

애플리케이션 빌드

빌드

1. 빌드 도구 : Ant, Maven, Gradle

- ✓ 초기의 JAVA 빌드 도구로 Ant라는 도구를 많이 사용하였으나 늘어나는 라이브러리를 관리하기 위해 Maven, Gradle등의 기존 Ant를 보완한 빌드 도구들도 생기게 되었다

1) Apache ANT

- ✓ Ant는 Java 기반의 빌드 도구로 다른 빌드 도구보다 역사가 오래되었다. Ant는 개발자가 원하는 것을 개발할 수 있다는 유연성에 큰 장점이 있다.

✓ [ANT의 특징]

- 각 프로젝트에 대한 XML기반 빌드 스크립트 개발
- 형식적인 규칙이 없음 : 결과물을 넣을 위치를 정확히 알려줘야 하며, 프로젝트에 특화된 Target과 Dependency를 이용해 모델링
- 절차적 : 명확한 빌드 절차 정의가 필요
- 생명주기를 갖지 않기 때문에 각각의 target에 대한 의존관계와 일련의 작업을 정의해 주어야 함

✓ [ANT의 단점]

- 유연성이 높으나 프로젝트가 복잡해질 경우 각각의 Build과정을 이해하기 어려움
- XML, Remote Repository를 가져올 수 없었음 (IVY 도입)
- 스크립트의 재사용이 어려움

빌드

1. 빌드 도구 : Ant, Maven, Gradle

2) Apache Maven

- ✓ Maven은 프로젝트에 필요한 모든 'Dependency (종속성)'를 리스트의 형태로 Maven에게 알려 관리 할 수 있도록 돕는 방식을 말한다.
Dependency를 관리하고, 표준화된 프로젝트(Standardized project)를 제공
 - ✓ XML, remote repository를 가져 올 수 있음 : 개발에 필요한 종속되는 'jar', 'class path'를 다운로드 할 필요 없이 선언만으로 사용 가능
 - ✓ 상속형 : 하위 XML이 필요 없는 속성도 모두 표기
 - ✓ 즉, 'POM.xml' 이라는 Maven 파일에 필요한 'Jar', 'Class Path'를 선언해 주면 직접 다운로드 할 필요 없이 Maven은 Repository에서 필요한 모든 파일들을 해당 프로젝트로 불러와 준다. 이러한 장점에도 불구하고, Maven은 몇가지 단점이 있는데 그것은 바로 아래와 같다.
-
- 라이브러리가 서로 종속할 경우 XML이 복잡해짐
 - 계층적인 데이터를 표현하기에 좋지만, 플로우나 조건부 상황을 표현하기엔 어려움
 - 편리하나 맞춤화된 로직 실행이 어려움

빌드

1. 빌드 도구 : Ant, Maven, Gradle

3) Apache Gradle

- ✓ 최근 소프트웨어개발 범위의 변화에 따라 빌드의 자동화에 대한 요구도 증가하게 되었다. [Gradle](#)은 JVM 기반의 빌드 도구로 기존의 Ant와 Maven을 보완하였다. 따라서 JAVA 혹은 Groovy를 이용해 logic을 개발자의 의도에 따라 설계할 수 있다.

오픈소스기반의 build 자동화 시스템으로 Groovy 기반 DSL(Domain-Specific Language)로 작성

- ✓ Build-by-convention을 바탕으로함: 스크립트 규모가 작고 읽기 쉬움
- ✓ Multi 프로젝트의 빌드를 지원하기 위해 설계됨
- ✓ 설정 주입 방식 (Configuration Injection)
- ✓ 따라서 초기 프로젝트 설정에 드는 시간을 절약할 수 있으며 기존의 Maven이나 Ivy등과 같은 빌드 도구들과도 호환이 가능하다는 점이다.

