

# 스프링(Spring) 개발

## (1) 개발환경 구성하기 [Eclipse+Maven+SVN]

### 1 개발환경 구성

#### 1.1 Eclipse 다운받기

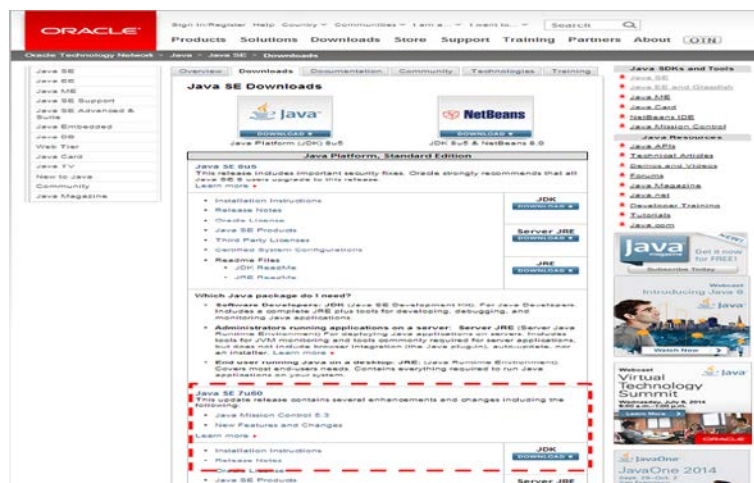
[www.eclipse.org](http://www.eclipse.org)에 들어가서 Java EE 버전의 이클립스를 다운받는다.



자신의 OS에 맞는 Eclipse를 다운받으면 된다.

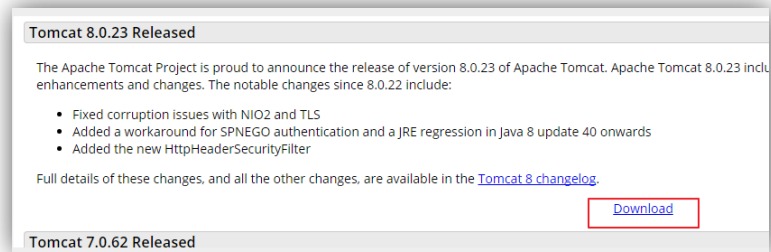
#### 1.2 JDK 다운받기

<http://www.oracle.com> (<http://www.oracle.com/technetwork/java/javase/downloads/index.html>)에 들어가서 Java SE 버전을 다운받는다.



## 1.3 톰캣 다운 받기

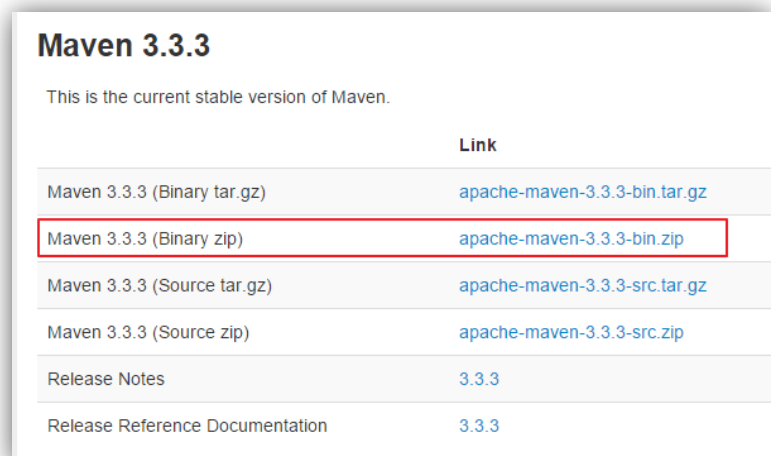
<http://tomcat.apache.org/>에 들어가서 Tomcat 을 다운받는다.



보통은 톰캣 설치파일을 받아서 설치를 진행하지만, 우리는 하나의 폴더에 다 담을거라서 그냥 압축파일을 받으면 된다.

## 1.4 Maven 다운받기

<http://maven.apache.org/>에 들어가서 Maven 를 다운받는다.



## 2. 환경 구성하기

### 2.1 압축풀기

C 드라이브 밑에 spring\dev라는 별도의 폴더를 만들고, 다운받은 프로그램들을 모두 모아 넣어 준다.

Eclipse, Tomcat, Maven은 dev 폴더 밑에 압축을 풀어준다.

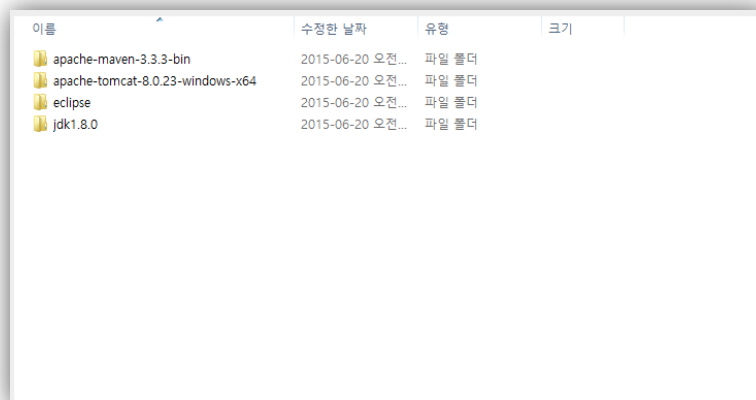
그리고 이클립스의 workspace로 사용할 폴더를 하나 만들어주자.

그 후, JDK를 설치하자.

그러면, C:\Program Files\Java 폴더에 jdk1.8.0 라는 폴더가 생겼을 것이다.

설치된 jdk1.x. 폴더를 아까 만든 dev 폴더에 복사 + 붙여넣기 한다.

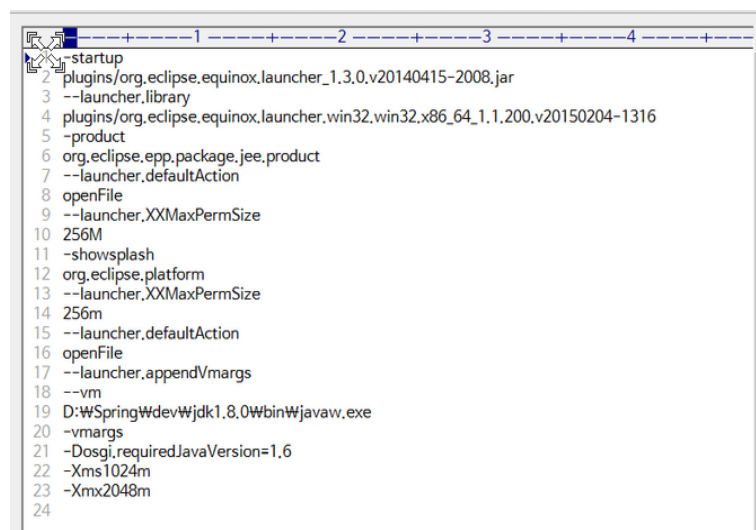
그러면 다음과 같은 폴더 구성이 되었을 것이다.



## 2.2 경로 설정하기

필요한 프로그램은 모두 다운받았으니, 이제 경로를 설정할 차례이다.

eclipse 폴더에 있는 eclipse.ini 파일을 열자.



이제 이클립스에서 우리가 설치한 JDK의 경로를 설정해 주고, JVM의 기본메모리 구성을 바꾸려고 한다.

첫번째로, JDK 설정이다.

