전자정부 표준프레임워크

개발환경



eGovFrame

Contents

1. _ 개발환경 개요
2. _ 구현도구
3. _ 개인 빌드 (Maven)





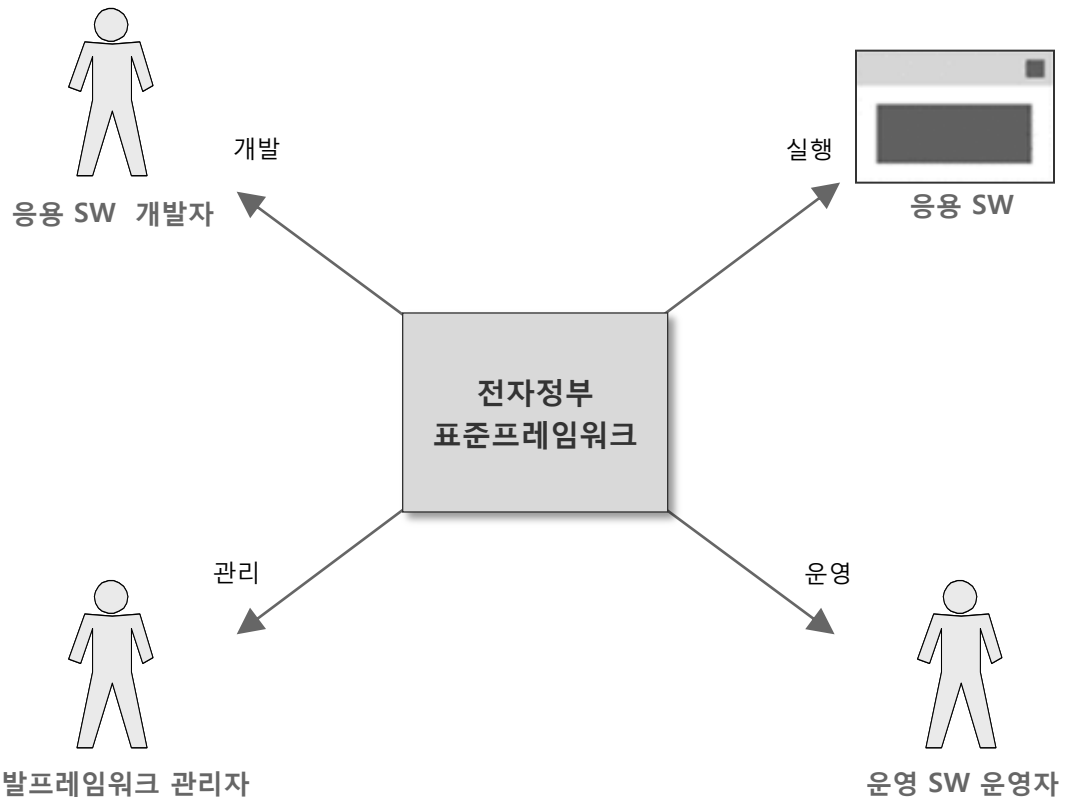
1. 개발환경 개요
2. 서비스 도출
3. 개발환경 구성요소
4. 적용 오픈 소스
5. 서비스 별 소개
6. 개발자 개발환경 구성
7. 서버 개발환경 구성

eGovFrame

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 개발프레임워크로부터의 개발 지원 기능이 필요함**



- 응용 SW는 전자정부 개발프레임워크를 기반으로 개발되는 어플리케이션으로써 어플리케이션의 효과적인 동작을 위한 전자정부 개발프레임워크의 실행 지원 기능이 필요함

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

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

2. 서비스 도출 (1/2)

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

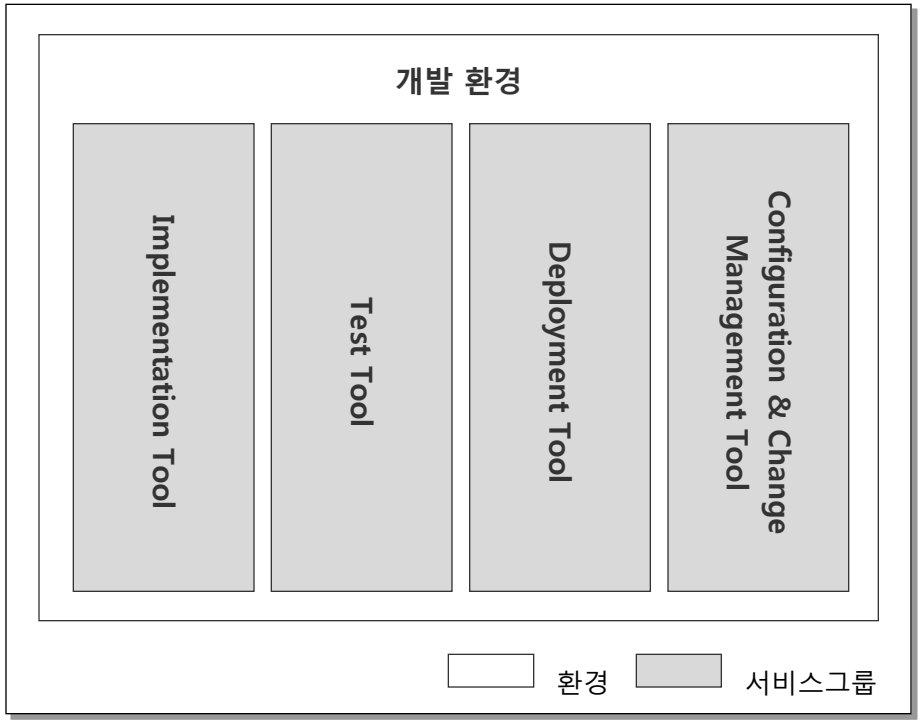
개발프레임워크 기술구조 사례 연구

개발프레임워크 (사업자 명)	요구분석자 지원 서비스	분석 설계자 지원 서비스 *	구현자 지원 서비스	테스터 지원 서비스	배포자 지원 서비스	형상/변경관리자 지원 서비스	프로젝트관리자 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, 성능 테스트	자동 빌드	형상관리	N/A
DevOn (LG CNS)	N/A	모델링, MDSD	개발환경 설치, IDE, Prototype, 코드 Inspection	단위 테스트, 성능 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 Transformation	디버깅, EMB Designer, ProMapper, DBIO Editor	Service Tester	Deploy Management	버전 관리, 개발자 권한 관리	N/A

2. 서비스 도출 (2/2)

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

개발 환경 서비스 그룹 도출

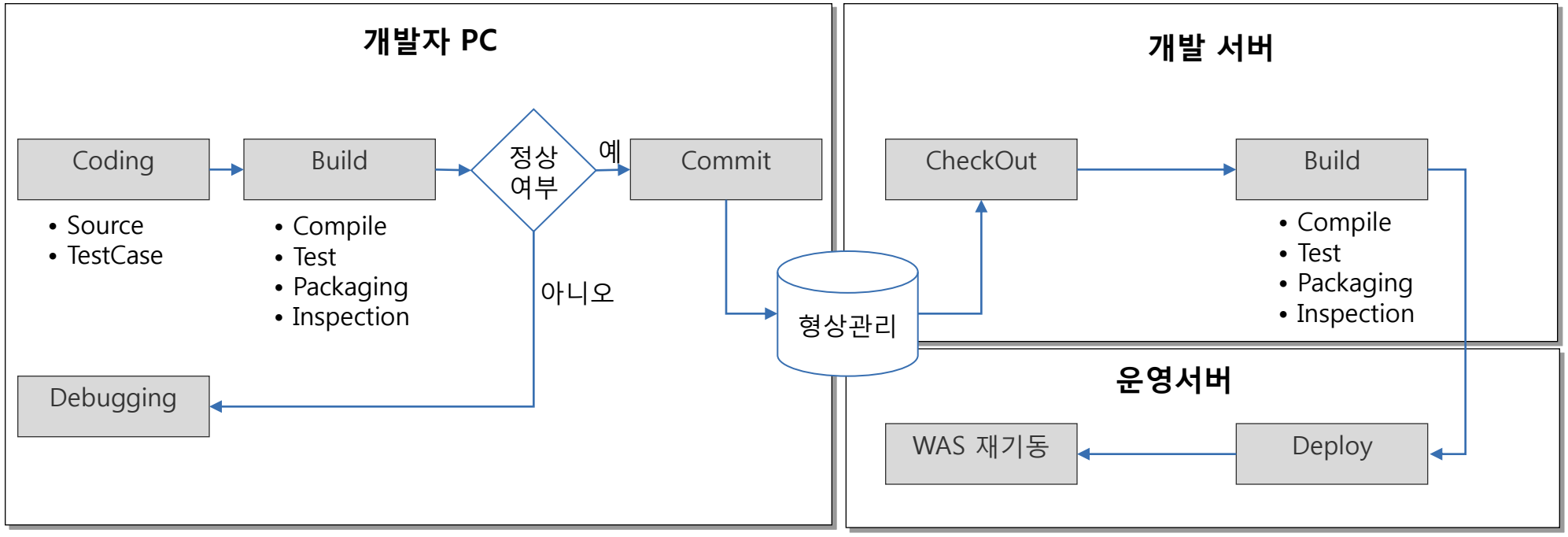


서비스 그룹	설명
Implementation Tool	• 업무 프로그램 구현을 지원하는 도구임
Test Tool	• 구현된 업무 프로그램의 테스트를 지원하는 도구임
Deployment Tool	• 구현 완료된 업무 프로그램을 실행 환경에 배포 가능한 형태로 패키징하고, 패키징된 업무 프로그램을 실행 환경에 배포하는 도구임
Configuration & Change Management Tool	• 형상 및 변경 관리 지원 도구임

3. 개발환경 구성요소

개발환경은 개발자 PC에서 개발서버, 운영서버로 연결되는 개발 흐름을 지원함

Implementation Tool		Test Tool	Deployment Tool	Conf. & Change Mgt.
Debug	Code Generation	Test Reporting	Build	Configuration Mgt.
Editor	Code Inspection	Unit Test	Deployment	Change Management
Methodology & Template	Batch	Batch Test		



4. 개발환경 선정 오픈소스(1/2)

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

4. 개발환경 선정 오픈소스(2/2)

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	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

5. 서비스 별 소개 - 구현도구

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

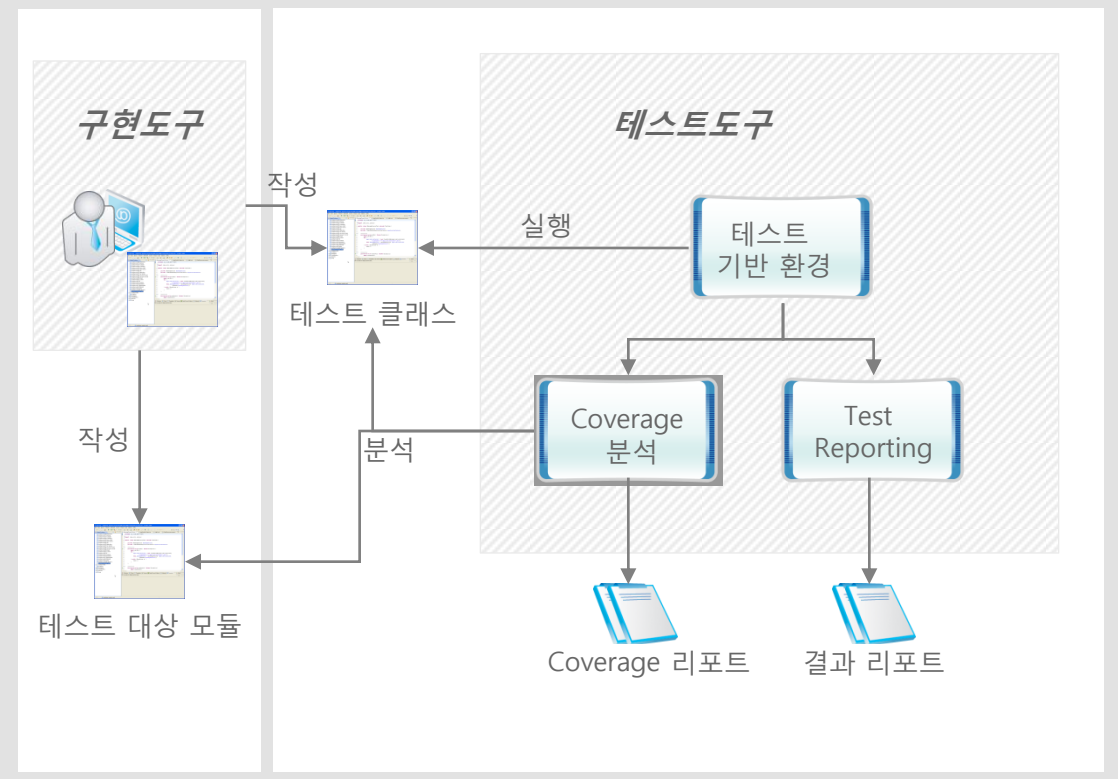
구성요소 별 상세

구성요소	설명
에디터	• 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 Edit
디버거	• 편집모드와 디버깅 모드의 전환을 통한 작업의 편의성 향상 • Remote 디버깅 제공
방법론(개발절차)	• 애플리케이션을 개발 가이드 제공
템플릿 프로젝트	• 애플리케이션 개발 시 자동으로 템플릿을 생성 마법사 제공
공통 컴포넌트	• 애플리케이션 개발 시 마법사를 통해 공통컴포넌트를 추가 기능 제공
맞춤형 개발환경	• 프로젝트에서 필요로 하는 기능만을 선택적으로 구성할 수 있는 도구 제공
배치 템플릿 프로젝트	• New eGovFrame Batch Template Project 생성 마법사 실행 • 배치 처리 구성 별 예제 제공
배치 Job 생성	• 배치 작업 파일 생성 마법사 실행 • 배치 Job / Step / Item 설정으로 XML 파일 생성
배치 Job Launcher	• 배치 작업 실행 파일 생성 마법사 실행 • 배치 처리 구성 시 최초에 필요한 Launcher, Repository의 설정 XML 파일 생성

5. 서비스 별 소개 – 테스트 도구

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

주요 메커니즘



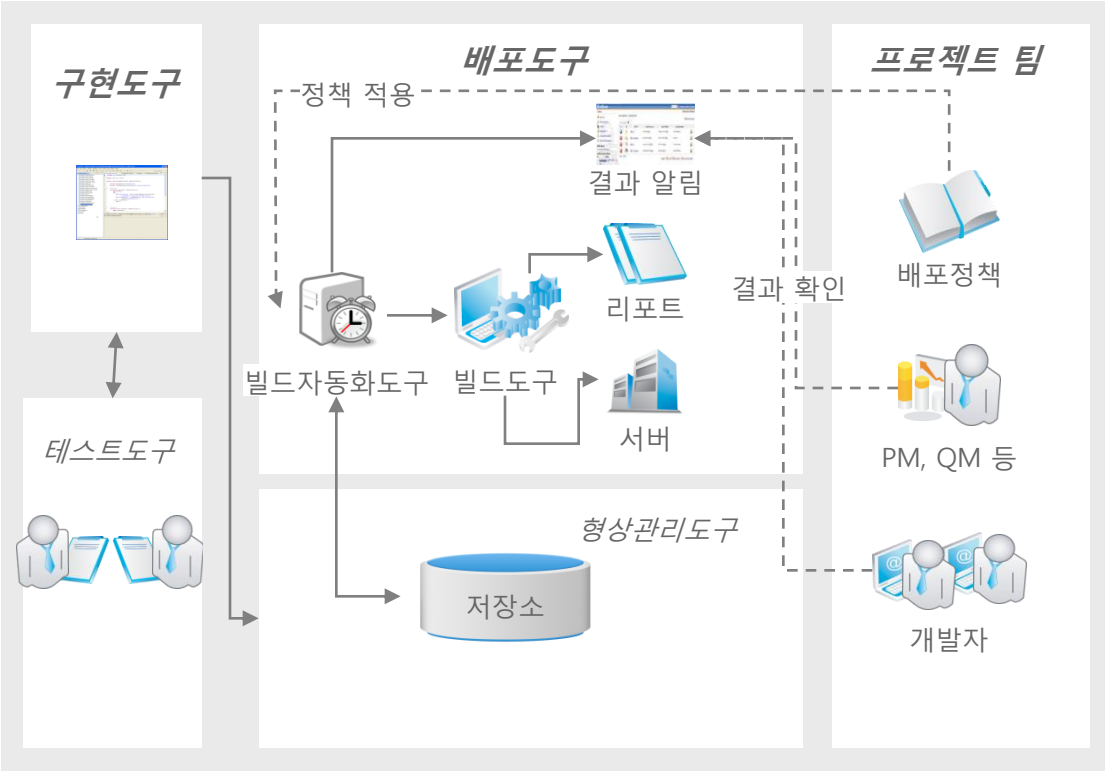
구성요소별 상세

구성요소	설명
테스트 기반환경	• Junit 기반의 Unit Test 작성 지원 • Mock 및 DB Support
Test Reporting	• Eclipse 기반의 Green Bar GUI • HTML/XML/Excel 형식의 Reporting 지원
Coverage 분석	• TestCase가 커버하는 정도를 수치 및 코드 영역을 통해 확인
Batch Job Test	• 배치 관련 파일을 활용하여 간단히 테스트할 수 있는 테스트 마법사 실행

5. 서비스 별 소개 – 배포 도구

Maven 기반의 개발자 빌드와 CI서버를 통한 지속적인 통합을 지향

주요 메커니즘



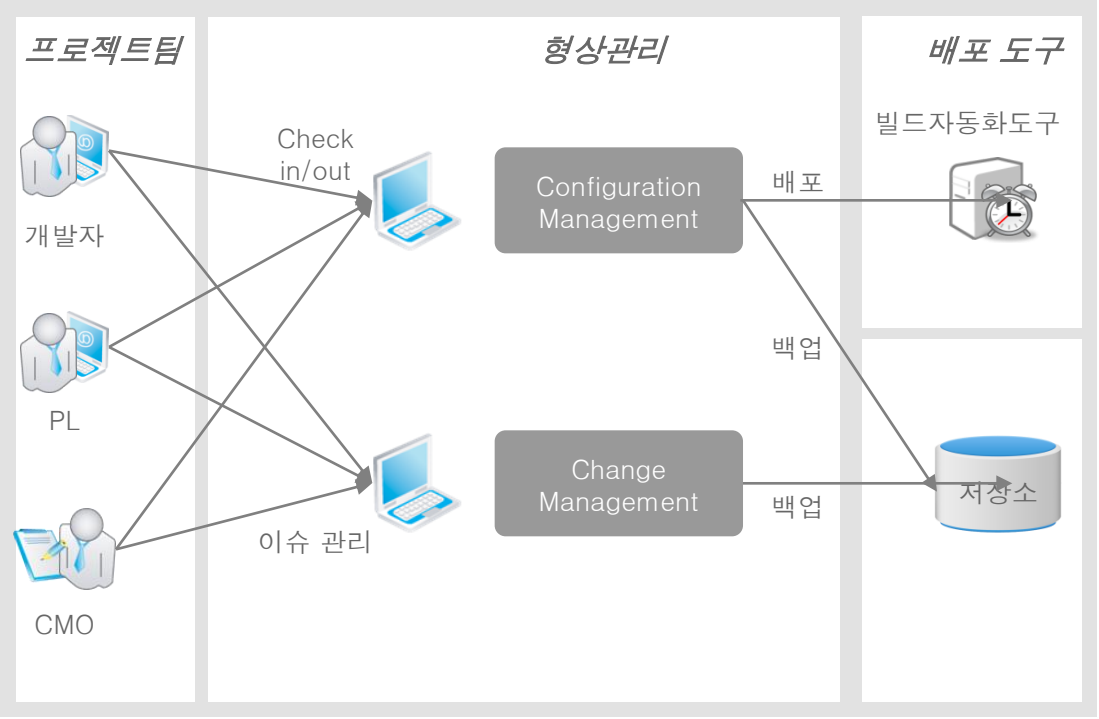
구성요소별 상세

구성요소	설명
빌드 도구	• Maven 기반의 빌드 수행 • 3rd Party 라이브러리 관리 편의성을 위한 Nexus Repository 제공
빌드 자동화 도구	• 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

5. 서비스 별 소개 – 형상관리

형상요소의 식별, 이력관리를 지원하는 형상도구와 JAVA기반의 이슈관리시스템 제공

주요 메커니즘



구성요소별 상세

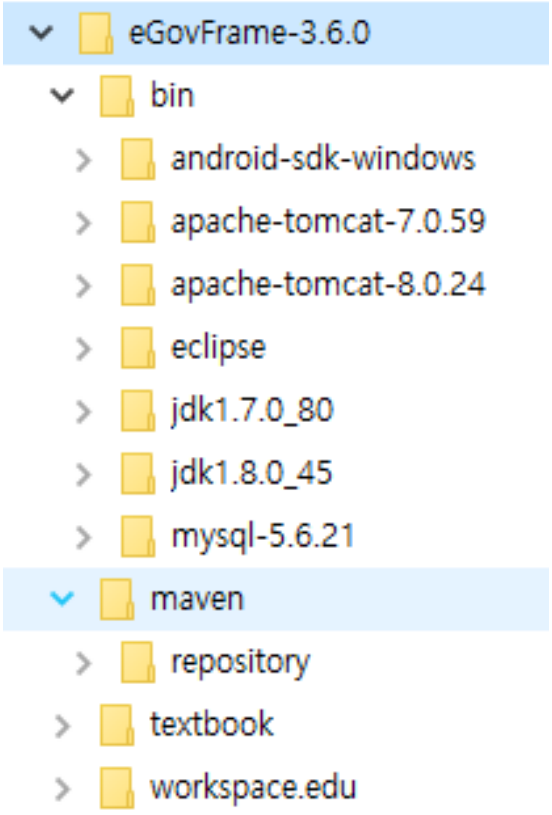
구성요소	설명
Config. Mgmt	• Subversion을 그대로 활용 • 이슈ID 등록을 통한 이슈 연계
Change Mgmt	• JAVA기반의 이슈관리 시스템 • 이슈 등록, 검색, 처리 기능 제공

6. 개발자 개발환경 구성 (1/9)

❑ 설치

– eGovFrame-3.6.0.exe 파일을 C:\ 드라이브에서 실행한다.

❑ 디렉토리 정보

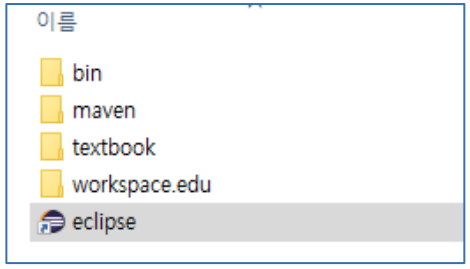


디렉토리	설명
bin	실행파일
android-sdk-windows	윈도우용 안드로이드 SDK 폴더
apache-tomcat-7.0.59	**Apache Tomcat**
apache-tomcat-8.0.24	
eclipse	개발환경이 포함된 **Eclipse JEE Mars.2 Release (4.5.2)**
jdk1.7.0_80	**JDK_HOME**
jdk1.8.0_45	
mysql-5.6.21	실습용 **DB**
maven/repository	**Local Maven Repository**
textbook	교육 교재파일
workspace.edu	**Eclipse Workspace**

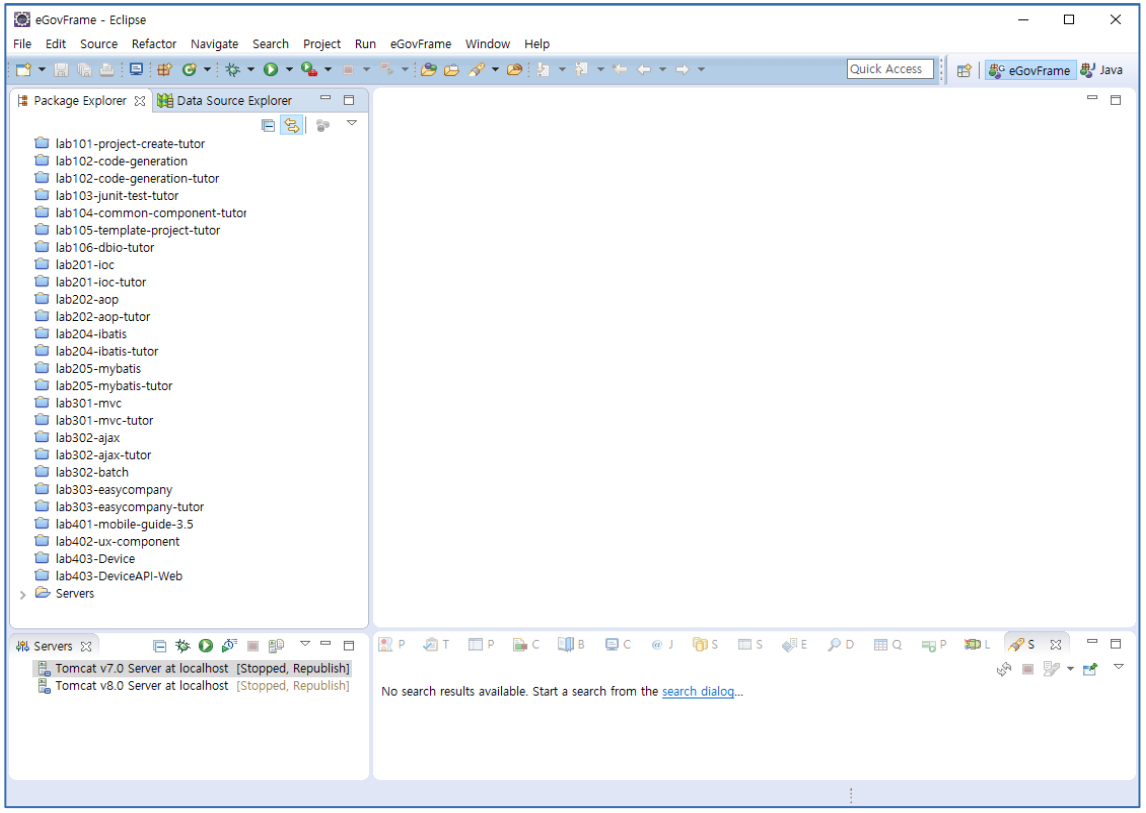
6. 개발자 개발환경 구성 (2/9)

❑ 실행

– “C:\eGovFrame-3.6.0\eclipse” 를 더블 클릭하여 Eclipse를 실행한다.