빌드

2. 이클립스-ant 사용하기

- ✓ 이클립스에서 자바 코딩을 하다보면 bin폴더에 class파일이 제대로 생성되지 않을 때가 있다.
- ✓ 프로젝트를 처음 가져올때는 class파일을 만들어 주는데 그 이후에 이클립스가 아닌 다른 경로로 파일을 가져오거나 class파일을 지우면 이클립스 build를 못하죠.
- ✓ 그 외에도 프로젝트 진행하다 보면 jar파일을 만들거나 jar파일에 서명을 하는 등의 작업을 해야할 때가 있는데 이걸 수작업으로 계속 할 순 없다.
- ✓ 이때 ant를 사용합니다.
- ✓ ant는 한마디로 리눅스의 make 파일이랑 같다고 보면 된다.
- ✓ build 못 할 때 build도 해서 class파일도 만들어 줍니다. jar파일도 만들어주고 서명도 해줍니다

2. ant 설치

<http://ant.apache.org/bindownload.cgi>

←

→

↺

http://ant.apache.org/bindownload.cgi


PIP... [Sp... 06... NC... PIP... 만... KISA 기... B [프... [An... N 이... 이... 달... A. X

Thanks

Sponsorship

◆ Project Management

- Contributors
- Apache Ant Mission
- Project Bylaws
- Legal
- Processes
- Archive



APACHECON
North America
Montréal, Canada
24-27 September, 2018
Register Now!

Mirror

You are currently using **http://apache.tt.co.kr/**. If you encounter a problem with this mirror, please select another mirror. If all mirrors are failing, there are *backup* mirrors (at the end of the mirror list) that should be available.

Other mirrors: Change

Current Release of Ant

The Apache Ant team currently maintains two lines of development. The 1.9.x releases require Java 7 runtime and 1.10.x requires Java8 at runtime. Both lines are based off of Ant 1.9.7 and the 1.9.x releases are mostly bug fix releases while additional new features are developed for 1.10.x. We recommend using 1.10.x unless you are required to use versions of Java prior to Java8 during the build process.

Currently, Apache Ant 1.9.13 and 1.10.5 are the best available versions, see the [release notes](#).

Note

Ant 1.9.13 and 1.10.5 have been released on 13-Jul-2018 and may not be available on all mirrors for a few days.

Tar files may require gnu tar to extract

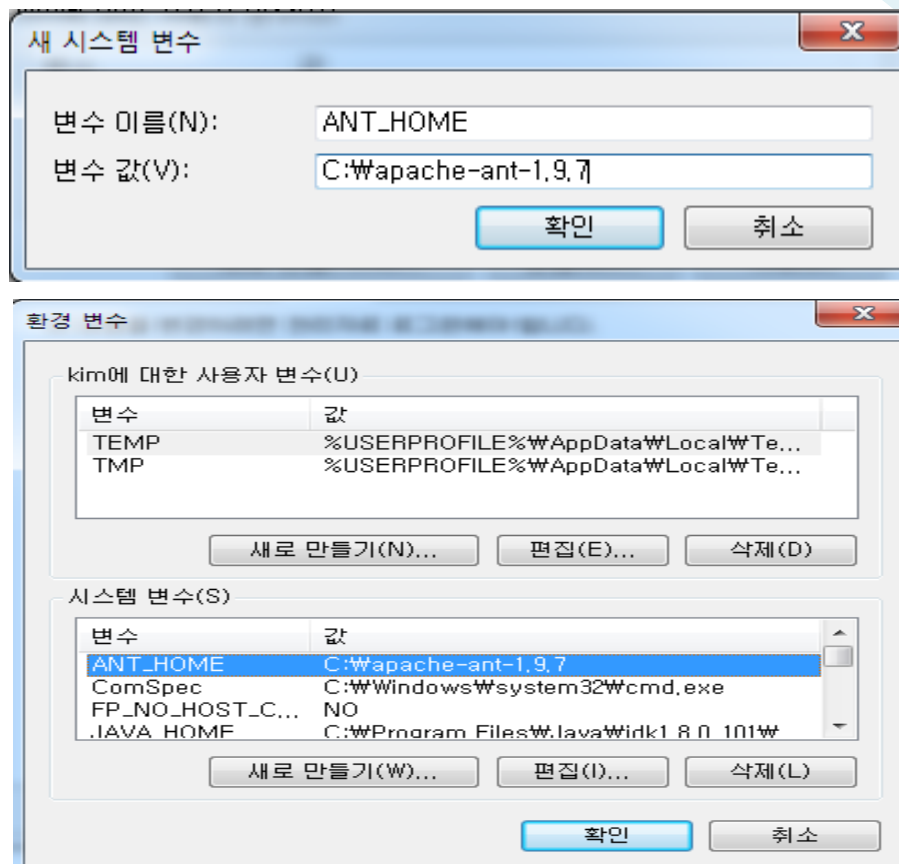
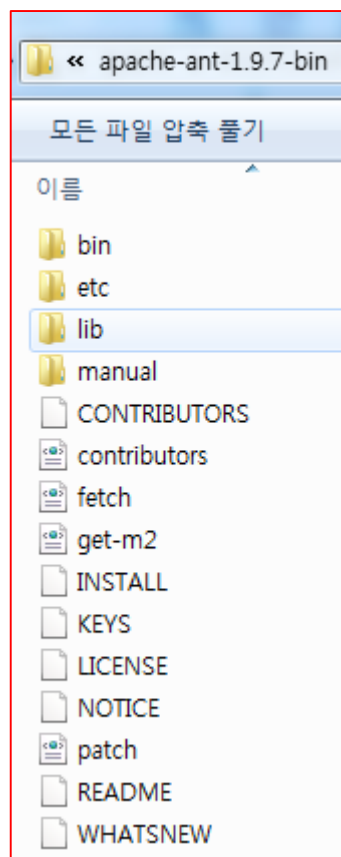
Tar files in the distribution contain long file names, and may require gnu tar to do the extraction.

