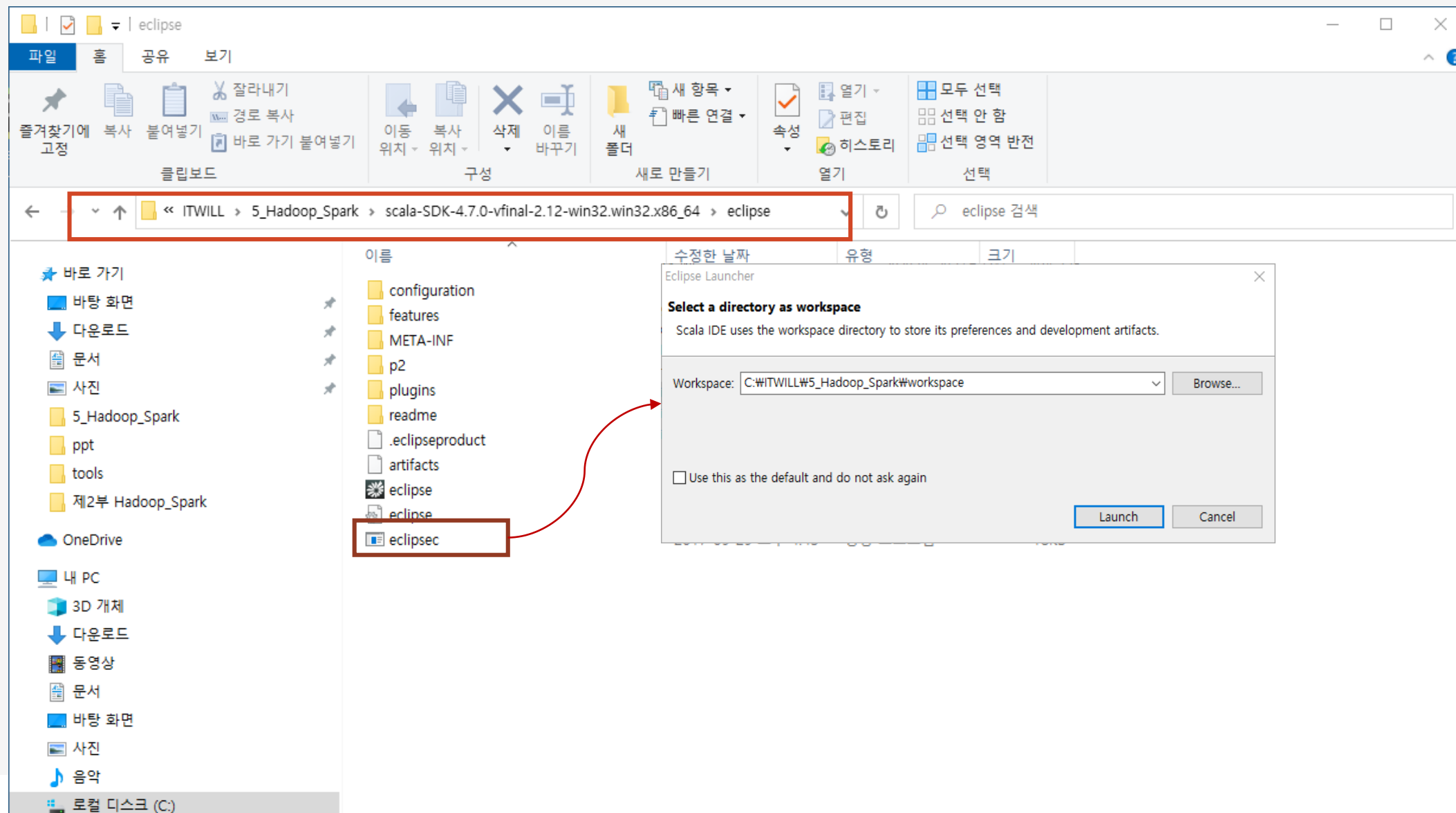


빅데이터 플랫폼 머신러닝 개발을 위한

SparkML 개발환경 구축(Maven Project)

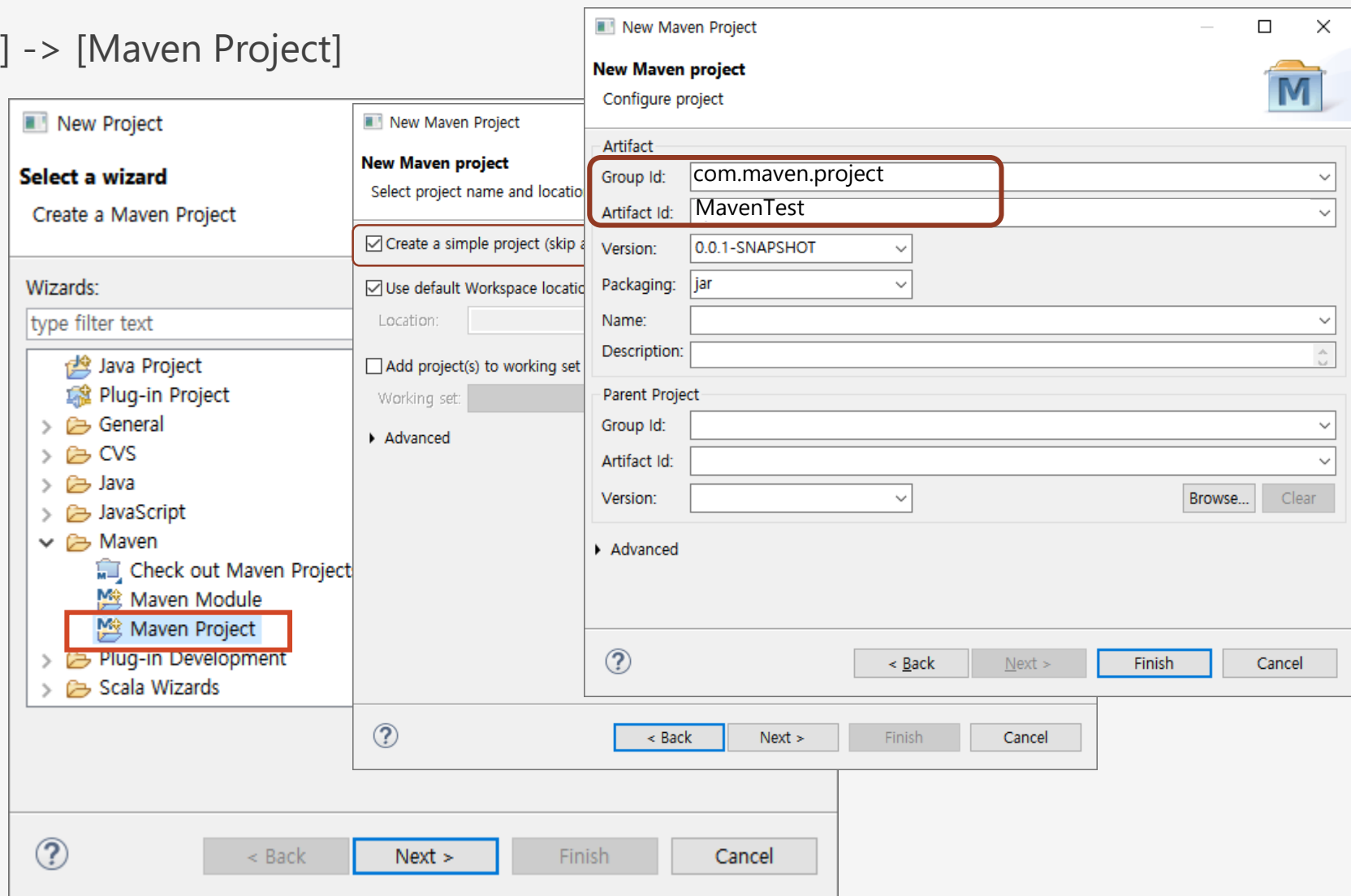
작성자 : 김진성

1. Eclipse 실행



2. maven project 생성

[File] -> [New] -> [Project] -> [Maven Project]