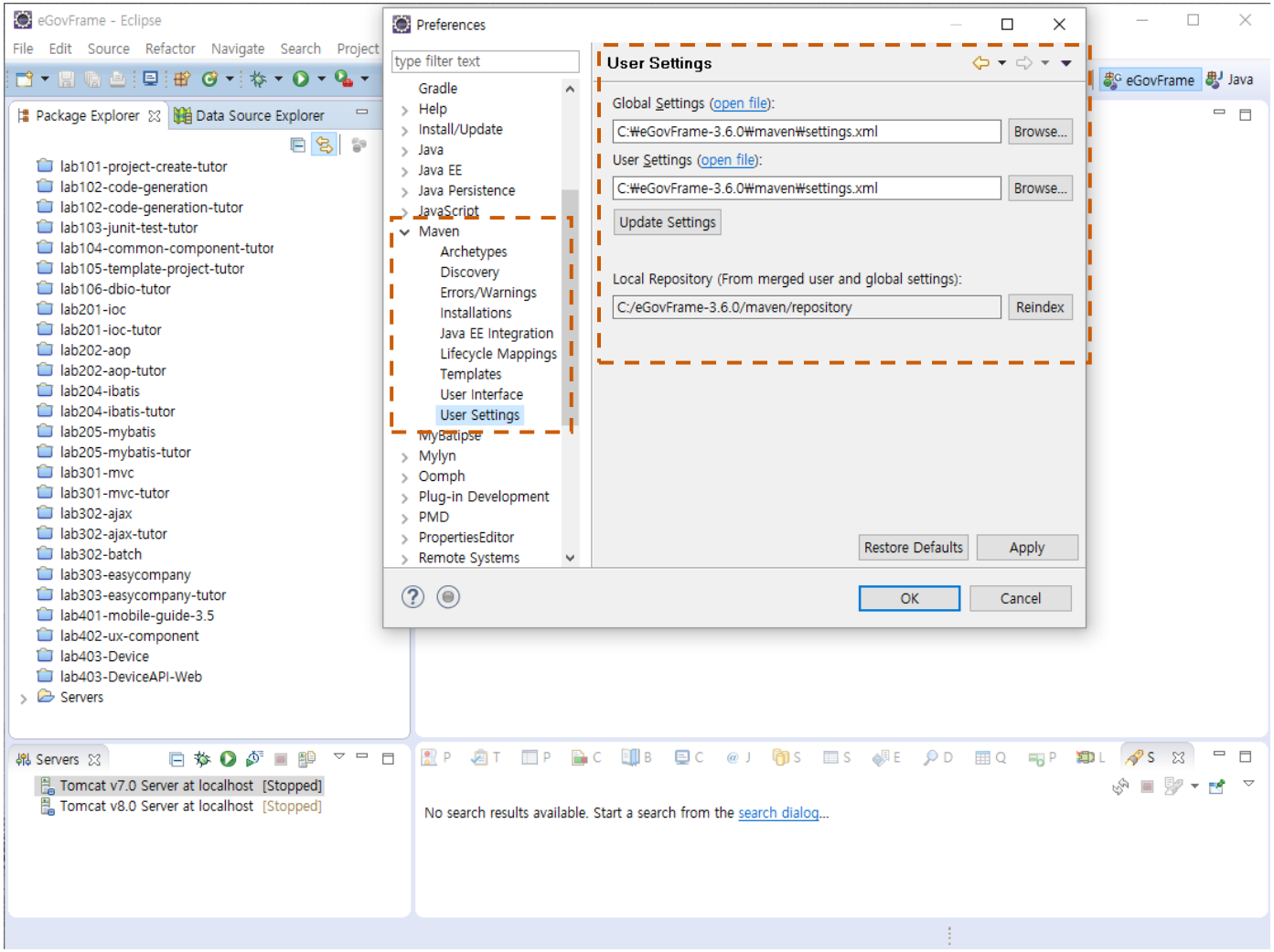




6. 개발자 개발환경 구성 (3/9)

❑ 설정

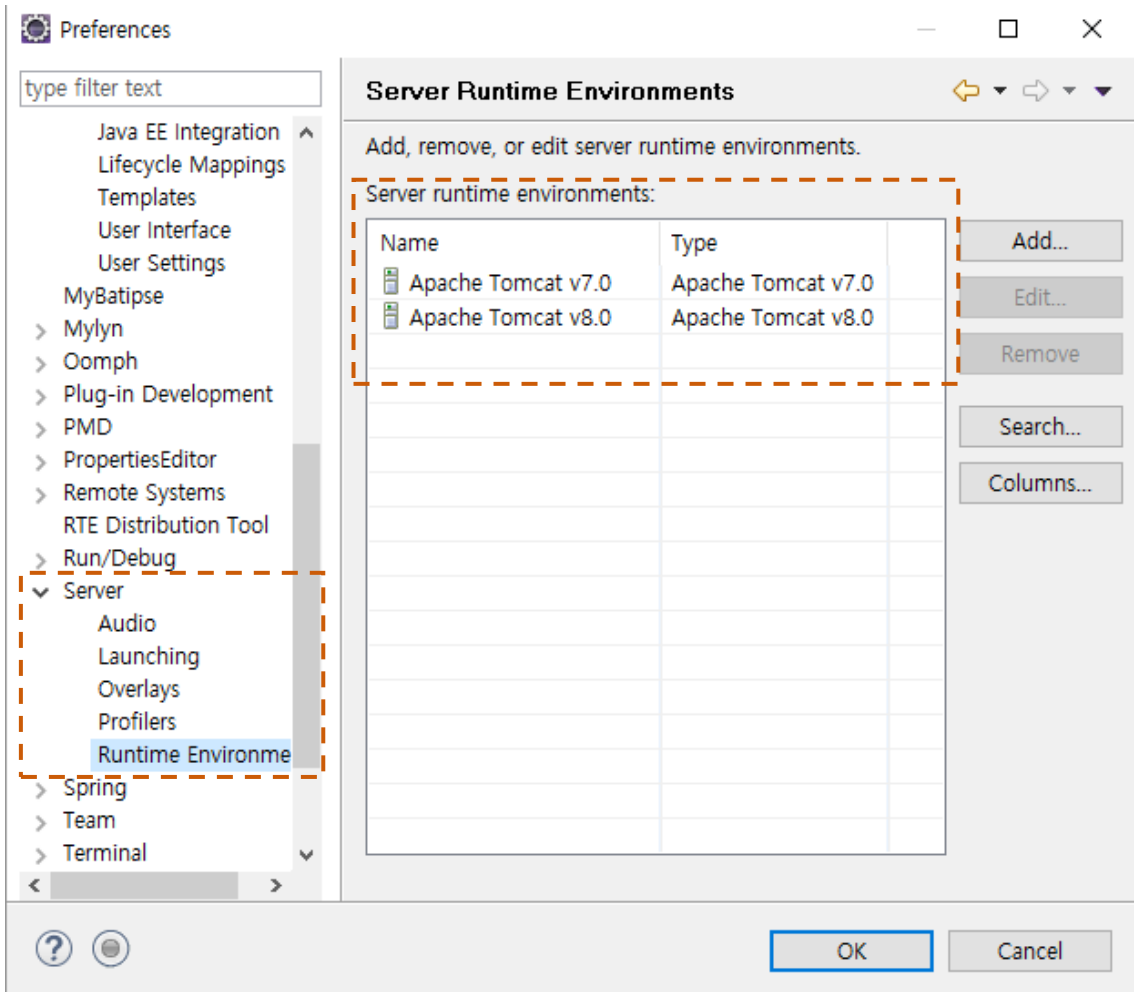
– Maven Installations 설정 확인



6. 개발자 개발환경 구성 (4/9)

❑ 설정

– Apache Tomcat 설정

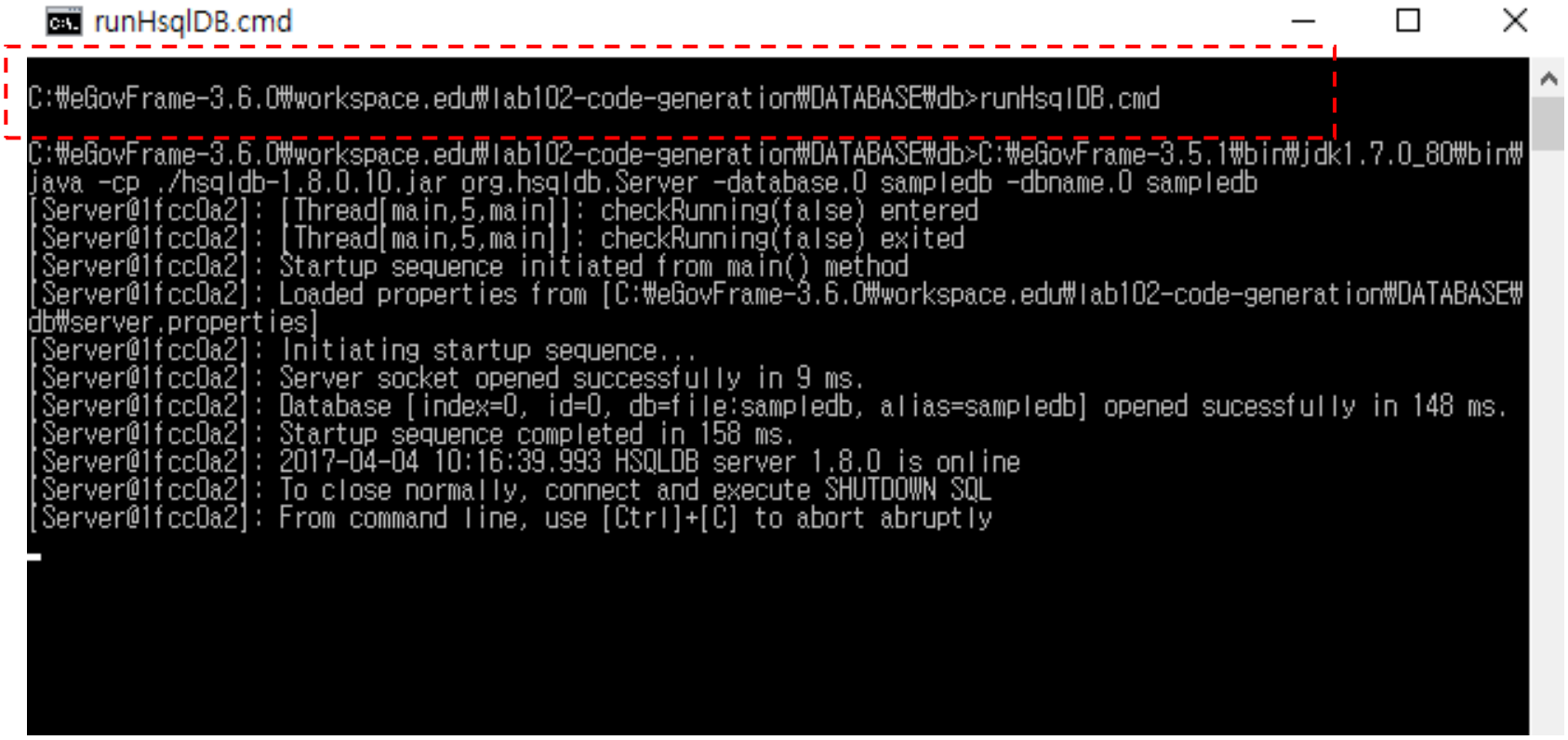


6. 개발자 개발환경 구성 (5/9)

❑ 설정

– 대상 프로젝트 DB 구동을 확인한다. (example project name: lab102-code-generation)

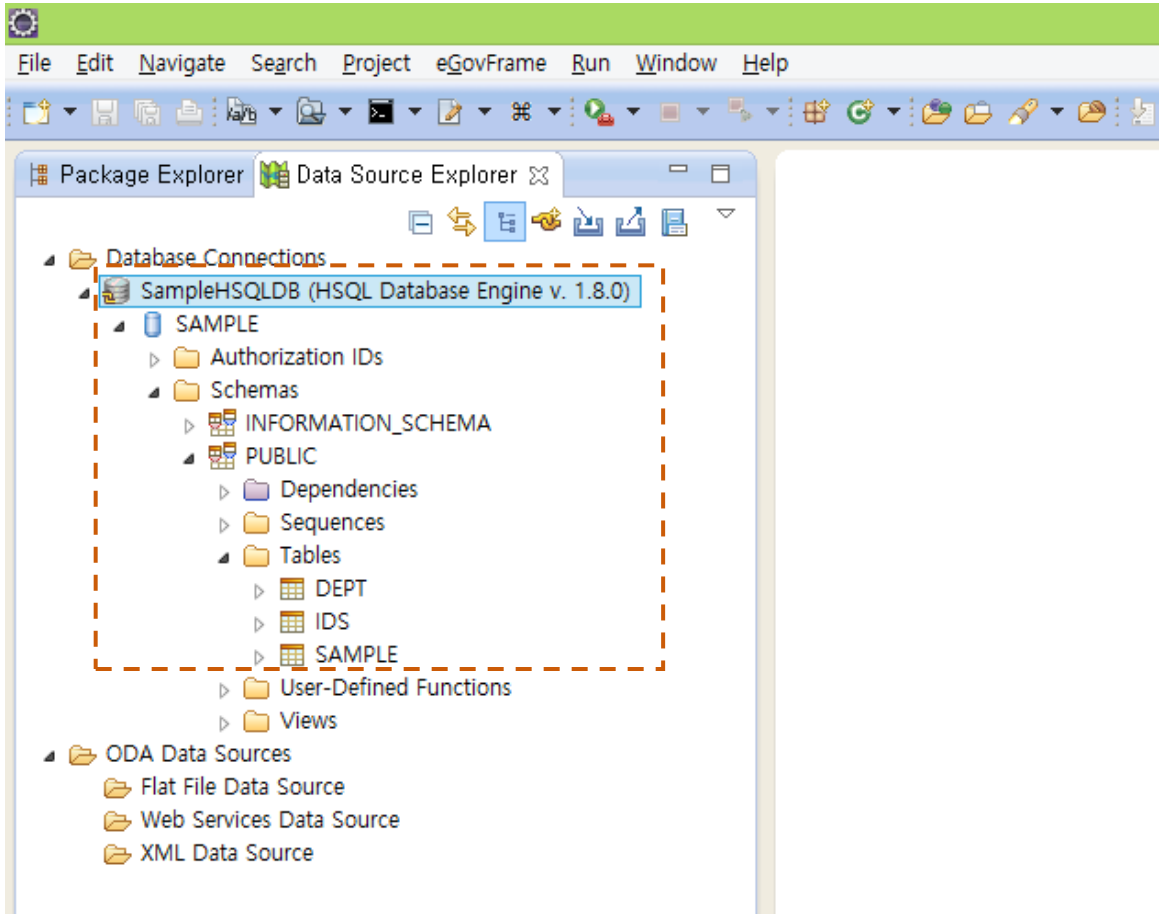
– C:\egovFrame-3.6.0\workspace.edu\lab102-code-generation\DATABASE\db\runHsqlDB.cmd”를 더블클릭하여 Sample용 DB를 띄운다.



6. 개발자 개발환경 구성 (6/9)

❑ 설정

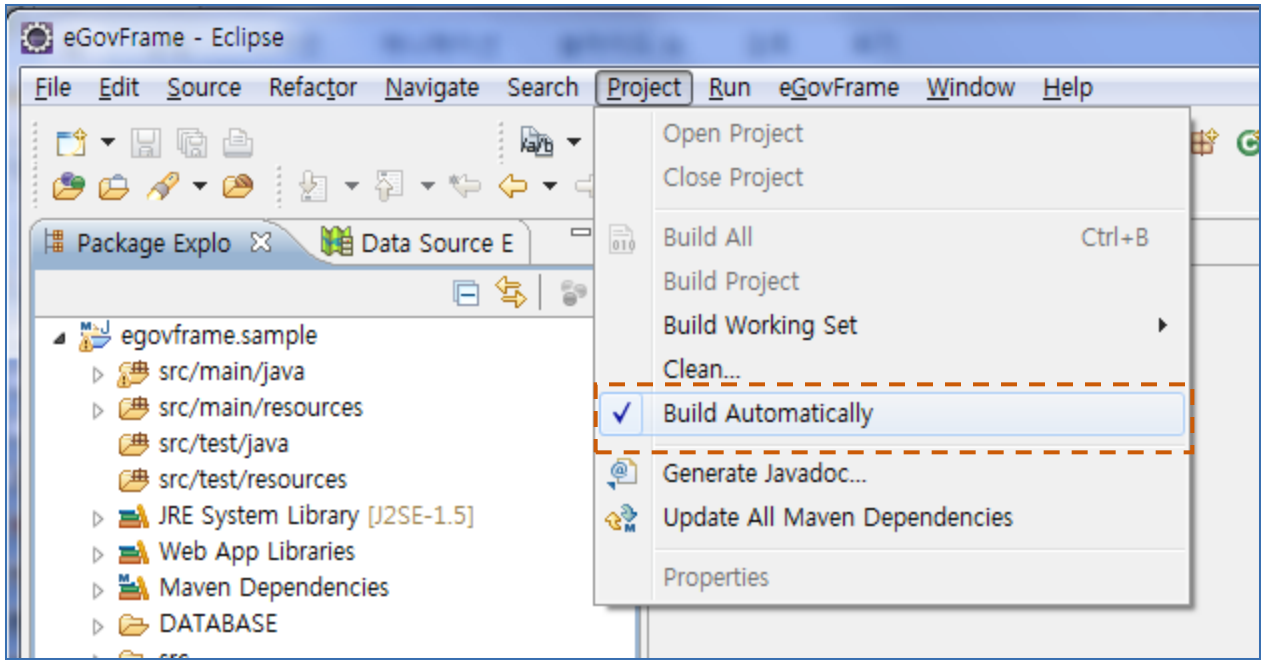
– Eclipse의 “Data Source Explorer” View에서 “SampleHSQLDB”에서 오른쪽 버튼의 “connect”를 선택하여 다음과 같이 테이블이 설정되어 있는지 확인한다.



6. 개발자 개발환경 구성 (7/9)

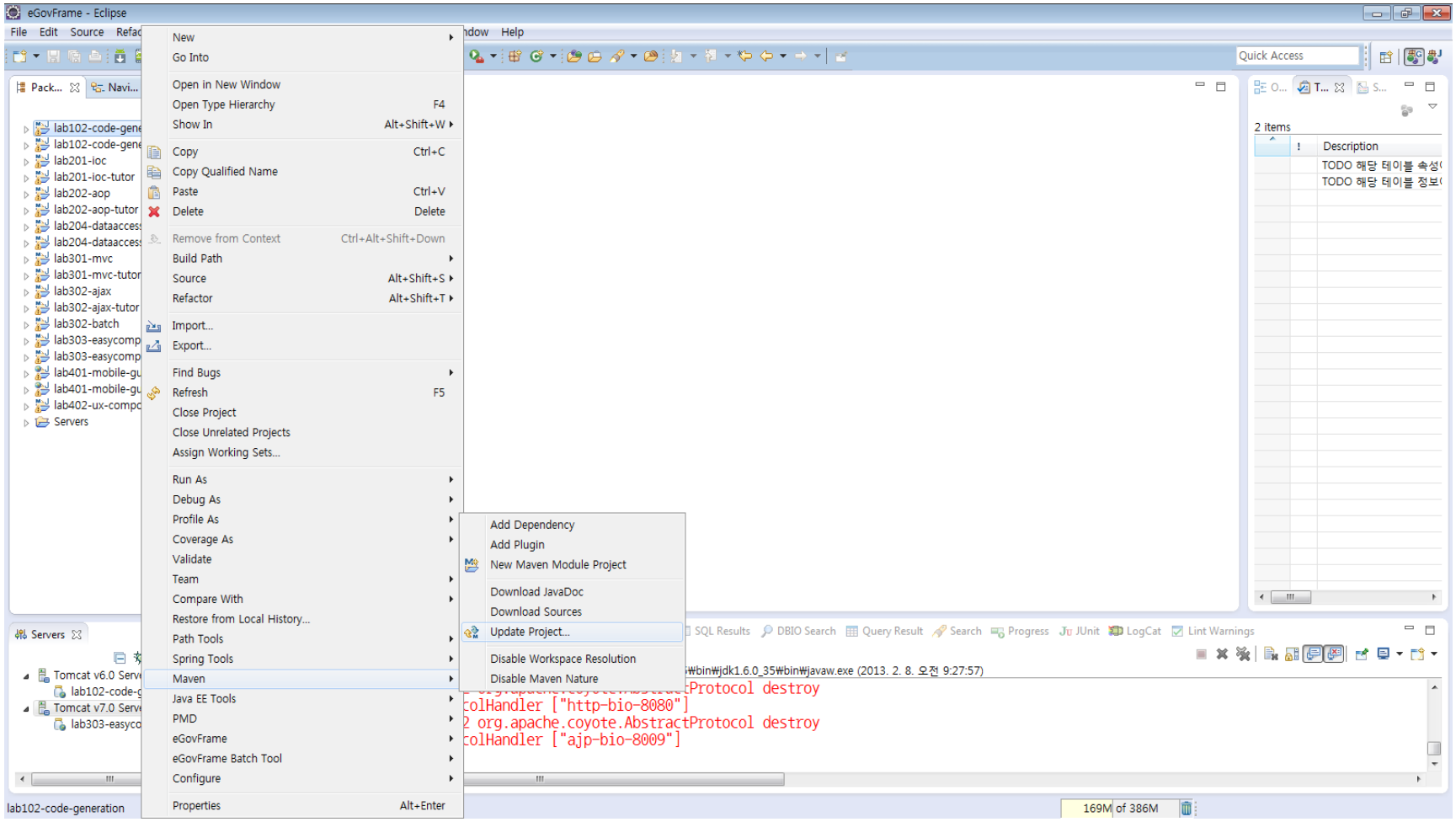
❑ 설정

– Eclipse에서 “Build Automatically”가 선택되어 있다.



❑ 설정

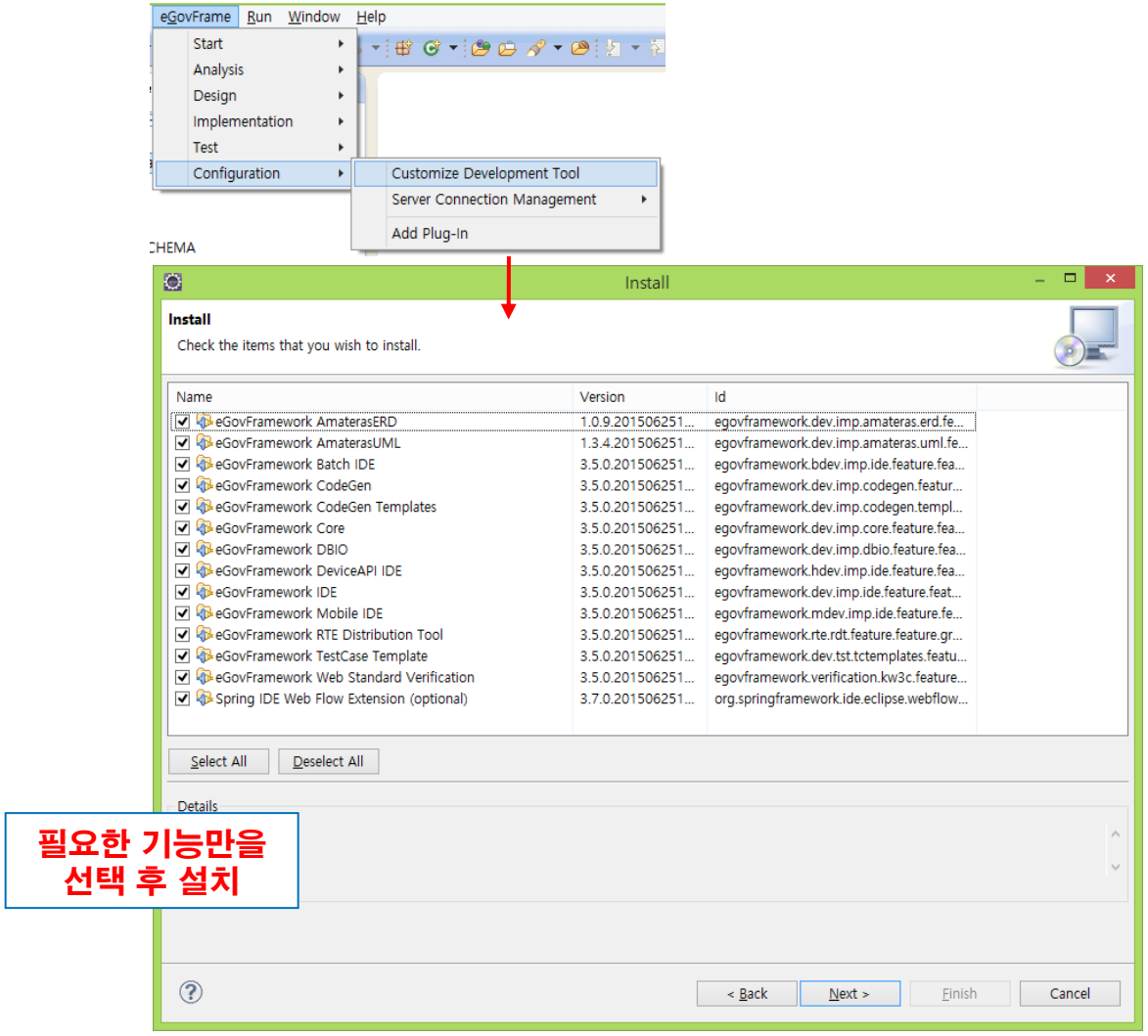
– 프로젝트 앞에 빨간 아이콘이 있어 오류가 있다면 Maven → Update Project 을 실행해본다.