**-vm**

**C:/spring/dev/jdk1.8.0\_45/bin/javaw.exe** 를 추가해준다.

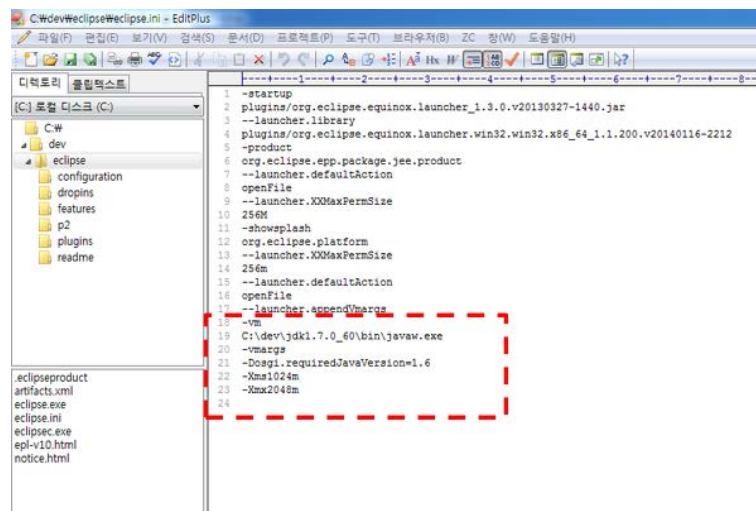
이 옵션은 vmargs 옵션 위쪽 아무 곳이나 위치하면 된다.

두번째로, -vmargs(Virtual MachineArguments) 옵션이다.

-Xms 40m, -Xmx51m로 되어있는 것을 **-Xms1024, -Xmx2048** 로 바꾼다.

(RAM이 8GB 이상일 경우만 2048m로 잡는다. 만약 RAM이 4GB 이하라면 1024 이하로 설정하자.)

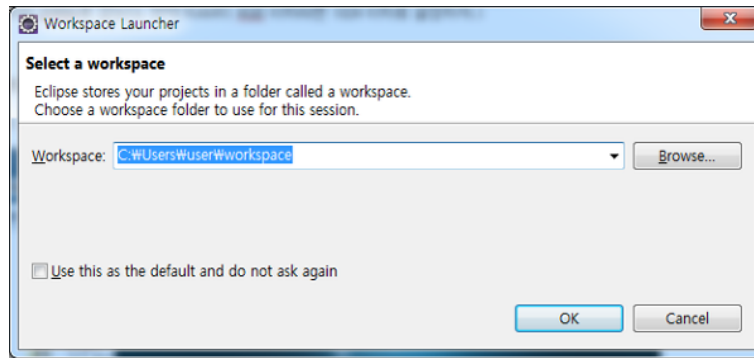
```
-vmargs
-Dosgi.requiredJavaVersion=1.6
-Xms512m
-Xmx1024m
```



이 설정을 저장하고, 이클립스를 실행시켜보자.

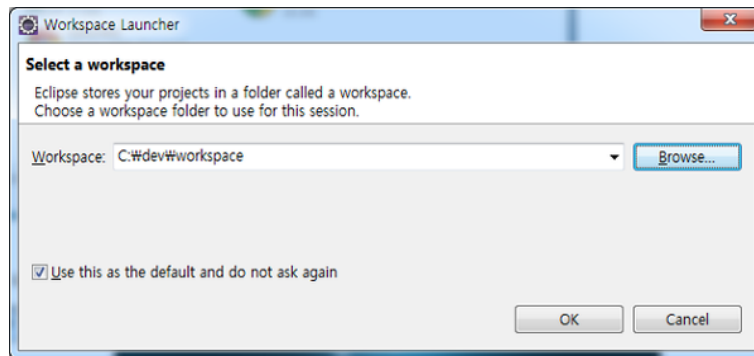
C:\spring\dev\eclipse에 있는 eclipse.exe를 실행시킨다.

여기까지 설정이 제대로 되었다면, 에러없이 실행되며 다음과 같은 창이 뜰것이다.



이 창은 이클립스의 기본 workspace를 설정하는 창이다.

우리는 spring\dev 밑에 workspace라는 폴더를 이미 만들었으니 그 폴더로 지정해 준다. 이 폴더를 계속 workspace로 사용할 것이니, Use this as the default and do not ask again도 체크해 준다.



이클립스가 실행되면 성공!

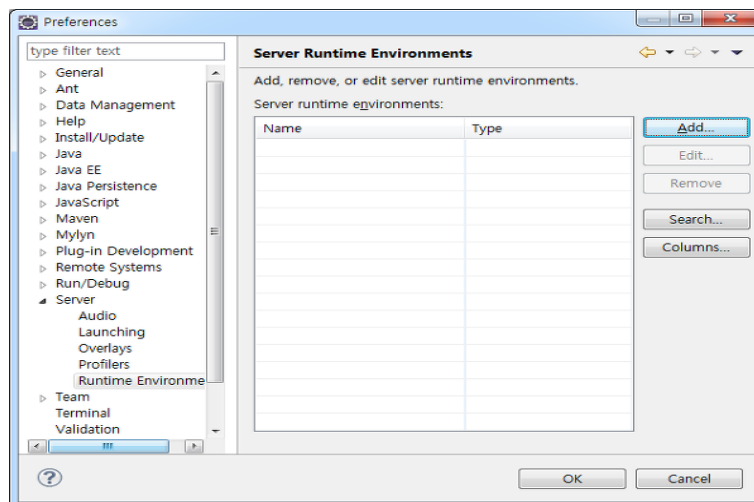
## (2) 웹 환경 구축하기

### 1. Tomcat 설정

이클립스를 실행 시키면 다음과 같은 화면이 나온다. 이제 다운받아왔던 톰캣과 메이븐의 설정을 하려고 한다.

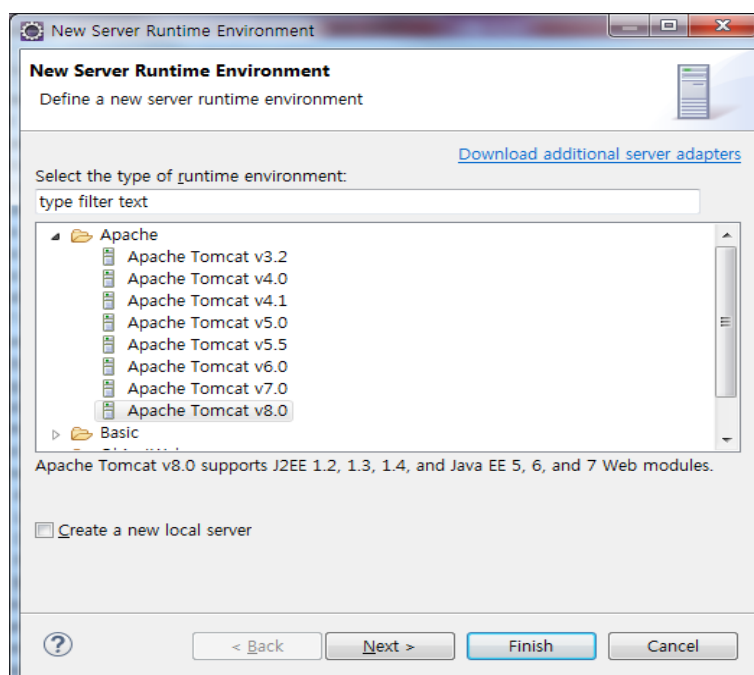
1. 상단의 Window > Preferences를 선택한다.