- 1.10.5 .zip archive: [apache-ant-1.10.5-bin.zip](#) [PGP] [SHA1] [SHA512]
- 1.9.13 .zip archive: [apache-ant-1.9.13-bin.zip](#) [PGP] [SHA1] [SHA512]
- 1.10.5 .tar.gz archive: [apache-ant-1.10.5-bin.tar.gz](#) [PGP] [SHA1] [SHA512]
- 1.9.13 .tar.gz archive: [apache-ant-1.9.13-bin.tar.gz](#) [PGP] [SHA1] [SHA512]
- 1.10.5 .tar.bz2 archive: [apache-ant-1.10.5-bin.tar.bz2](#) [PGP] [SHA1] [SHA512]
- 1.9.13 .tar.bz2 archive: [apache-ant-1.9.13-bin.tar.bz2](#) [PGP] [SHA1] [SHA512]

빌드

2. ant 설치

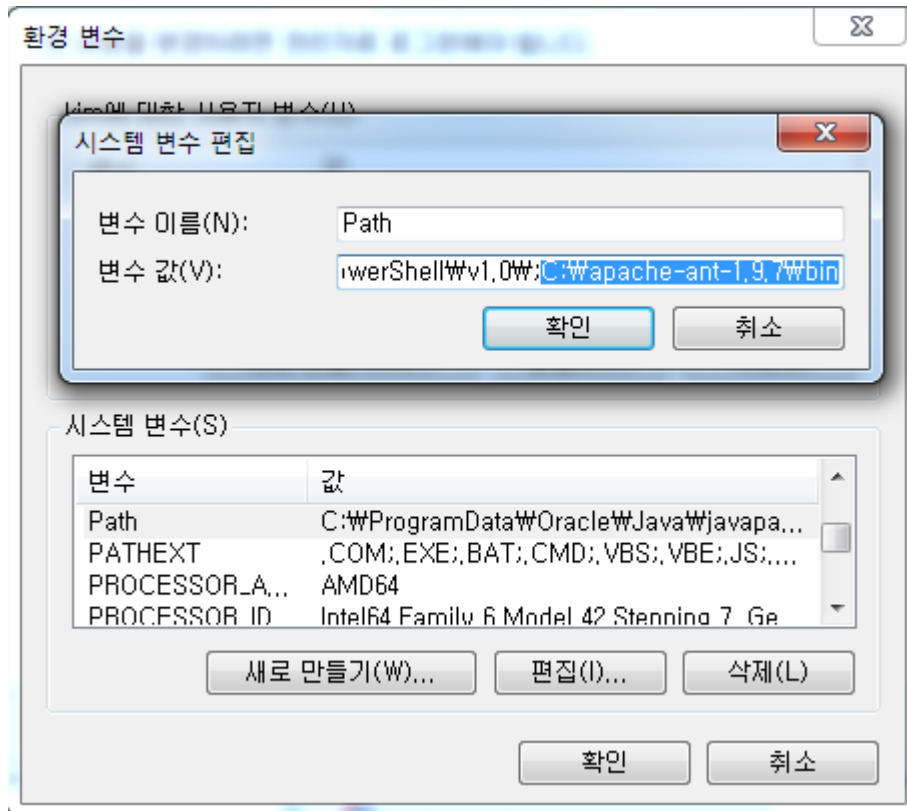
- ❖ Ant의 설치를 진행한다.
- ✓ 다운받기가 완료되면 압축을 해제하고 환경변수를 설정한다.



