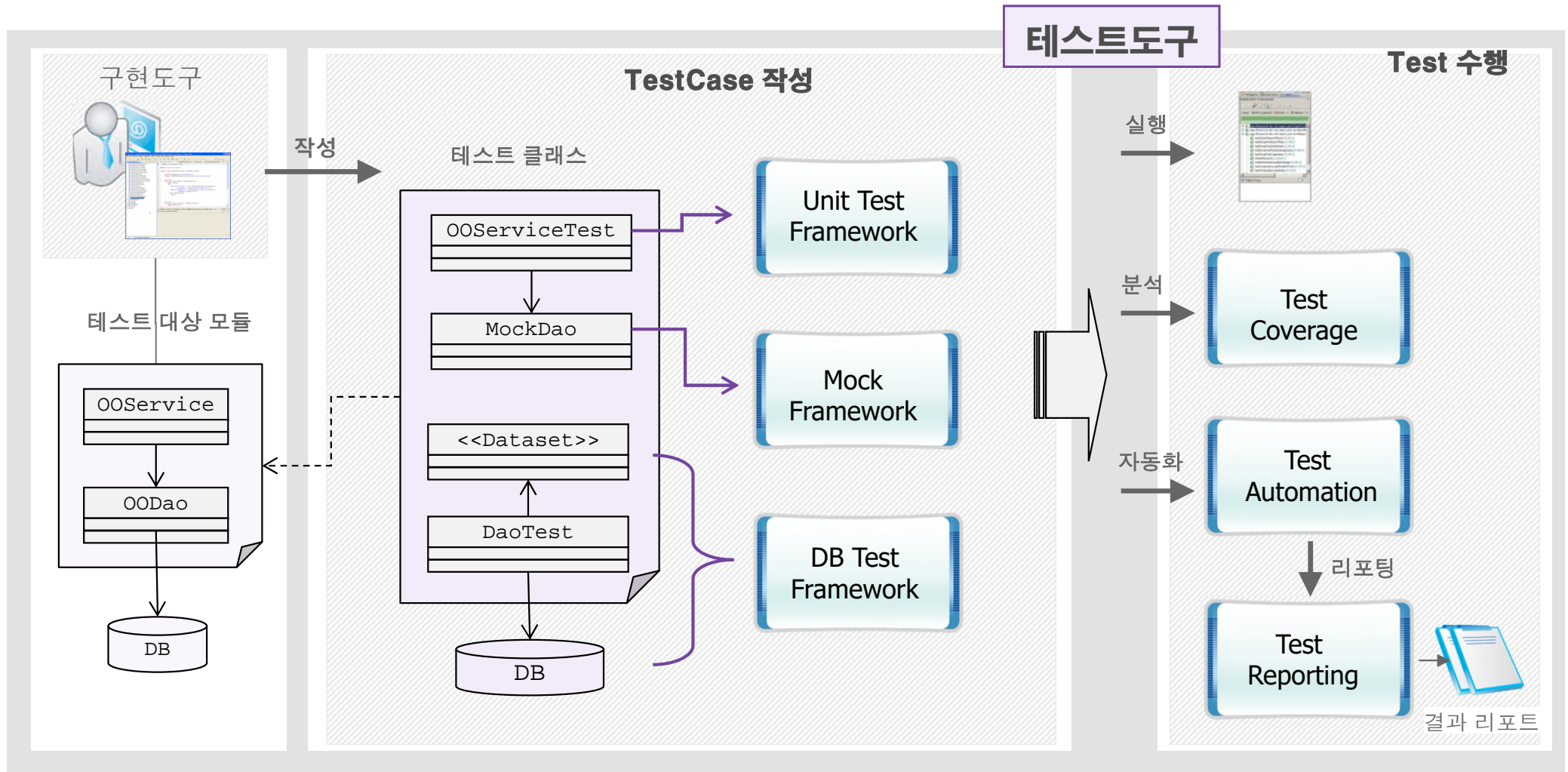


### □ 수동 테스트 vs. 자동 테스트

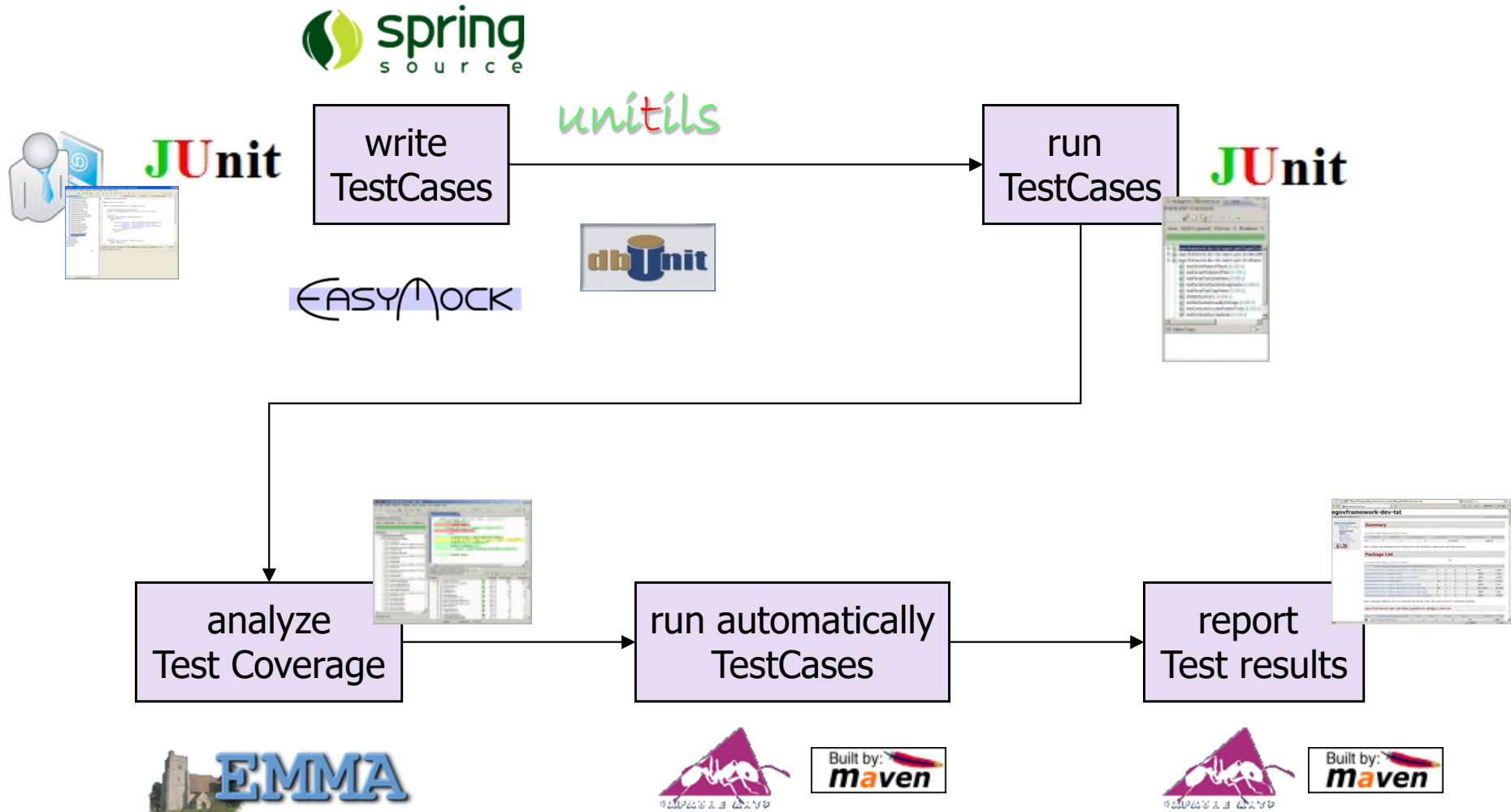


	수동 테스트	자동 테스트
장점	<ul style="list-style-type: none"> <li>□ 쉽다. 간편하다.</li> <li>□ 테스트 불가능한 상황이 별로 없다.</li> </ul>	<ul style="list-style-type: none"> <li>□ 언제든지 같은 테스트를 여러 번 수행 가능</li> <li>□ 기존 테스트는 새 테스트를 작성하는 발판이 된다. (지식축적)</li> <li>□ 빠르므로 자주 돌려볼 수 있다.</li> <li>□ 개발자가 코드 개선 활동을 수행할 때에도 회귀테스트(Regression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

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



### □ 테스트 도구 사용 프로세스



**Unit Test**란 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

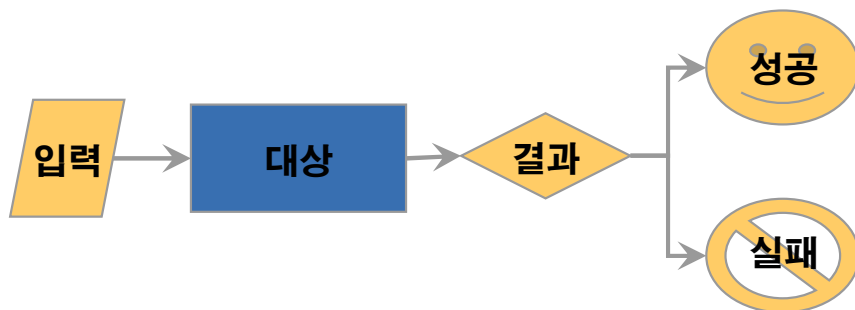
- 작성한 코드의 설계 개선 작업 시, 코드 품질에 대한 확신
- 코드 수정 시 버그를 쉽게 찾을 수 있게 해줌
- 자동화된 회귀 테스트 (Regression Test)를 가능하게 해주는 Source가 됨