2. Server > Runtime Environment를 선택한다.

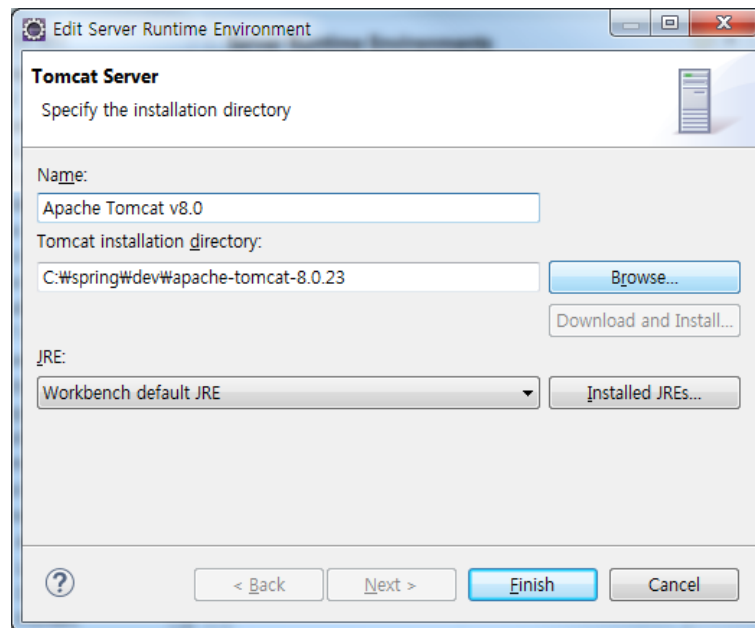


3. Add... 버튼을 누른다.

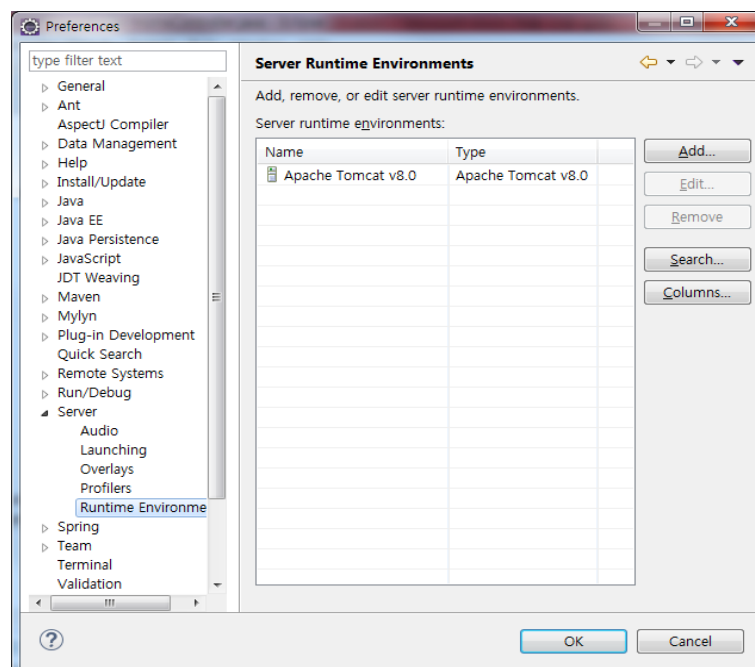
4. Apache tomcat 8.0을 선택하고 "Next >" 를 누른다.



5. JRE 및 톰캣 설정을 한다. 우리는 dev 폴더 밑에 tomcat이 있으며, 설치된 JRE는 JDK 1.8.x이므로, 이를 선택하여준다.



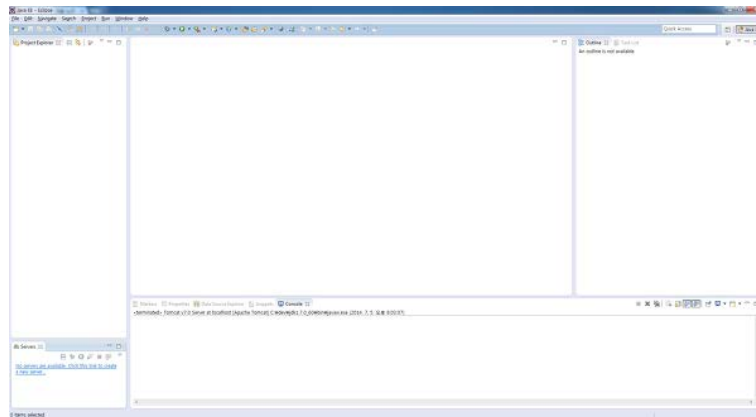
6. 다음과 같이 Tomcat이 추가된 것을 확인할 수 있다.



7. Tomcat이 구동 되는지 확인해 보자.

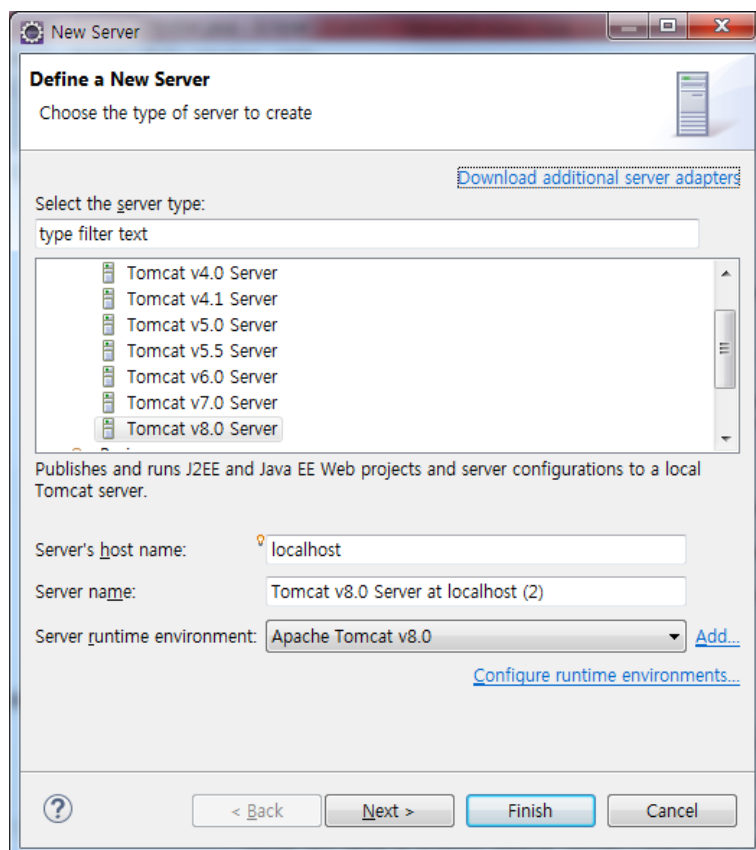
하단의 Servers Tab을 선택하자.

앞으로 개발하는데 이 Servers Tab은 굉장히 많이 쓰이게 되므로, 위치를 이동시켜 주는 것이 좋다. Servers 탭을 클릭하고 왼쪽의 Project Explorer 밑에 위치시켜 준다.



↑ 이렇게 Servers 탭 위치가 바뀌었다. (서버 탭의 크기는 각자 적절하게 조절하면 된다.)

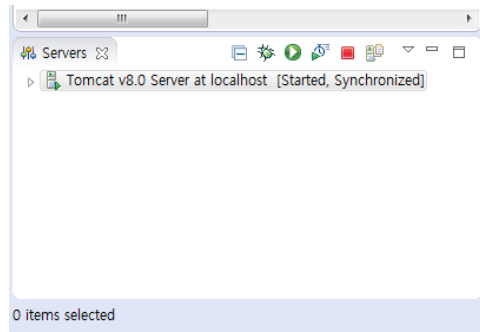
파란색으로 되어있는 No Servers are available. Click this link to create a new server를 누르자.



그러면 이와 같은 화면이 나온다.

여기서 Tomcat v8.0 Server를 선택하고 Finish를 누르자.





그럼 이렇게 서버가 생성된다. 이제 서버를 구동시켜 주자.

- 1) 서버 위에 녹색 동그라미에 ▷ 삼각형이 있는 아이콘을 클릭하거나
- 2) 서버를 우클릭하고 Start를 누르거나
- 3) 서버 선택하고 Ctrl + Alt + R을 눌러주자.



↑ 이클립스의 Console 탭에서 다음과 같이 빨간색으로 뭐라고 뭐라고 하면 끝.

(정보: Server startup in XXX ms 문구가 나오는걸 확인한다.)

## 2. Maven 설정

이제 메이븐을 설정해 줘야 한다. 메이븐은 Apache project의 하나로, 프로젝트의 관리와 관련된 여러가지 기능을 제공한다.

### 2.1 Maven Repository 변경

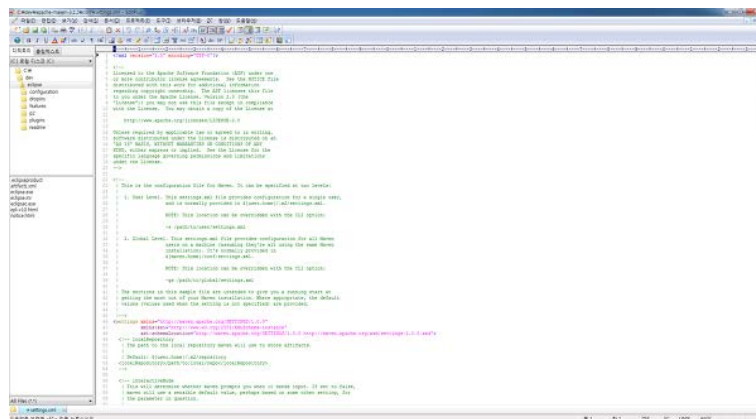
메이븐을 사용하는 여러가지 이유 중 가장 큰 이유는 프로젝트의 라이브러리들을 관리하기 쉽다는 점이다. 기존에는 라이브러리를 다운 받아서 이클립스에서 라이브러리를 추가해서 사용했는데, 버전이 바뀌거나 다른 라이브러리로 교체할 경우, 완전히 새로운 라이브러리를 요구하는 경우가 있는 등, 관리하기가 어려웠다.