빌드

2. ant 설치

- ❖ Ant의 설치를 진행한다.

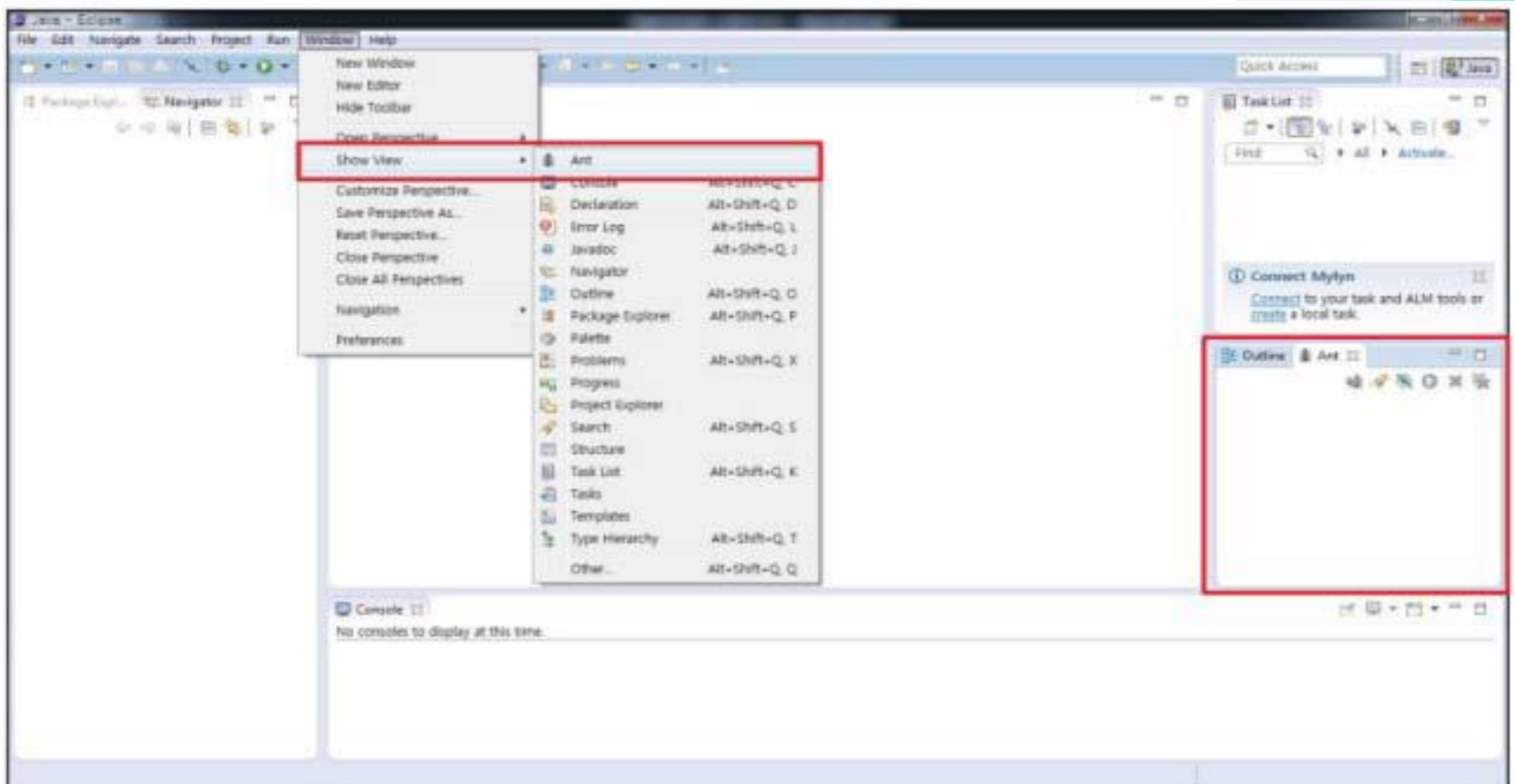


빌드

2. ant 설치

. Ant의 설치 여부를 확인한다.