#### □ 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

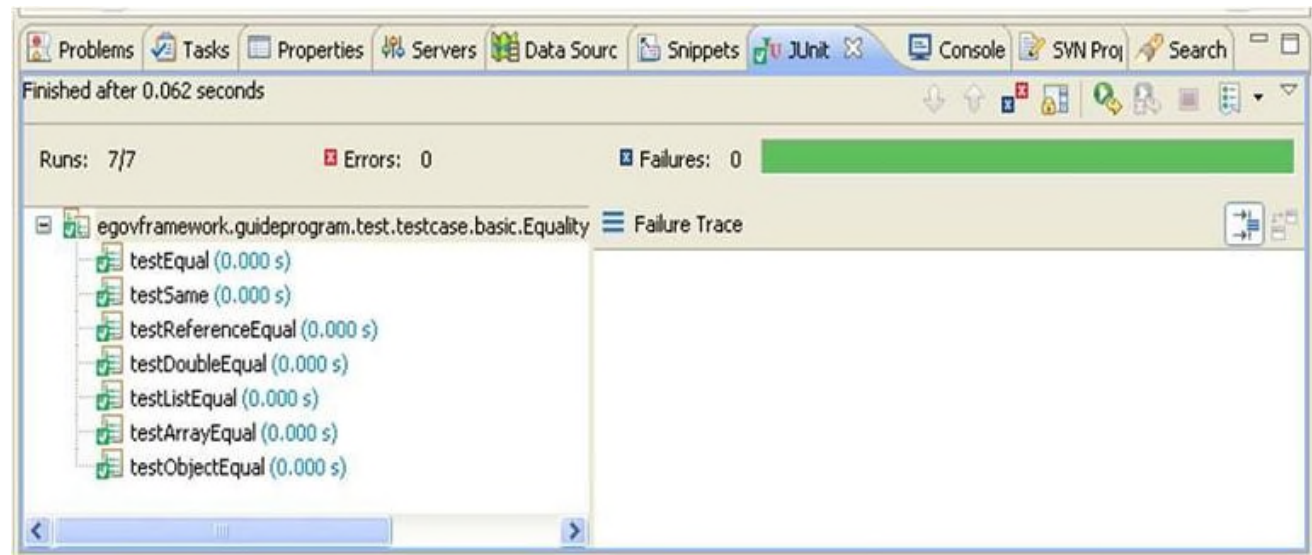
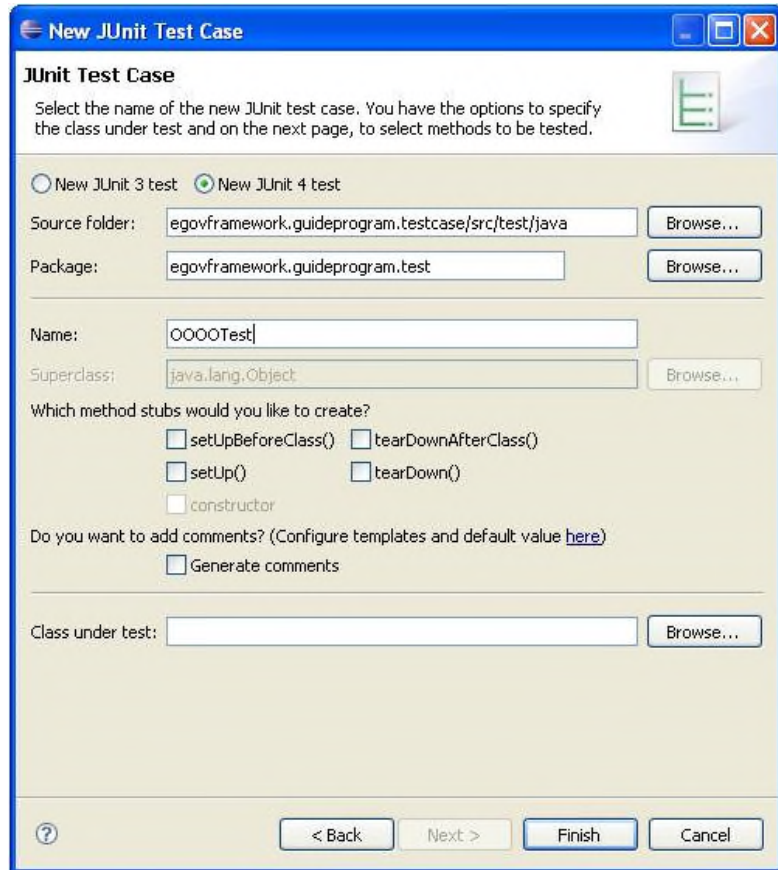
- 테스트 프레임워크를 사용하는 Class
- 공용으로 사용하는 테스트 데이터 (test fixture)
- 테스트 데이터 준비 (Setup of test data)
- 테스트 메소드 (testXXX)
  - (테스트 별 준비)
  - 테스트 대상 메소드 실행
  - assert 문을 이용한 결과 확인 (assertTrue, assertEquals etc.)
- (내부 메소드)



```

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



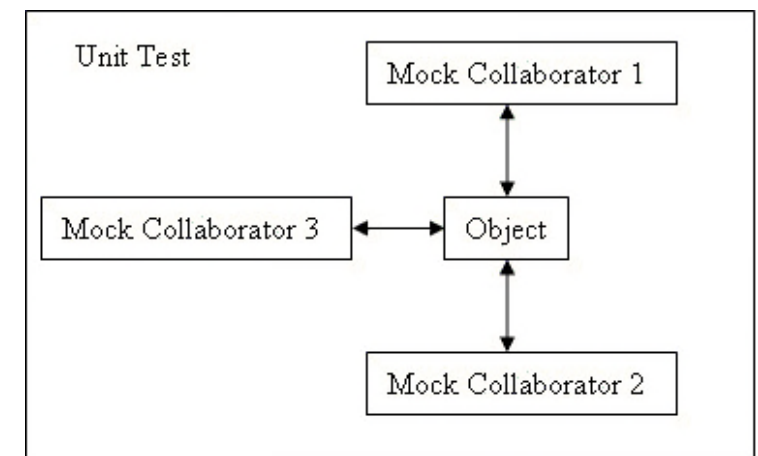
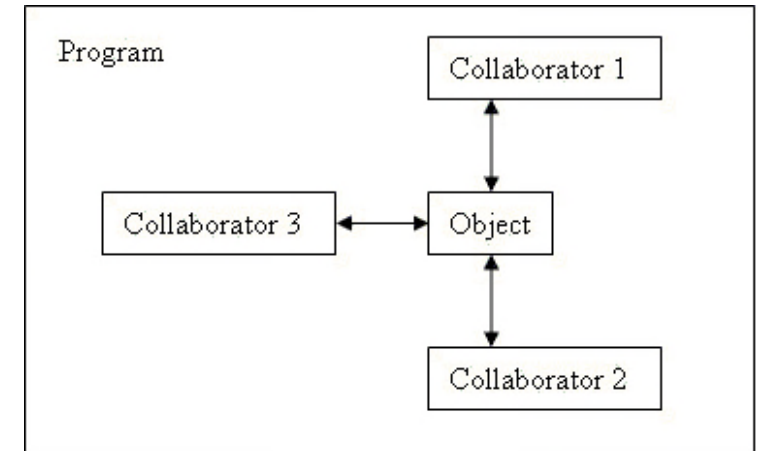
**Mock 객체**는 **Unit Test**의 독립성(isolation)을 높여주기 위해 사용하며, 테스트하고자 코드와 관련이 있는 객체(collaborator)를 흉내내어 **Unit Test**를 수행할 수 있도록 도와주는 객체임

### □ 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



## □ 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.

배치 테스트 도구를 통해 기 개발된 일괄처리 모듈에 대한 Junit 테스트 파일생성 및 테스트 수행과 함께, 기존에 생성한 Junit 테스트 파일을 재실행할 수 있는 기능을 제공

The screenshot shows the 'eGovFrame Batch Job Test' dialog box. It has a title bar and a main area with the following sections:

- Customize Batch Job Test Option**: Subtitle 'Batch Job Test를 수행할 조건을 설정하십시오.' (Set conditions for executing Batch Job Test).
- Project Name**: Set to 'file.sche.egov.temp'.
- Job List**: A list of jobs including 'TestJob', 'delimitedToDelimited', 'fixedLengthToFixedLength', 'fixedLengthToBatisJob', and 'fixedLengthToJdbcJob'.
- Job Launcher Information**: A table with columns 'File Name' and 'File Location'. It lists 'context-batch-job-launcher.x...' and 'newJobLauncher.xml'.