6. 개발자 개발환경 구성 (9/9)

❑ 개발자 개발환경 선택적 구성

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



필요한 기능만을 선택 후 설치

❑ 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)

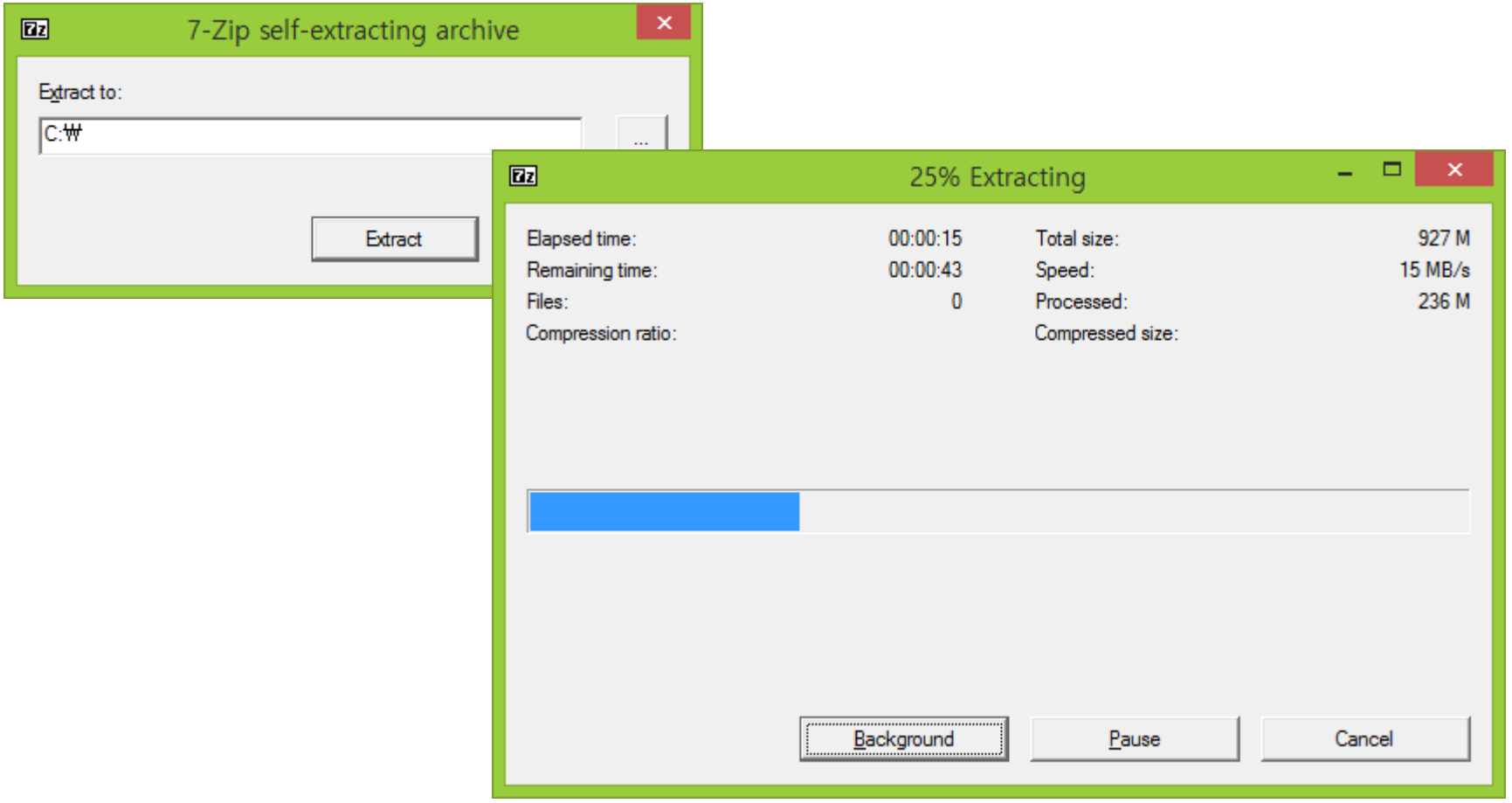
❑ 다운로드

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



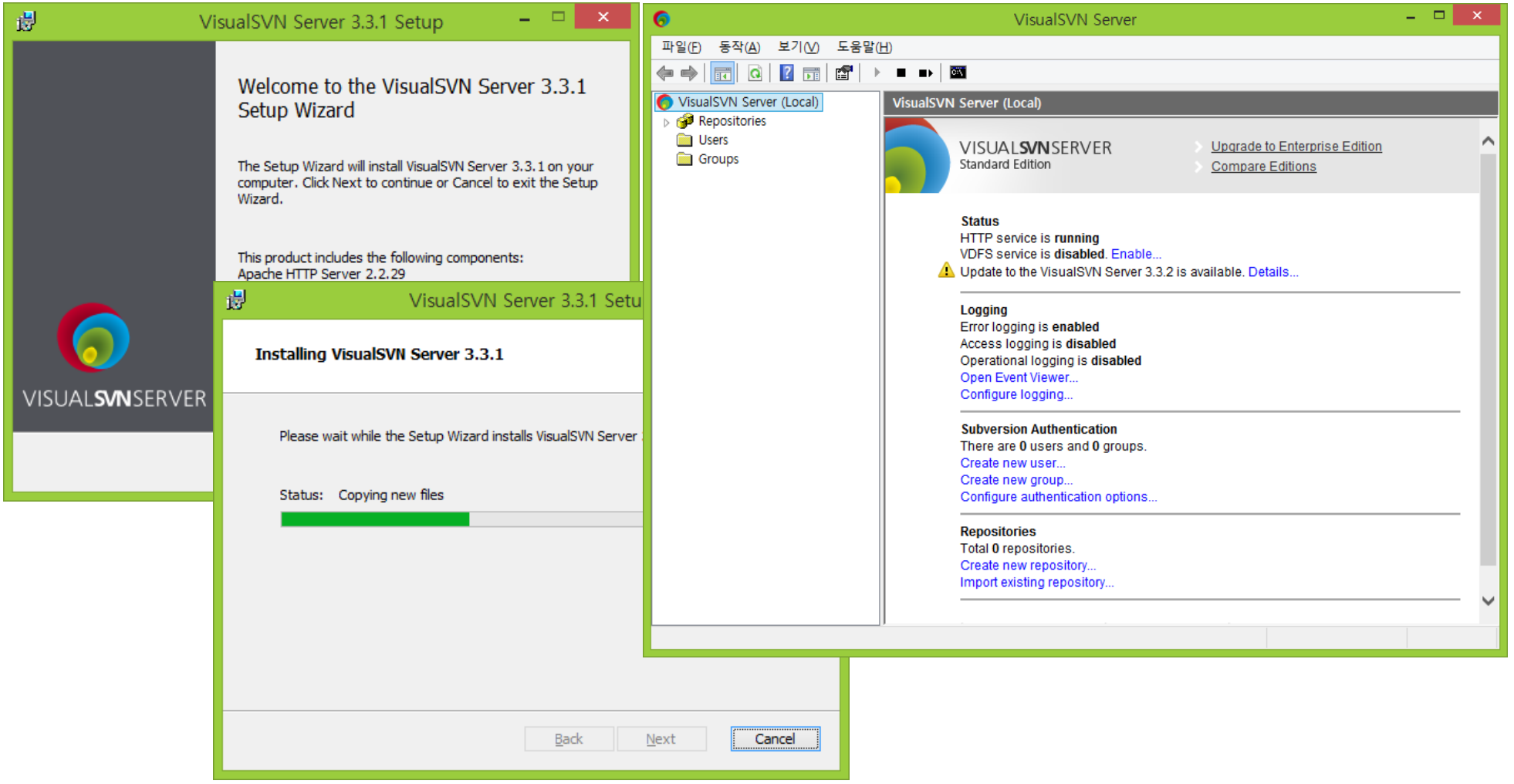
❑ 설치

– eGovCI-3.5.0_XXbit.exe 파일을 실행하여 압축을 해제한다.



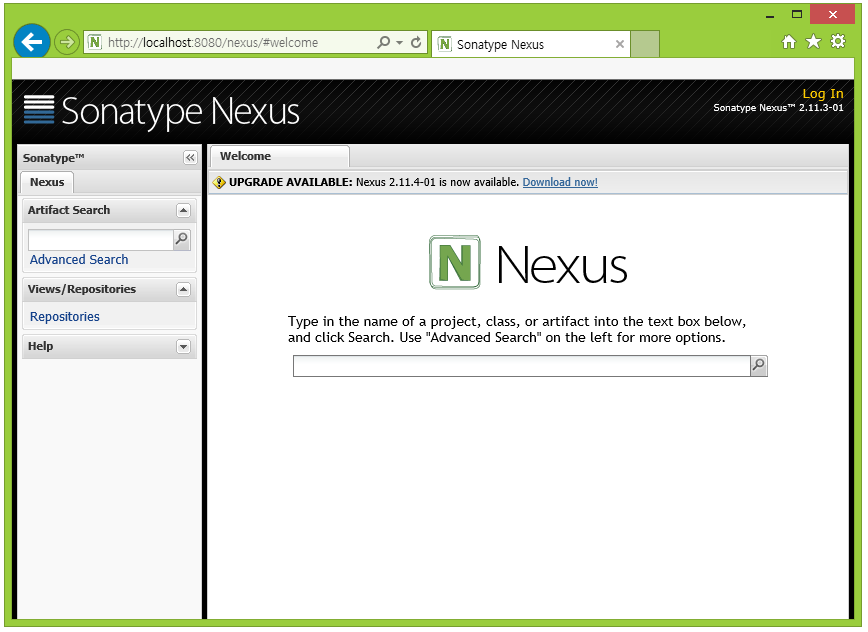
❑ 설치

– C:\eGovCI-3.5.0_XXbit\bin\VisualSVN-Server-3.3.1-x64.msi 파일을 실행하여 VisualSVN을 설치한다.

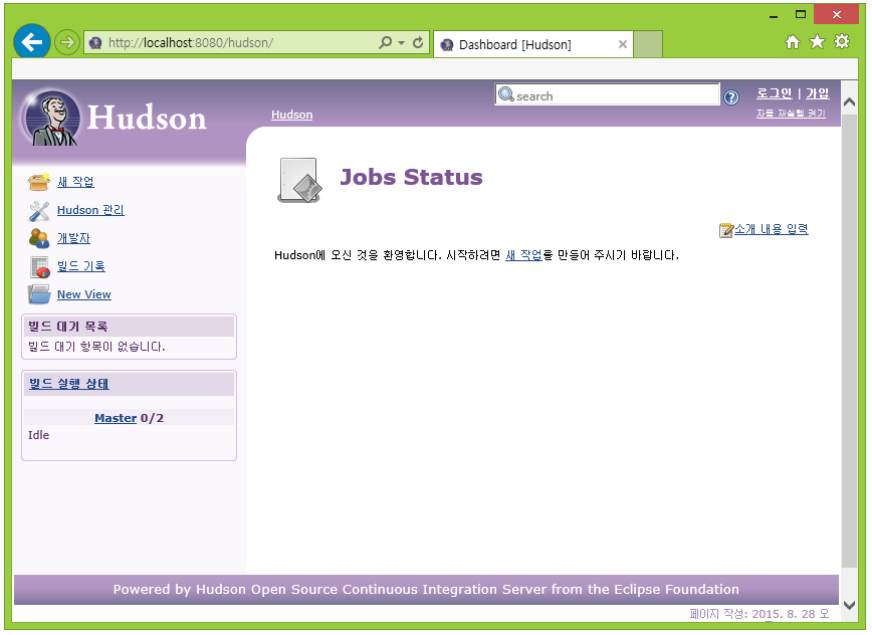


❑ 실행

- C:\eGovCI-3.5.0_XXbit\start.bat 파일을 실행하여 진행한다.
- 브라우저 주소 입력창에 Nexus, Hudson가 설치된 URL을 입력하여 설치 여부를 확인할 수 있다.



[그림 1] Nexus가 설치된 모습

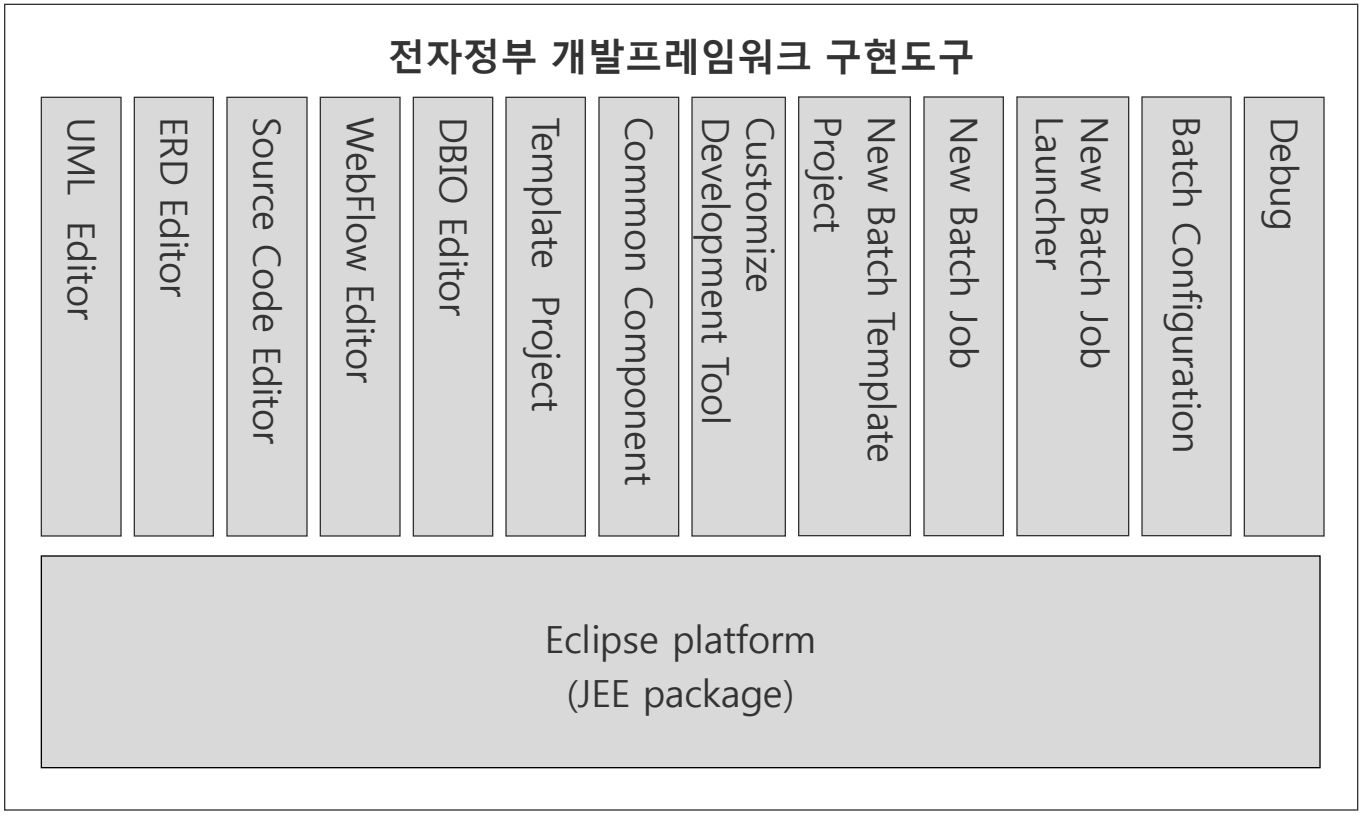


[그림 2] Hudson이 설치된 모습



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. Sever Connection Management

구현도구는 **UML, ERD, Source Code, WebFlow, DBIO Editor**와 **Template Project, Common Component, Customize Development Tool,** 배치 **Template** 및 **Job** 생성 그리고 **Debug**로 구성



1. 개요 - Package 구성

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.

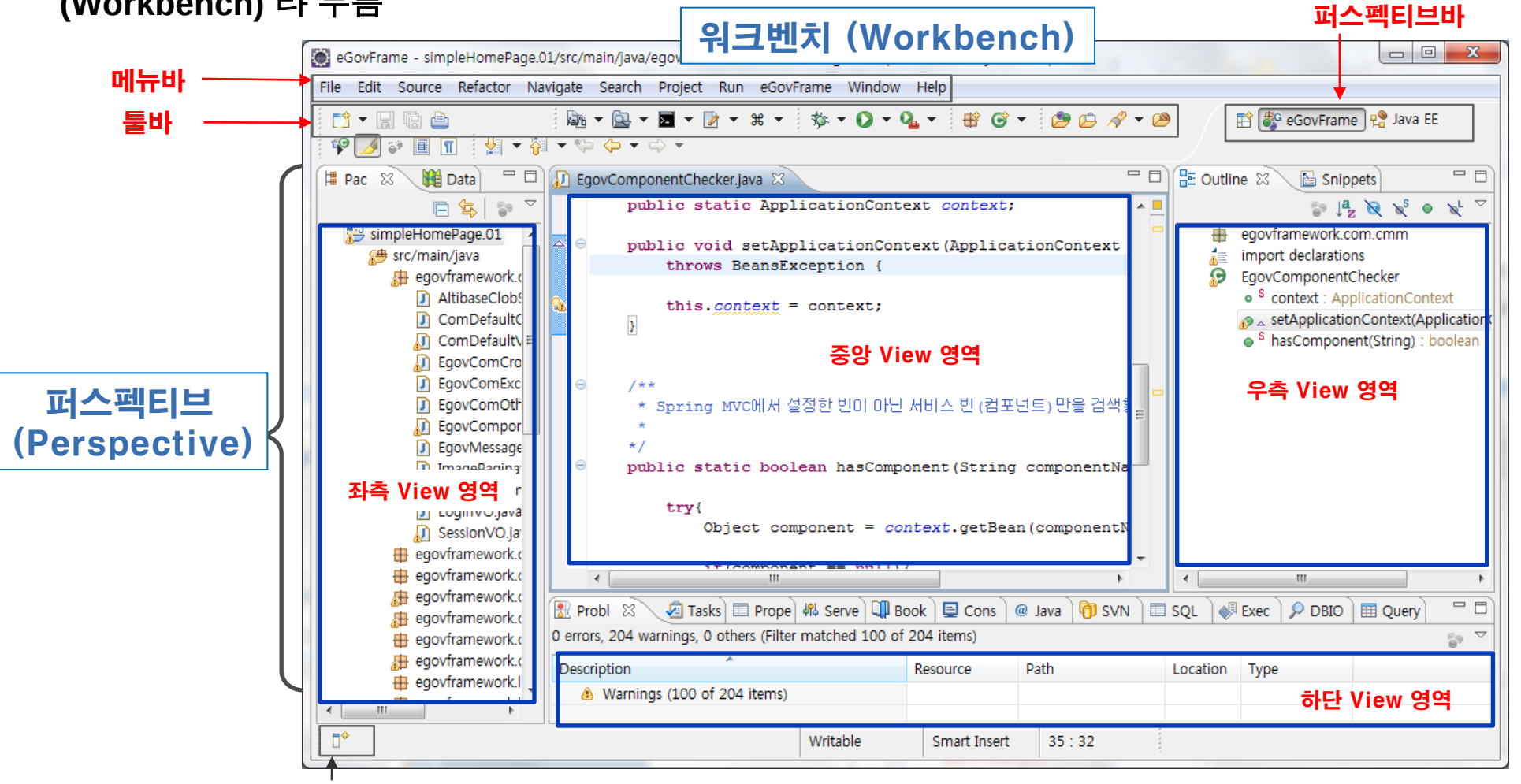
- RCP/Platform(Rich Client Platform)
- CVS(Concurrent Versions System)
    - EMF(Eclipse Modeling Framework)
- GEF(Graphical Editing Framework)
- JDT(Eclipse Java development tools)
- Mylyn
- WTP (Web Tools Platform)
  : Web Tools, Java EE Tools , XML Tools
- RSE(RSE, Target Management)
- Eclipse Link
- 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

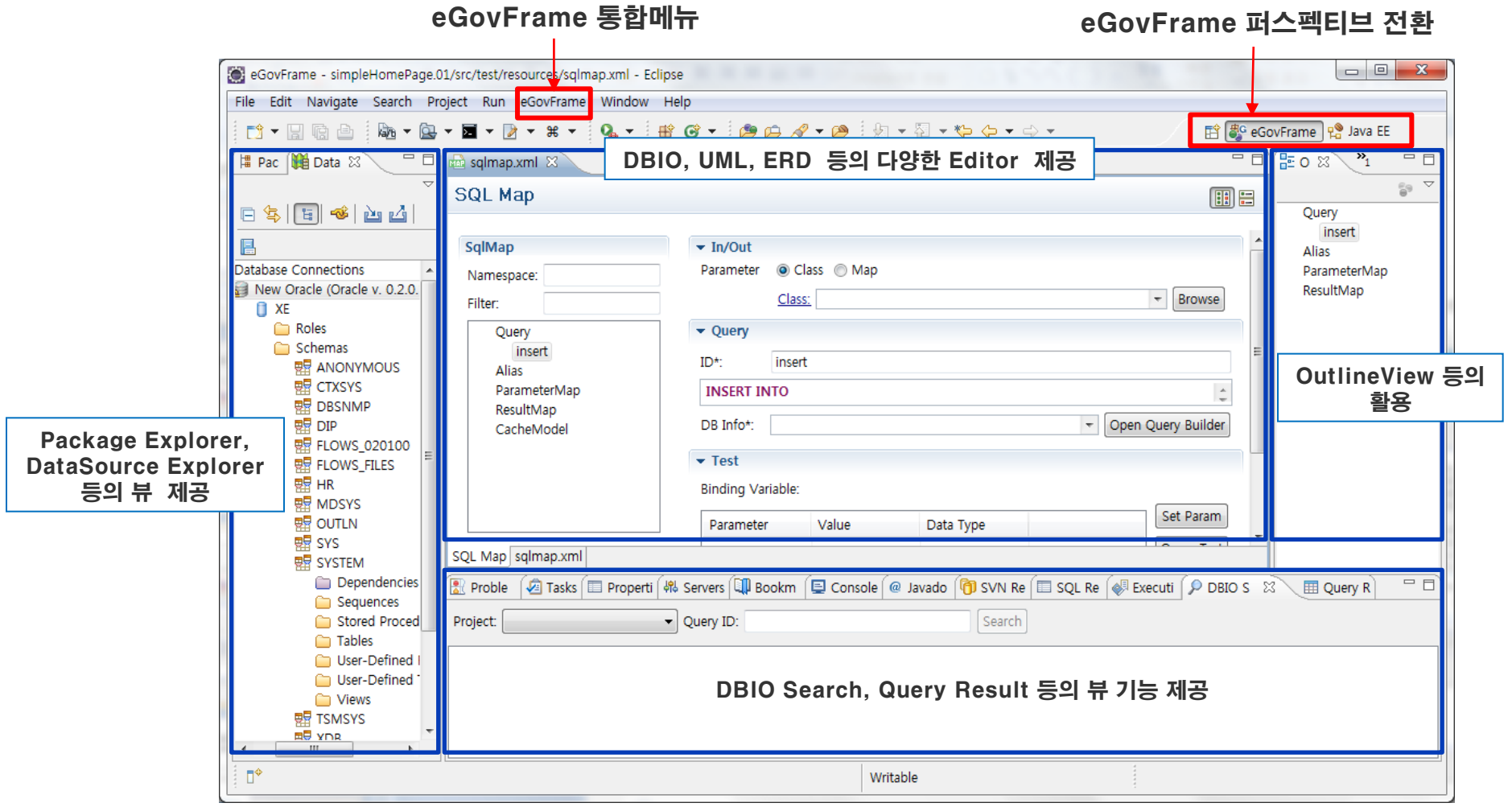
Eclipse 기본 화면은 메뉴바, 툴바, 퍼스펙티브바, 4개의 뷰영역으로 크게 구분된다. 이 4개의 뷰를 포함하는 전체 영역을 퍼스펙티브(**Perspective**) 라고 하고, 이 **Eclipse Window** 전체를 워크벤치(**Workbench**) 라 부름