메이븐을 사용할 경우, 어떤 라이브러리를 사용하겠다 라고 명시하면, 인터넷에서(사설 Repository를 관리하면 그 곳에서) 필요한 라이브러리를 자동으로 찾아서 추가해 준다. 다시 말해, 우리가 사용하고자 하는 라이브러리가 다른 어떠한 라이브러리를 필요로 하는지 고민할 필요가 없게 만들어준다.

그러므로 라이브러리를 다운 설치받는 폴더를 명시적으로 지정해 주는 것이 좋다 이 설정을 해

놓지 않으면 엉뚱한 곳에 라이브러리가 다운받아 진다.

**C:\spring\dev\apache-maven-3.3.3\conf** 폴더로 이동하여 **settings.xml** 을 열어보자.

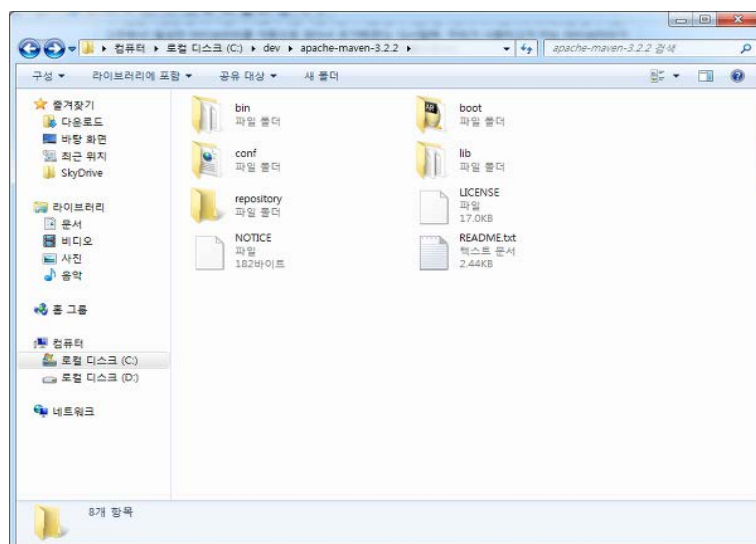


중간 썸에 주석 처리된 **localRepository** 란 부분을 찾자.

아마 Default: `${user.home}/.m2/repository` 라고 되어 있는 것을 볼 수 있을 것이다. 우리가 설정하지 않으면, 사용자 폴더에 `.m2`라는 폴더 밑에 `repository`라는 폴더에서 라이브러리가 된다. 우리는 개발환경이 설치되어 있는 `dev` 폴더에서 모든 걸 관리하려고 하니 이 `repository` 위치를 변경하려고 한다.

라이브러리는 메이븐을 통해서 관리되니까, `maven` 폴더 안에 `repository` 폴더를 생성한다.

**C:\spring\dev\apache-maven-3.3.3** 폴더 밑에 **repository** 폴더를 생성한다.



그 다음, `settings.xml`에서 **<localRepository>** 부분을 적당한 위치에 복사한 다음

**<localRepository>/path/to/local/repo</localRepository>** 라고 되어있는 부분을 방금 생성한 `repository` 폴더로 변경해 준다.

```
<settings xmlns="http://maven.apache.org/SETTINGS/1.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/SETTINGS/1.0.0 http://maven.apache.org/xsd/settings-1.0.0.xsd">

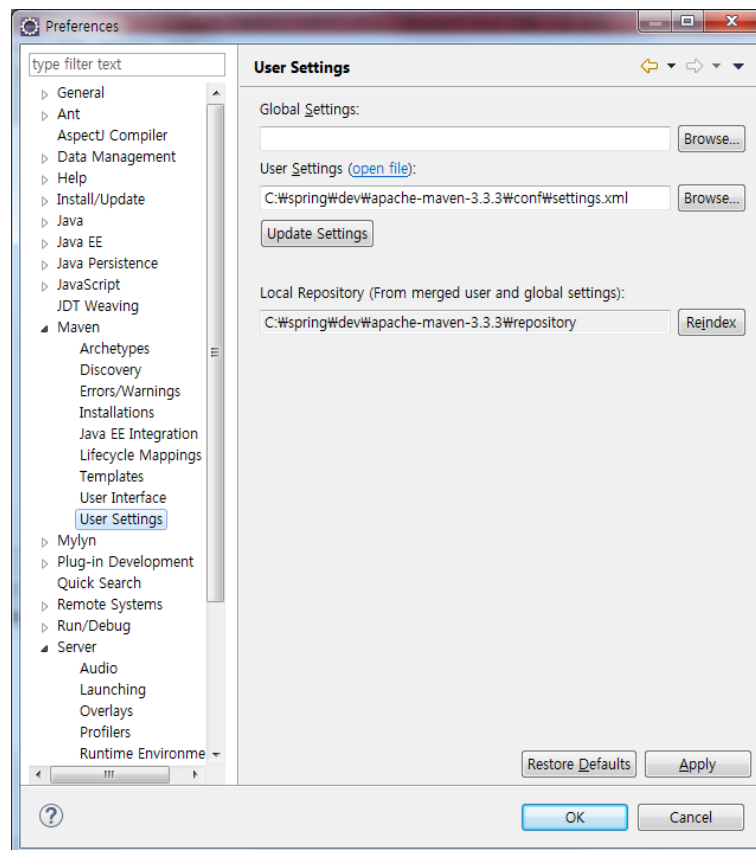
  <localRepository>C:\dev\apache-maven-3.2.2\repository</localRepository>
  <!-- localRepository
    | The path to the local repository maven will use to store artifacts.
    |
    | Default: ${user.home}/.m2/repository
  <localRepository>/path/to/local/repo</localRepository>
  -->
```

**<localRepository>C:\spring\dev\apache-maven-3.3.3\repository</localRepository>**

## 2.2 이클립스 내에서 Maven 설정

1. 상단의 window > Preference를 선택한다.

2. Maven > User Settings를 선택한다.

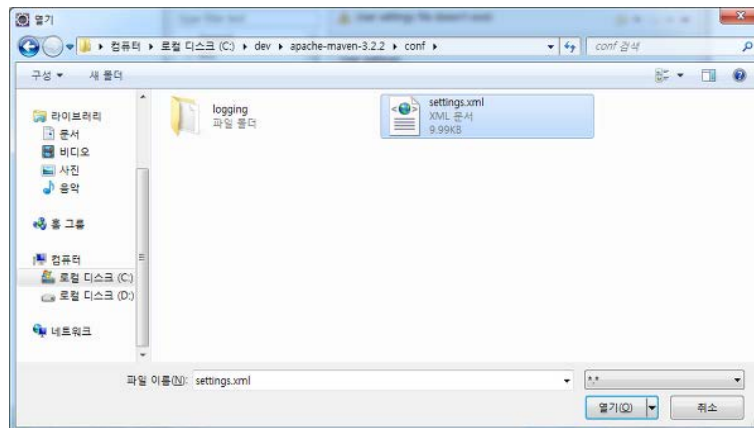


위에서 maven의 기본 설정(Default)는 사용자 폴더 \ .m2 \ repository라고 되어있었던 것을 확인할 수 있다.