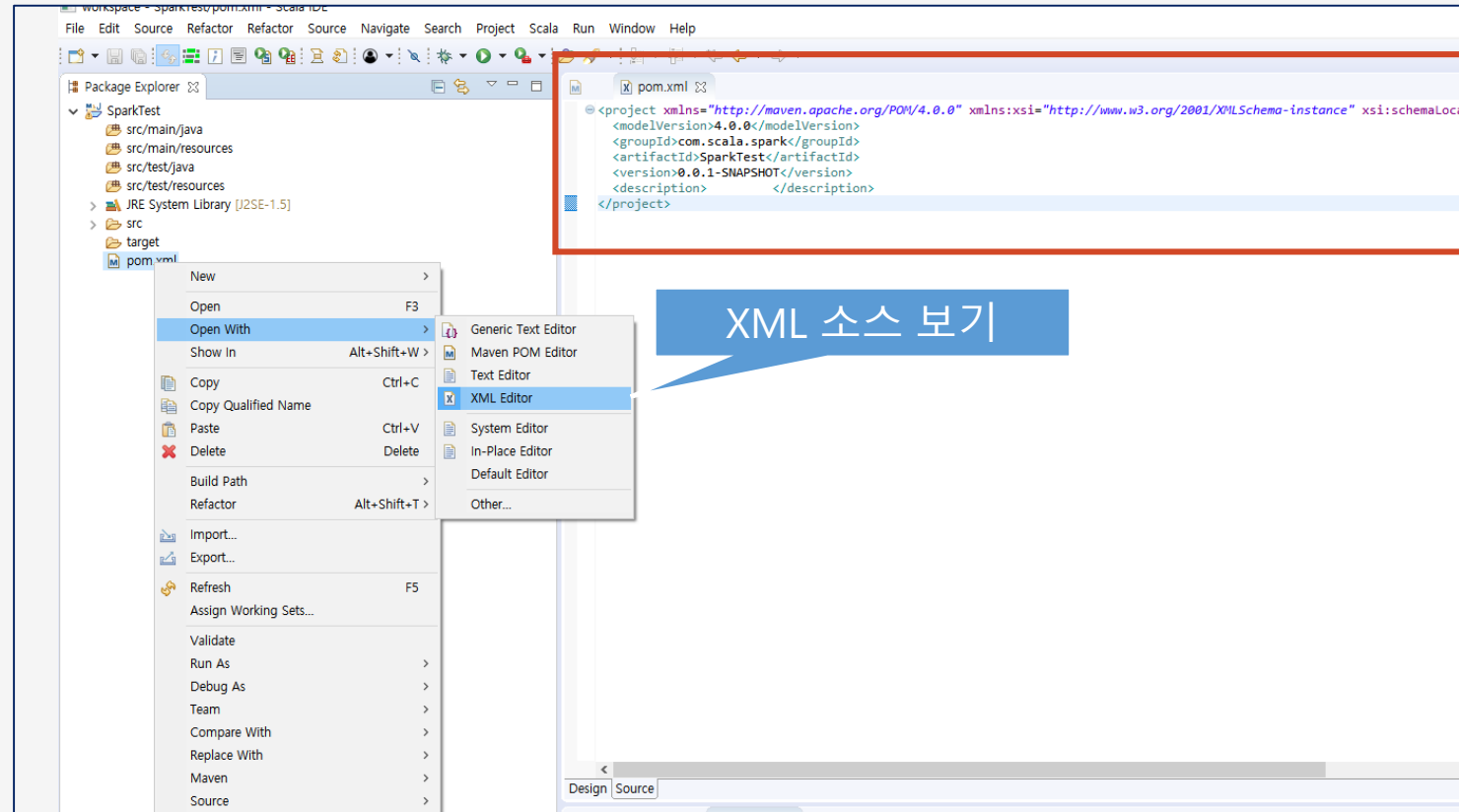


1) pom.xml 설정

Maven Project에서

개발에 필요한 라이브러리를

관리하는 설정파일

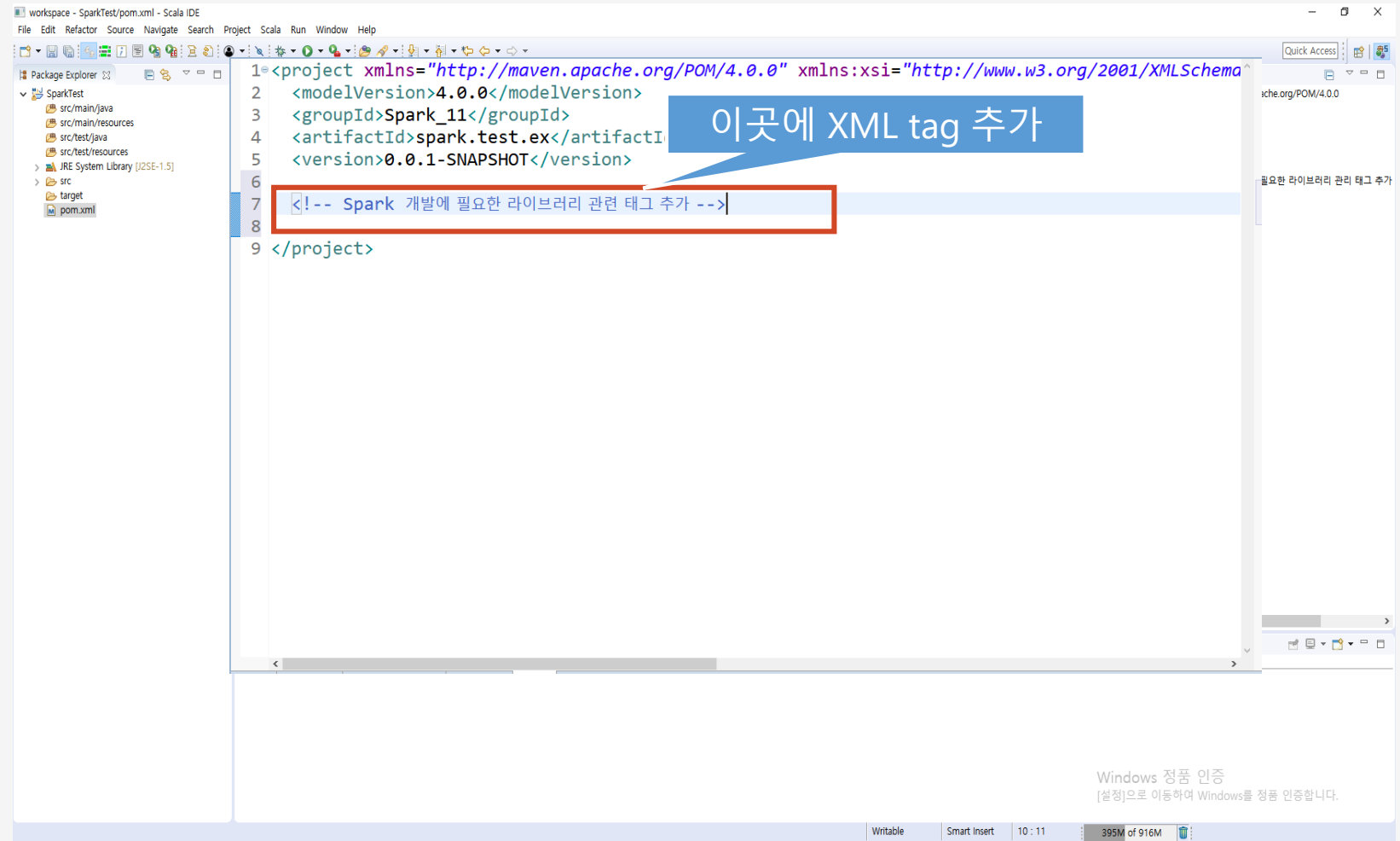


1) pom.xml 설정

Salac, Spark 개발에 필요한

라이브러리 관련 설정

태그 추가



1) pom.xml 설정

라이브러리 관련 설정