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

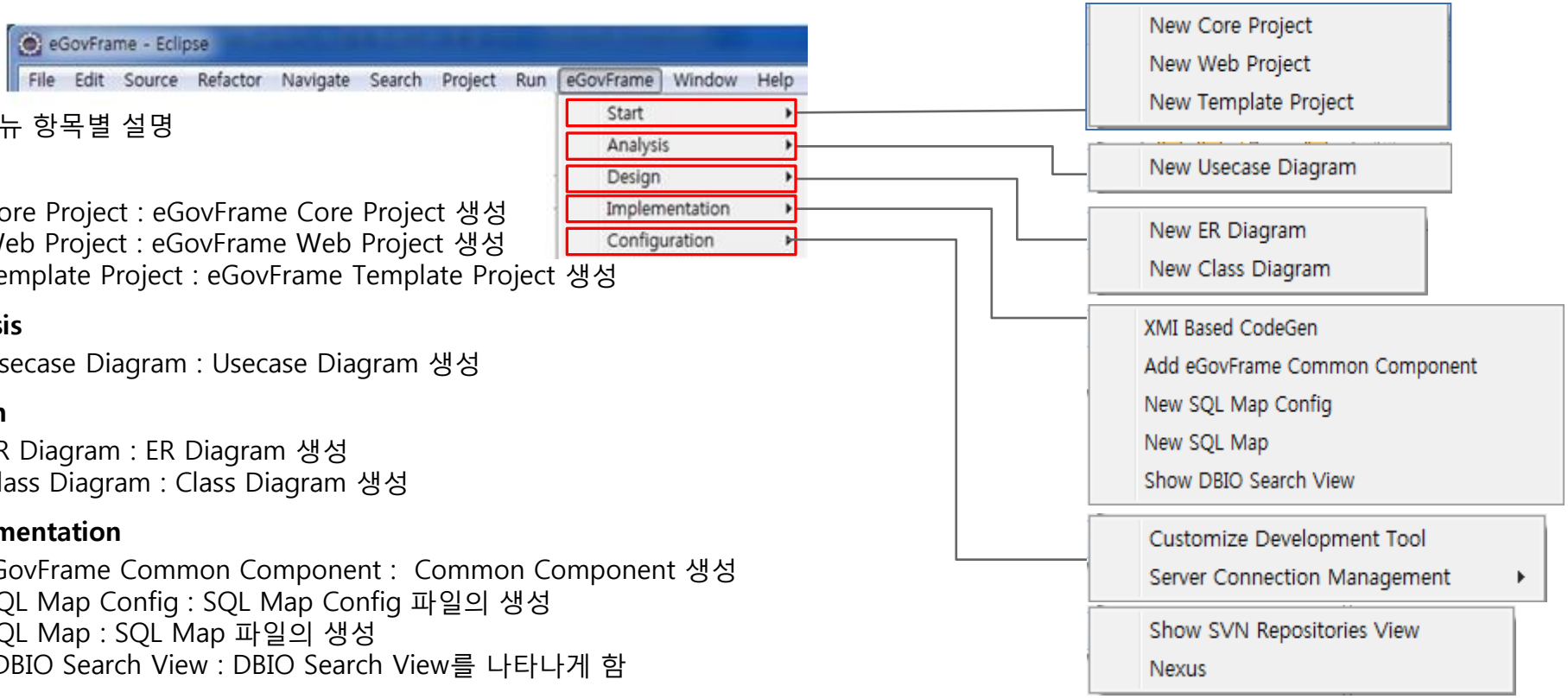
1. 개요 – 화면 구성

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



2. eGovFrame IDE (1/2) - eGovFrame Menu

eGovFrame Perspective에서만 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

❑ **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

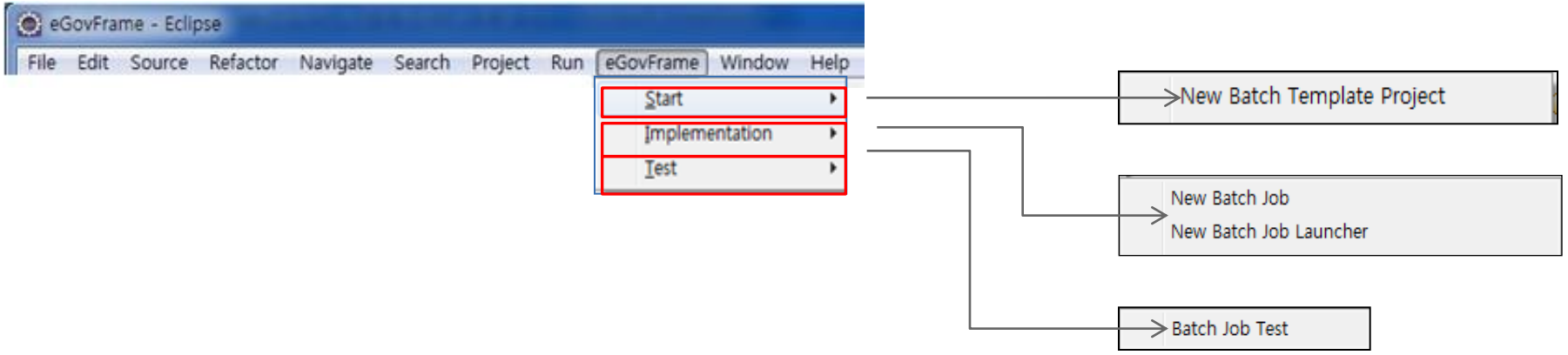
❑ **Design**
- New ER Diagram : ER Diagram 생성
- New Class Diagram : Class Diagram 생성

❑ **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

2. eGovFrame IDE (2/2) – eGovFrame Batch Menu

eGovFrame Perspective에서만 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

3. UML 작성

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

❑ UseCase Diagram 작성

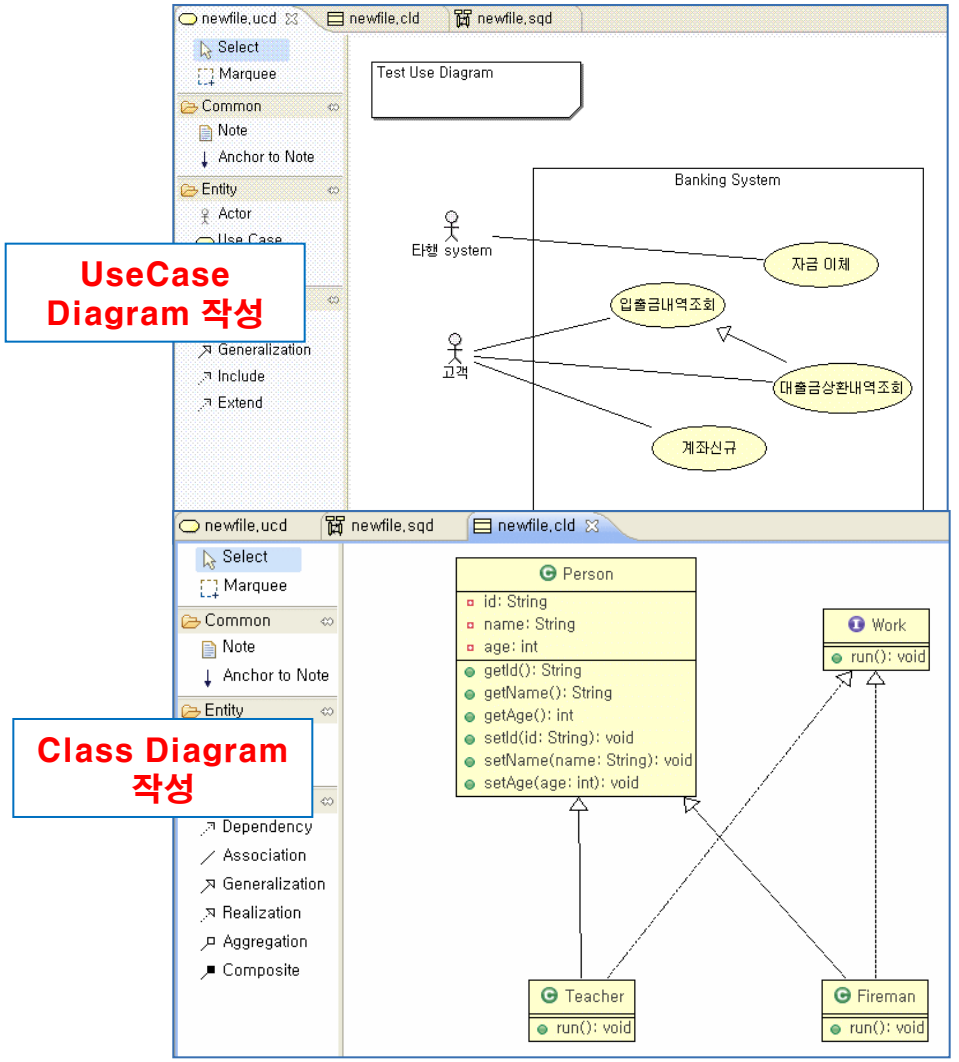
- Use Case Diagram 작성할 수 있는 기능을 제공

❑ Class Diagram 작성

- Class Diagram 작성할 수 있는 기능을 제공

❑ 제약사항

- UML 1.4 기반으로 설계작업 가능
- UML 2.0 지원하지 않음



4. ERD 작성

오픈소스 중 선정된 **ERD** 작성도구는 **AmaterasERD**로서 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

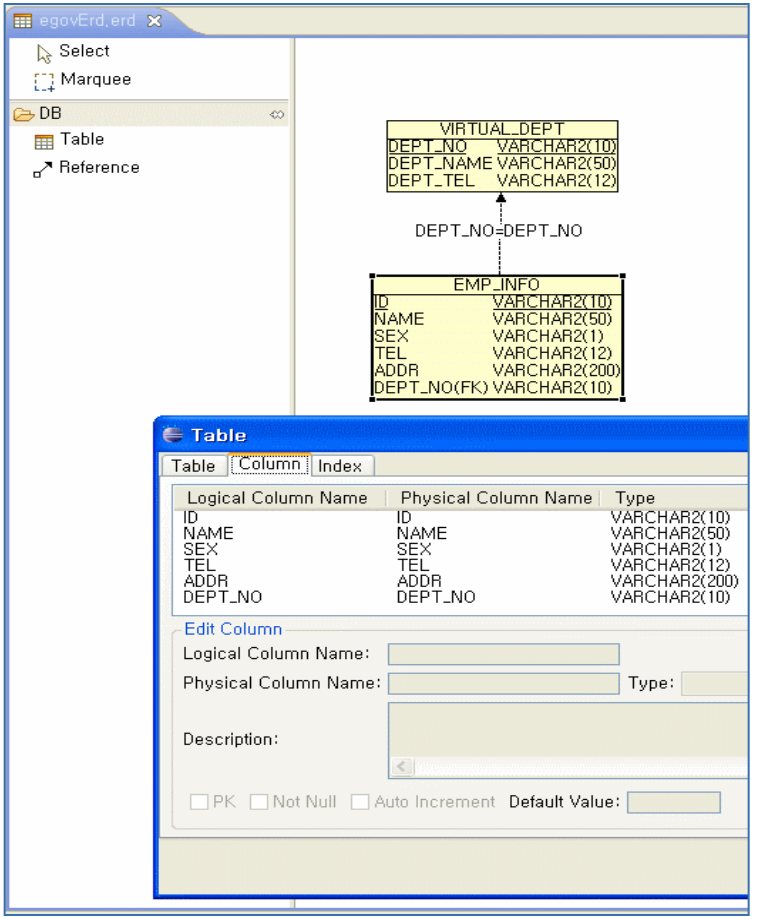
– Oracle, PostgreSQL, MySQL, hsqldb 등의 Database로부터 테이블 Import가 가능

❑ 테이블 명세서 Export

– 도구를 이용해 작성된 Table 들의 명세서를 HTML 형식 Export

❑ 제약사항

– ERWIN 모델링 도구와 호환 불가

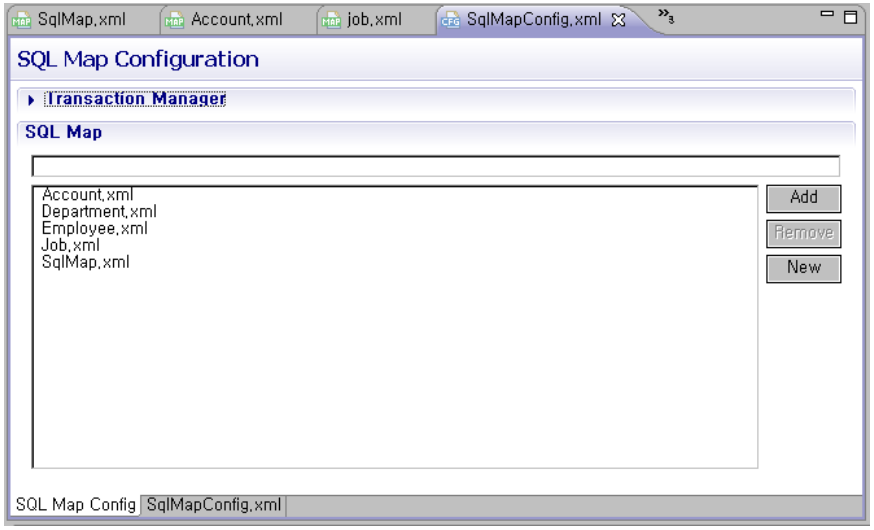


5. DBIO 작성(1/6)

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)

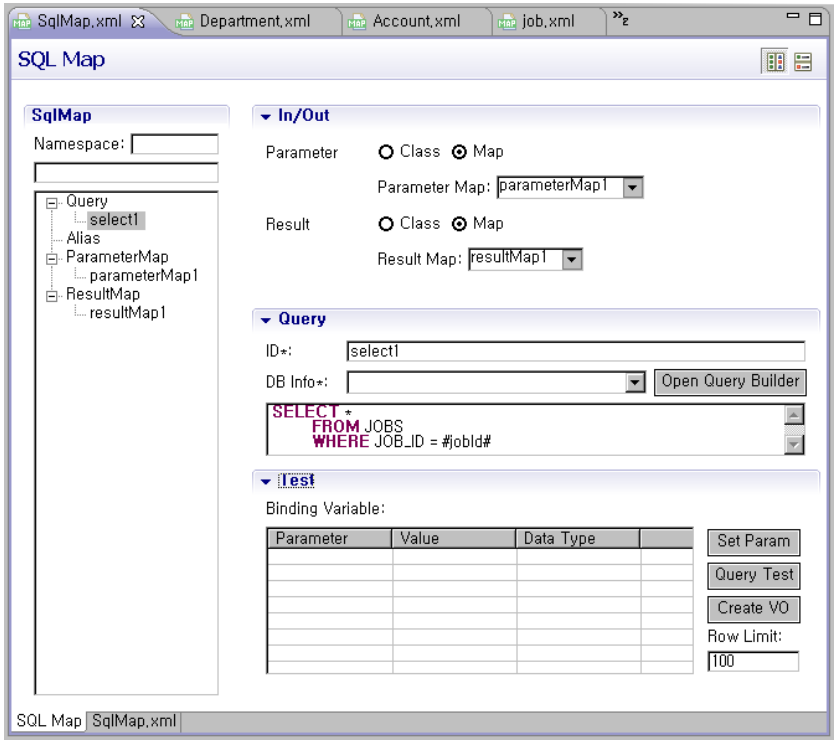
❑ SQL Map Config Editor

- 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.**

5. DBIO 작성(2/6)- SqlMap Config Editor

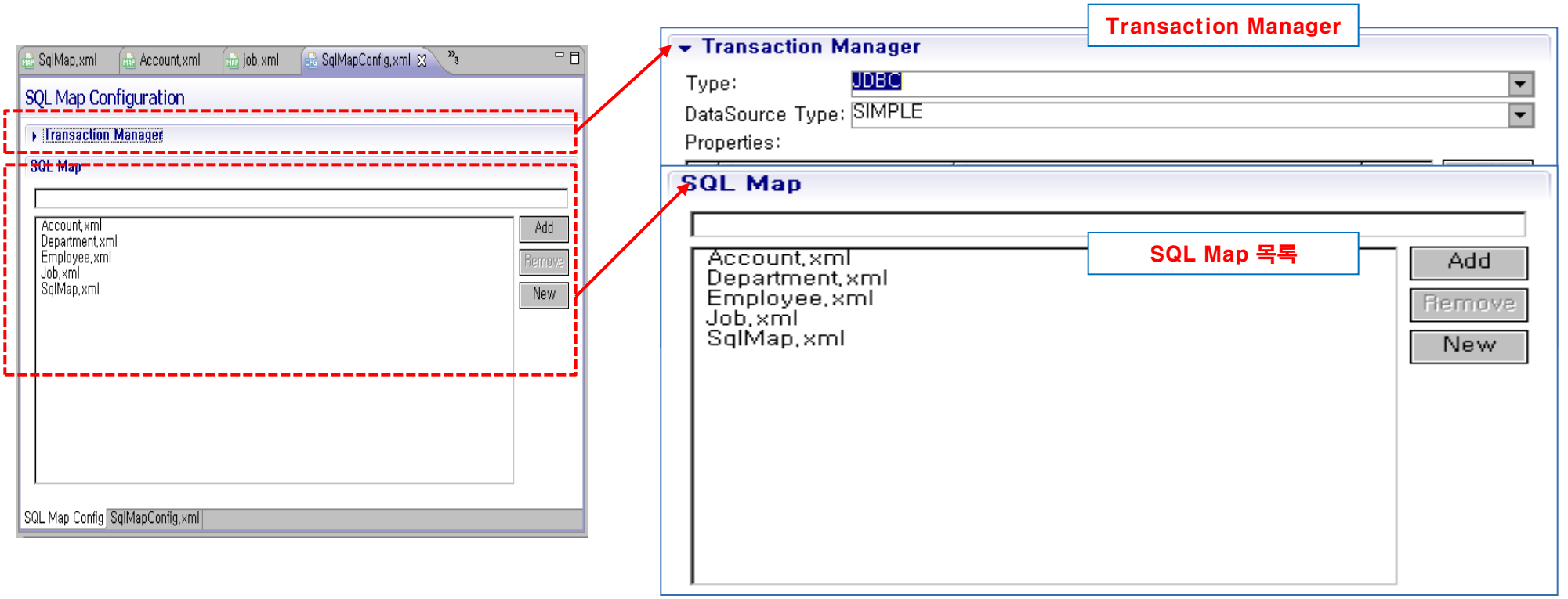
SqlMap Config Editor는 iBATIS SQL Maps 설정 파일을 UI 형태로 편집 가능하도록 지원

❑ Transaction Manager

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

❑ SqlMap 목록

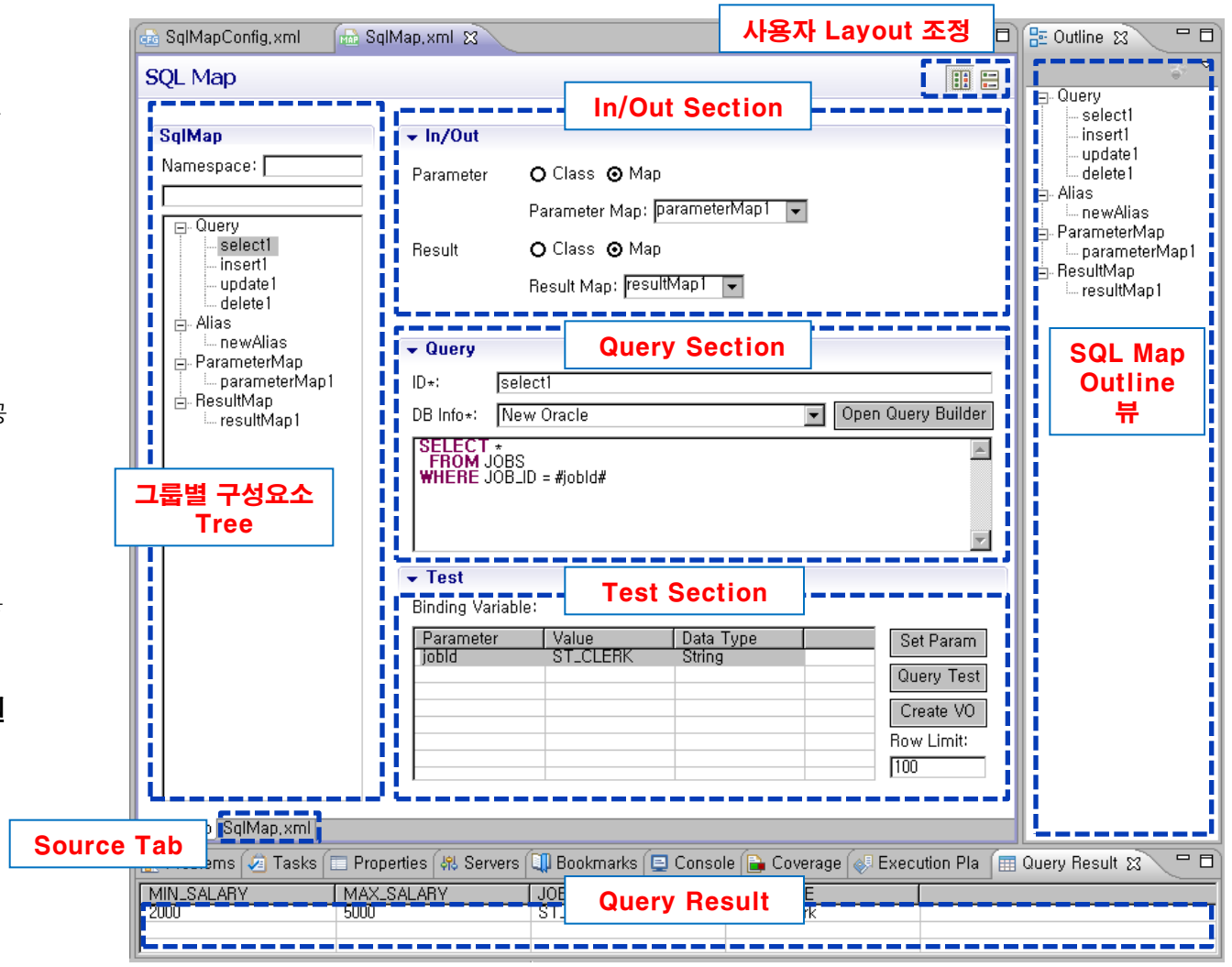
– SqlMap 항목을 추가, 삭제, 신규 생성하고 SQLMapConfig 파일 내 등록을 지원한다.



5. DBIO 작성(3/6)- SqlMap Editor 화면구성

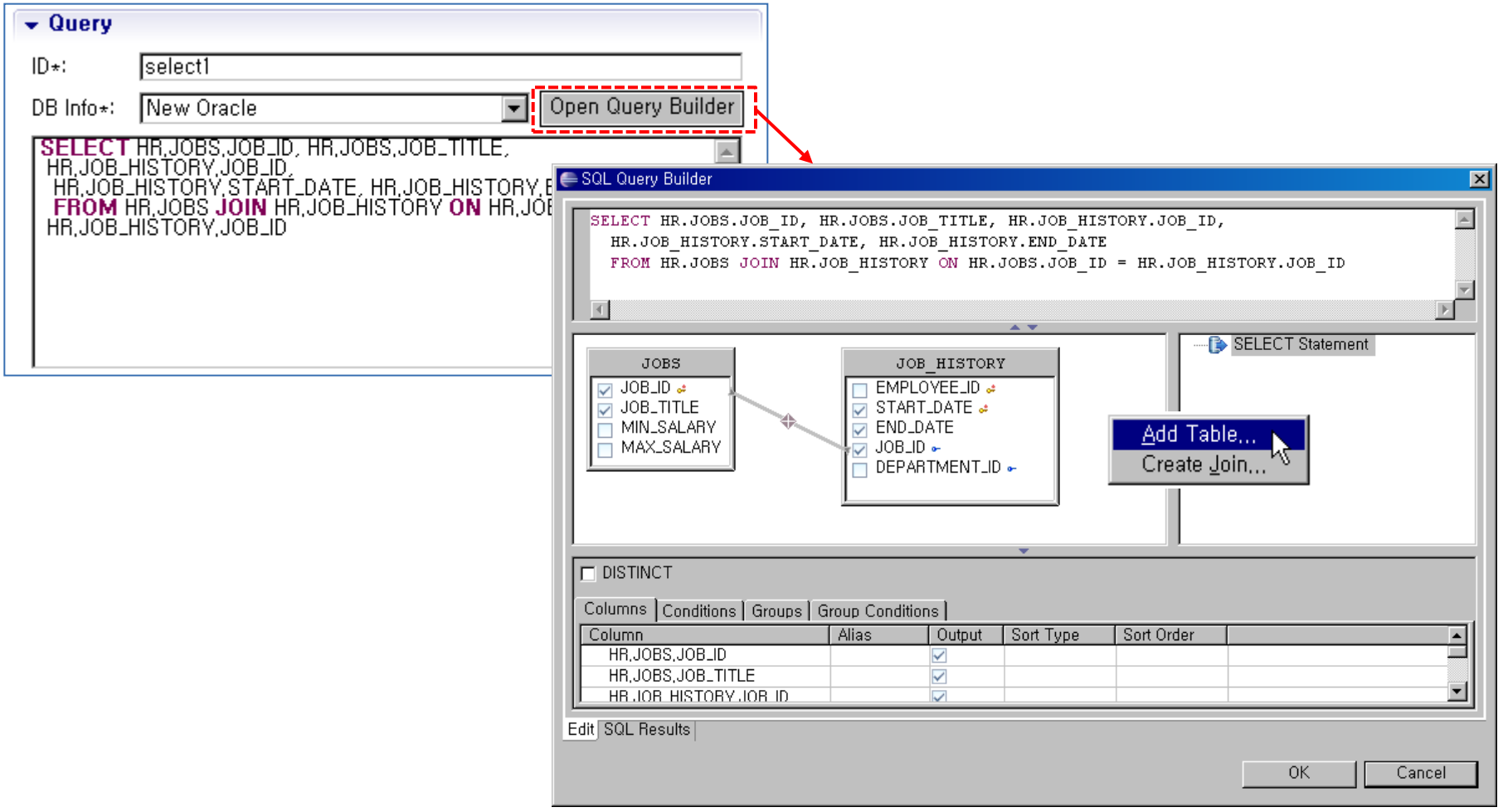
SqlMap Editor는 iBatis SQL Maps 파일을 UI 형태로 편집 가능하도록 지원하는 도구

- ❑ **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



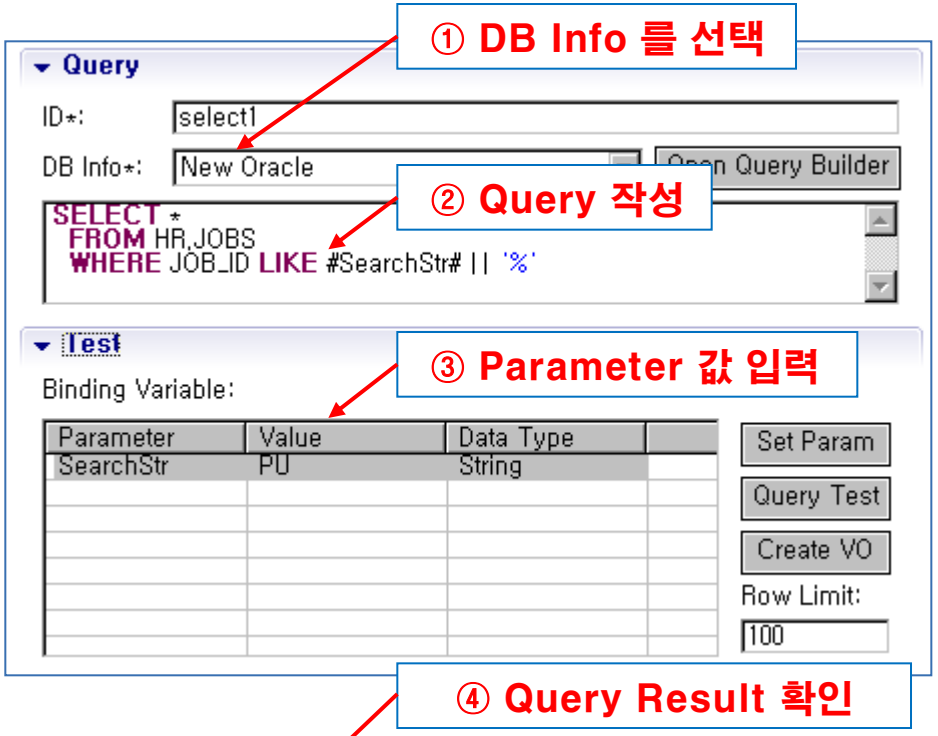
5. DBIO 작성(4/6)- 쿼리 작성

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



5. DBIO 작성(5/6)- 쿼리 Test 및 결과 확인

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



❑ 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.