- Job Parameter Information (Selection Item)**: A table with columns 'Name', 'Value', and 'Data Type'. It shows a parameter 'name' with value 'eGovFrame' and data type 'String'. There are 'Add' and 'Remove' buttons.
- Batch Test File Generation**: A button labeled 'Generate Batch Test File'.
- Batch Test Execution**: A button labeled 'Test'.
- Result**: A text area showing '<eGovFramework Batch Test>' and 'Batch Execution Status ..... [ OK ]'.
- Buttons**: 'Finish' and 'Cancel' buttons at the bottom.

Numbered annotations (1-5) are placed over the dialog box:

- Job 정보 선택**: Points to the Job List.
- Job Launcher 정보 선택**: Points to the Job Launcher Information table.
- Job Parameter 정보 선택 (선택사항)**: Points to the Job Parameter Information table.
- 배치 테스트 파일 생성**: Points to the 'Generate Batch Test File' button.
- 배치 테스트 수행**: Points to the 'Test' button.

### □ 배치 테스트 절차

- ① 배치 Job 정보를 선택한다.
- ② Job Launcher 정보를 선택한다.
- ③ 필요 시 Job Parameter 정보를 선택한다.
- ④ Generate Batch Test File 버튼을 클릭하여 배치 테스트 파일을 생성한다.
- ⑤ Test 버튼을 눌러 배치 테스트를 수행한다.

**DB Test**는 **DAO**와 **DB** 모두를 통틀어 **Persistence layer**를 테스트하는 것을 말하며, 테스트**DB**와 테스트 데이터를 준비하여 실제로 **DB**를 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

작성된 다수의 **TestCase**를 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)

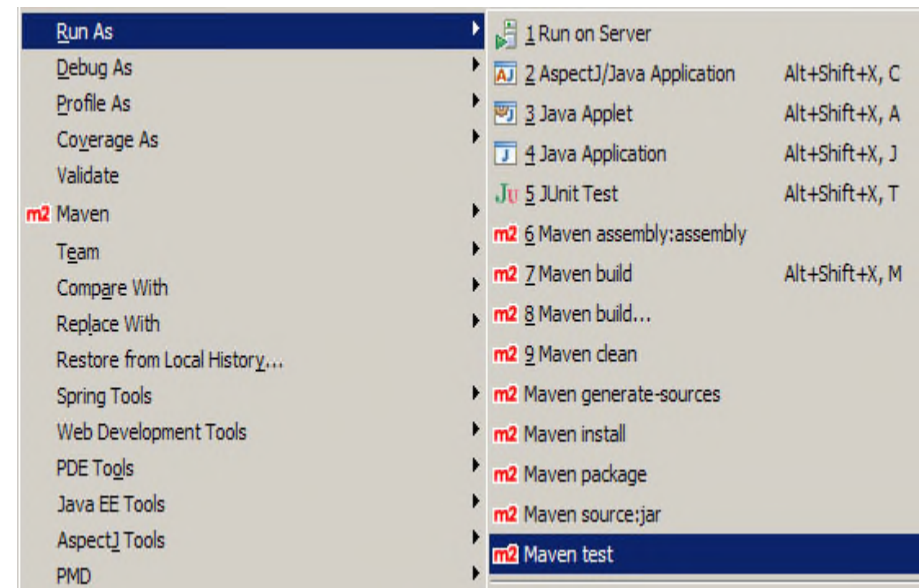
```
<junit forkmode="perBatch" printsummary="true" haltonfailure="yes" haltonerror="yes">
  <classpath refid="master-classpath" />
  <classpath refid="test-classpath" />
  <classpath path="${testbuild.dir}" />