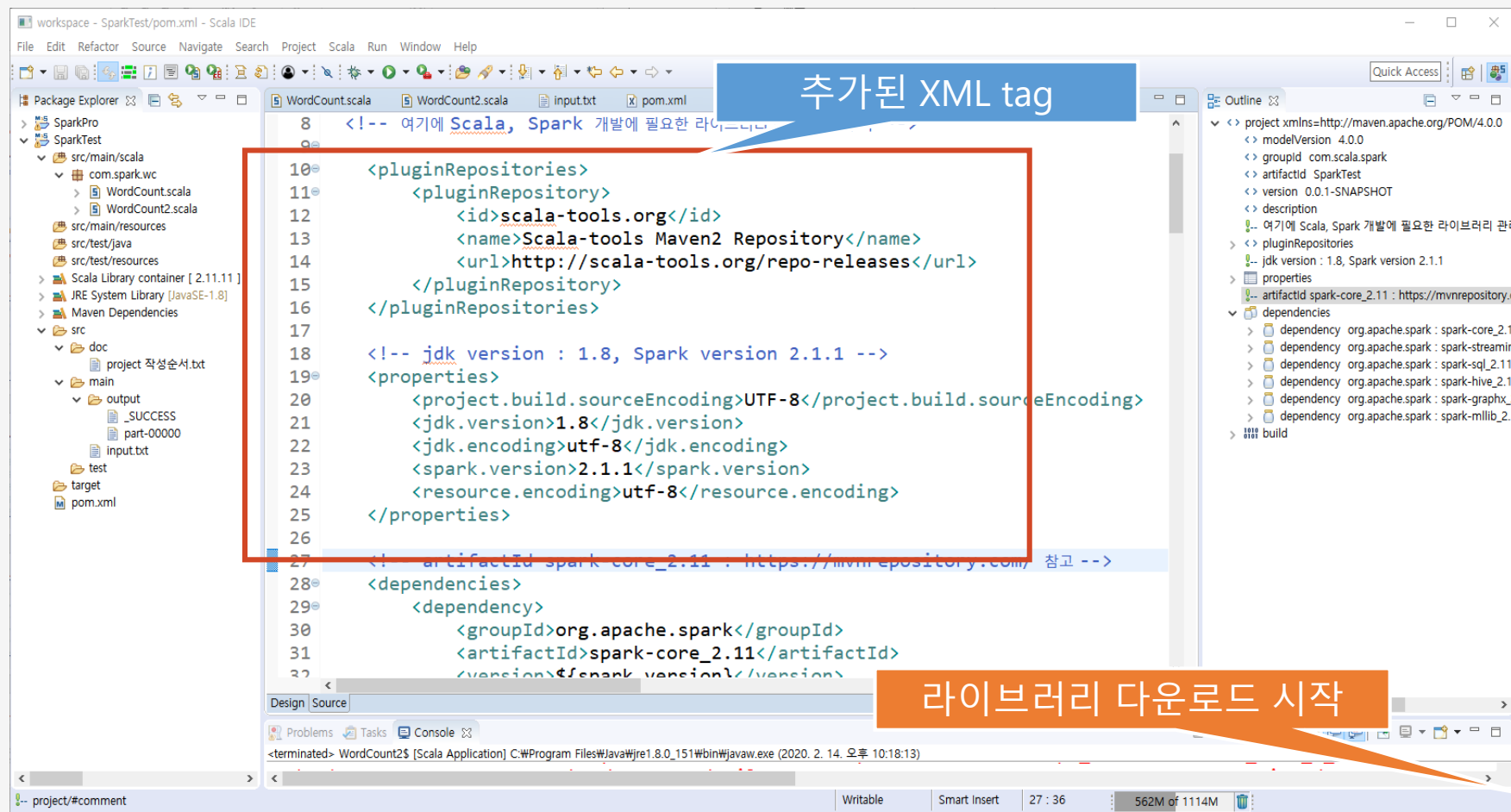
태그를 추가한 후

저장(Save)를 누르면

관련 라이브러리가

다운로드를 시작

Building workspace(100%)
될 때 까지 대기

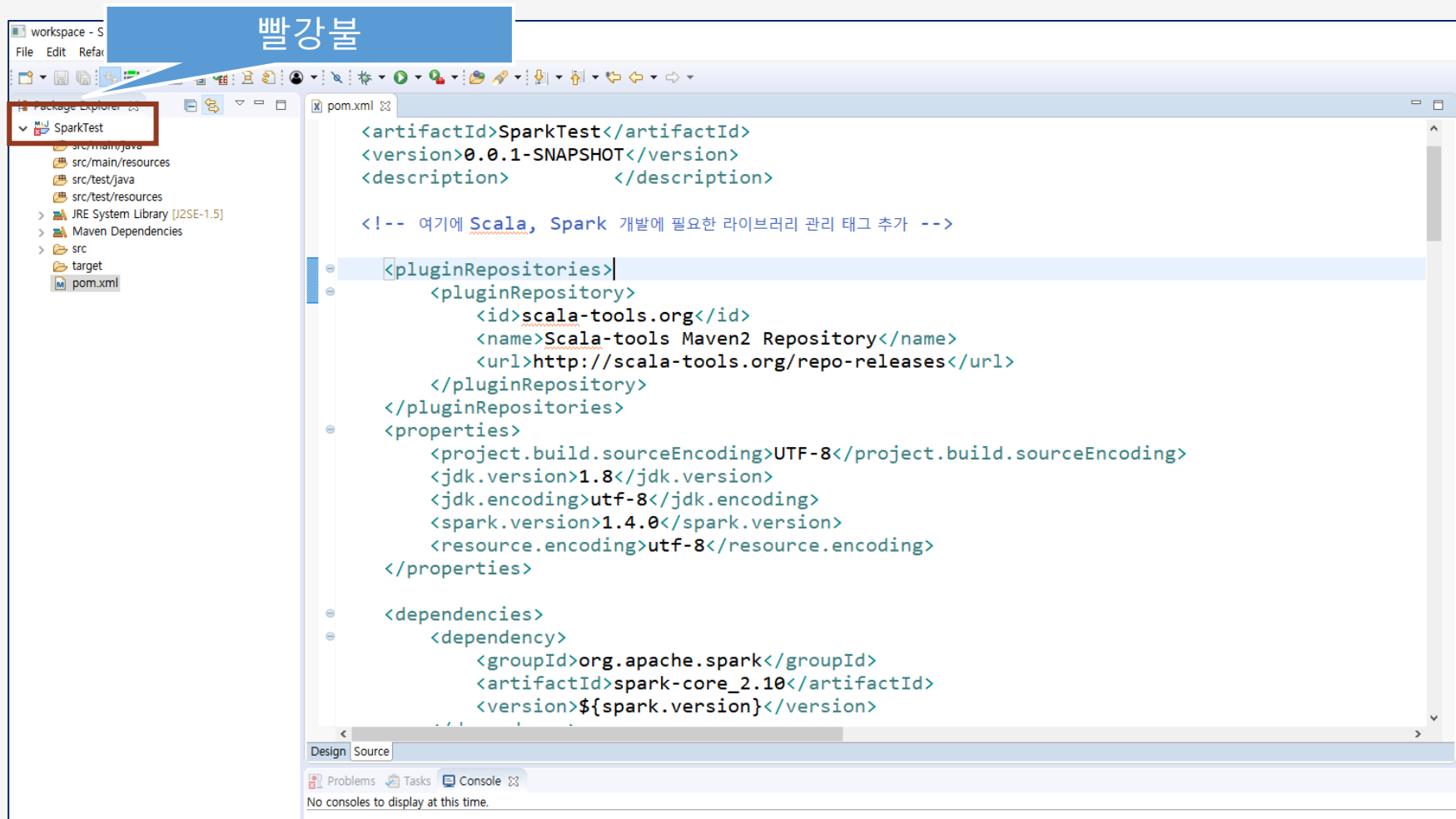


1) pom.xml 설정

[Project] 마우스 오른 버튼

[Maven] -> [Update Project]