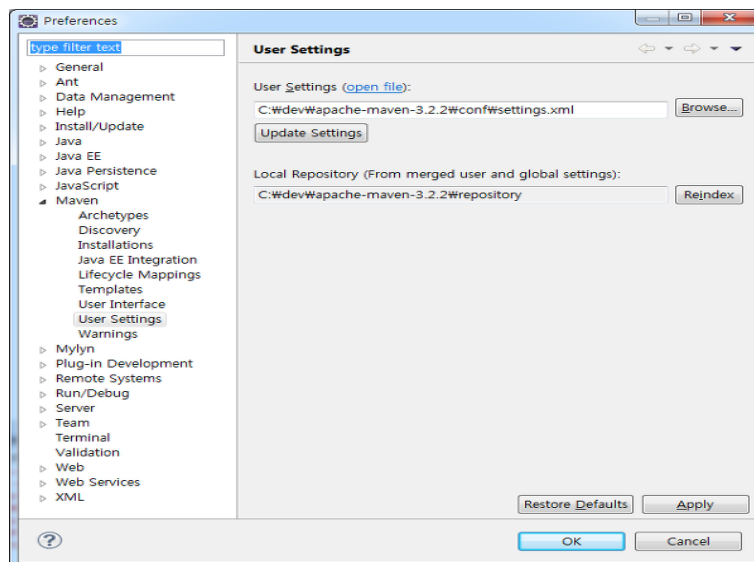
3. Browse를 클릭해서 setting.xml 파일을 선택한다.

이 setting.xml 파일은 우리가 아까 변경했던 그 파일이다.

C:\dev\apache-maven-3.2.2\conf\settings.xml 을 선택하자.



4. 변경된 설정을 확인하고 Ok를 누르자.

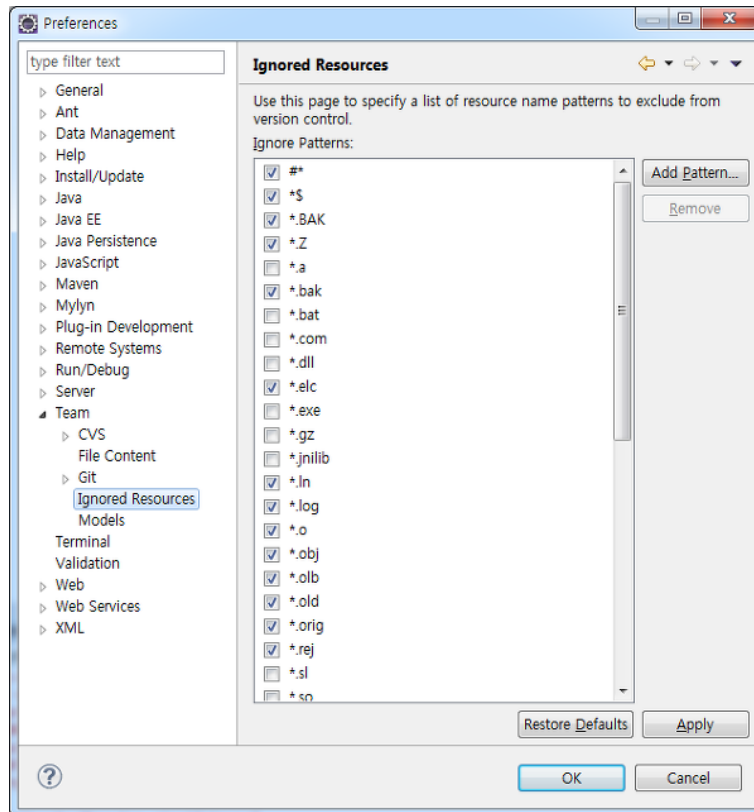


## 2.3 target 제외

maven을 이용하여 프로젝트를 진행하면 target이라는 폴더가 나온다. 프로젝트를 컴파일 하면 target/classes 디렉토리에 컴파일된 클래스 파일들이 생긴다. 추후 SVN(Subversion), GIT 등을 이용하여 프로젝트의 형상관리를 할 경우, 컴파일된 결과까지 공유 서버에 올라 갈 필요는 없으므로 컴파일된 결과는 제외하고 개발 소스만 올라가도록 설정하는 것이다.

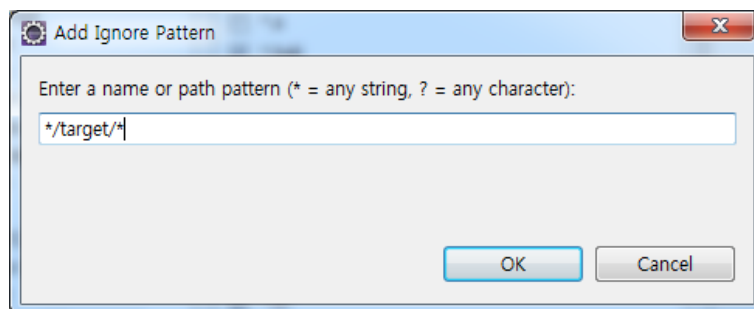
1. 상단의 Window > Preferences를 선택한다.

2. Team > Ignored Resources를 선택한다.



3. Add Pattern을 누른다.

4. \*/target/\* 을 추가한다.



5. Ok를 눌러서 적용해 주자.

### (3) 이클립스 플러그인 설치

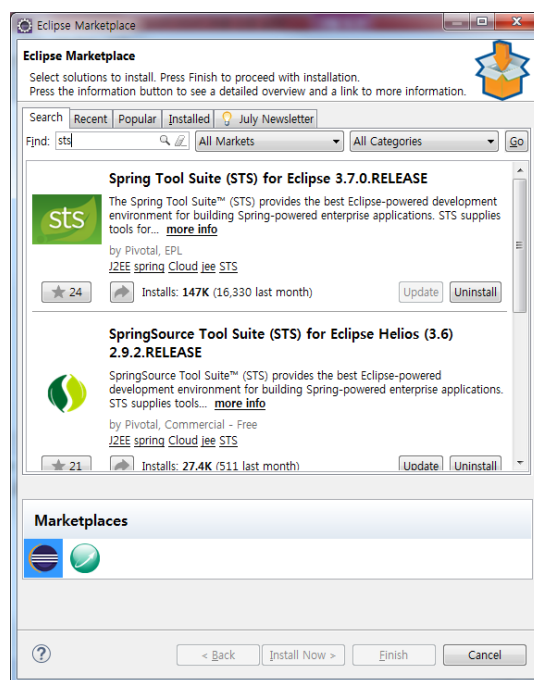
#### 1. STS(Spring Tool Suite) 설치

STS는 이클립스에서 스프링을 개발을 편리하게 하는 개발 도구이다.

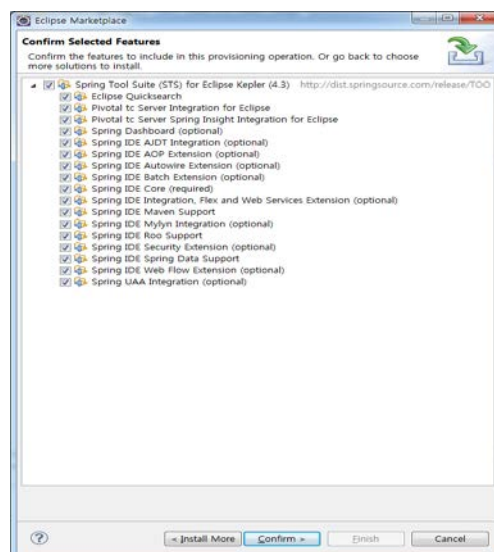
그럼 이클립스에서 STS를 설치해보자.

1. 상단의 Help > Eclipse Marketplace를 선택한다.

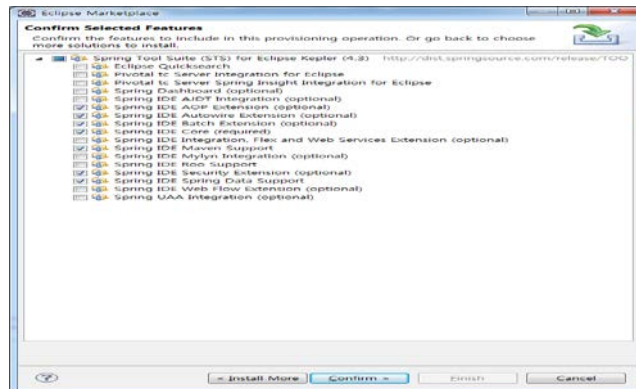
2. Marketplace에서 STS를 검색한다.