  <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.ipa	39	26	0	51.327	2009-03-11T02:22:44	lgcns-juroh

egovframework-dev-tst

Last Published: 2009-03-12

**Summary**

[Summary] [Package List] [Test Cases]

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**

[Summary] [Package List] [Test Cases]

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
EgovTxServiceImpTest	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

The image displays two side-by-side screenshots of Microsoft Excel templates used for test reporting. Both templates are titled 'Microsoft Excel - template-kr.xls'.

**Left Screenshot (JUnit Test):**

- The worksheet is named '이름' (Name).
- Row 6 is highlighted in green and labeled 'Summary'. An annotation points to cell B6 with the text 'summary: 0, B, 6'.
- Row 7 is labeled 'all classes'.
- Row 11 is highlighted in yellow and labeled 'Package'. An annotation points to cell B11 with the text 'packages: 0, B, 11'.
- Row 12 contains the text 'egovframework.dev.samples.guestbook.category.service'.
- The bottom status bar shows the sheet name 'Summary' and the selected cell range 'Lists/'.

**Right Screenshot (Test):**

- The worksheet is named 'Package'.
- Row 5 is highlighted in orange and labeled 'Package'. An annotation points to cell A5 with the text 'lists 위치 : 1, A, 5'.
- Row 6 contains the text 'egovframework.dev.samples.guestbook.category.service'.
- Row 7 is labeled 'Class' and contains the text 'EgovTxServiceImplTest'.
- The bottom status bar shows the sheet name 'Summary' and the selected cell range 'Lists/'.