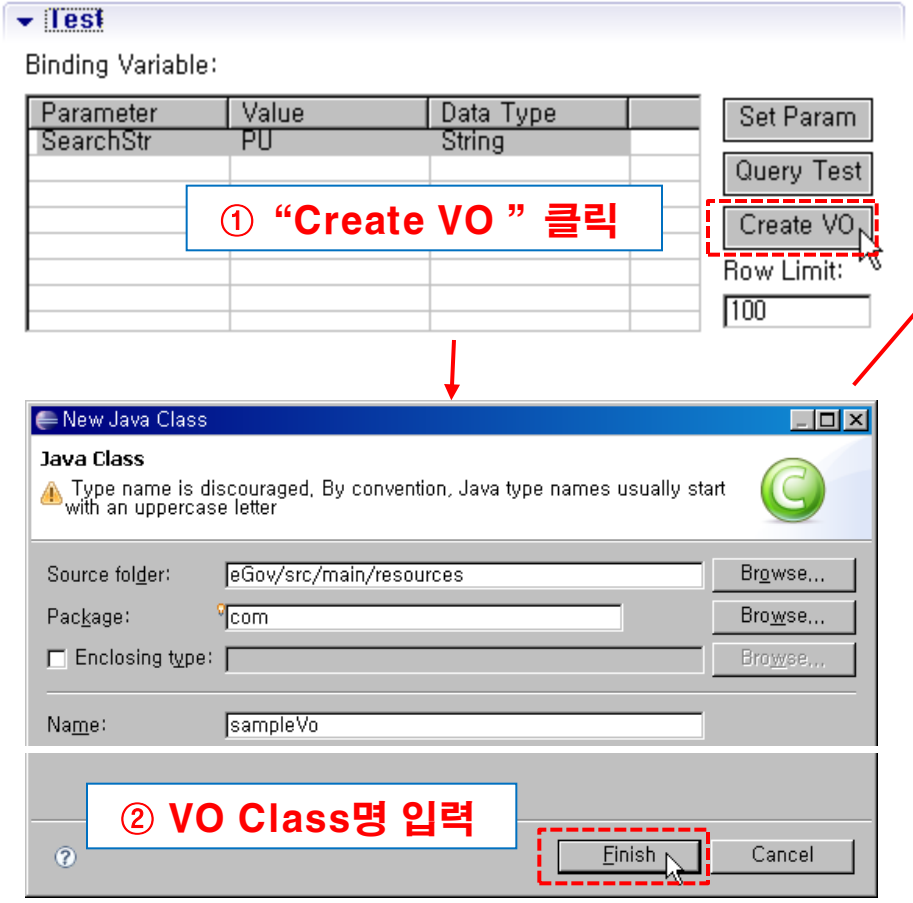
④ 결과 값을 확인한다

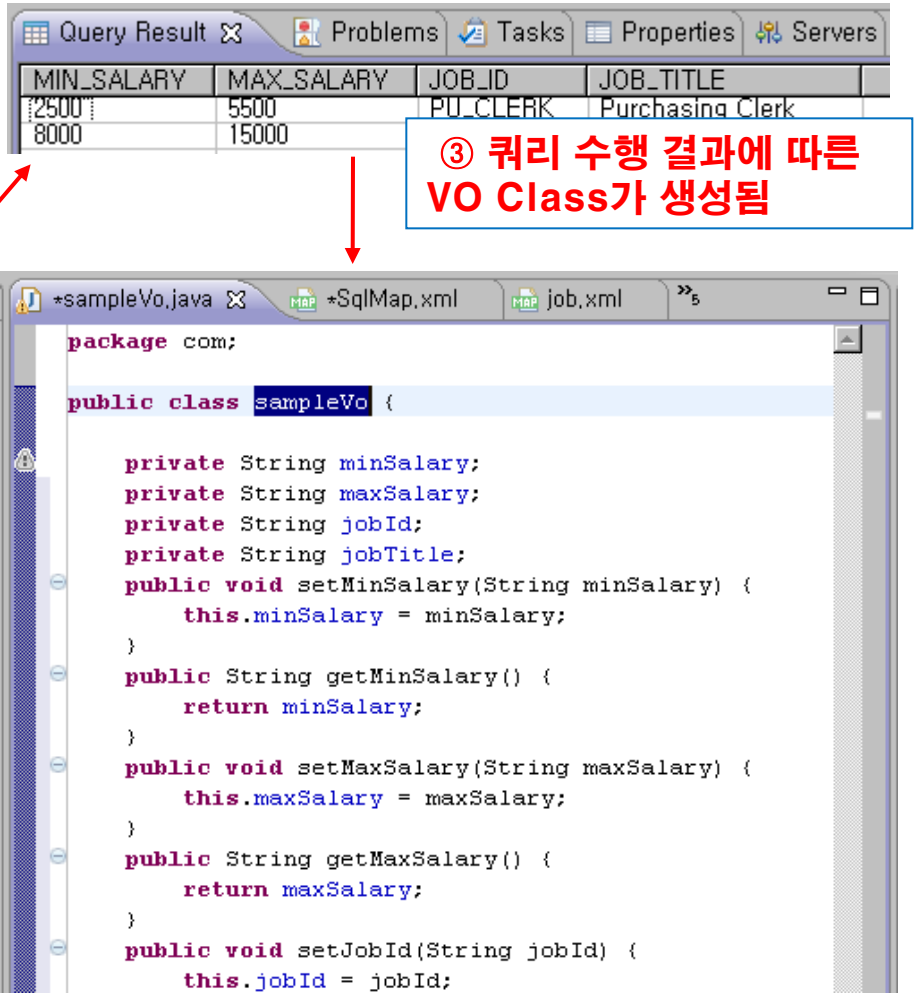
❑ Test Section 버튼 설명

- 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)

5. DBIO 작성(6/6)- VO 자동 생성

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



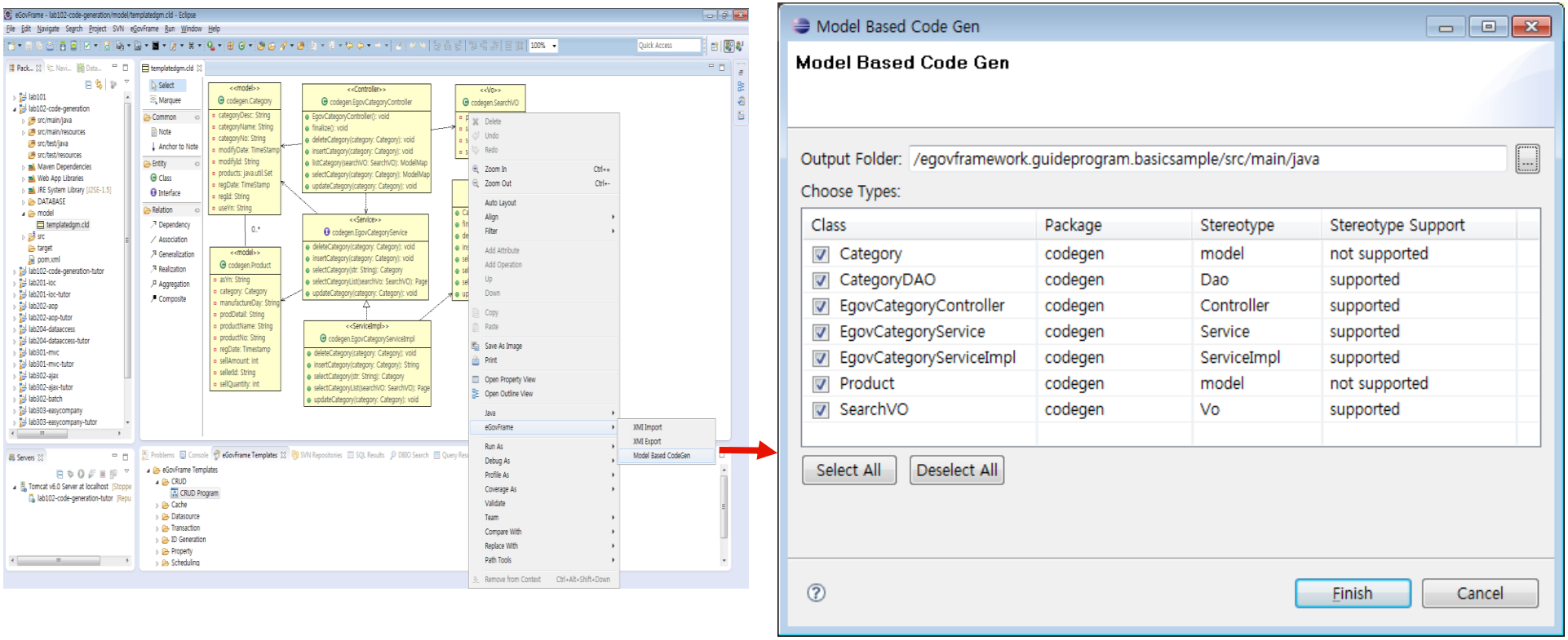


6. Code Generation – 모델기반

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

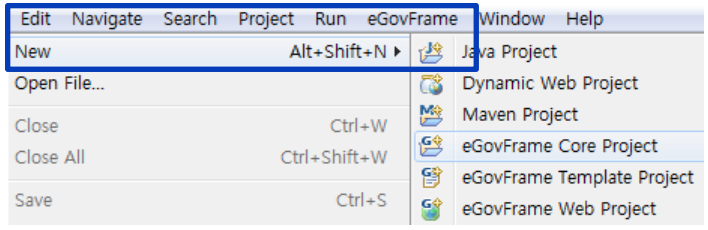
프로젝트 생성 마법사는 일반 혹은 Web 기반 프로젝트 자동 생성 기능을 제공

❑ eGovFrame Core Project

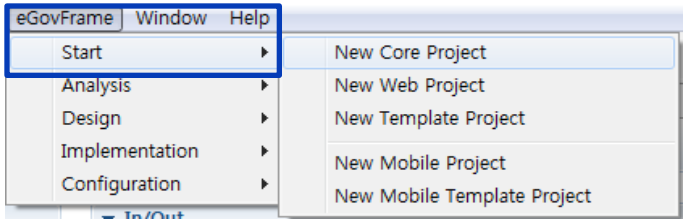
- 일반적인 JAVA Project 생성

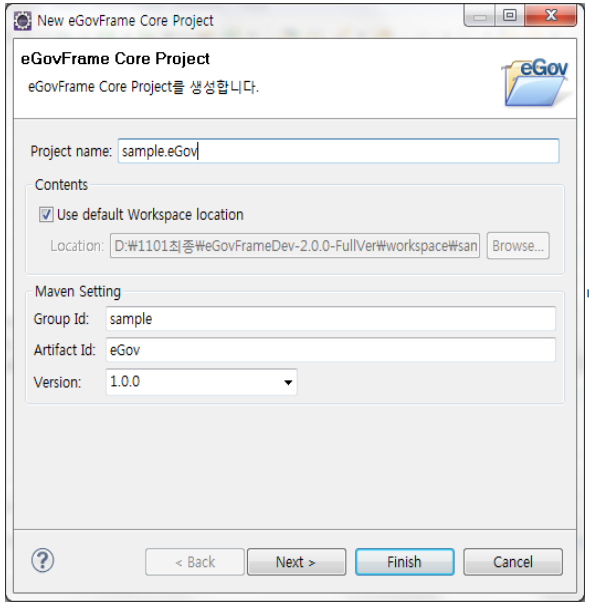
❑ eGovFrame Web Project

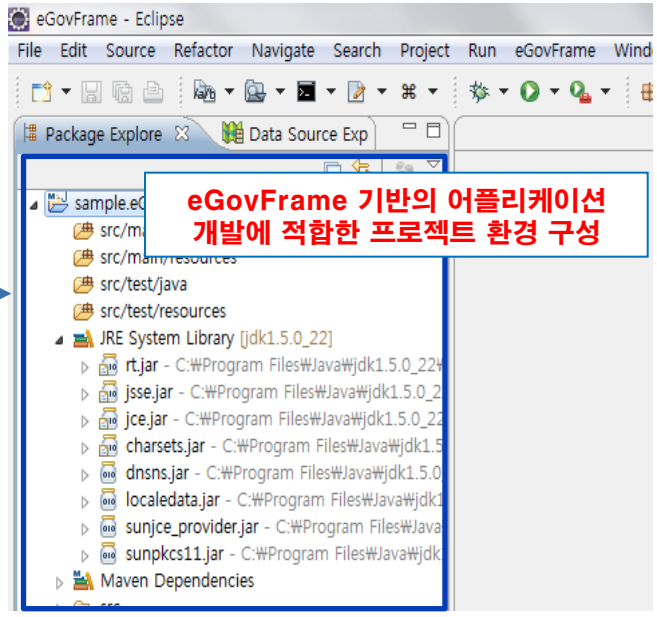
- Web 기반 프로젝트 생성



2가지 생성 방식

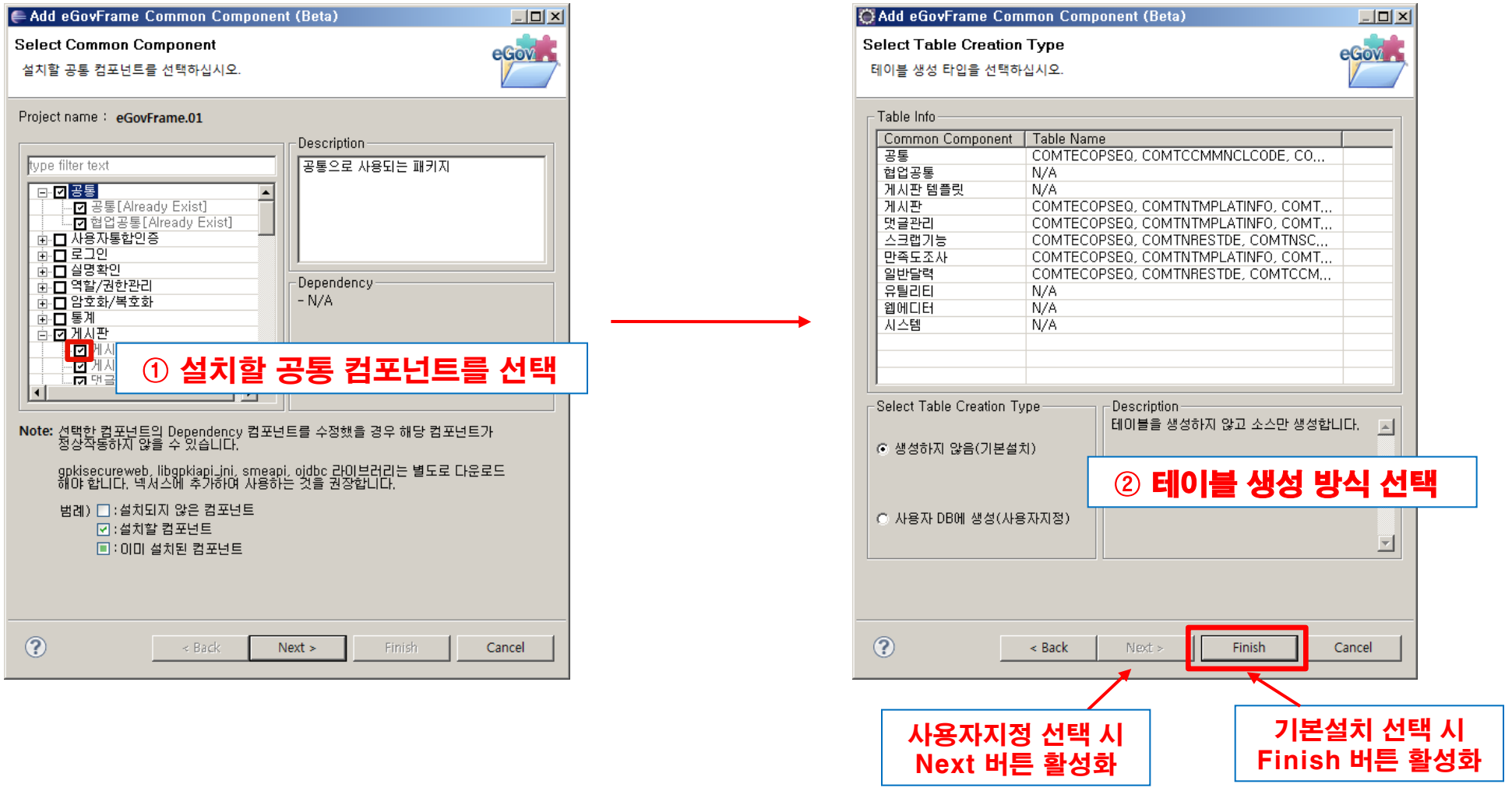






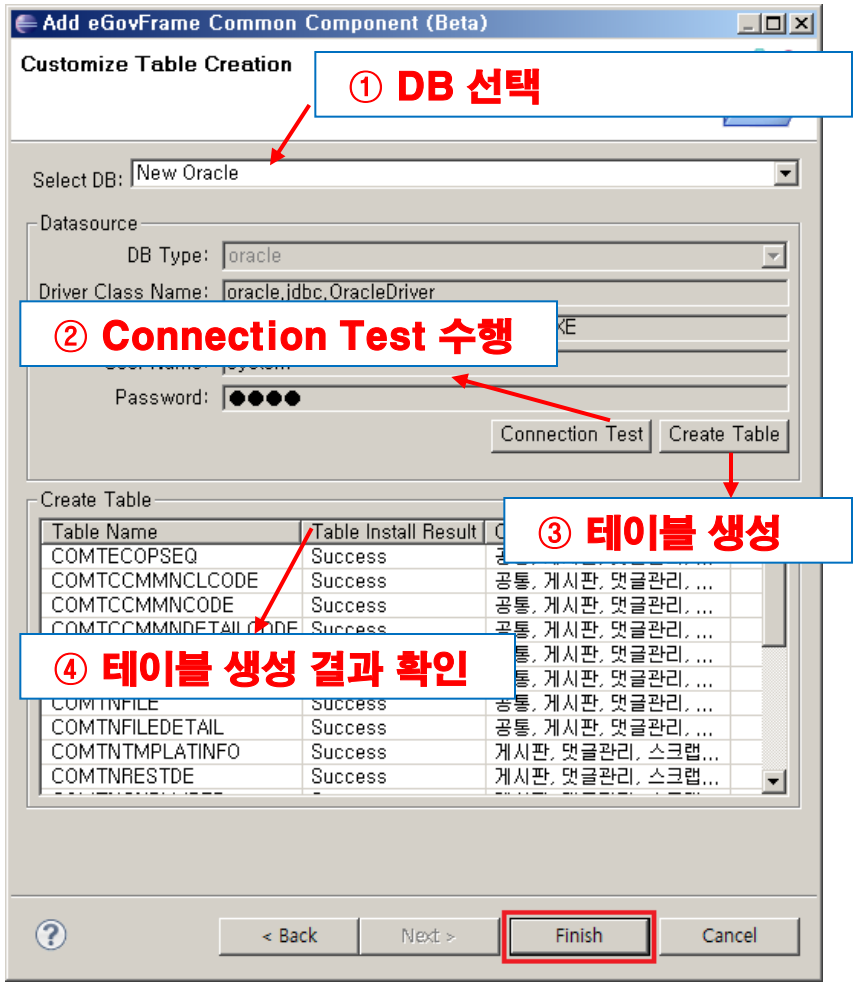
8. Common Component(1/3) - 기본설치

개발자의 편의성을 위하여 전자정부 표준 프레임워크에서 제공하는 공통컴포넌트 251종의 설치 마법사를 제공



8. Common Component(2/3) – 사용자 지정

개발자의 편의성을 위하여 전자정부 표준 프레임워크에서 제공하는 공통컴포넌트 251종의 설치 마법사를 제공



❑ **Table 생성 절차**

① **Select DB**에서 사용할 **DB**를 선택한다.

② **Connection Test** 를 수행한다.

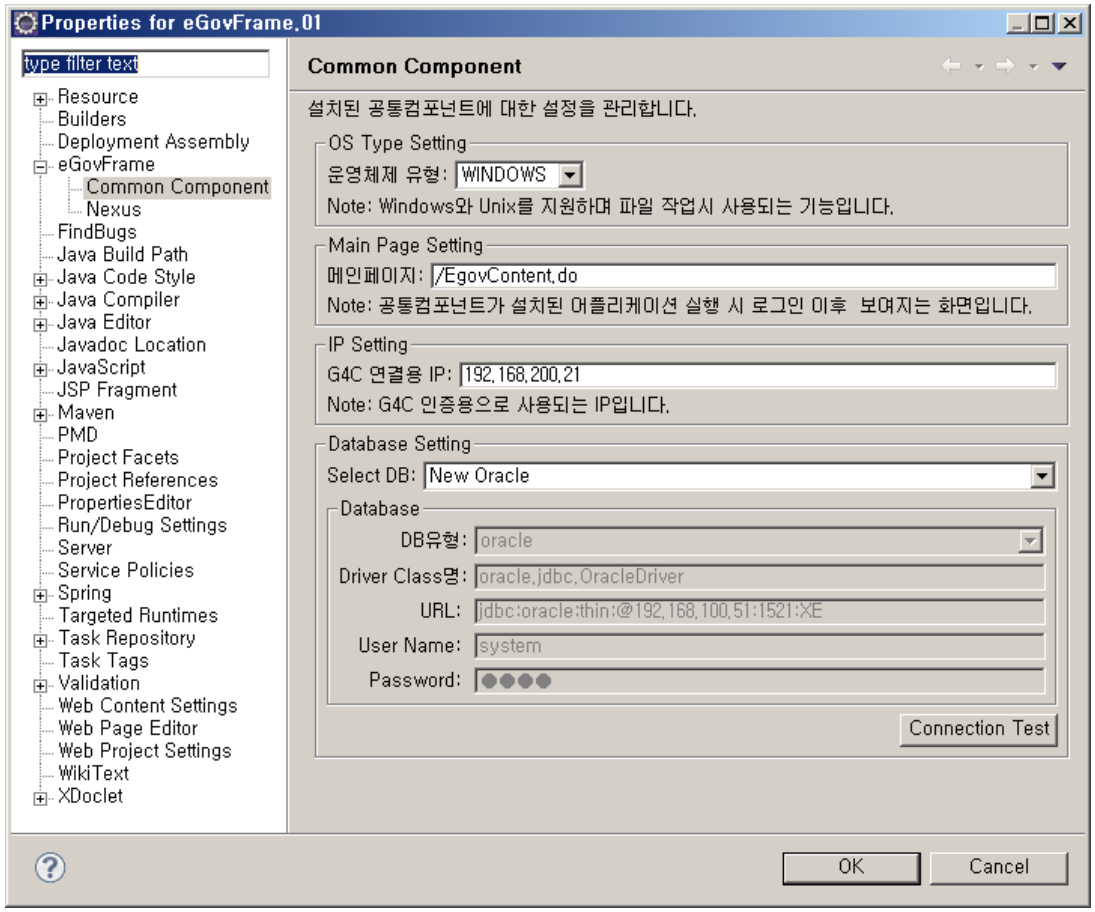
③ **Create Table** 버튼을 눌러 테이블을 생성한다.

④ 테이블 생성 결과를 확인한다.

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

8. Common Component(3/3) – 공통 컴포넌트 설정

개발자의 편의성을 위하여 전자정부 표준 프레임워크에서 제공하는 공통컴포넌트 251종의 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.

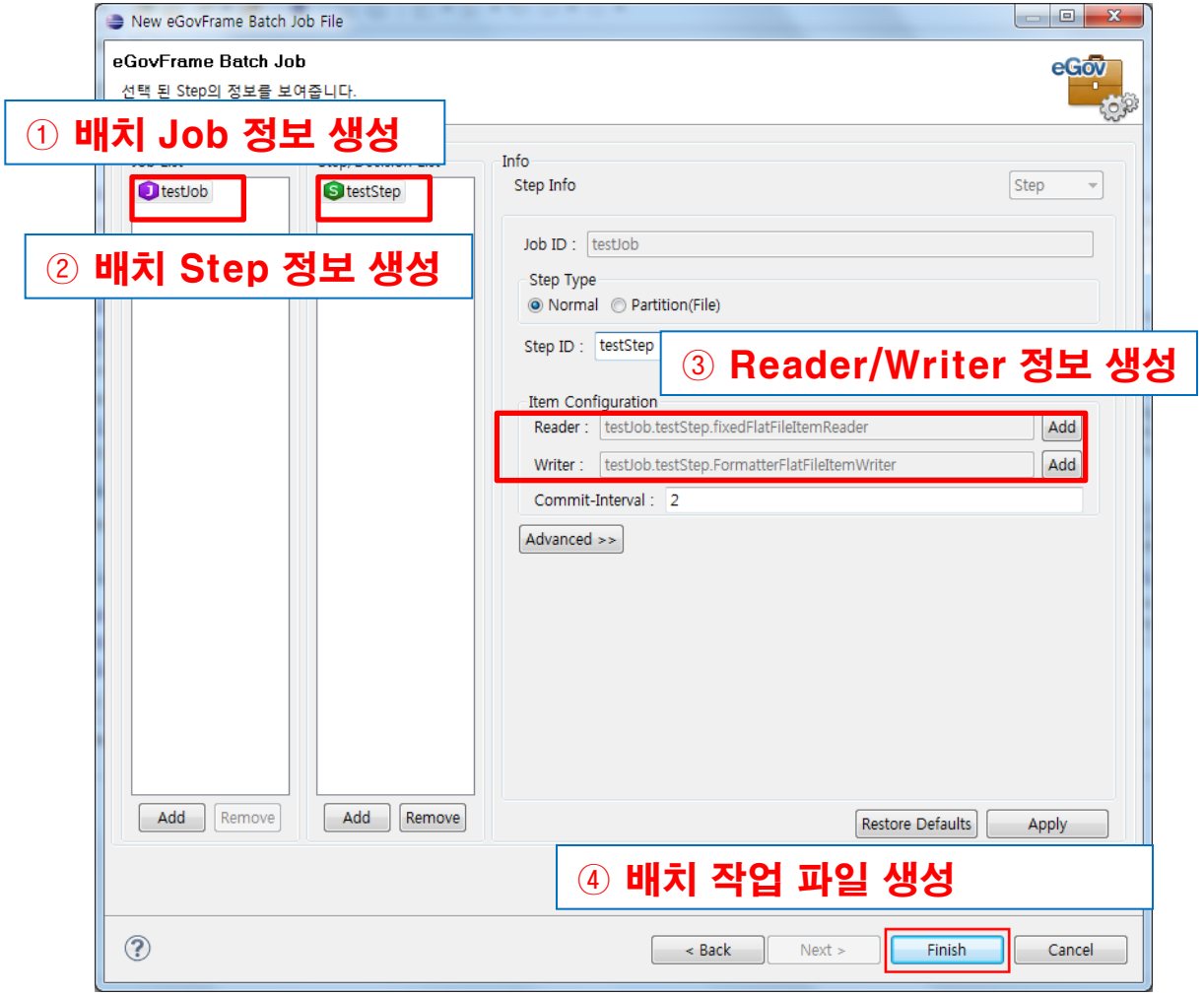
❑ **Select DB**

– 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.