메뉴 선택으로 빨강불 제거



1) pom.xml 설정

다운로드 라이브러리 확인

workspace - SparkTest/nmm.xml - Scala IDE

File Edit Source

다운로드 된 Spark 관련 라이브러리

Package Explorer

- IDE System Library [Eclipse 4.5.1]
- Maven Dependencies
 - spark-core_2.11-2.11.jar - C:\W\Users\Wu\Downloads\spark-core_2.11-2.11.jar
 - avro-mapred-1.7.7-hadoop2.jar - C:\W\Users\Wu\Downloads\avro-mapred-1.7.7-hadoop2.jar
 - avro-ipc-1.7.7.jar - C:\W\Users\Wu\Downloads\avro-ipc-1.7.7.jar
 - avro-ipc-1.7.7-tests.jar - C:\W\Users\Wu\Downloads\avro-ipc-1.7.7-tests.jar
 - jackson-core-asl-1.9.13.jar - C:\W\Users\Wu\Downloads\jackson-core-asl-1.9.13.jar
 - chill_2.11-0.8.0.jar - C:\W\Users\Wu\Downloads\chill_2.11-0.8.0.jar
 - kryo-shaded-3.0.3.jar - C:\W\Users\Wu\Downloads\kryo-shaded-3.0.3.jar
 - minlog-1.3.0.jar - C:\W\Users\Wu\Downloads\minlog-1.3.0.jar
 - objenesis-2.1.jar - C:\W\Users\Wu\Downloads\objenesis-2.1.jar
 - chill-java-0.8.0.jar - C:\W\Users\Wu\Downloads\chill-java-0.8.0.jar
 - xbean-asm5-shaded-4.4.jar - C:\W\Users\Wu\Downloads\xbean-asm5-shaded-4.4.jar
 - hadoop-client-2.2.0.jar - C:\W\Users\Wu\Downloads\hadoop-client-2.2.0.jar
 - hadoop-common-2.2.0.jar - C:\W\Users\Wu\Downloads\hadoop-common-2.2.0.jar
 - commons-math-2.1.jar - C:\W\Users\Wu\Downloads\commons-math-2.1.jar
 - xmlenc-0.52.jar - C:\W\Users\Wu\Downloads\xmlenc-0.52.jar
 - commons-configuration-1.6.jar - C:\W\Users\Wu\Downloads\commons-configuration-1.6.jar
 - commons-collections-3.2.1.jar - C:\W\Users\Wu\Downloads\commons-collections-3.2.1.jar
 - commons-digester-1.8.jar - C:\W\Users\Wu\Downloads\commons-digester-1.8.jar
 - commons-beanutils-1.7.0.jar - C:\W\Users\Wu\Downloads\commons-beanutils-1.7.0.jar
 - commons-beanutils-core-1.8.0.jar - C:\W\Users\Wu\Downloads\commons-beanutils-core-1.8.0.jar
 - protobuf-java-2.5.0.jar - C:\W\Users\Wu\Downloads\protobuf-java-2.5.0.jar
 - hadoop-auth-2.2.0.jar - C:\W\Users\Wu\Downloads\hadoop-auth-2.2.0.jar
 - hadoop-hdfs-2.2.0.jar - C:\W\Users\Wu\Downloads\hadoop-hdfs-2.2.0.jar
 - jetty-util-6.1.26.jar - C:\W\Users\Wu\Downloads\jetty-util-6.1.26.jar
 - hadoop-mapreduce-client-app-2.2.0.jar - C:\W\Users\Wu\Downloads\hadoop-mapreduce-client-app-2.2.0.jar
 - hadoop-mapreduce-client-common-2.2.0.jar - C:\W\Users\Wu\Downloads\hadoop-mapreduce-client-common-2.2.0.jar
 - guice-3.0.jar - C:\W\Users\Wu\Downloads\guice-3.0.jar
 - javax.inject-1.jar - C:\W\Users\Wu\Downloads\javax.inject-1.jar
 - aopalliance-1.0.jar - C:\W\Users\Wu\Downloads\aopalliance-1.0.jar
 - hadoop-yarn-server-common-2.2.0.jar - C:\W\Users\Wu\Downloads\hadoop-yarn-server-common-2.2.0.jar
 - hadoop-mapreduce-client-shuffle-2.2.0.jar - C:\W\Users\Wu\Downloads\hadoop-mapreduce-client-shuffle-2.2.0.jar
 - hadoop-yarn-api-2.2.0.jar - C:\W\Users\Wu\Downloads\hadoop-yarn-api-2.2.0.jar
 - hadoop-mapreduce-client-core-2.2.0.jar - C:\W\Users\Wu\Downloads\hadoop-mapreduce-client-core-2.2.0.jar
 - hadoop-yarn-common-2.2.0.jar - C:\W\Users\Wu\Downloads\hadoop-yarn-common-2.2.0.jar
 - hadoop-mapreduce-client-jobclient-2.2.0.jar - C:\W\Users\Wu\Downloads\hadoop-mapreduce-client-jobclient-2.2.0.jar

WordCount.scala

```
32         <version>${spark.version}</version>
33     </dependency>
34     <dependency>
35         <groupId>org.apache.spark</groupId>
36         <artifactId>spark-streaming_2.11</artifactId>
37         <version>${spark.version}</version>
38     </dependency>
39
40     <dependency>
41         <groupId>org.apache.spark</groupId>
42         <artifactId>spark-sql_2.11</artifactId>
43         <version>${spark.version}</version>
44     </dependency>
45     <dependency>
46         <groupId>org.apache.spark</groupId>
47         <artifactId>spark-hive_2.11</artifactId>
48         <version>${spark.version}</version>
49         <scope>provided</scope>
50     </dependency>
51     <dependency>
52         <groupId>org.apache.spark</groupId>
53         <artifactId>spark-graphx_2.11</artifactId>
54         <version>${spark.version}</version>
55     </dependency>
56 </dependencies>
```

Outline

- project xmlns=http://maven.apache.org/POM/4.0.0
 - modelVersion 4.0.0
 - groupId com.scala.spark
 - artifactId SparkTest
 - version 0.0.1-SNAPSHOT
 - description
 - 여기에 Scala, Spark 개발에 필요한 라이브러리 관리
 - pluginRepositories
 - jdk version : 1.8, Spark version 2.1.1
 - properties
 - dependencies
 - artifactId spark-core_2.11 : https://mvnrepository.com/artifact/org.apache.spark/spark-core_2.11
 - dependency org.apache.spark : spark-core_2.11
 - dependency org.apache.spark : spark-streaming_2.11
 - dependency org.apache.spark : spark-sql_2.11
 - dependency org.apache.spark : spark-hive_2.11
 - dependency org.apache.spark : spark-graphx_2.11
 - dependency org.apache.spark : spark-mllib_2.11
 - build

Design | Source

Problems Tasks Console

<terminated> WordCount2\$ [Scala Application] C:\Program Files\Java\jre1.8.0_151\bin\javaw.exe (2020. 2. 14. 오후 10:18:13)