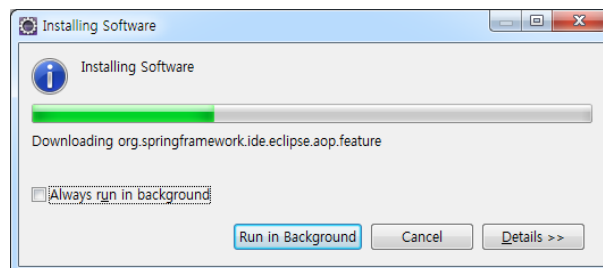
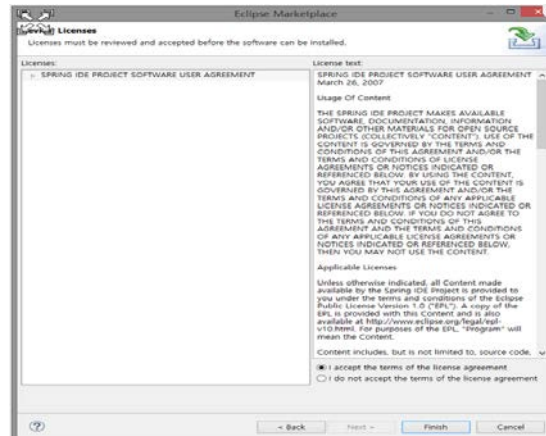
자신이 사용하는 이클립스의 버전에 맞는걸 설치하면 된다.



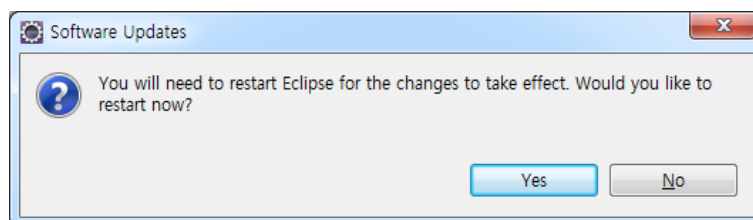
그러면 이러한 화면이 나온다. 다 설치해도 무방하다.



3. 라이선스에 동의하고 나면, 설치가 진행된다.

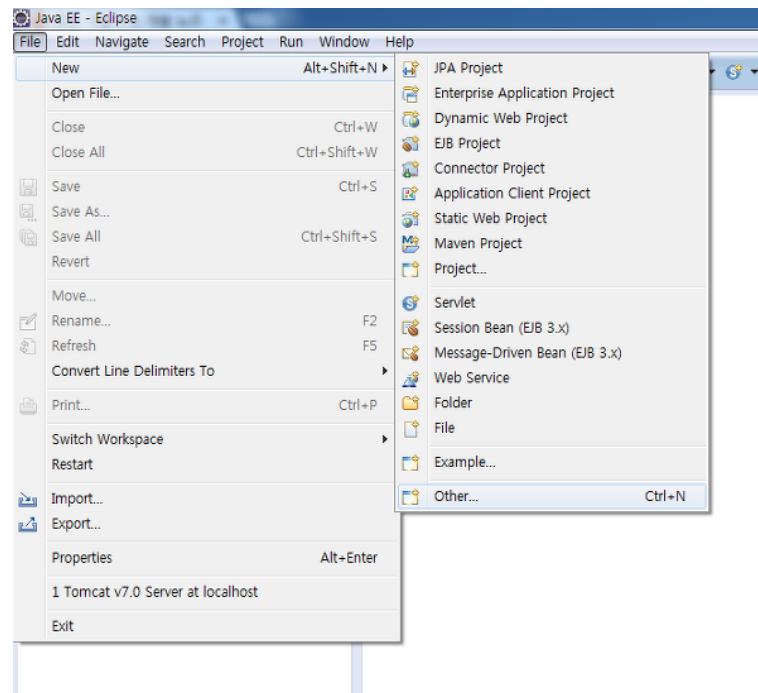


4. 설치가 완료되고 나면, 이클립스를 재시작 할거냐고 물어본다. Yes를 눌러주자.

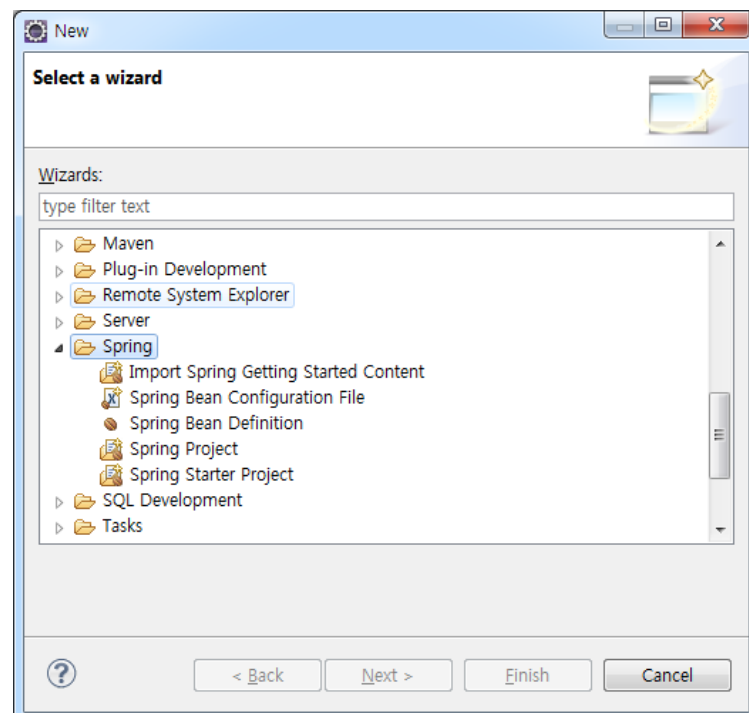


## 5. STS가 잘 설치 되었는지 확인해 보자.

1) File > New > Others를 선택한다.



2) 아래쪽에 Spring 이 보이는지 확인한다.

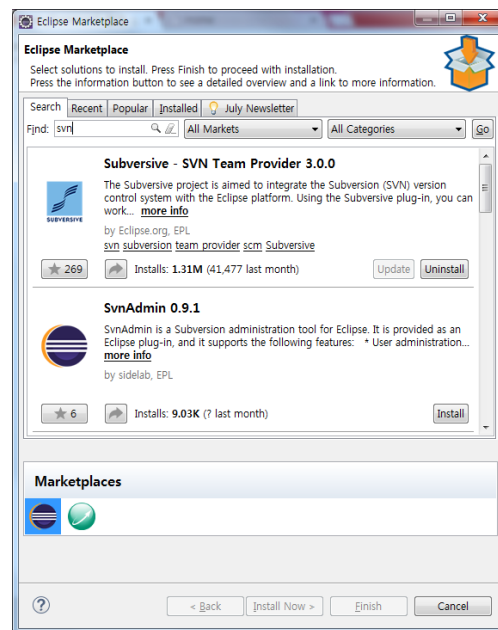




## 2. SVN(Subversion) 설치

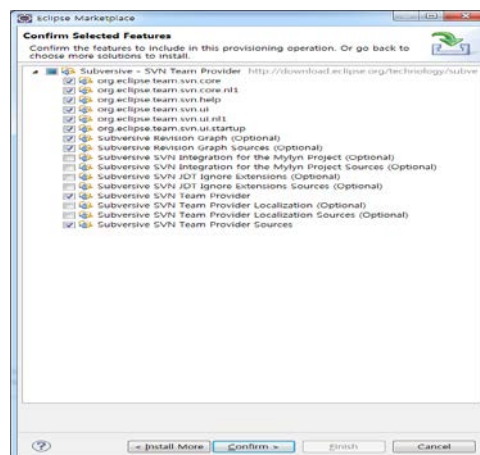
SVN은 여러명의 개발자들이 참여하는 프로젝트의 특성상, 꼭 필요한 도구이다. 아파치 프로젝트 (Apache project)중의 하나로, 지금은 전세계에서 가장 많이 사용되는 도구중 하나로, 프로젝트의 소스 관리 및 백업이 가능한 도구이다.