9. Batch Job

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



❑ 배치 작업 생성 절차

① 배치 Job 정보를 생성한다.

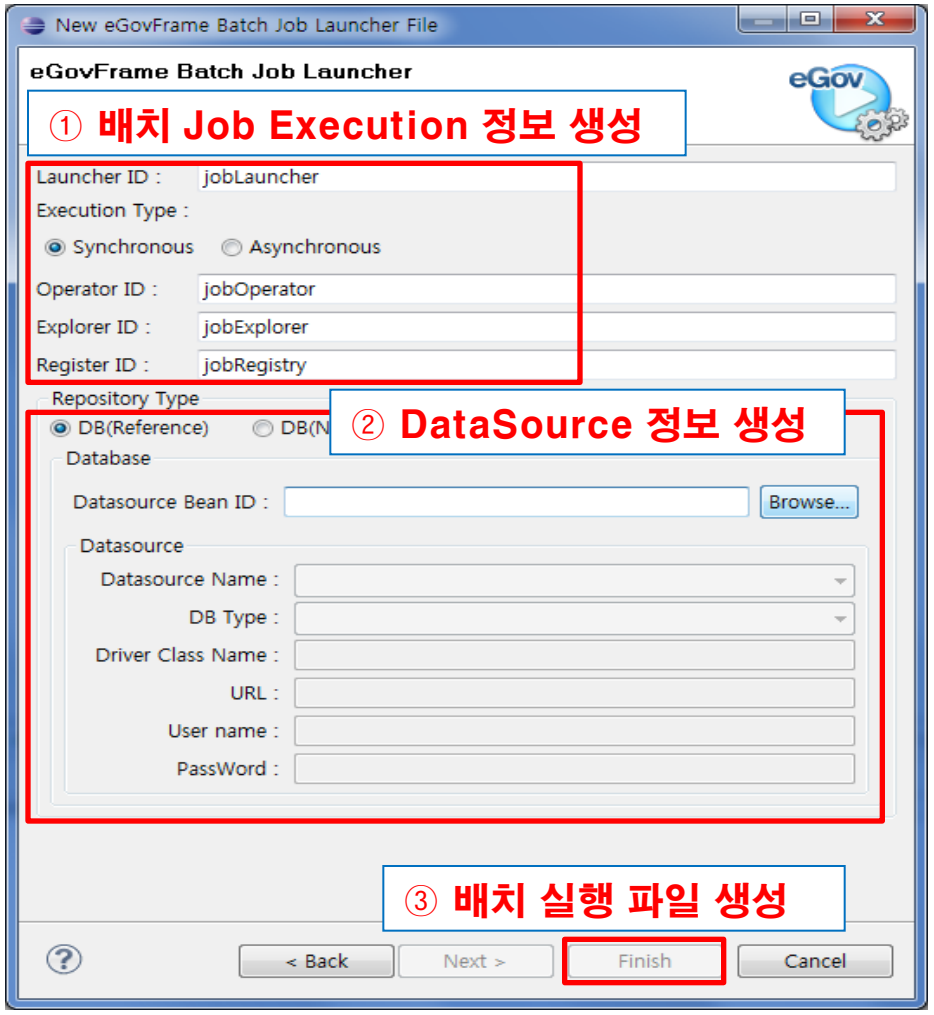
② Batch Step 정보를 생성한다.

③ Reader/Writer 정보를 생성한다.

④ 배치 작업 파일을 생성한다.

10. Batch Job Launcher

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



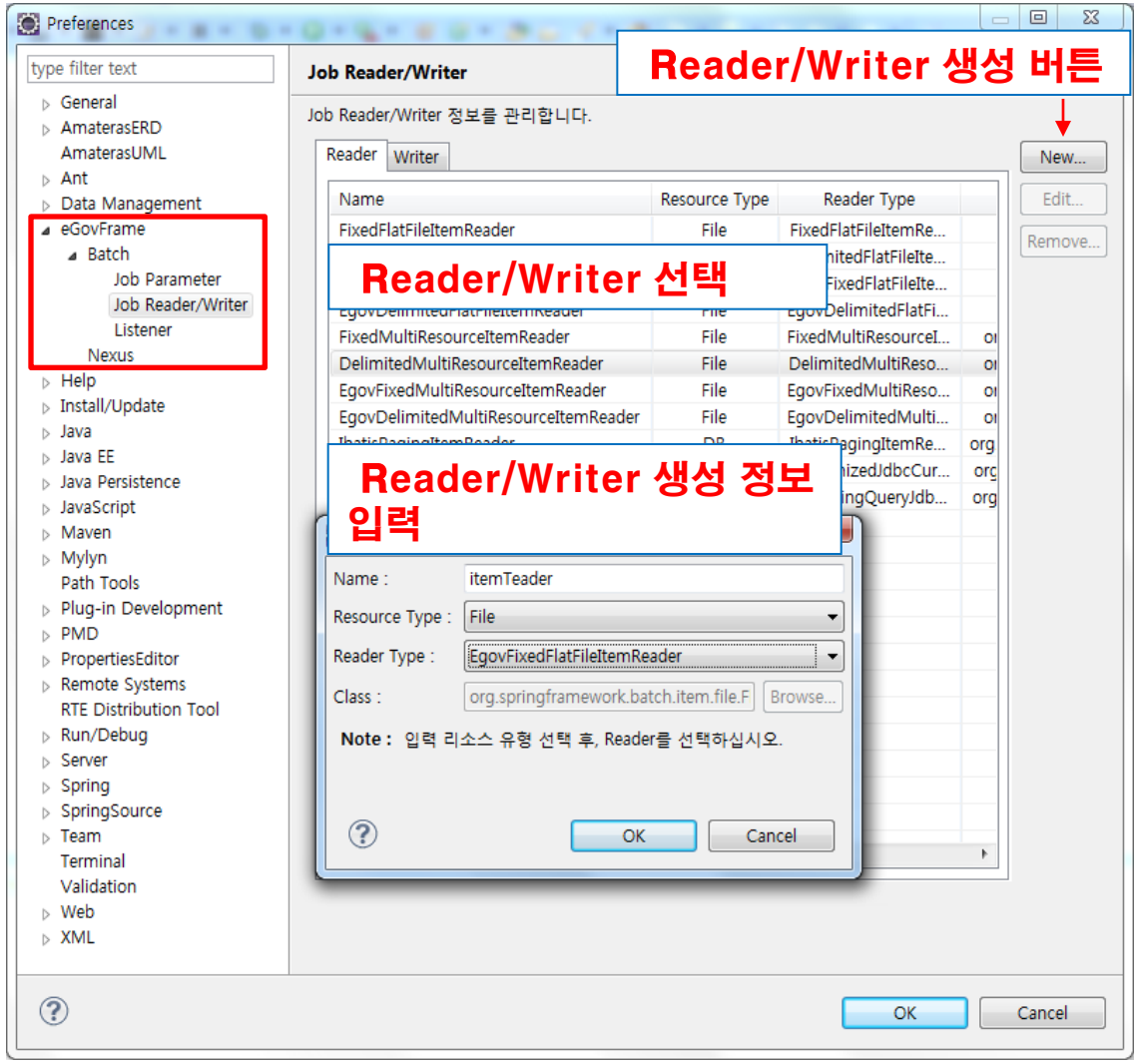
❑ 배치 실행 작업 생성 절차

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

12. eGovFrame Template Project(1/2)

유형별 템플릿 프로젝트를 자동 생성하는 기능을 제공

❑ Simple Homepage

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

❑ Portal Site

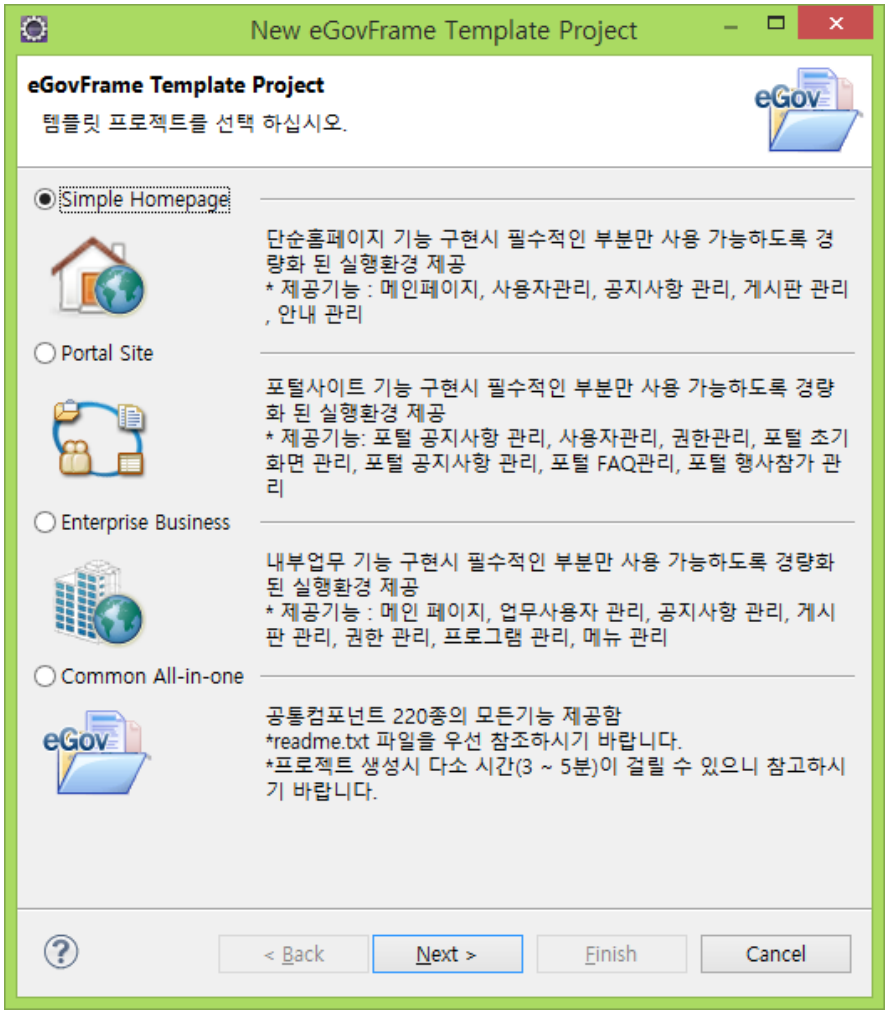
- 포털 공지사항 관리, 사용자 관리, 권한 관리,
  포털 초기화면 관리, 포털 공지사항 관리,
  포털 FAQ관리, 포털 행사참가 관리 기능 제공

❑ Enterprise Business

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

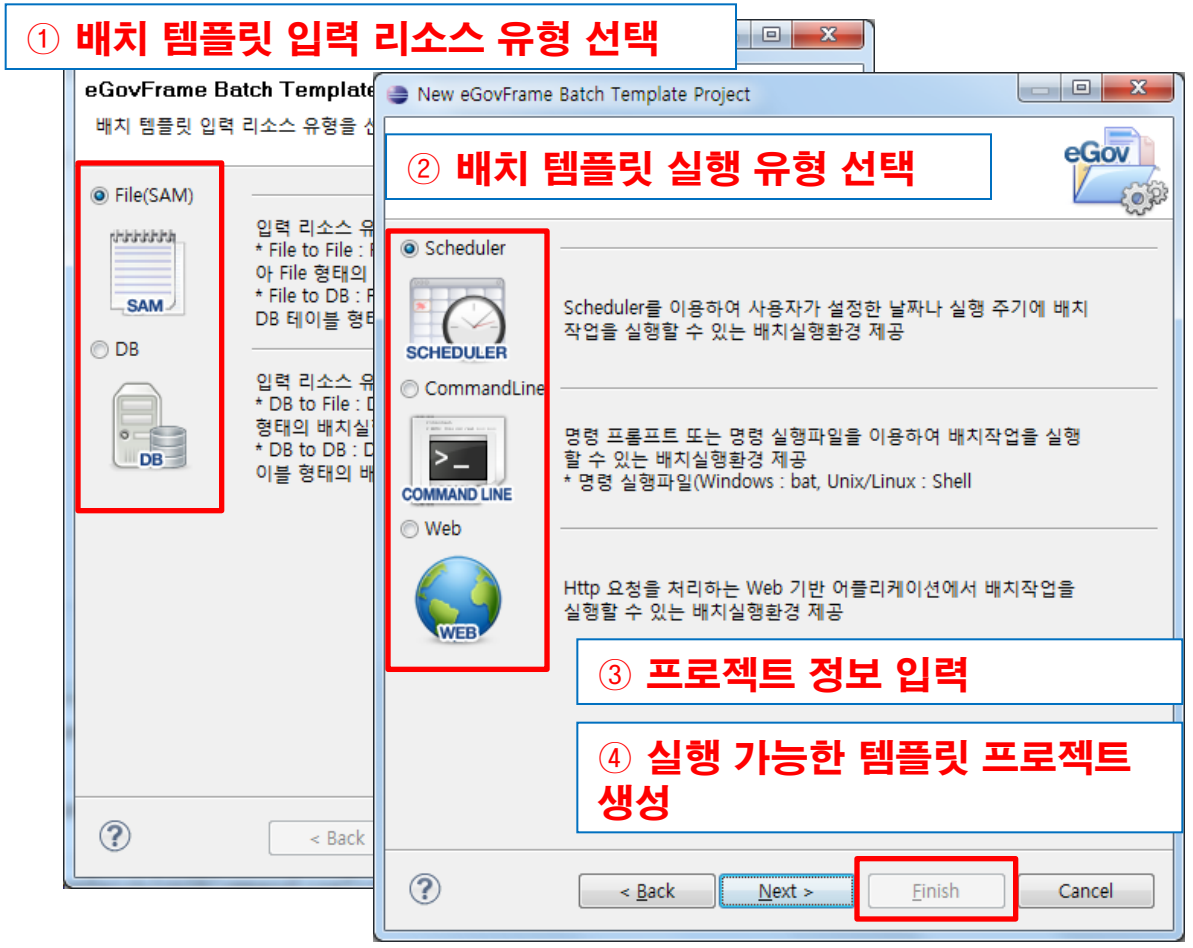
❑ Common All-in-one

- 공통컴포넌트 251종의 모든 기능 제공



12. eGovFrame Template Project(2/2) - Batch

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



❑ 배치 실행 작업 생성 절차

① 배치 Job Execution 정보를 생성한다.

② DataSource 정보를 생성한다.

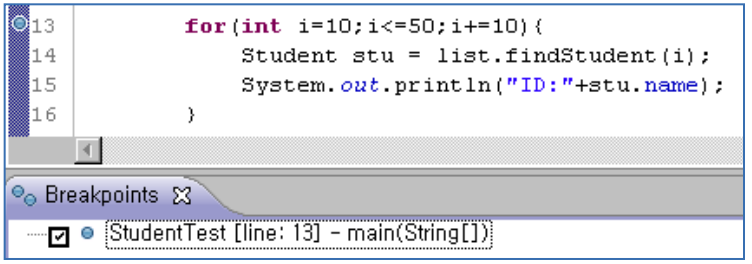
③ 배치 실행 파일을 생성한다.

13. Debug(1/2) - Local Debugging

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**

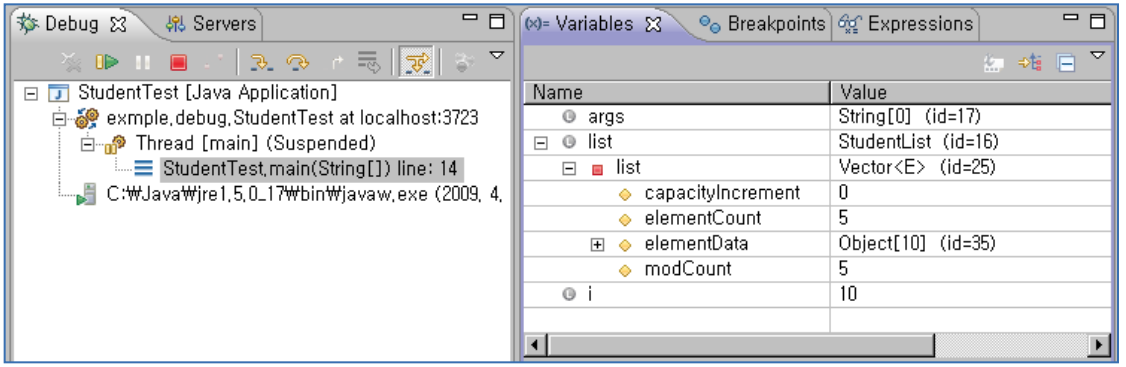
- 필터링 대상 프로그램의 경우 Step Over와 같이 동작하고, 필터링 대상이 아니면 Step Into와 같이 동작한다. Debug View의 Use Step Filters 버튼을 눌러 활성화

❑ **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



13. Debug(2/2) - Remote Debugging

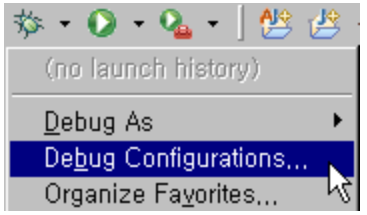
원격 컴퓨터의 WAS(Jboss, JEUS, WebLogic)에서 실행되는 어플리케이션을 디버깅

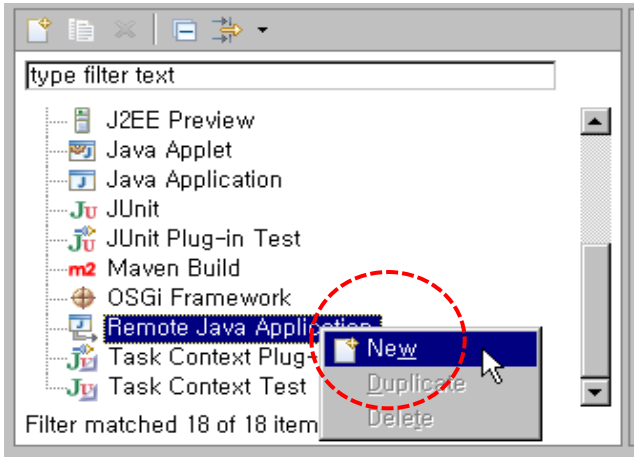
❑ 디버깅 환경 설정

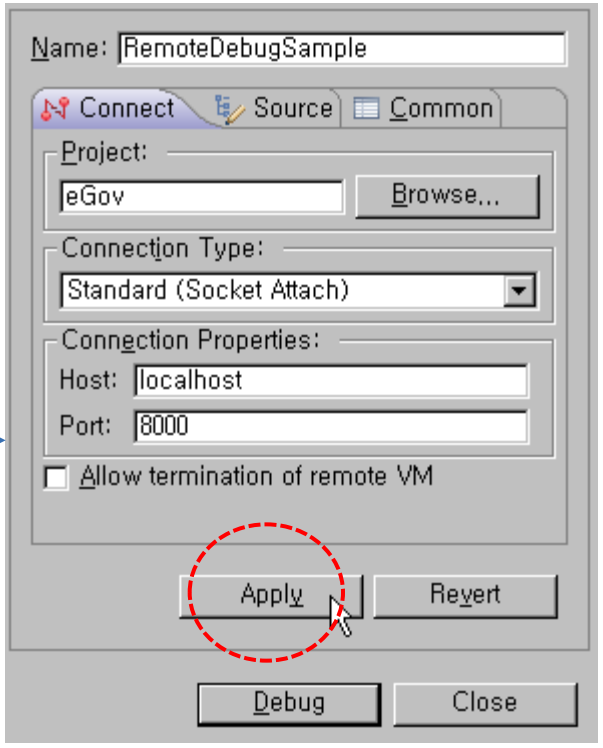
– WAS 환경설정파일에 "-Xdebug -Xnoagent -Xrunjdwp:transport=dt_socket,address=8000,server=y,suspend=n –server" 를 정의 (Jeus 서버 기준)

❑ 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.







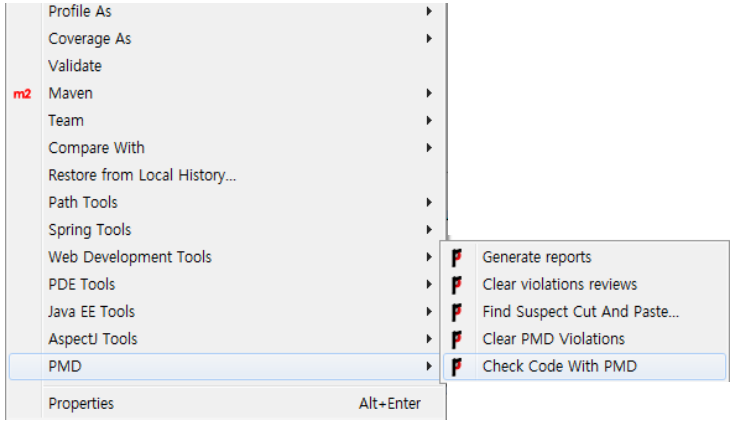
* 디버그 방법(BreakPoint 등)은 로컬 디버깅과 동일하다.

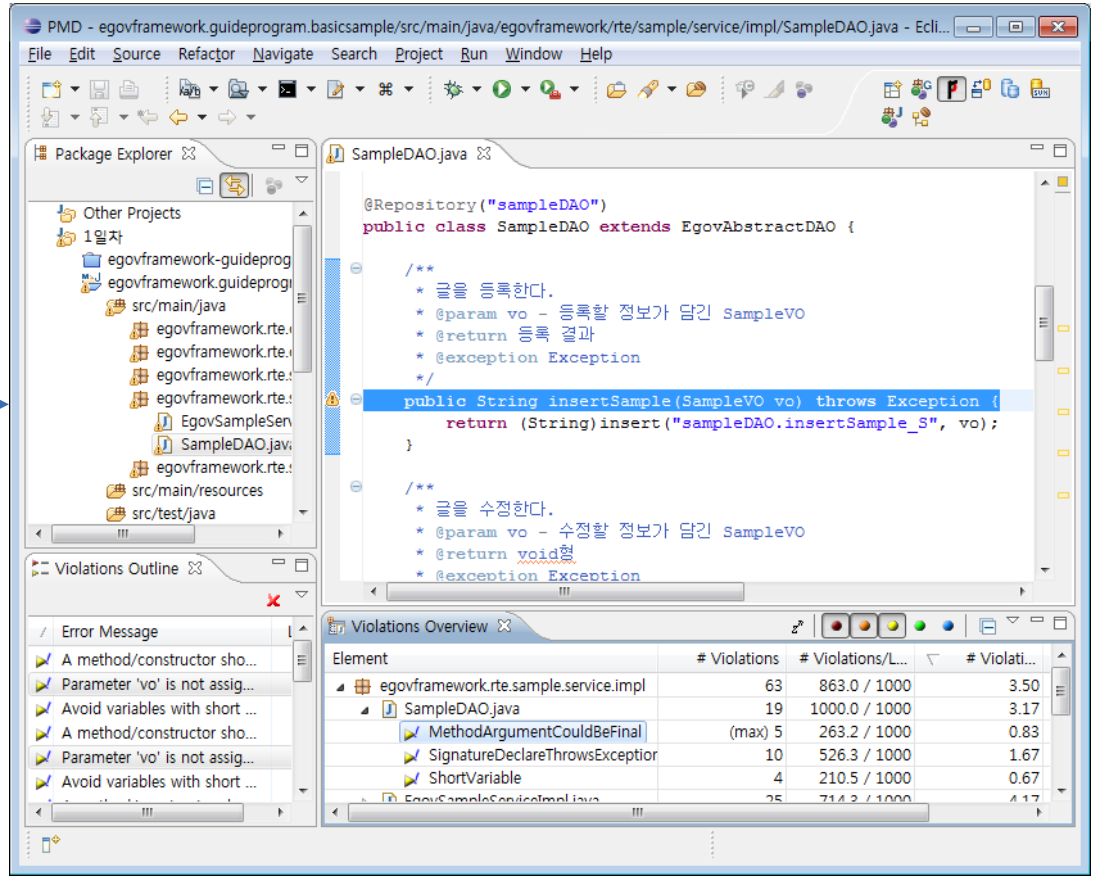
14. Code Inspection

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.