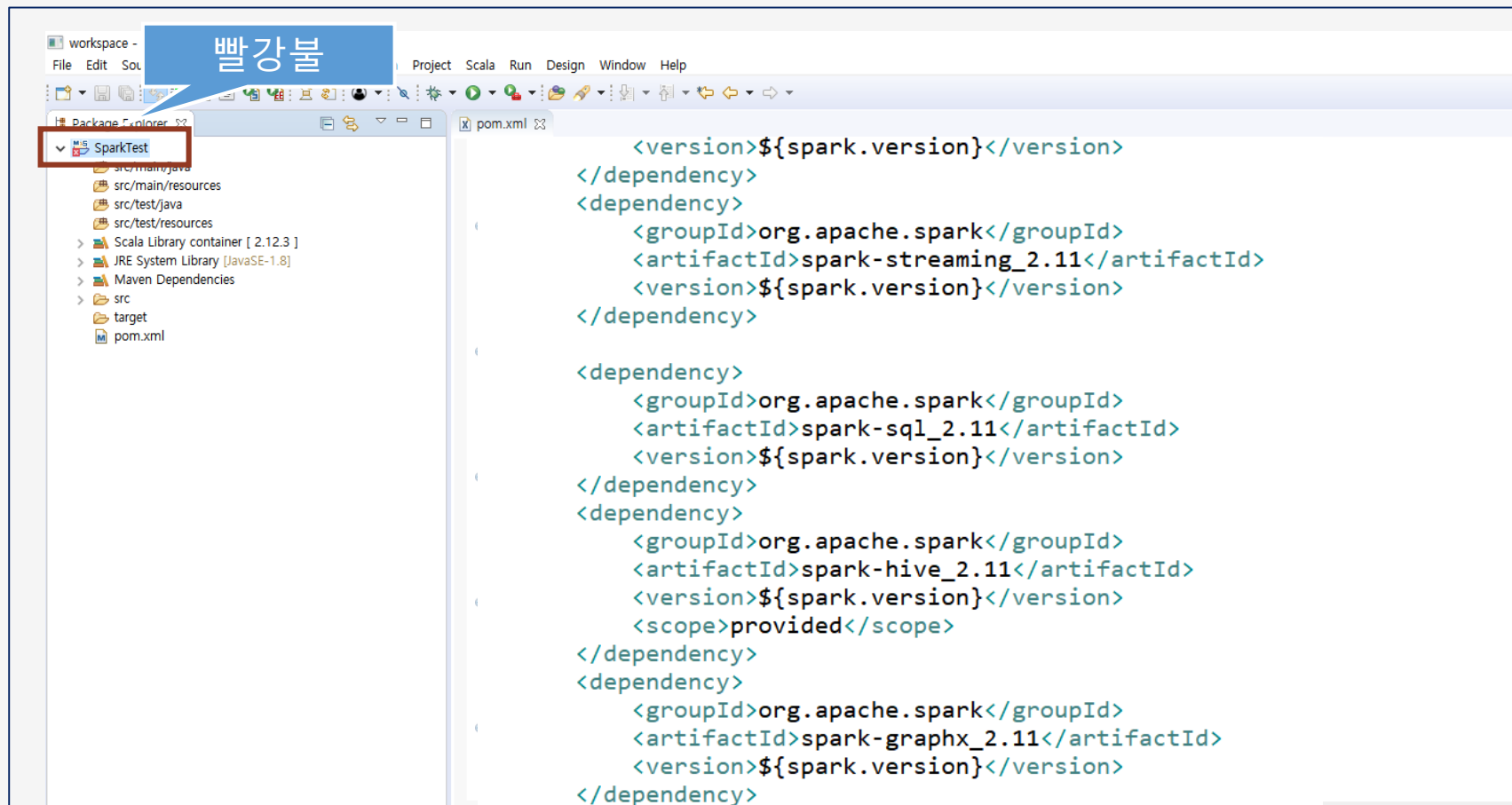
585M of 1114M

2) Scala Nature 추가 : Scala 환경으로 수정

Project에서 오른쪽 마우스 버튼

Configure -> Add Scala Nature

메뉴 선택 -> Scala Nature를 추가

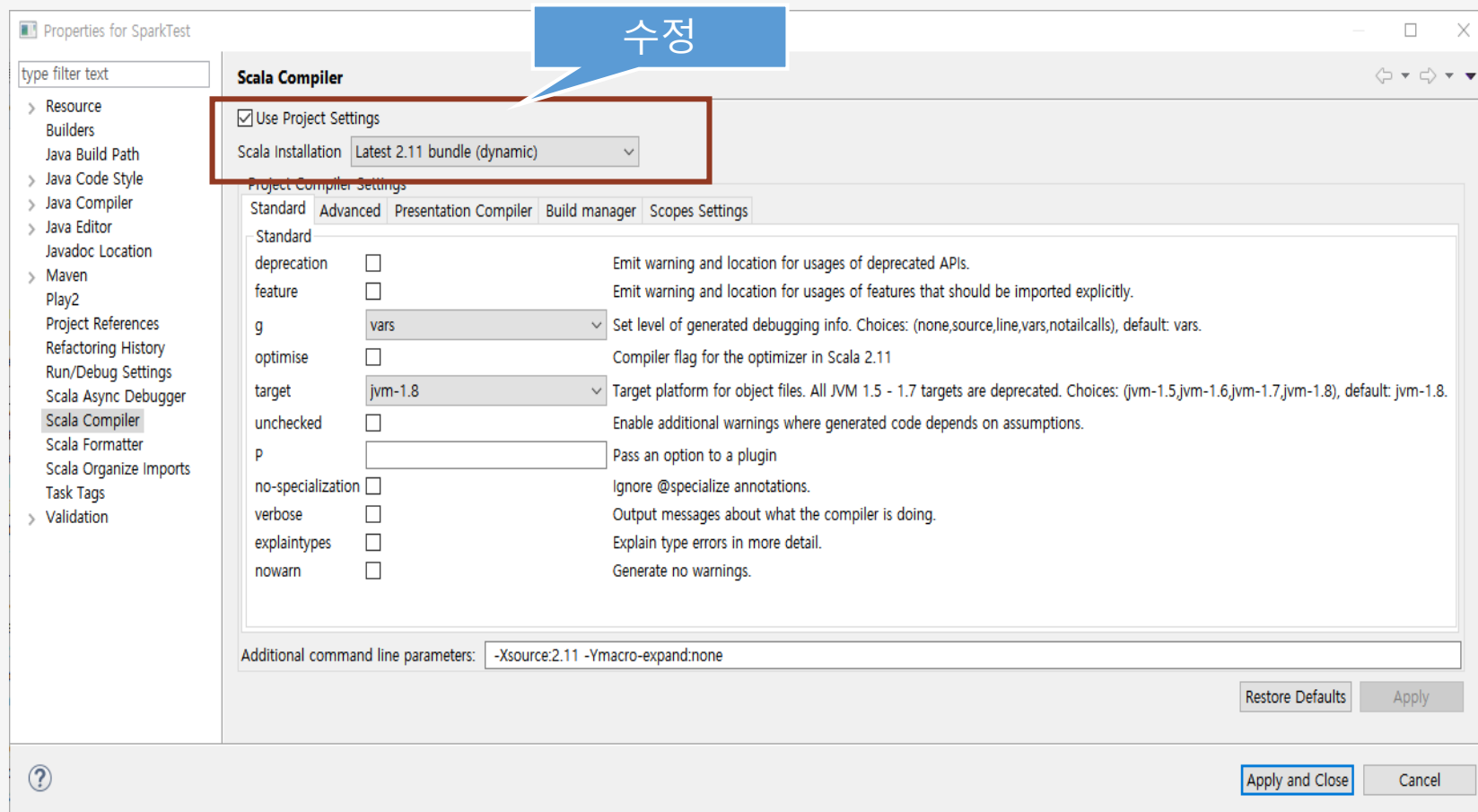


2) Scala Nature 추가

Project에서 오른쪽 마우스 버튼

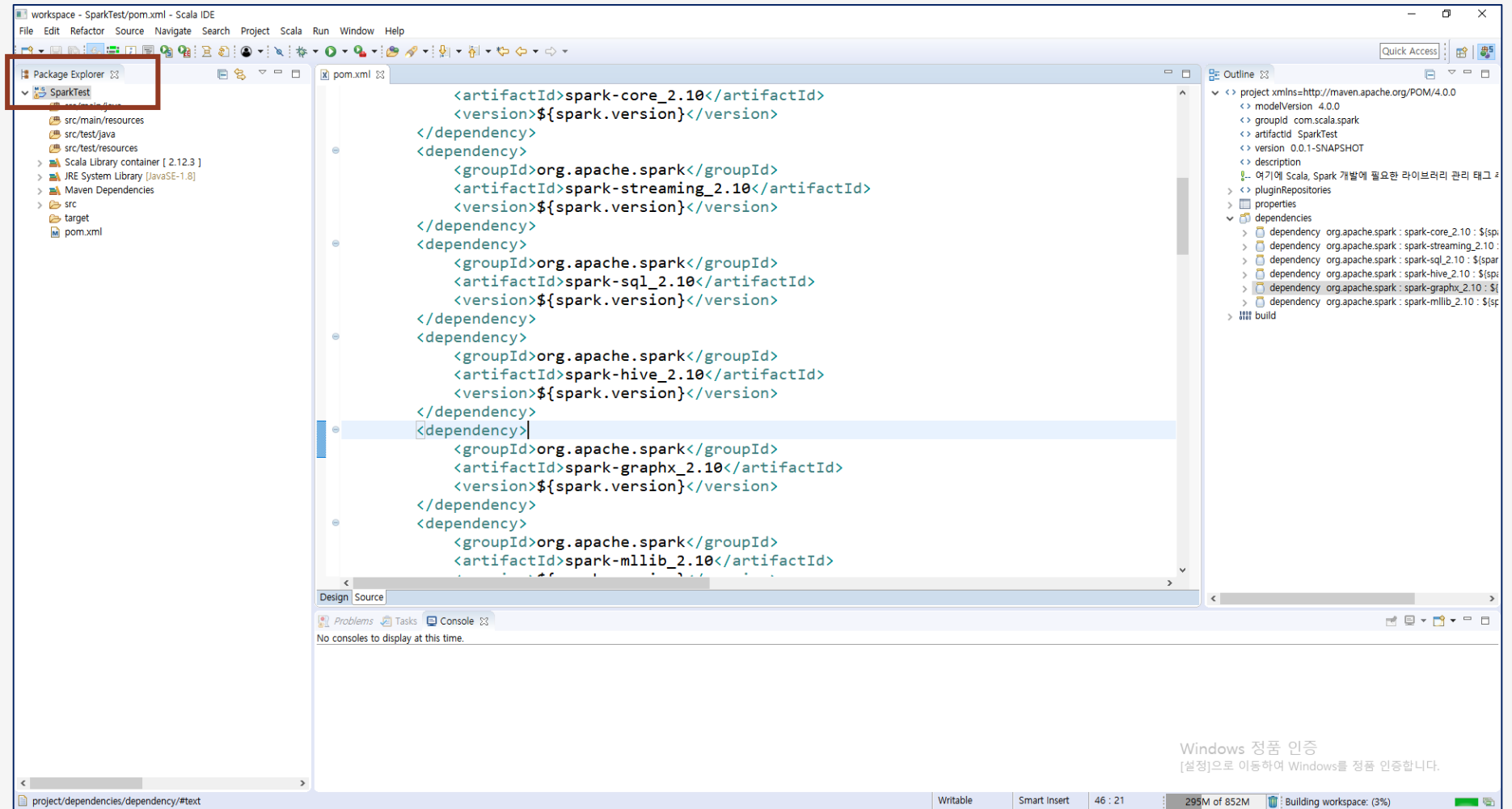
[Properties] -> [Scala compiler]
에서 Latest 2.11 bundle
(dynamic) 선택

Scala IDE는 기본적으로 Latest
2.12 bundle (dynamic)을 사용하고
있으나 안정화된 Scala 2.11을
사용하기 위해서 Compiler를
변경한다.