### 1. Eclipse Market플레이스에서 SVN을 검색한다.

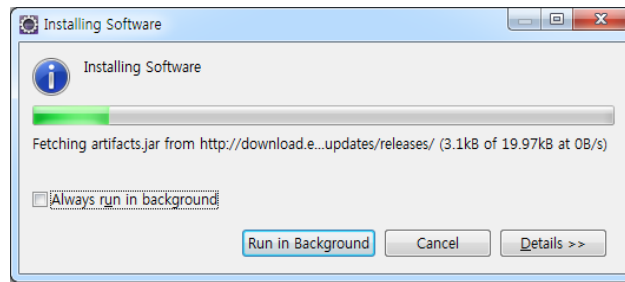


### 2. Subversive -SVN Team Provider 최신 버전을 설치하자.

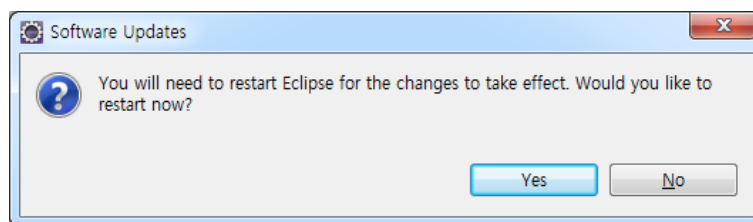
다 설치하는 게 가장 편함.



### 3. 라이선스 동의하고 설치를 진행한다.



### 4. 또 이클립스를 재시작할 거냐고 물어본다. 재시작을 해 주자.



### 5. 이클립스가 재시작 되면 SVN Connector를 설치 하라고 창이 뜬다.

이클립스에서 Subversion 을 설치하고 이클립스를 재시작하면 SVN Connector 를 설치하라는 팝업이 뜬다.

문제는 그걸 실수로 취소하거나, 아니면 이클립스의 버그인지, SVN 의 버그인지는 모르겠지만, 팝업이 아예 안뜨는 경우도 있다.

이럴 경우, 이클립스를 껐다 켜도, SVN 을 재설치 해도 Connector 를 설치하라는 팝업이 안뜬다. 이 SVN Connector 가 없을 경우, 이클립스와 SVN 이 연결이 안 되기때문에 설치한 SVN 플러그인은 아무 쓸모가 없게 된다.

그럴 경우, SVN Connector 를 직접 설치할 수 있다.

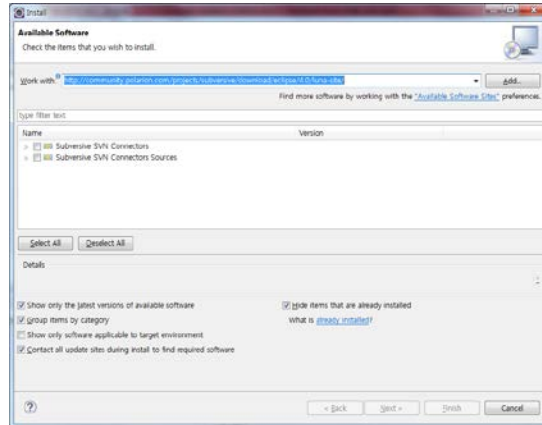
#### 1. Help > Install New Software 를 선택한다.

<Luna 의 경우>

<http://community.polarion.com/projects/subversive/download/eclipse/4.0/luna-site/>

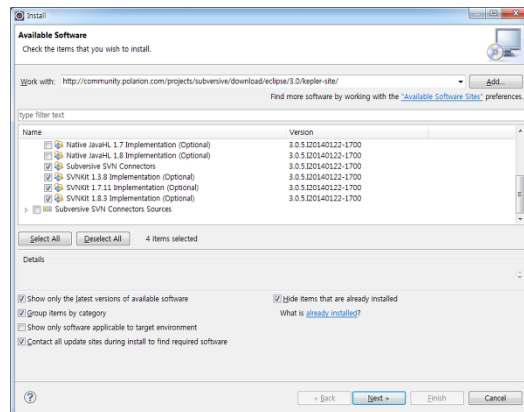
#### 2. Work with : 에 위에 있는 주소를 복사해 넣는다. >> Add 버튼 클릭함

>> 대화상자에서 그냥 ok 하면 아래에 항목이 나타남

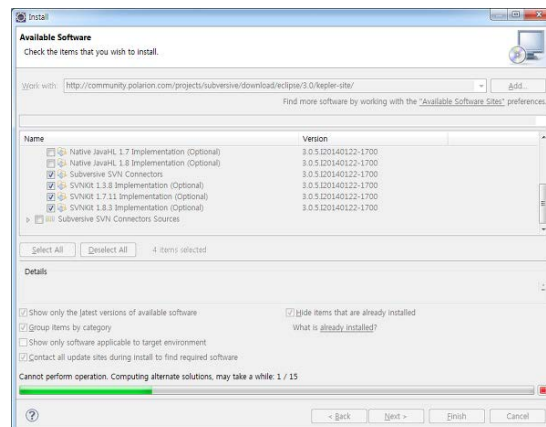


3. SVN Connectors 를 설치한다.

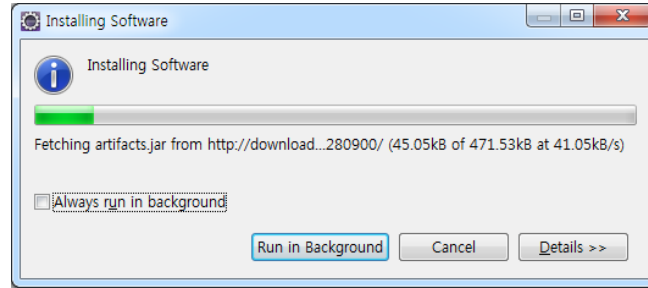
SVN Connectors를 수동으로 설치하는 모습이다. 우리는 Sources는 필요없으므로 Subversive SVN Connectors를 설치한다.



Subversive SVN Connectors의 하위 목록에는 위와 같이 여러가지 설치 목록이 있다. 몇몇 상황에서 안되는 경우도 있으므로 **전부 설치해** 주자.



설치하고, 경고가 나오는데 그냥 넘어가면 된다. 그 후, 이클립스를 재시작 하라고 한다.



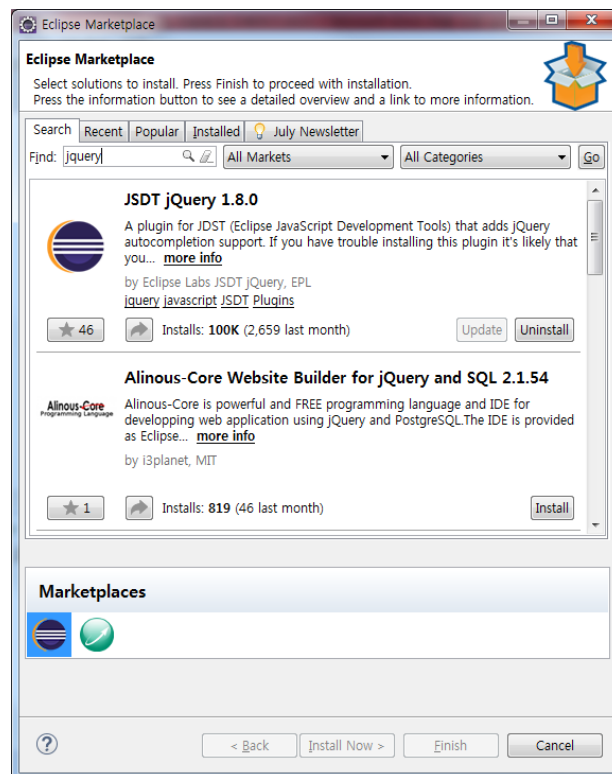
### 3. JSDT jQuery 설치

웹 개발을 하다보면 jQuery를 굉장히 많이 사용하게 된다. 문제는, jQuery의 수많은 기능들을 우리가 다 알지 못하고, IDE에 익숙해진 우리는 Ctrl + Space를 눌렀을때, 아무것도 안 나오면 살짝 패닉이 오는 경우도 있다.