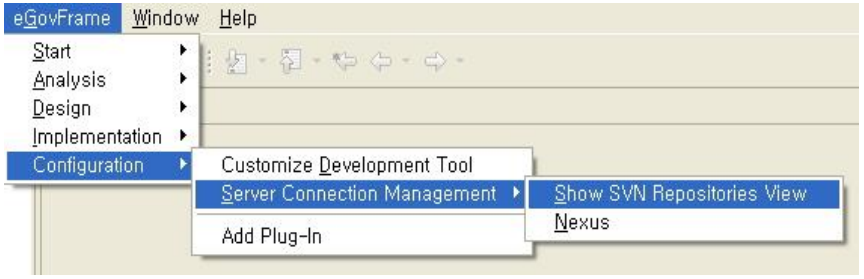


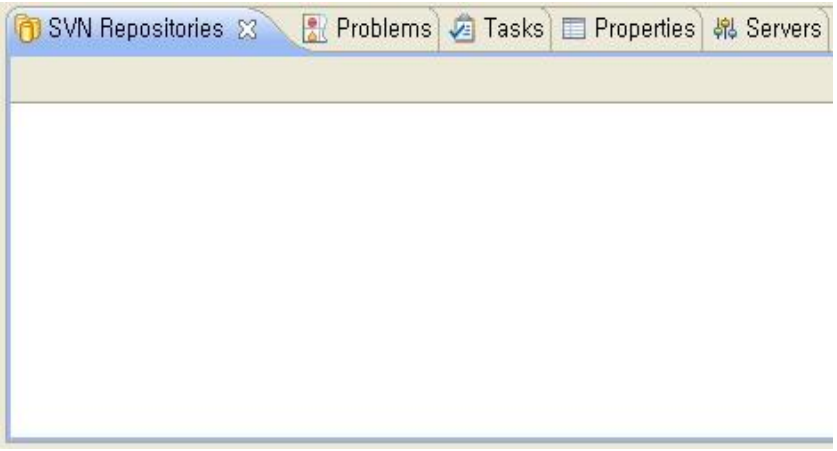


eGovFrame 통합메뉴를 통해 SVN Repositories View를 연동하는 기능을 제공한다.

❑ 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.



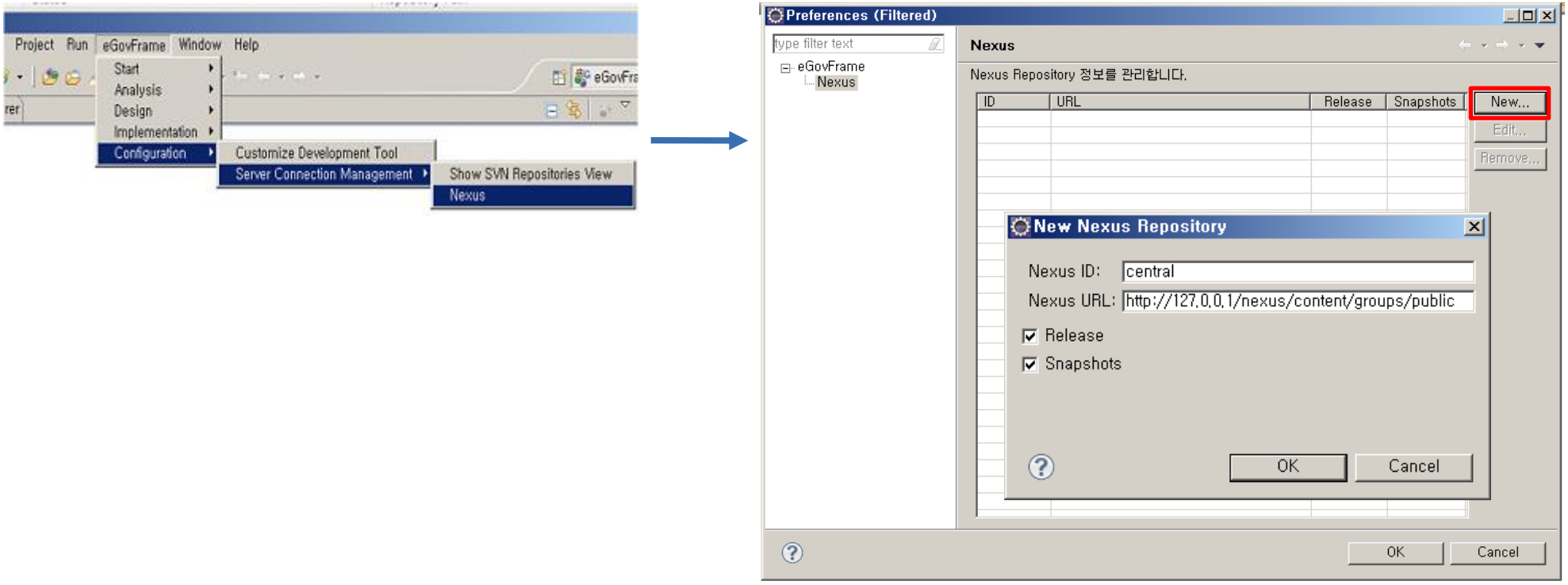


15. Server Connection Management(2/3)(서버 연동 정보 관리)

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.



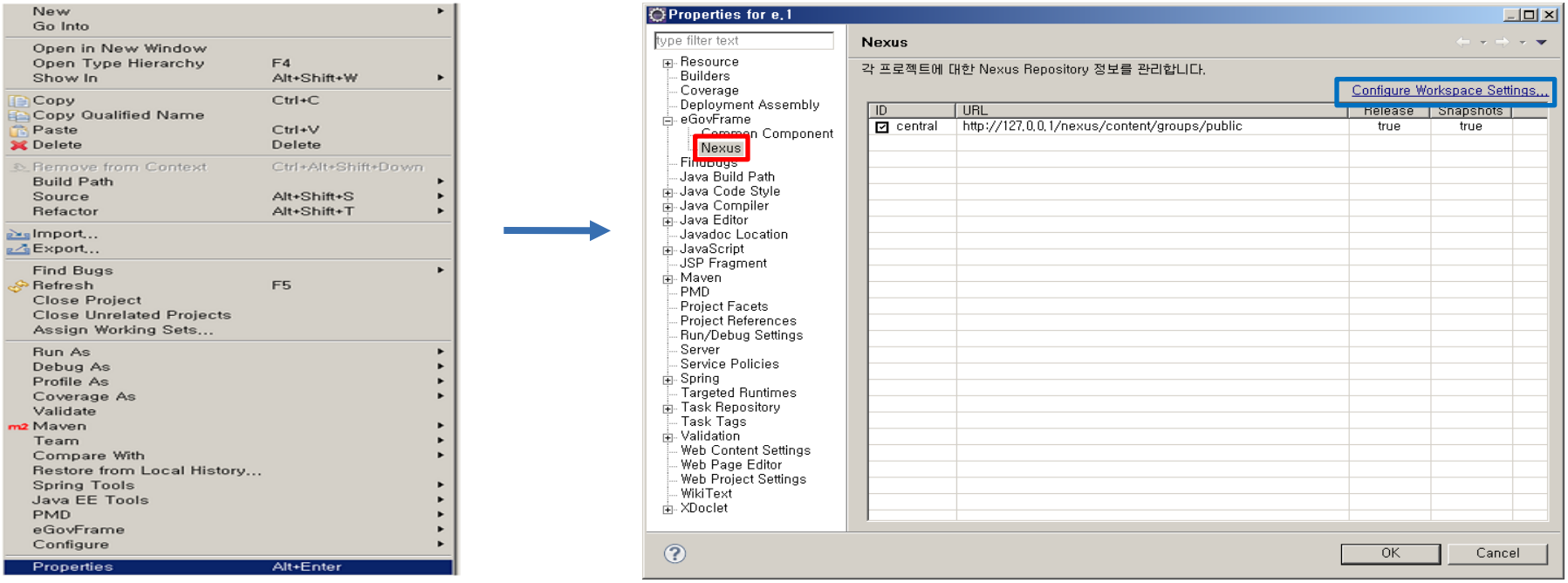
15. Server Connection Management(3/3)(서버 연동 정보 관리)

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.





3. 개인빌드

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

1. Maven 개요 (1/2)

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

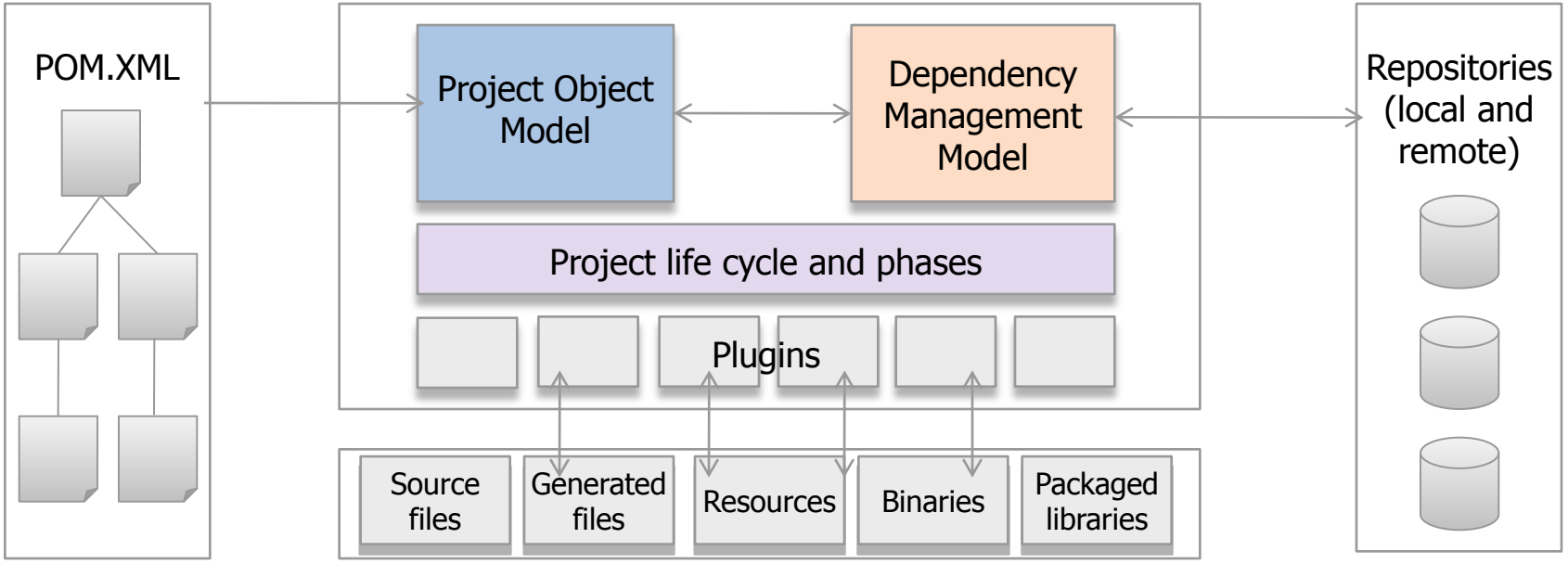
❑Maven 특징

Maven 장점	Maven 단점
• 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	• repository 관리의 불편함 – Maven 프로젝트의 급속한 발전으로 central repository가 제공하는 라이브러리들이 급속히 증가하고 있으나 아직 3rd 파티 라이브러리 등 미제공 라이브러리들이 있음 • pom.xml 파일 관리 –메이븐 프로젝트 관리에 대한 모든 내용이 pom.xml 파일에 담기게 되므로 길고 장황하게 될 수 있음 • 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

1. Maven 개요 (2/2)

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

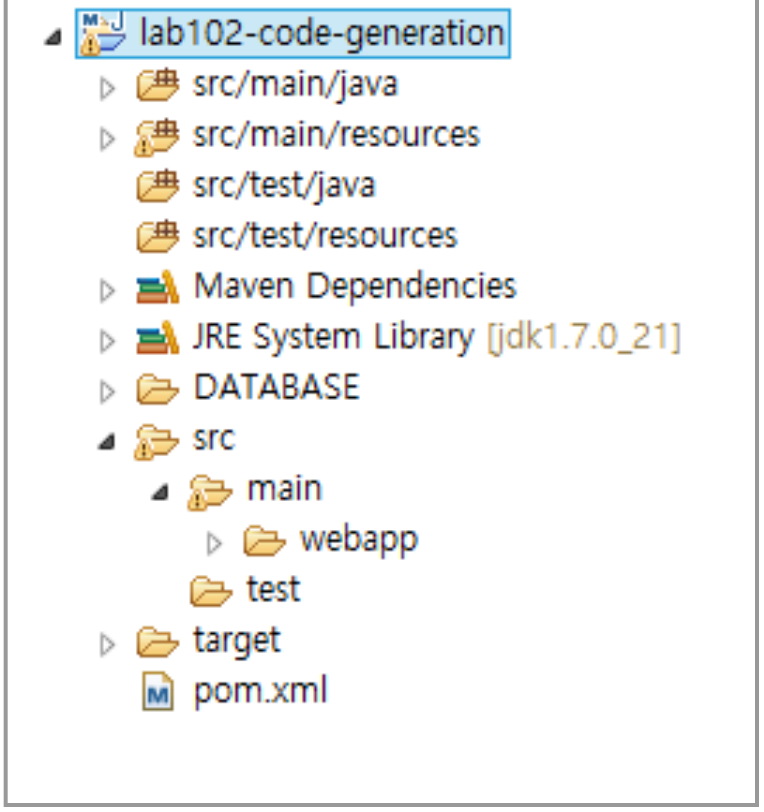


- **프로젝트 객체 모델(POM)**: 메이븐 엔진 내장 + POM.XML 파일에서 선언적으로 제공
- 의존성 관리 모델: 로컬 및 리모트 저장소를 이용하여 관리
- 빌드 생명주기와 각 단계: 잘 정의된 단계들과 빌드 사이클에 따라 플러그인들을 조율

2. Maven 디렉터리 구조

Best practices을 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

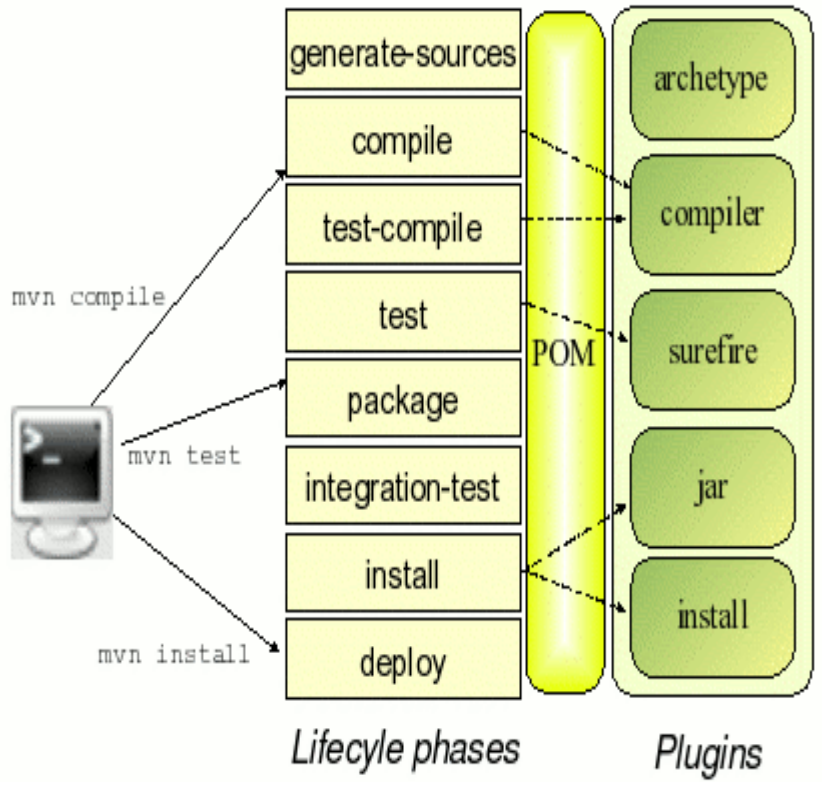
메이븐 빌드는 소프트웨어 프로젝트의 핵심적인 빌드 라이프 사이클 개념을 따르고 있으며 빌드 부터 **artifact**의 배포까지의 라이프 사이클을 정의하고 있음

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 타깃 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.

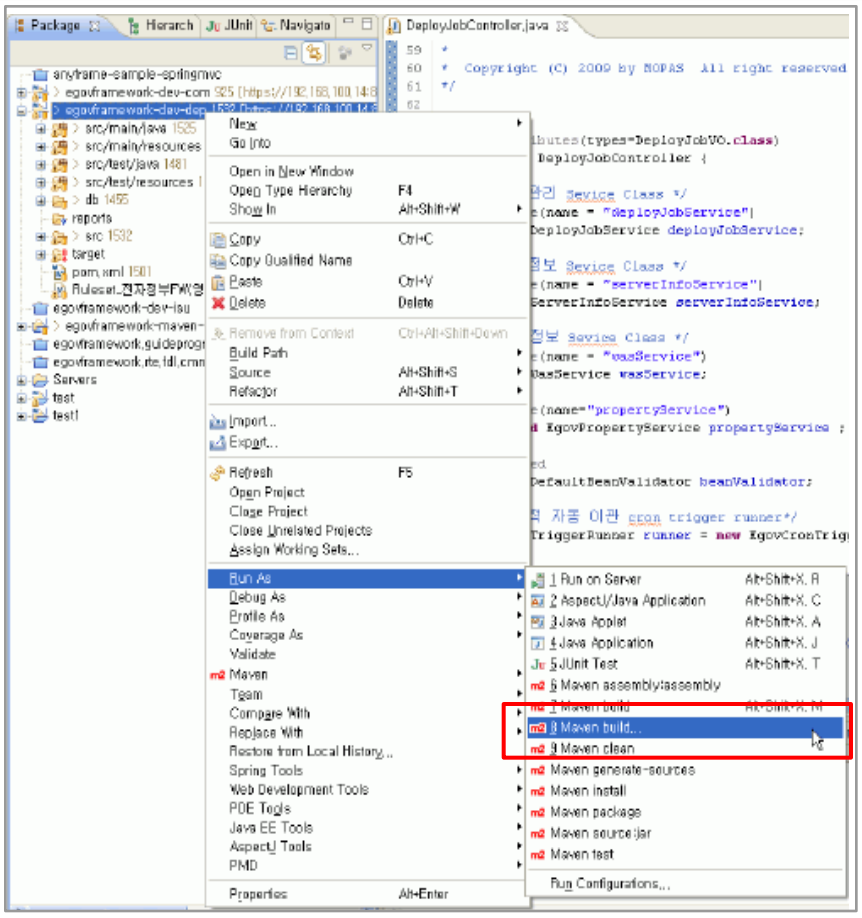
```
$mvn compile
```

Maven2 라이프 사이클 단계(phase)

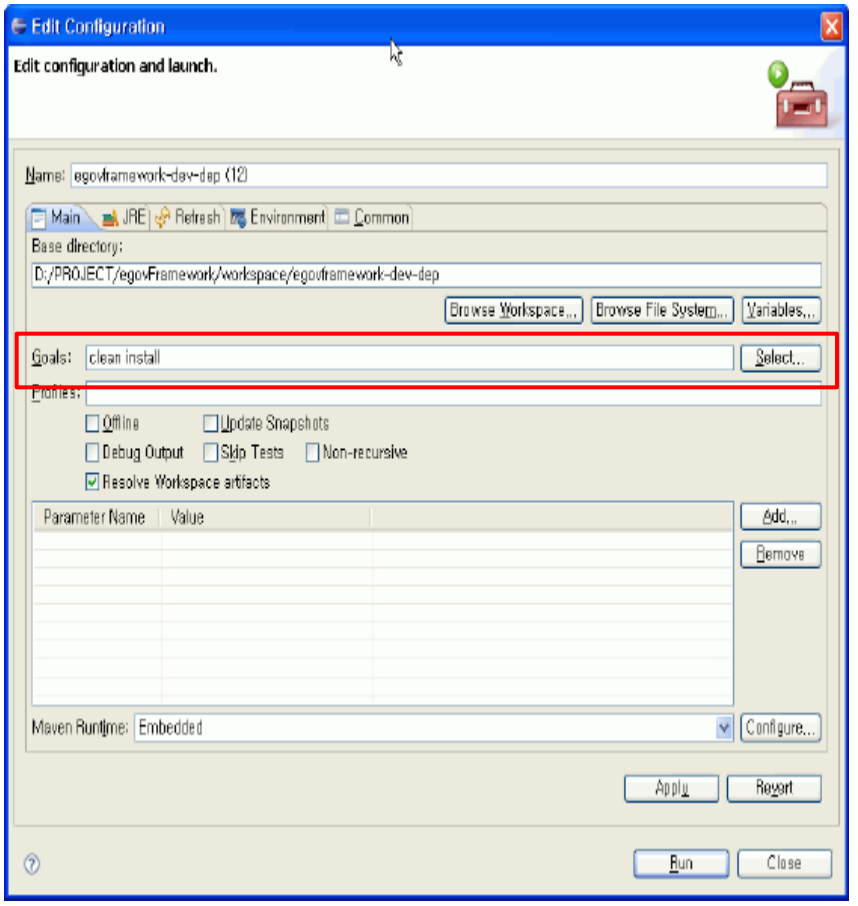


3. 빌드 Lifecycle (2/2)

이클립스에서 빌드 라이프 사이클에서 정의한 각 단계들 실행하기



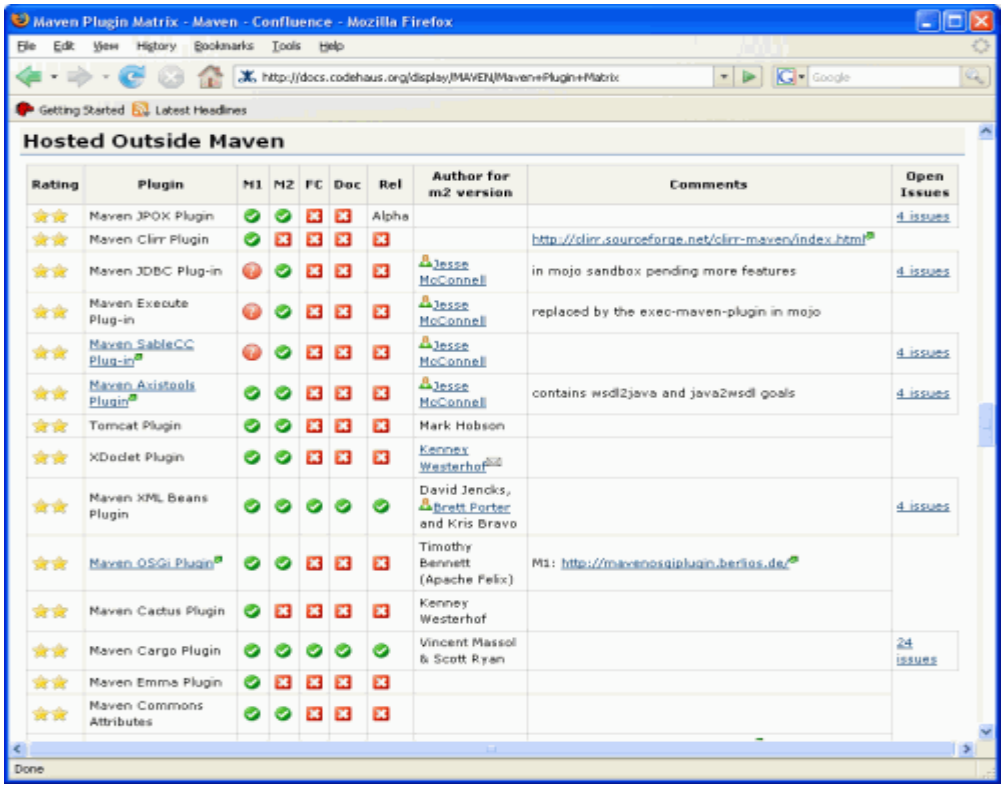
메이븐 프로젝트 -> Run As -> Maven build..