2) Scala Nature 추가

빨강불 제거 확인

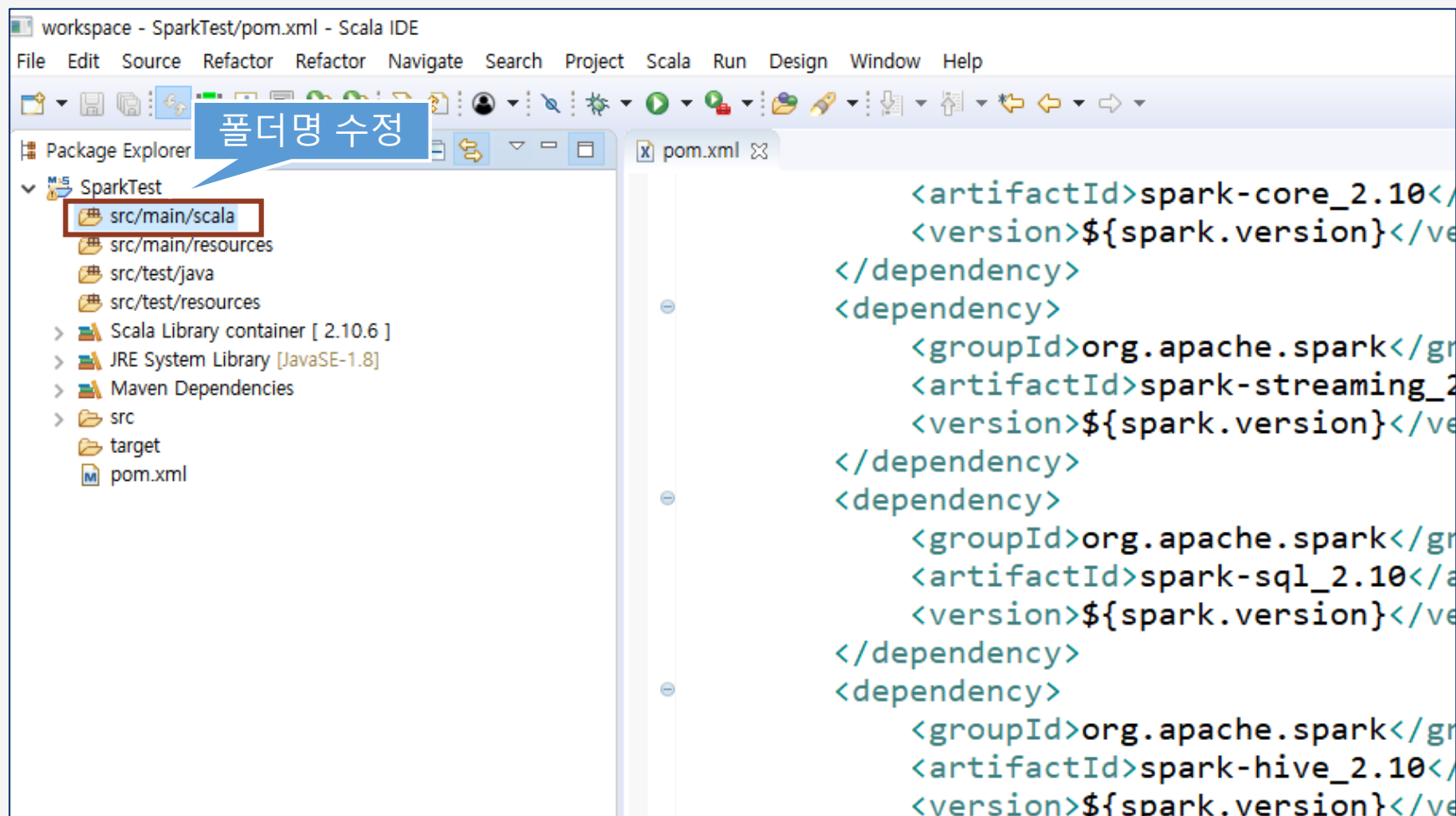


3) java -> scala 수정

폴더에서 오른쪽 마우스 버튼 클릭

-> Refactor -> Rename 선택

scala로 변경



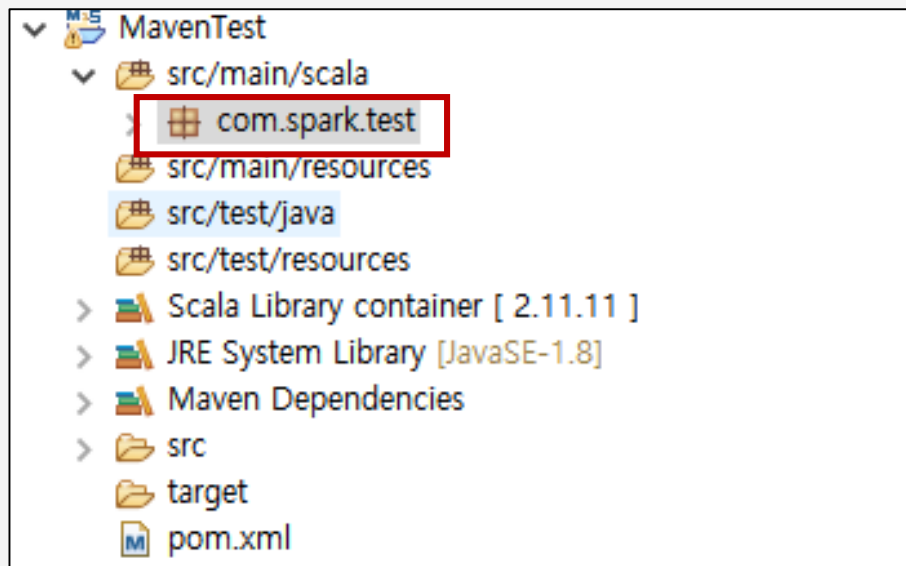
3. Spark 애플리케이션 작성

1) 패키지 만들기

scala 폴더에서 오른쪽 마우스 버튼 클릭

-> New -> Package 선택

-> com.spark.test 패키지 만들기

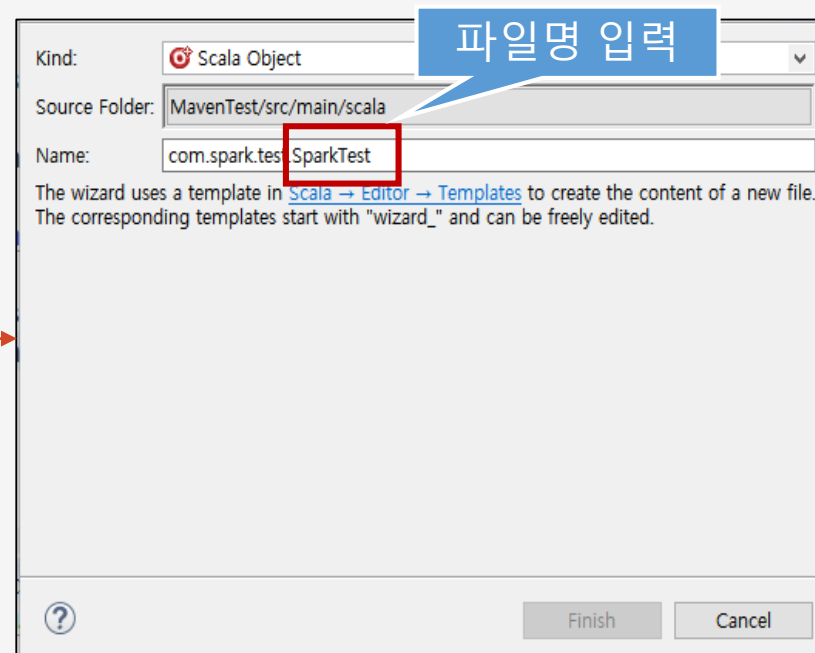
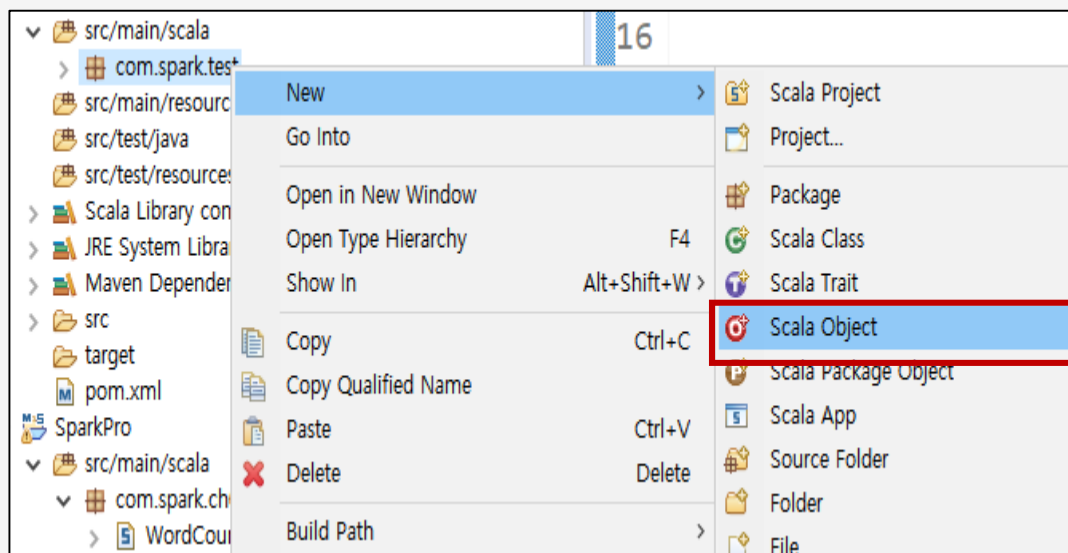