## ❑ 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>
```

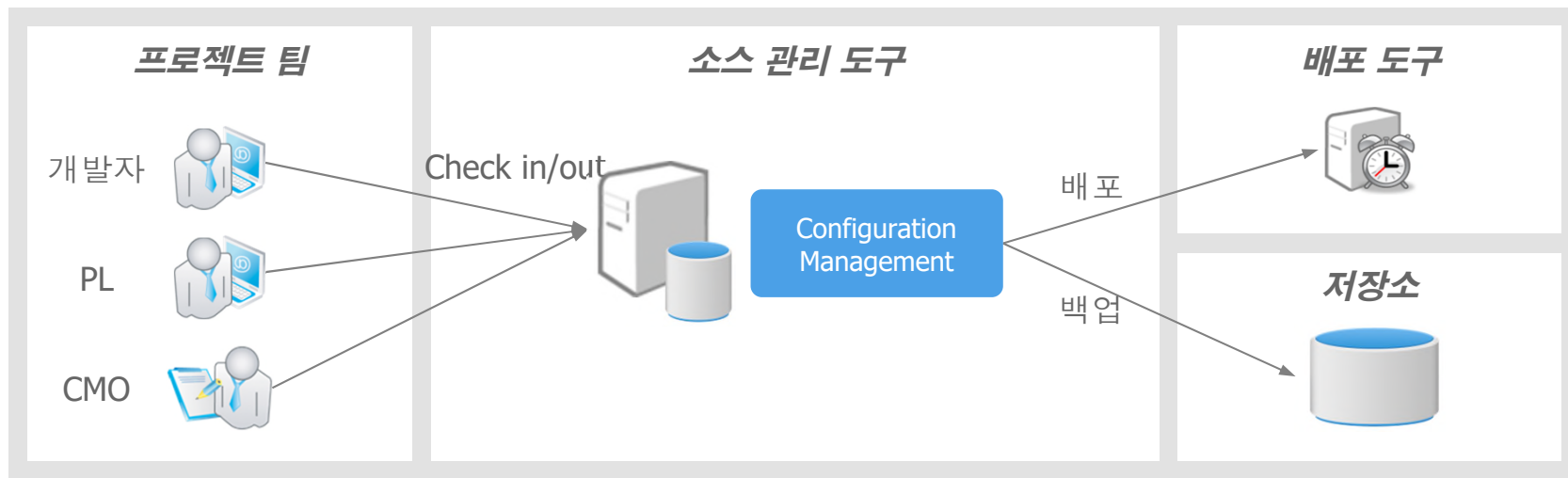
JUnit Test Results Summary						
2009-03-27						
Summary						
name	Tests	Errors	Failures	Skipped	Success Rate	Time (s)
all classes	44	6	0	4	77.27%	36.172
Package						
name	Tests	Errors	Failures	Skipped	Success Rate	Time (s)
egovframework.dev.samples.guestbook.category.service	6	6	0	0	0.00%	2.984
egovframework.dev.samples.junit4	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



## 5. 형상관리

- 1. 개요
- 2. Subversion 개요
- 3. 소스 공유
- 4. Checkout
- 5. Update
- 6. Commit
- 7. History 조회
- 8. 충돌해결 (Merge)

- ❑ 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/>

### ❑ 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

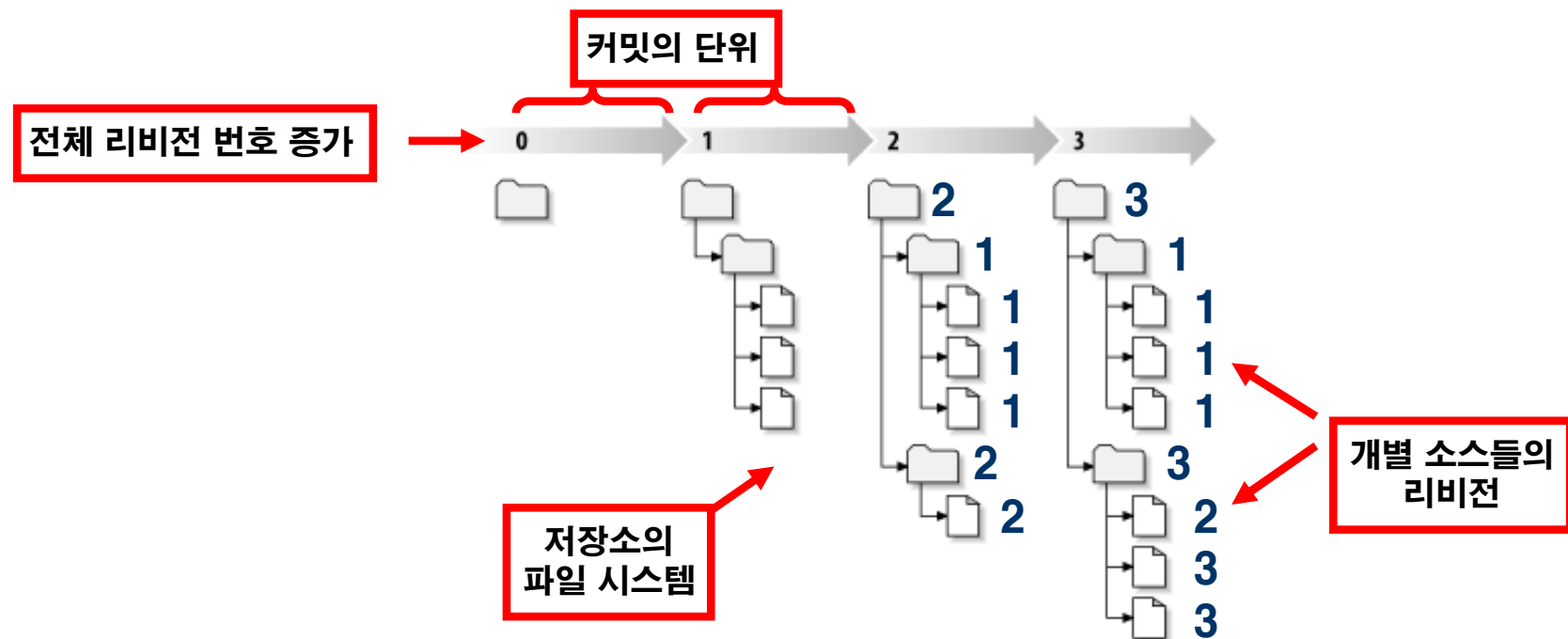
② Eclipse 메뉴,  
File > New >  
Repository Location

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

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

Share Project Wizard

Share Project with SVN repository

Select an existing repository location or create a new location.

This wizard will help you to share your files with the SVN repository for the first time. Your project will automatically be imported into the SVN repository.

☐ Create a new repository location

☒ Use existing repository location:

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

< Back Next > Finish Cancel

#### □ 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...

② 팝업 창에서 내용 확인 > OK

**Add to SVN Version Control**

Selected resources will be added to the source control