JSDT jQuery는 이클립스 내에서 jQuery에 관련된 코드 어시스트 기능을 사용할 수 있도록 도와준다.

1. Marketplace에서 jquery를 검색한다.

2. JSDT jQuery 최신 버전을 설치한다.



3. 라이선스 동의하고, 설치하다 보면 뭐라고 경고 한번 나오면 ok하고 나면, 재시작하라고 한다.

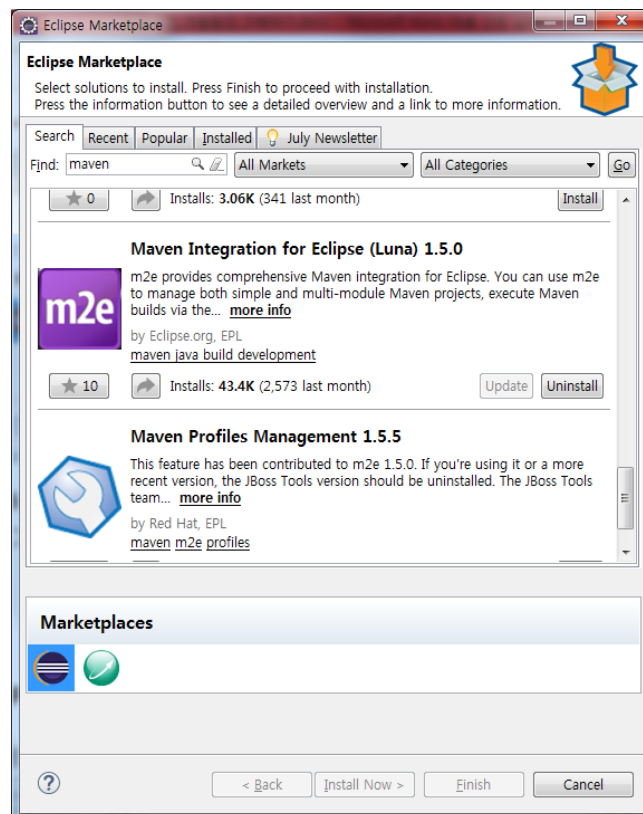
이클립스가 재시작 되면 JSDT는 설치가 된 것이다.

## 4. Maven Integration 설치

1. Marketplace에서 maven을 검색한다.

2. 검색 결과에서 밑으로 스크롤을 내리면 중간쯤에 " Maven Integration for Eclipse (Luna) 1.5.0" 라는 게 보인다.

Maven Integration도 이클립스 버전에 따라 몇 가지 존재하기 때문에, 자신이 사용하는 이클립스 버전을 확인하고, 설치해야 한다.



그런데 이클립스 Indigo 이상 부터는 Maven 플러그인이 설치되어서 나온다. 따라서, 자신의 이클립스에 메이븐 플러그인이 설치되어 있는지 확인하고, 설치되지 않은 경우에만 설치를 진행하면 된다.

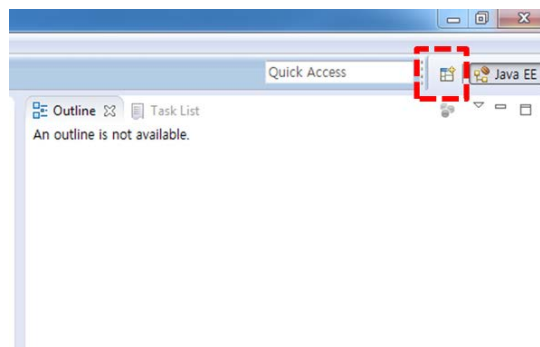
**설치 확인 :** Help 메뉴 >> About Eclipse 선택 >> 아래쪽의 Installation Details 선택해서 확인 가능함



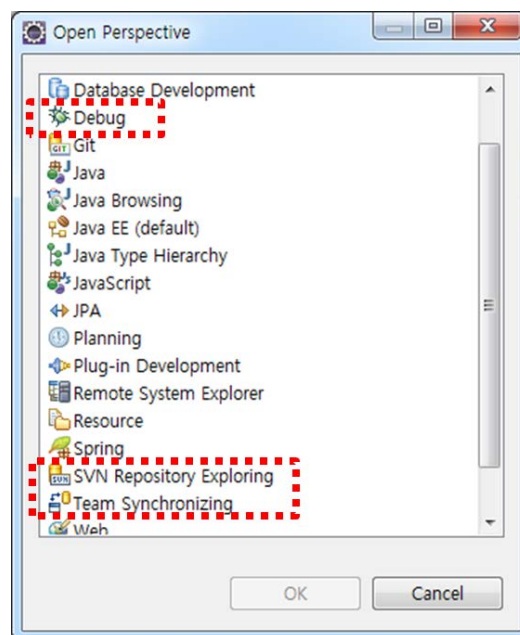
## 5. 이클립스 Perspective 변경

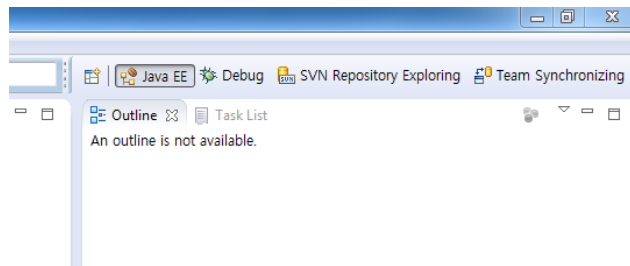
이클립스의 뷰 배치를 조금 바꿔서, 개발 하기에 편하게 하려는 것이다.

1. Eclipse의 오른쪽 상단에 있는 Open Perspective 버튼을 누른다.

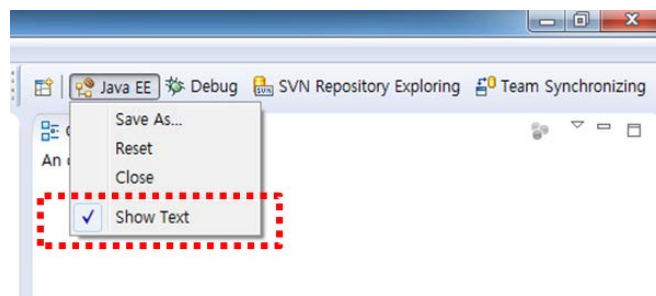


2. 다음 3가지를 추가하려고 한다. Debug, SVN Repository Exploring, Team Synchronizing.





3. 이대로 사용해도 무방하지만, 글자가 길다보니 자리를 너무 많이 차지하는 것 같다. 따라서, 글자는 없애고 이미지만 보려고 한다. 글자 위에서 우클릭을 한다.

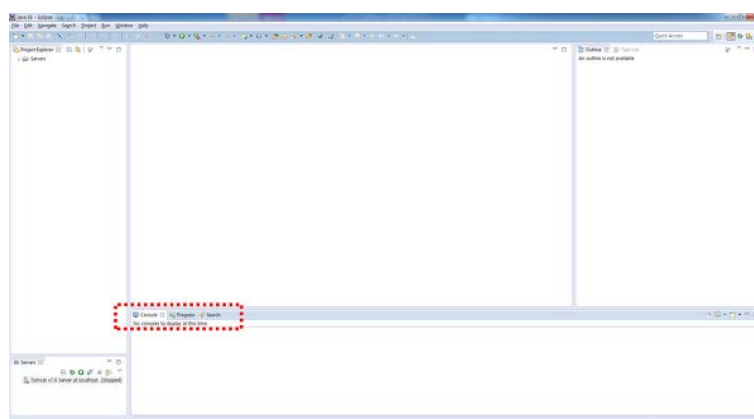


그러면 다음과 같이 나오면, **Show Text**를 체크 해제 한다.

여기까지 진행하면 기본적인 환경설정은 거의 끝났다.

나머지는 개인적으로 필요한 플러그인이나 이클립스의 메뉴 배치 정도가 남았다.

이클립스 하단의 View도 살짝 바꿔본다.



상단의 Window > Open View를 선택하고, 자신이 필요한 View를 추가하면 된다. 밑에 있는 Console, Progress, Search, Problems 등 가장 많이 사용하는 view를 놓는다.