2) Scala Object 만들기

com.spark.wc 패키지에서 오른쪽 마우스 버튼 클릭

-> New -> Scala Object 선택

-> SparkTest 파일명 입력



3) SparkTest 어플리케이션 작성

```
package com.spark.test

// Maven에서 제공하는 library 추가
import org.apache.spark.SparkConf
import org.apache.spark.SparkContext

object SparkTest {
  def main(args: Array[String]) = {

    // 1. SparkContext object 생성
    val conf = new SparkConf()
      .setAppName("SparkTest")
      .setMaster("local") // Spark 환경 객체

    val sc = new SparkContext(conf) // 분산 파일 읽기

    // 2. Local or Hadoop file 읽기
    val data = sc.textFile("file:/C:/hadoop-2.6.0/README.txt")

    // 3. 줄단위 출력
    data.foreach(println)
    sc.stop // 객체 닫기
  }
}
```

For the latest information about Hadoop, please visit our website at:

<http://hadoop.apache.org/core/>

and our wiki, at:

<http://wiki.apache.org/hadoop/>

This distribution includes cryptographic software. The country in which you currently reside may have restrictions on the import, possession, use, and/or re-export to another country, of encryption software. BEFORE using any encryption software, please check your country's laws, regulations and policies concerning the import, possession, or use, and re-export of encryption software, to see if this is permitted. See <<http://www.wassenaar.org/>> for more information.

The U.S. Government Department of Commerce, Bureau of Industry and Security (BIS), has classified this software as Export Commodity Control Number (ECCN) 5D002.C.1, which includes information security software using or performing cryptographic functions with asymmetric algorithms. The form and manner of this Apache Software Foundation distribution makes it eligible for export under the License Exception ENC Technology Software Unrestricted (TSU) exception (see the BIS Export Administration Regulations, Section 740.13) for both object code and source code.

The following provides more details on the included cryptographic software:

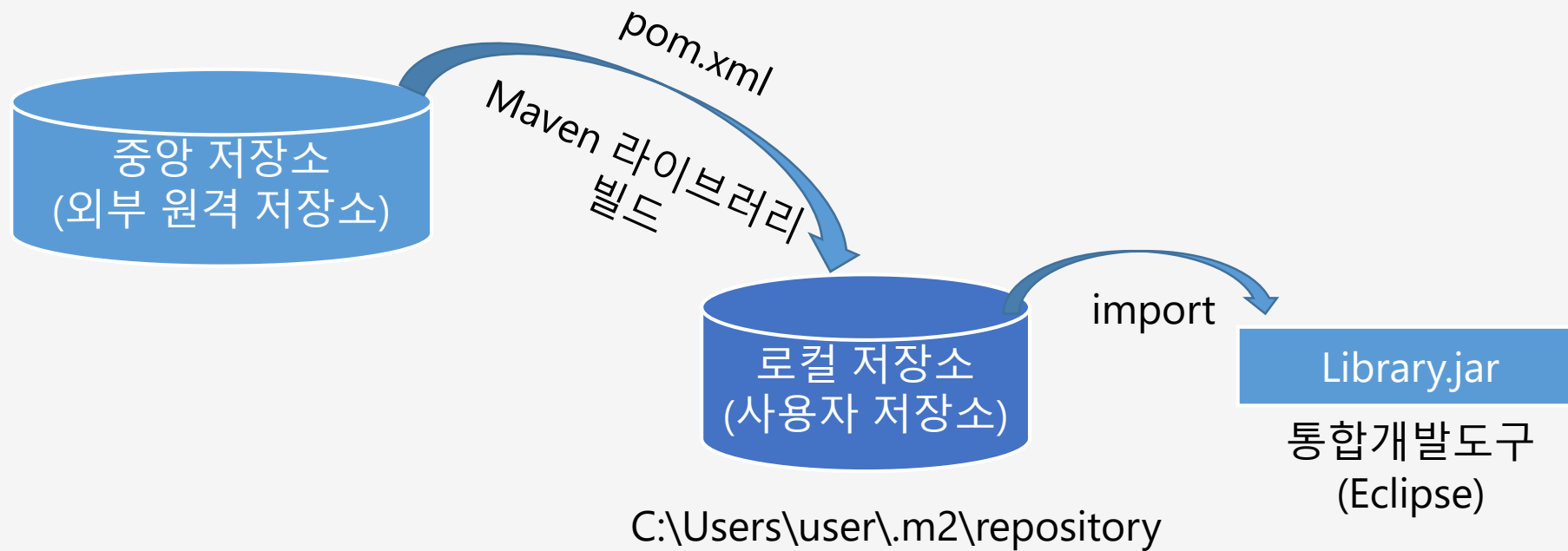
Hadoop Core uses the SSL libraries from the Jetty project written by mortbay.org.

4. Maven 기반 라이브러리 자동 관리

The screenshot displays an IDE interface with two main panels. On the left, the 'Project Explorer' shows a project named 'chap02_DI' with a 'Maven Dependencies' folder highlighted. This folder contains a list of downloaded JAR files, including 'spring-context-3.2.3.RELEASE.jar', 'spring-aop-3.2.3.RELEASE.jar', 'spring-beans-3.2.3.RELEASE.jar', 'spring-core-3.2.3.RELEASE.jar', 'commons-logging-1.1.1.jar', 'spring-expression-3.2.3.RELEASE.jar', 'spring-tx-3.2.3.RELEASE.jar', 'aopalliance-1.0.jar', 'slf4j-api-1.7.5.jar', 'logback-classic-1.0.13.jar', 'logback-core-1.0.13.jar', 'hibernate-entitymanager-4.2.1.Final.jar', 'jboss-logging-3.1.0.GA.jar', 'hibernate-core-4.2.1.Final.jar', 'antlr-2.7.7.jar', 'dom4j-1.6.1.jar', 'jboss-transaction-api_1.1_spec-1.0.1.Final.jar', 'hibernate-jpa-2.0-api-1.0.1.Final.jar', 'javassist-3.15.0-GA.jar', 'hibernate-commons-annotations-4.0.1.Final.jar', 'spring-test-3.2.3.RELEASE.jar', 'junit-4.11.jar', and 'hamcrest-core-1.3.jar'. A red box highlights the 'pom.xml' file in the 'target' folder. On the right, the 'Main.java' file is open, showing the contents of the 'pom.xml' file. The XML includes project information, properties for Java version, source encoding, and Spring/Hibernate versions, and a list of dependencies. A blue cloud-shaped callout with the text '라이브러리 다운로드/버전관리' (Library Download/Version Management) points to the 'pom.xml' file. A red arrow points from the 'pom.xml' file in the Project Explorer to the 'pom.xml' file in the Main.java editor.