There is 1 resource that is not under SVN version control. Are you sure you want to add it? The operation will be applied only to the resources which are selected in the list represented below.

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

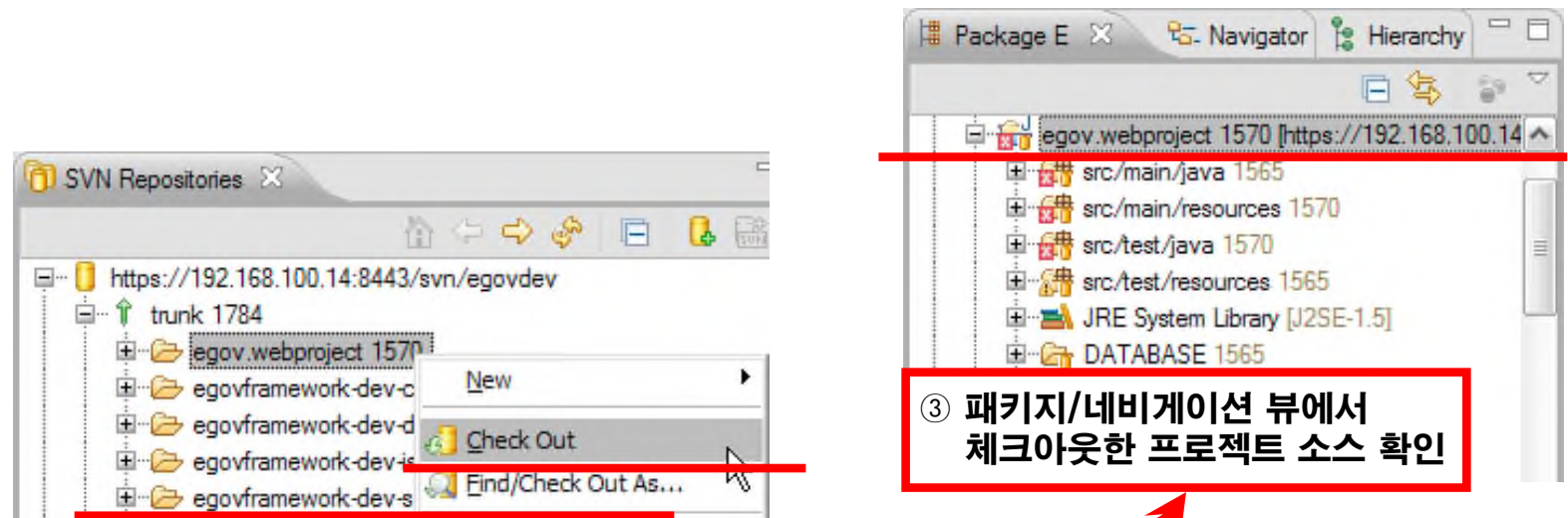
Select All Clear Selection

Selected: 1 of 1

OK Cancel

### ❑ 초기 checkout

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



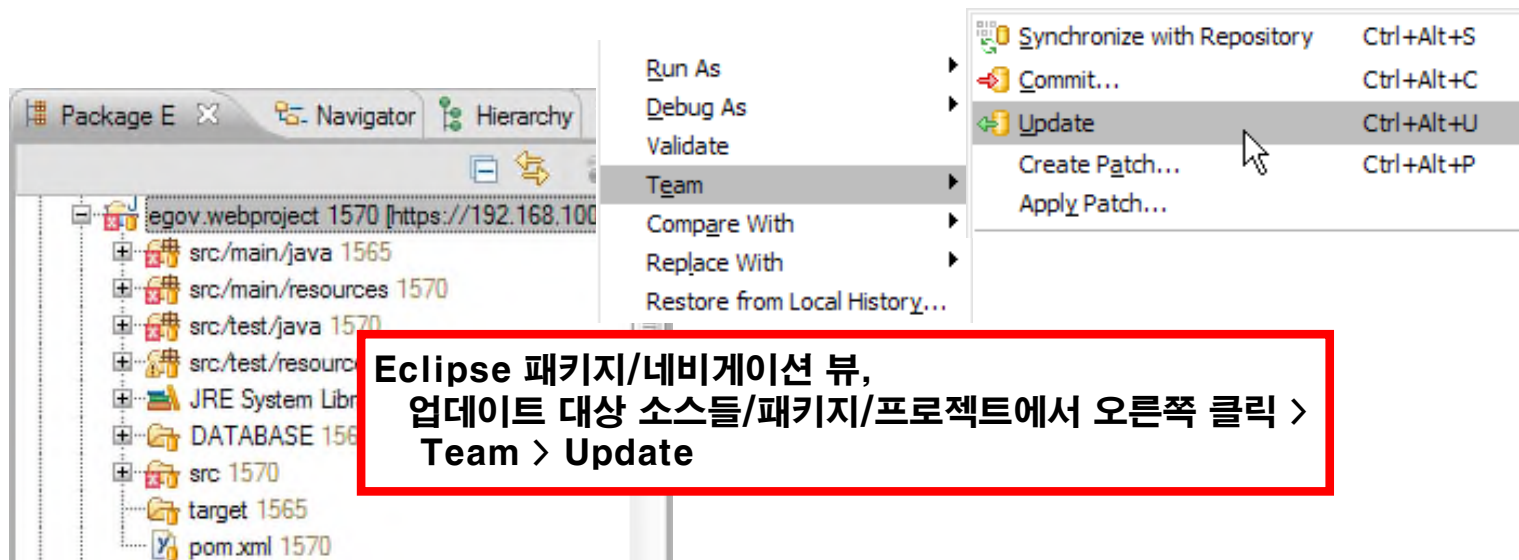
② SVN Repositories 뷰,  
저장소를 확장하여 trunk 아래의  
프로젝트 선택 > Check Out

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

③ 패키지/네비게이션 뷰에서  
체크아웃한 프로젝트 소스 확인

### □ 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)

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

② 팝업 창에서 반영 내용 확인 및 Comment 입력

Commit dialog box details:

- Comment: [Empty text field]
- Choose a previously entered comment or template: [Dropdown menu]
- Keep Locks: ☐
- Paste selected names: [Button]
- Resource list:

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	

Select All Clear Selection Selected: 5 of 5

OK Cancel

### □ History

- SVN에 의해 관리되는 특정 소스와 관련된 과거 모든 리비전의 내역

### □ History 조회

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

The image shows two screenshots from the Eclipse IDE. The left screenshot shows the 'Team' menu with 'Show History' selected. The right screenshot shows the 'History' view with a table of revisions and a file tree below it.

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

**리비전 로그 목록**

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]