Goals: clean install

4. Maven 플러그인 (1/2)

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



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

4. Maven 플러그인 (2/2)

❑ 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>
```

5. 의존성 관리 메커니즘

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 dirvers
- **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

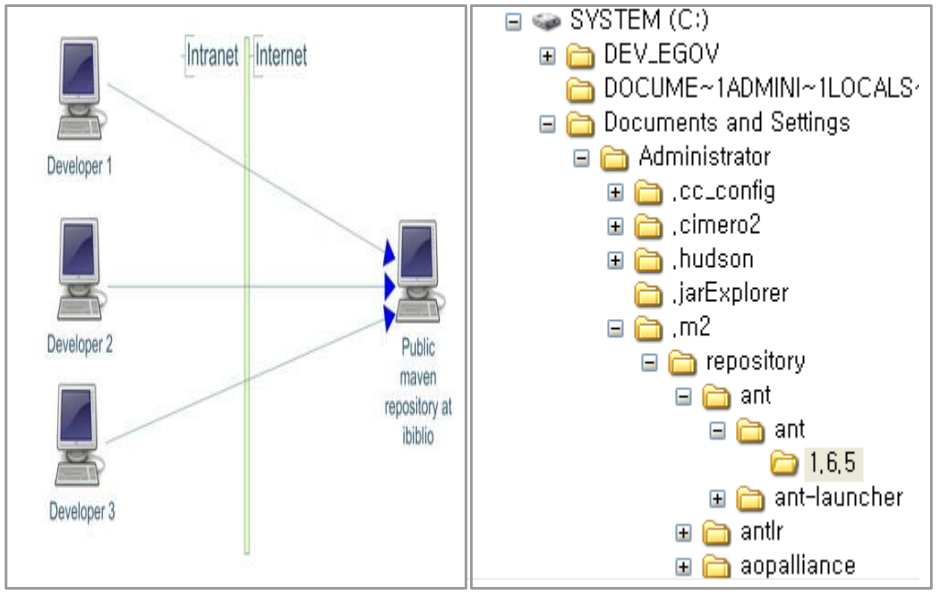
의존성 선언 항목

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 artifcat의 일관성을 추구할 수 있음

6. Maven Repository

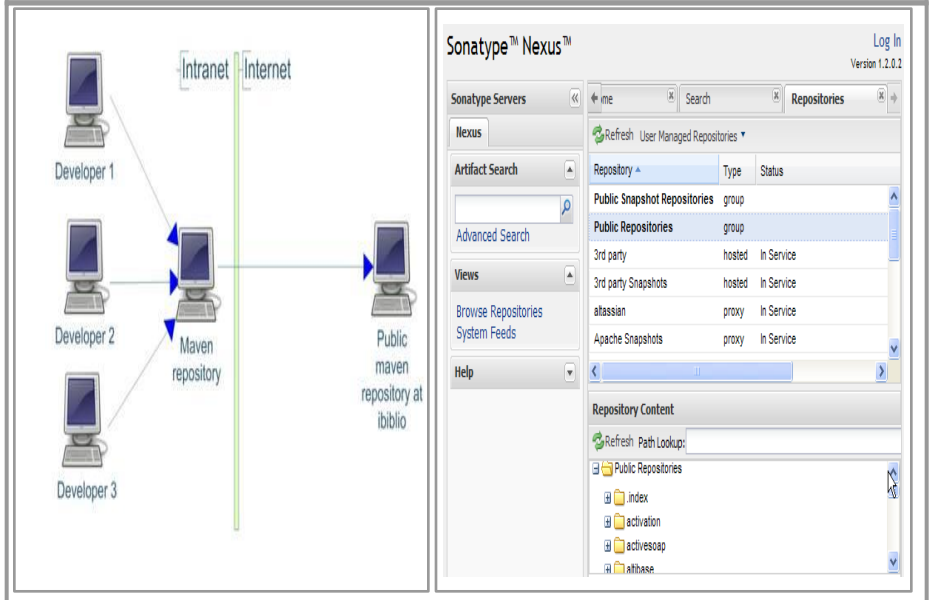
artifact들의 저장소로 로컬 및 리모트 repository로 구성되며 프로젝트는 pom.xml에서 선언한 dependency들을 저장소로부터 불러와서 사용함

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.

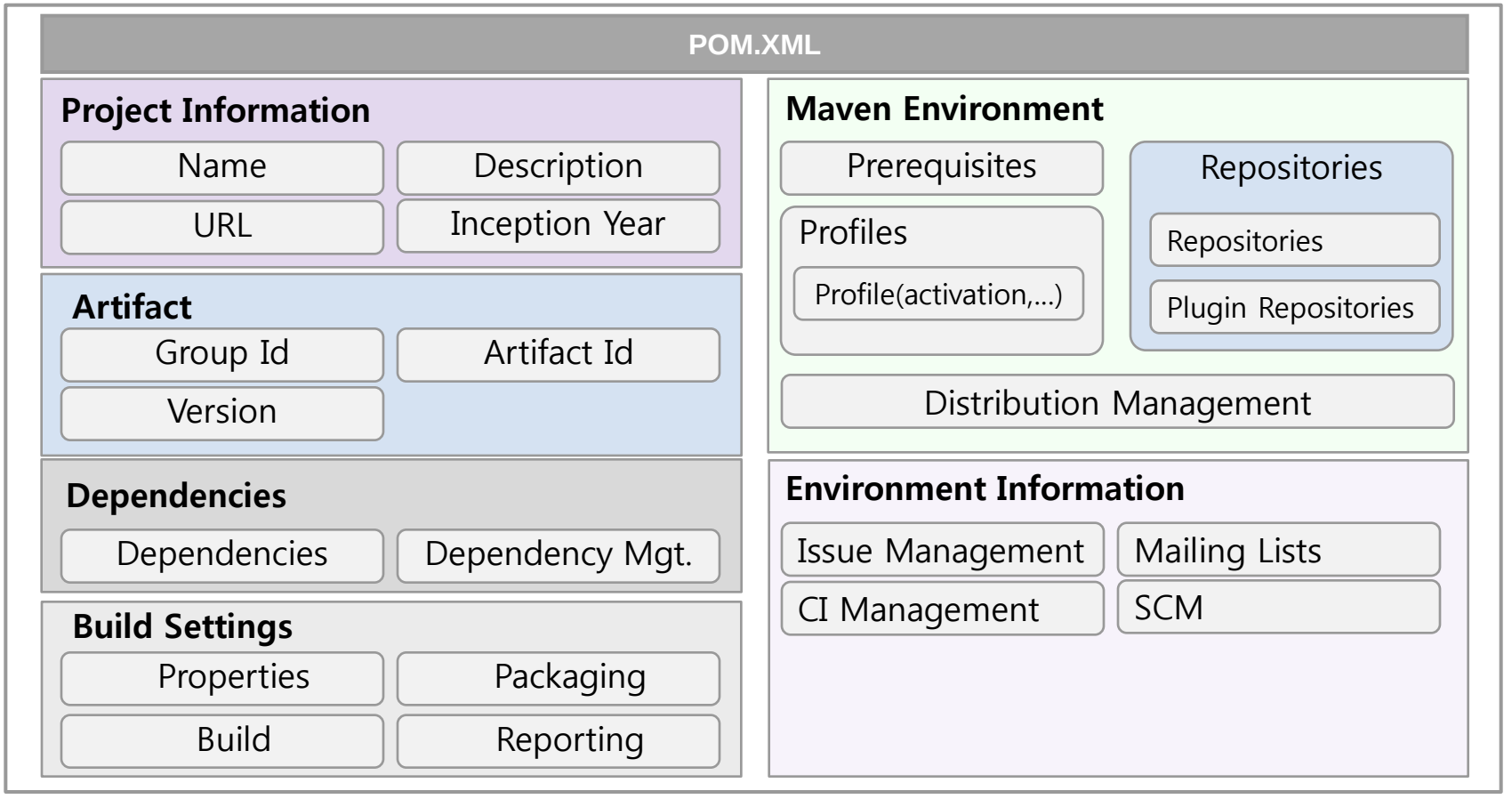
Nexus 연동



• **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.

7. 프로젝트 객체 모델 (POM) (1/2)

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

7. 프로젝트 객체 모델 (POM) (2/2)

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>1.6</source>
        <target>1.6</target>
      </configuration>
    </plugin>
  </plugins>
</project>
```

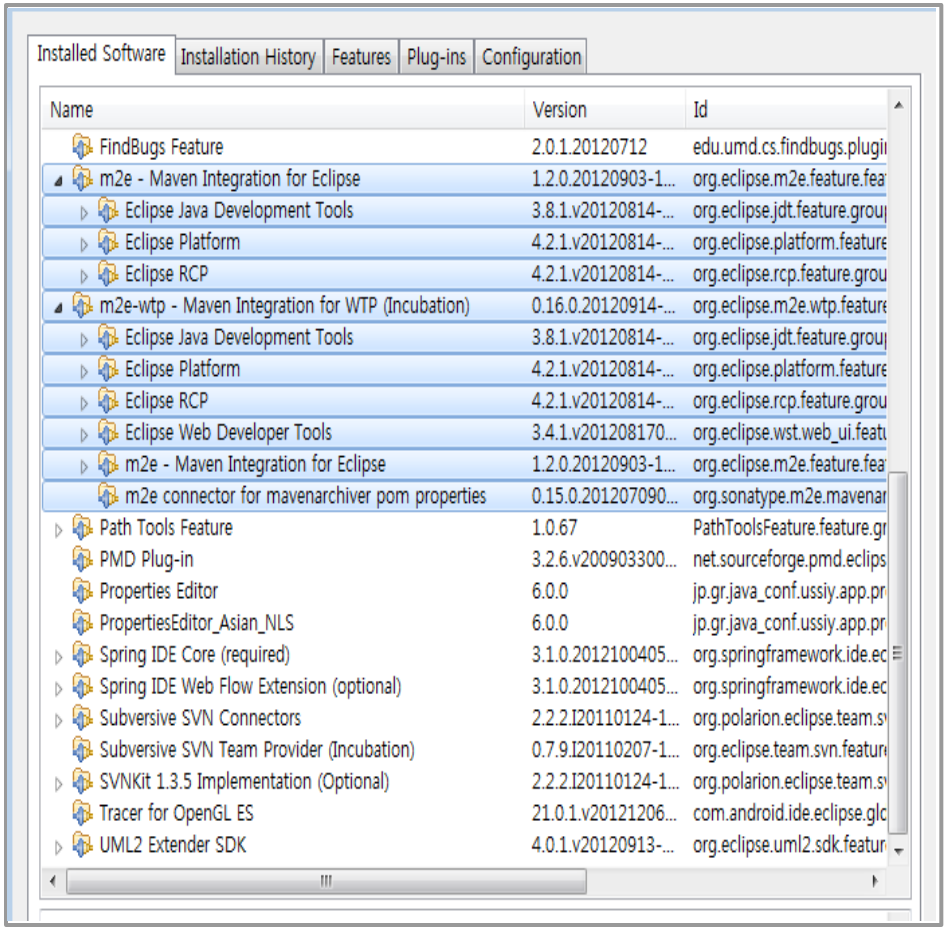
Artifact 정보

의존성 관리

플러그인 선언

8. Maven 이클립스 통합 : m2eclipse

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

8. Maven 이클립스 통합 : m2eclipse – 프로젝트 생성 (1/2)

❑ 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

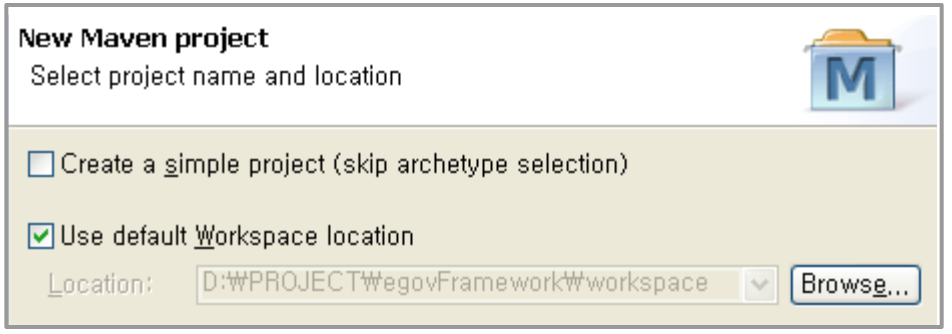
2. Maven **archetype**을 이용한 프로젝트 생성
 : 'Archetype is a Maven project templating toolkit.' - 프로젝트에 특화된 pom.xml 및 resource들을 포함한 Maven 프로젝트 생성

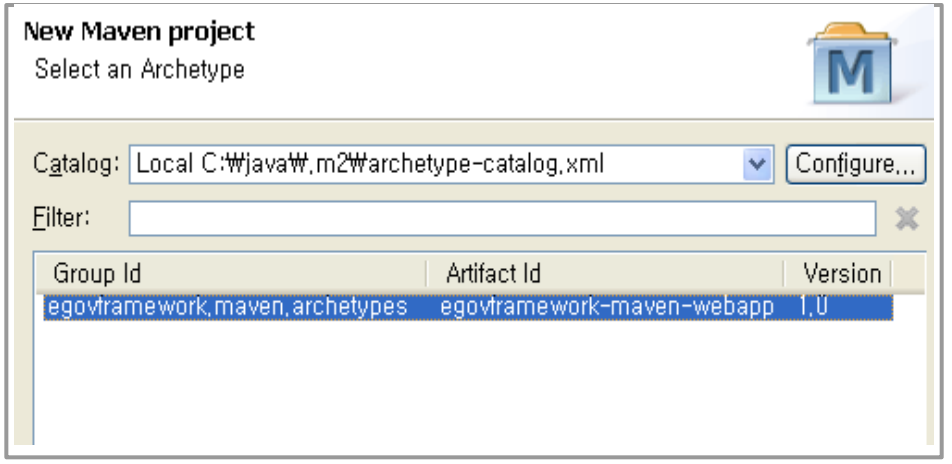
Maven Archetype 종류
1. 표 준 Archetype
- maven-archetype-j2ee-simple
- maven-archetype-quickstart
- maven-archetype-portlet

2. eGovFramework Archetype
- egovframework-maven-webapp

* Maven 제공 Archetype 종류:
http://repo1.maven.org/maven2/org/apache/maven/archetypes/

Maven 프로젝트의 archetype 선택

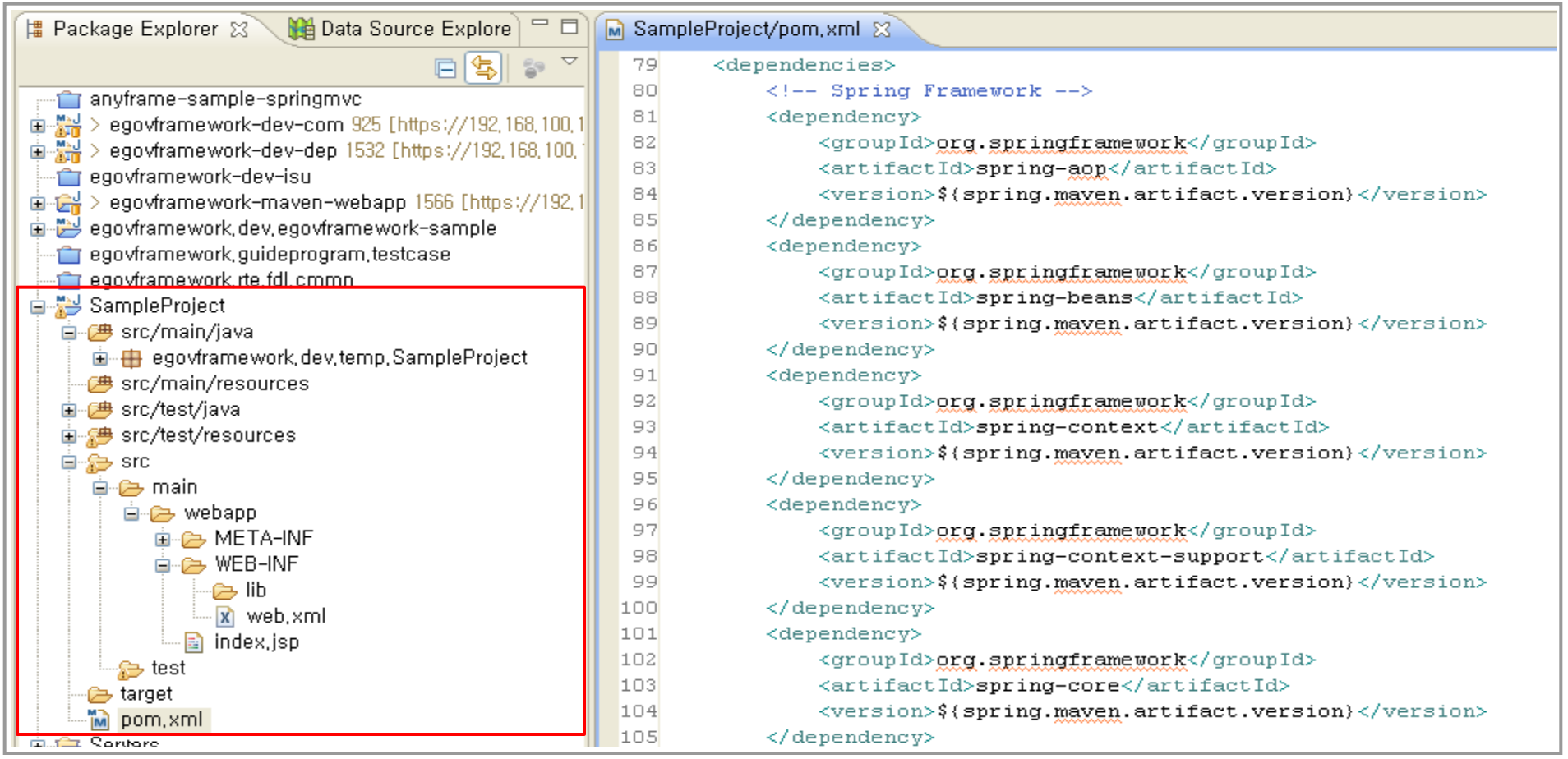




configure.. 버튼 클릭으로 제공된 archetype을 선택

8. Maven 이클립스 통합 : m2eclipse – 프로젝트 생성 (2/2)

❑ eGovFramework Archetype을 이용한 Maven 프로젝트

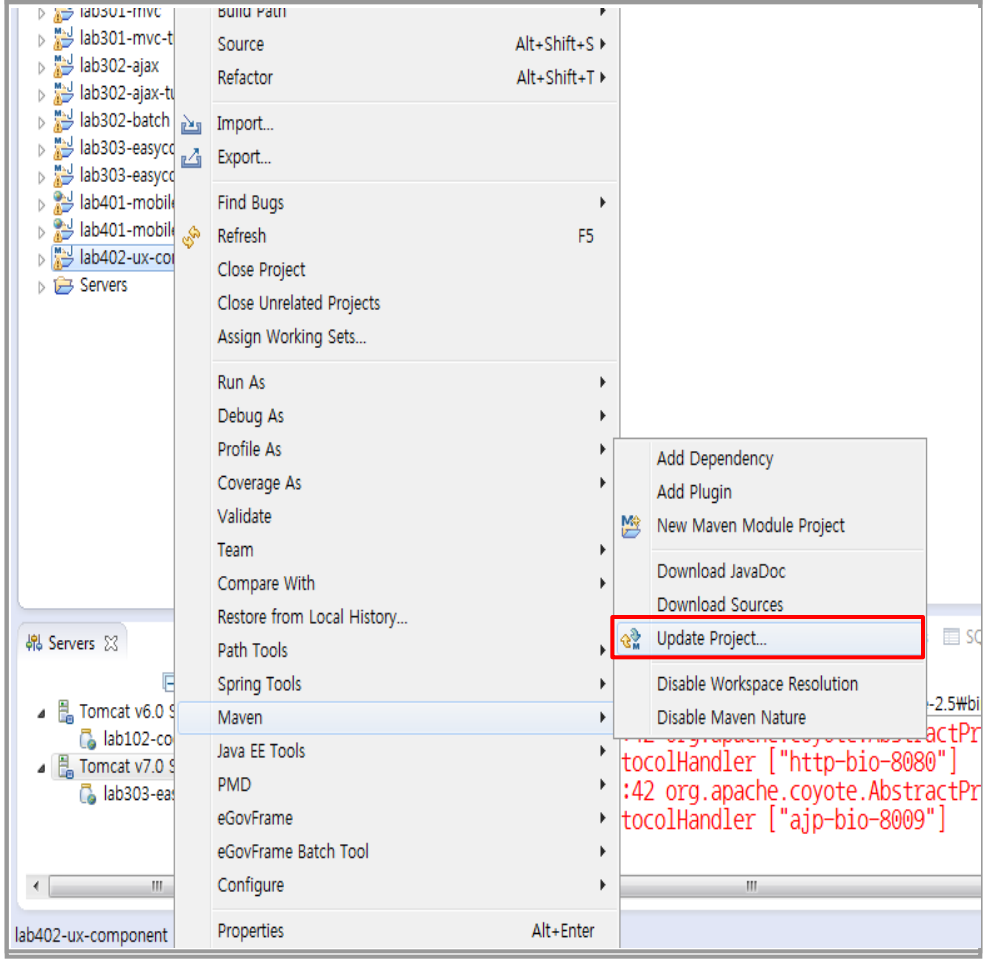


– Spring 4.0.9 기반의 dependency들을 기본으로 제공함

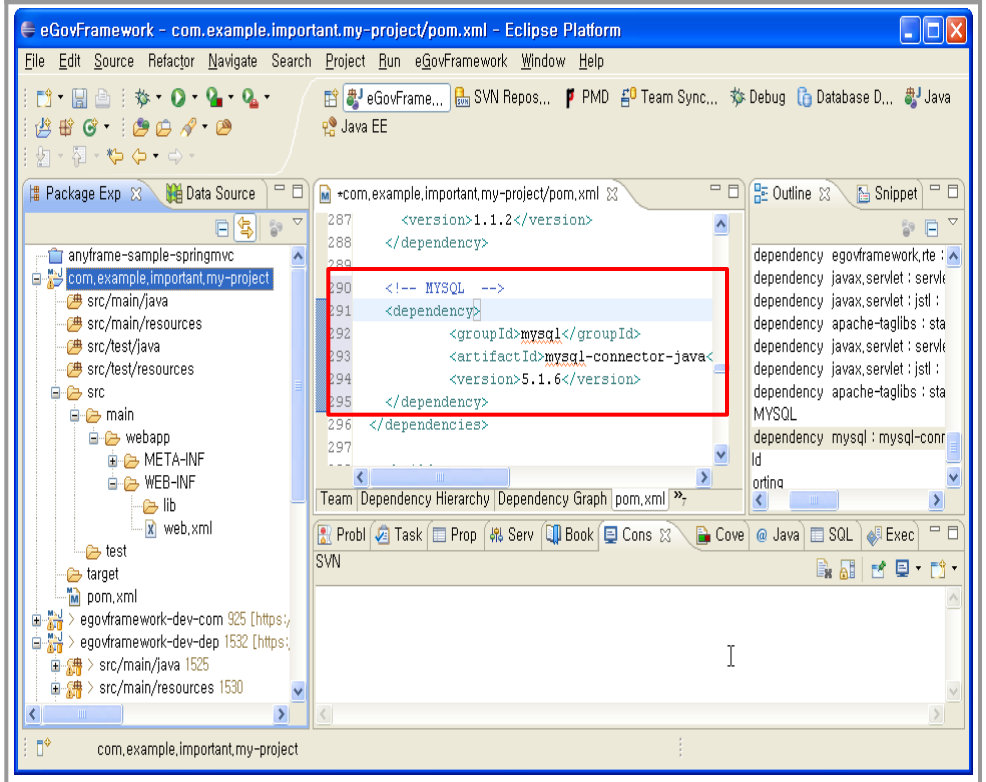
8. m2eclipse – Maven 기본 기능

❑ **m2eclipse에서 제공하는 메뉴에서 메이븐 설정 및 연동 기능을 호출할 수 있음**

m2eclipse Maven 기능



Update Dependencies 기능

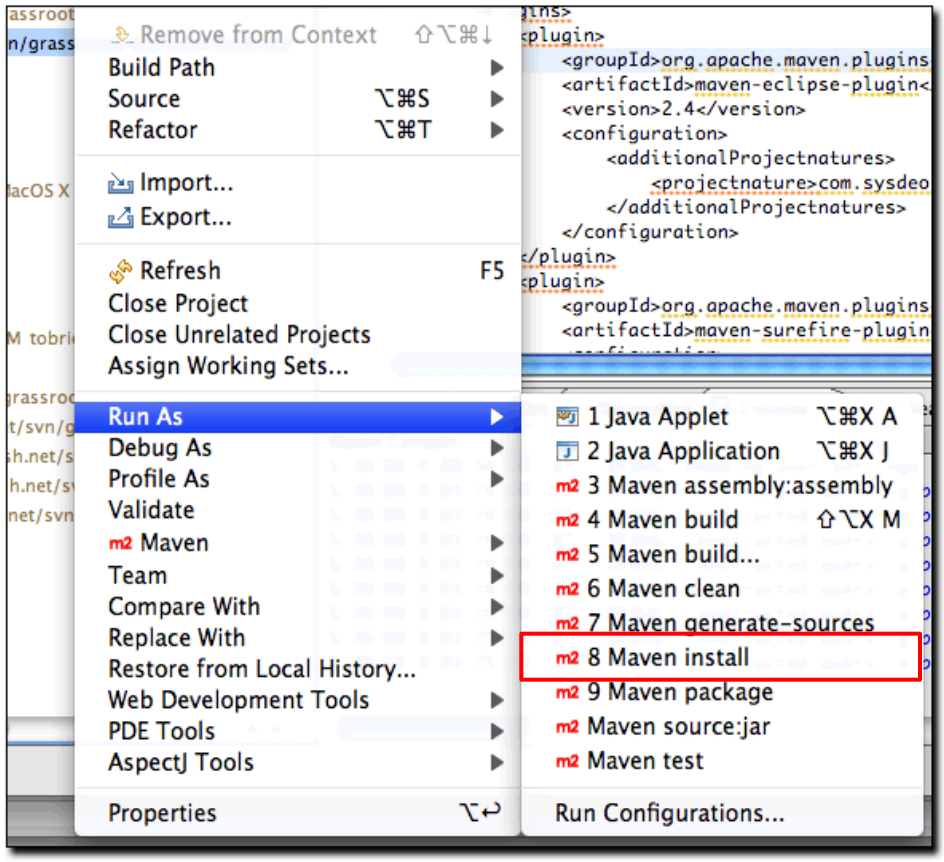


pom.xml에 dependency를 추가하고 **update dependencies** 메뉴를 선택하면 저장소에서 해당 artifact를 찾아 프로젝트에 Cache함

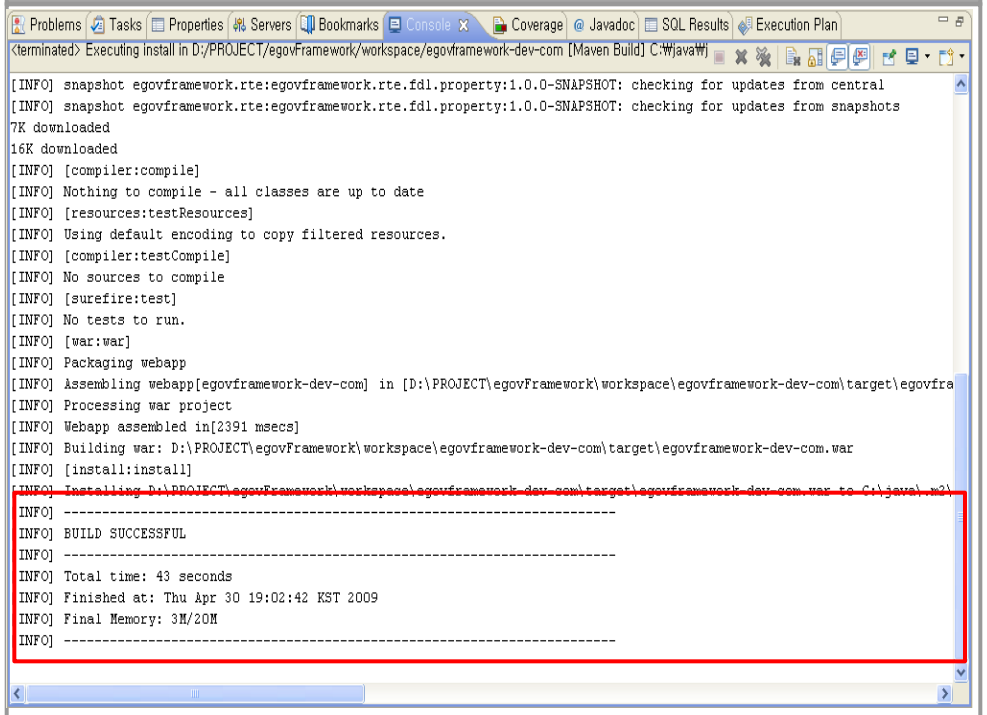
8. m2eclipse – Maven 기본 빌드 기능

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