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001
2   <modelVersion>4.0.0</modelVersion>
3   <groupId>org.springframework.samples</groupId>
4   <artifactId>chap02_DI</artifactId>
5   <version>0.0.1-SNAPSHOT</version>
6
7   <properties>
8
9       <!-- Generic properties -->
10      <java.version>1.6</java.version>
11      <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
12      <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>
13
14      <!-- Spring -->
15      <spring.framework.version>3.2.3.RELEASE</spring.framework.version>
16
17      <!-- Hibernate / JPA -->
18      <hibernate.version>4.2.1.Final</hibernate.version>
19
20      <!-- Logging -->
21      <logback.version>1.0.13</logback.version>
22      <slf4j.version>1.7.5</slf4j.version>
23
24      <!-- Test -->
25      <junit.version>4.11</junit.version>
26
27  </properties>
28
29  <dependencies>
30      <!-- Spring and Transactions -->
```


Maven 저장소



Maven 저장소 설정

- Maven 저장소에 대한 설정
 - pom.xml 의 <repositories /> 에서 설정
 - 최상위 POM에는 중앙 저장소의 정보가 설정되어 있으므로, 중앙 저장소에서 제공하지 않는 라이브러리가 있다면 <repositories /> 내에 설정
 - Maven 라이브러리를 다운로드할 때 <repositories /> 에 설정되어 있는 저장소 순서로 진행

Maven 저장소 설정

Maven 라이브러리 관리

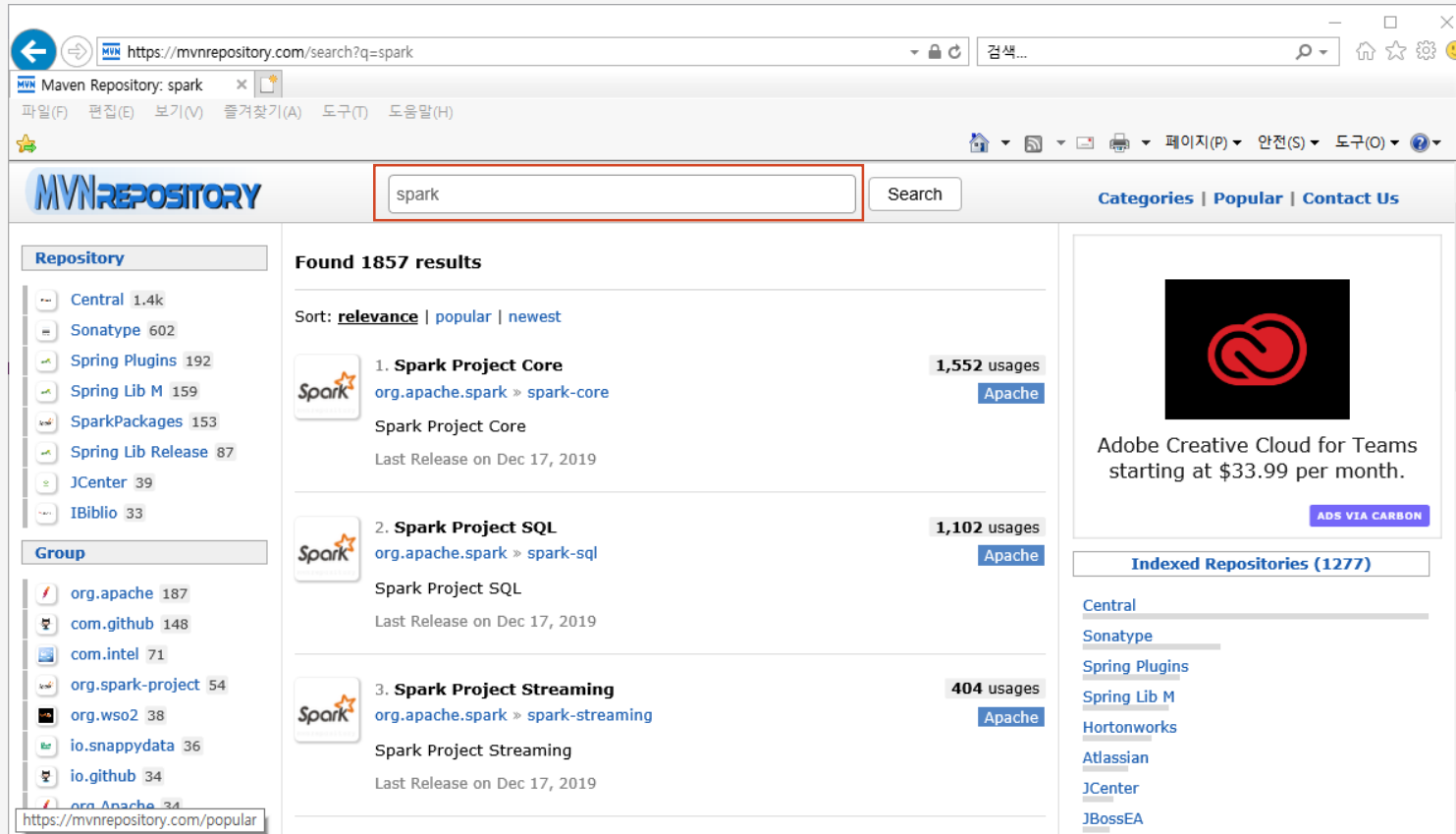
- pom.xml 의 <dependencies /> 설정을 통해서 필요한 라이브러리 로컬 저장소 다운로드

```
<dependencies>
  <dependency>
    <groupId>org.apache.spark</groupId>
    <artifactId>spark-core_2.11</artifactId>
    <version>${spark.version}</version>
  </dependency>
  <dependency>
    <groupId>org.apache.spark</groupId>
    <artifactId>spark-mllib_2.11</artifactId>
    <version>${spark.version}</version>
    <scope>runtime</scope>
  </dependency>
</dependencies>
```

pom.xml에 라이브러리 등록

● Spark 라이브러리 등록 예

단계1: <http://mvnrepository.com> -> spark 검색어 입력



pom.xml에 라이브러리 등록

단계2: spark-mllib 선택

The screenshot shows the Maven Repository website for the Apache Spark group. The page is titled "Group: Apache Spark" and lists three artifacts. The third artifact, "Spark Project ML Library", is highlighted with a red box. The page also includes a sidebar with "Popular Categories" and a list of "Indexed Repositories".

Indexed Artifacts (17.2M)

Projects (millions) vs Year graph showing an upward trend from 2006 to 2018.

Popular Categories

- Aspect Oriented
- Actor Frameworks
- Application Metrics
- Build Tools
- Bytecode Libraries
- Command Line Parsers
- Cache Implementations
- Cloud Computing
- Code Analyzers
- Collections
- Configuration Libraries
- Core Utilities

Group: Apache Spark

Sort: popular | newest

Artifact	Usage	Apache
1. Spark Project Core org.apache.spark » spark-core Spark Project Core Last Release on Dec 17, 2019	1,552 usages	Apache
2. Spark Project SQL org.apache.spark » spark-sql Spark Project SQL Last Release on Dec 17, 2019	1,102 usages	Apache
3. Spark Project ML Library org.apache.spark » spark-mllib Spark Project ML Library Last Release on Dec 17, 2019	454 usages	Apache

Indexed Repositories (1277)

- Central
- Sonatype
- Spring Plugins
- Spring Lib M
- Hortonworks
- Atlassian
- JCenter
- JBossEA

pom.xml에 라이브러리 등록

단계3: groupId, artifactId, version 정보 등 확인

The screenshot shows the Maven Repository website for the artifact `org.apache.spark/spark-mllib_2.12/2.4.5`. The page layout includes a sidebar on the left with categories like Bytecode Libraries, Command Line Parsers, etc. The main content area has a 'Scala Target' dropdown set to 'Scala 2.12'. A yellow box highlights a 'Note: There is a new version for this artifact' with a 'New Version' button and the text '3.0.0-preview2'. Below this, there are tabs for different build systems: Maven, Gradle, SBT, Ivy, Grape, Leiningen, and Buildr. The 'Maven' tab is active, showing a code block with the following XML snippet:

```
<!-- https://mvnrepository.com/artifact/org.apache.spark/spark-mllib -->
<dependency>
  <groupId>org.apache.spark</groupId>
  <artifactId>spark-mllib_2.12</artifactId>
  <version>2.4.5</version>
  <scope>runtime</scope>
</dependency>
```

Below the code block, there is a checkbox labeled 'Include comment with link to declaration' which is checked. At the bottom, a red banner says 'Copied to clipboard!'. On the right side of the page, there is an advertisement for Microsoft 365 with the text 'Try now' and 'FREE FOR SMALL TEAMS FOREVER'.

pom.xml에 라이브러리 등록

단계4: pom.xml에 복사 & 붙여넣기