**선택된 리비전의 Comment**

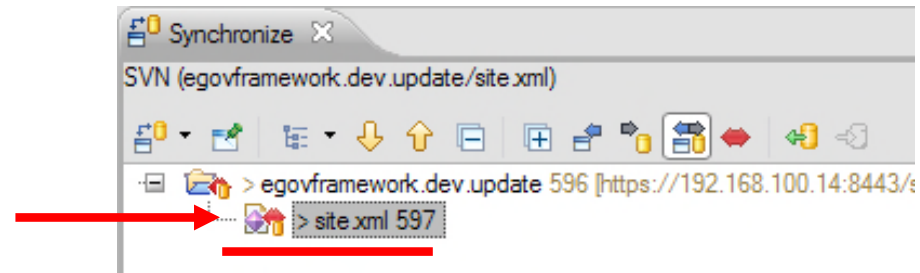
**선택된 리비전의 구조와 변경 파일들**

**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)

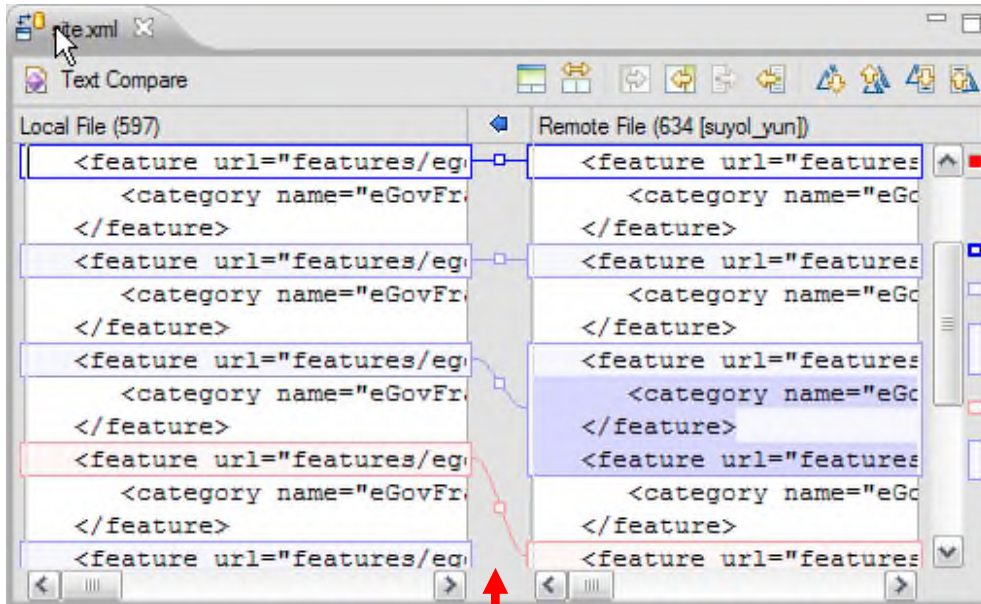
충돌 발생상황 예:  
로컬 리비전 = 597 < 저장소 리비전 = 634



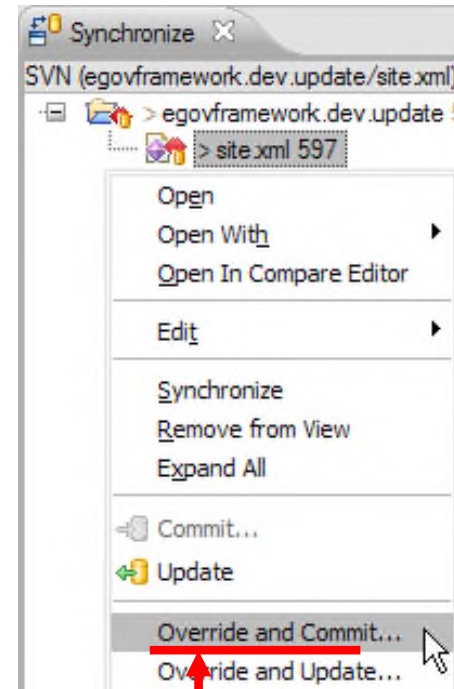
### ❑ 충돌 해결 방안

- 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

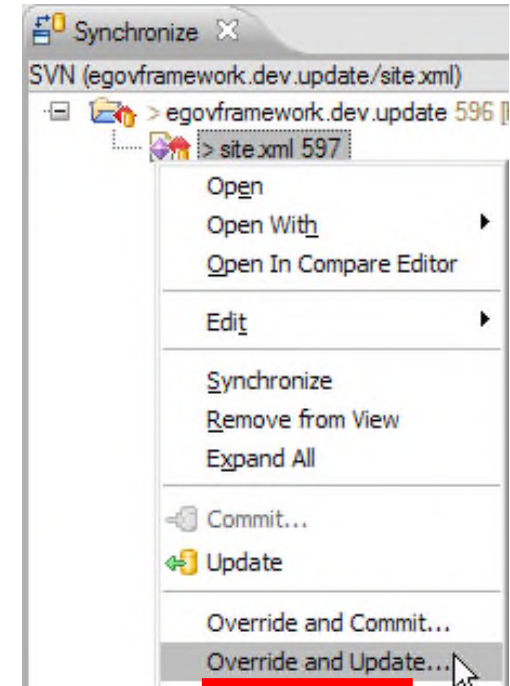
### ❑ 충돌 해결 방안 예



**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

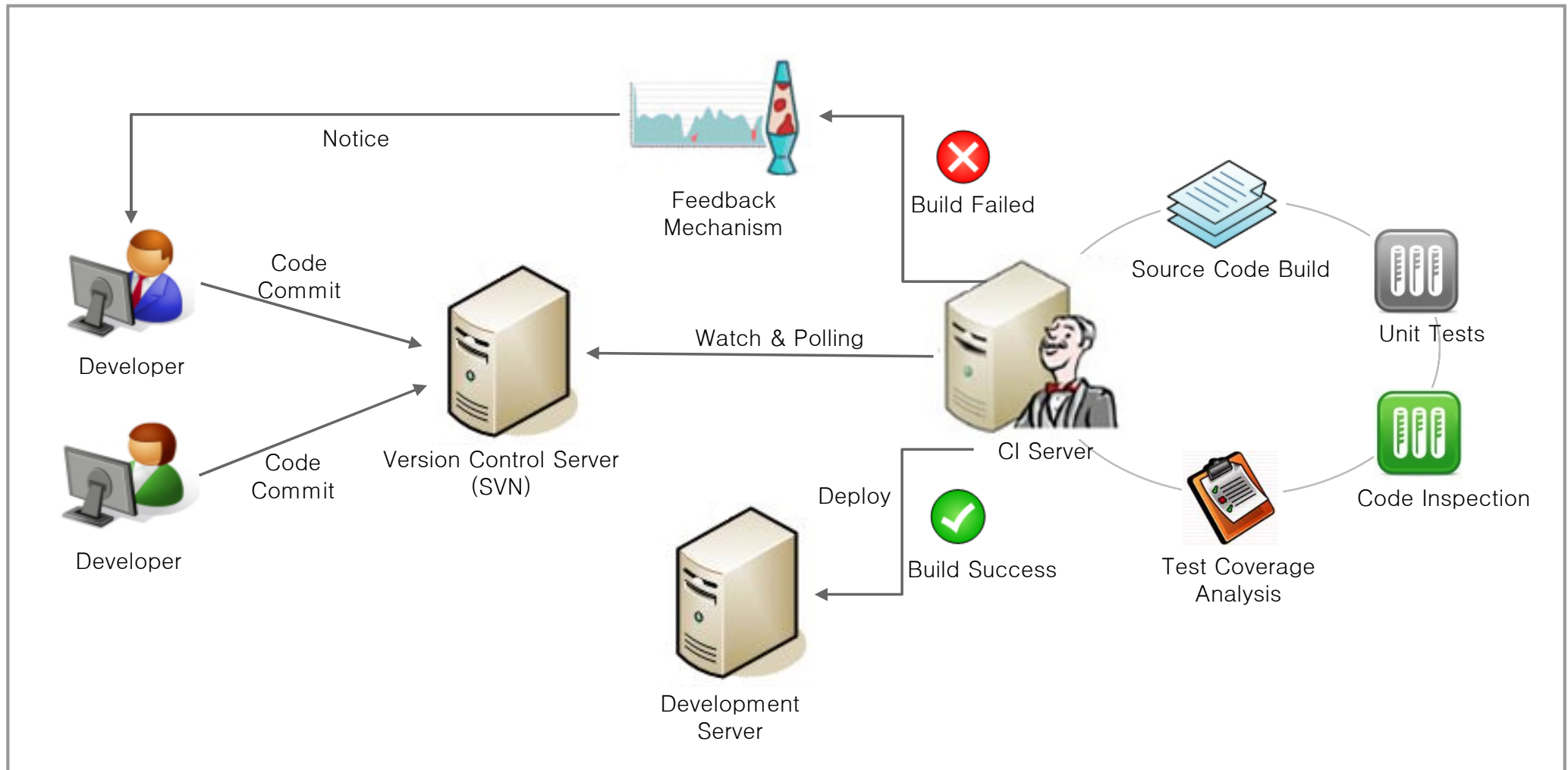
## 6. CI 서버



1. CI 정의
2. CI 효과와 특징
3. Jenkins 개요
4. Jenkins 특징 및 주요기능
5. Jenkins 프로젝트 메뉴
6. Jenkins - JUnit 테스트 연동
7. 참고자료

# 1. CI(Continuous Integration) 정의

여러 명으로 구성된 팀이 작업한 것을 자주-각 팀원이 하루에 한 번 이상, 매일 여러 번-통합하는 것을 가리키는 소프트웨어 개발 **Practice**로, 개발자, 버전 관리 저장소, 통합 빌드 머신 등으로 구성 됨



### □ 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.

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

The screenshot shows the Jenkins web interface. The top navigation bar includes the Jenkins logo, a search bar, and the user 'admin'. The left sidebar contains various links like '새로운 Item', '사람', '빌드 기록', etc. The main content area displays a table of build jobs. One job, 'sample.bulid', is shown with a successful status (green sun icon) and a duration of 3 min 14 sec. Below the table, there are links for 'Legend', 'RSS 모두', 'RSS 실패', and 'RSS 최근 빌드'.

S	W	Name ↓	최근 성공	최근 실패	최근 소요 시간
		sample.bulid	3 min 14 sec - #1	—	19 sec

아이콘: [S](#) [M](#) [L](#)

[Legend](#) [RSS 모두](#) [RSS 실패](#) [RSS 최근 빌드](#)

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

### □ Jenkins 특징

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

### □ Jenkins 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 스위트 실행을 통한 코드 품질 검증
- 빌드 후속 절차 자동화
  - 컴파일 된 코드의 패키징 및 테스트 리포팅 기능 제공

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

### 기본 메뉴

- **변경사항(Changes)**  
: 빌드 버전별로 SCM에서 지난 버전에 비해서 변경된 내용에 대해 변경자와 커밋시에 개발자가 추가한 comment를 확인할 수 있음
- **작업공간(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

Jenkins은 JUnit 테스트의 결과정보를 목록으로 보여주며, 단위테스트 결과 및 테스트 소요시간 등에 관한 리포트 기능을 제공

### JUnit 테스트 결과

**Test Result**

0 실패

1 테스트

Module	Fail	(diff)	Total	(diff)
egovframework.guide	0		1	+1

[상세 내용 입력](#)

**모든 테스트**

Package	실행시간	실패 (비교)	건너뛴 (비교)	Pass (비교)	총 (비교)
egovframework.guide.helloworld	0.5 sec	0	0	1 +1	1 +1

프로젝트 빌드에 따른 JUnit 테스트 빌드 결과를 보여줌

### JUnit 테스트 리포트

#### Test Result : egovframework.guide.helloworld

0 실패

1 테스트  
Took 0.5 sec.  
[상세 내용 입력](#)

#### 모든 테스트

Class	실행시간	실패 (비교)	건너뛴 (비교)	Pass (비교)	총 (비교)
HelloWorldServiceTest	0.5 sec	0	0	1 +1	1 +1

#### Test Result : HelloWorldServiceTest

0 실패

1 테스트  
Took 0.5 sec.  
[상세 내용 입력](#)

#### All Tests

Test name	Duration	Status
testSayHello	0.5 sec	Passed

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

## 7. 참고 자료

---

### ❑ CI 정의

- [http://en.wikipedia.org/wiki/Continuous\\_Integration](http://en.wikipedia.org/wiki/Continuous_Integration)

### ❑ Continuous Integration (Martin Fowler)

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

### ❑ Jenkins Home

- <https://jenkins.io/>