- ✓ Eclipse를 실행하여 Windows -> Show View -> Ant를 선택하여 Ant 윈도우즈가 생성되는 것을 확인한다.

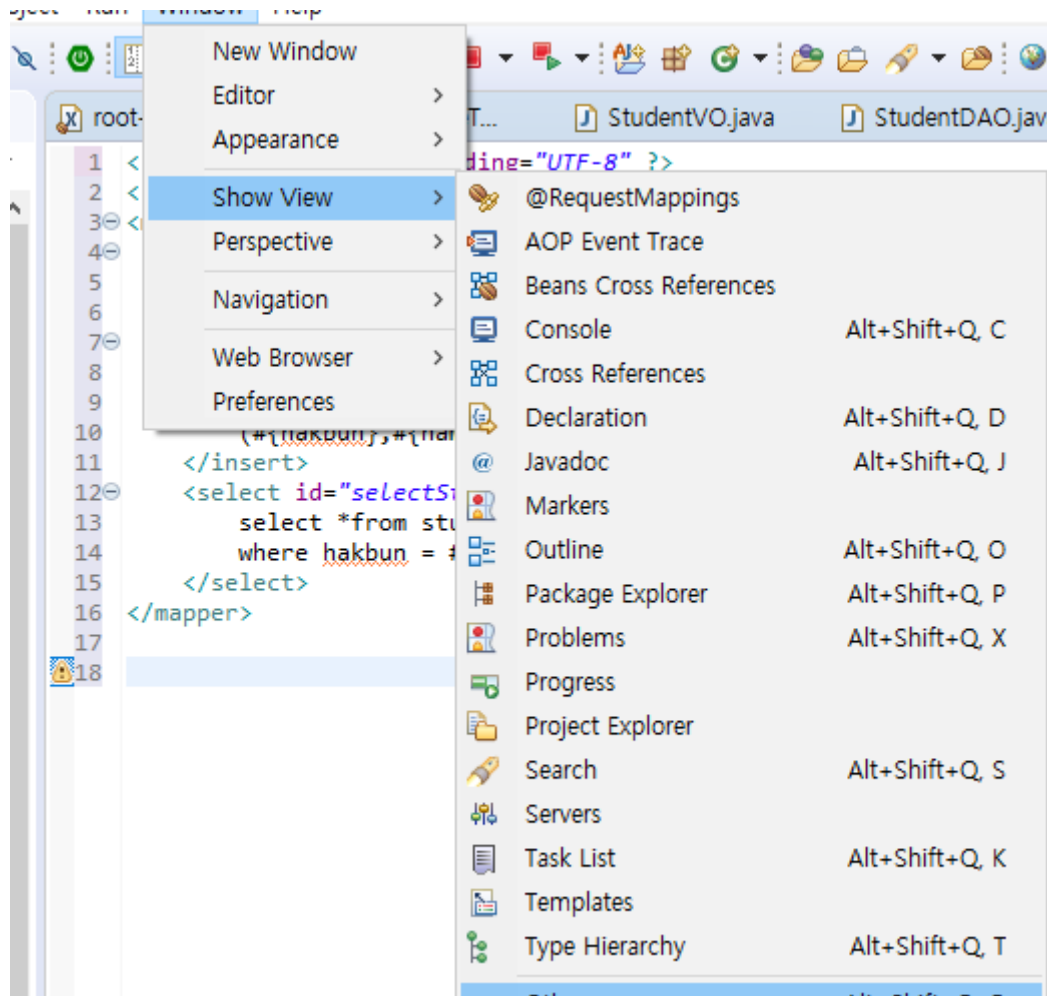


빌드

2. 이클립스-ant 사용하기

✓ ant파일 불러오기

메뉴-Window-Show View-Other를 선택

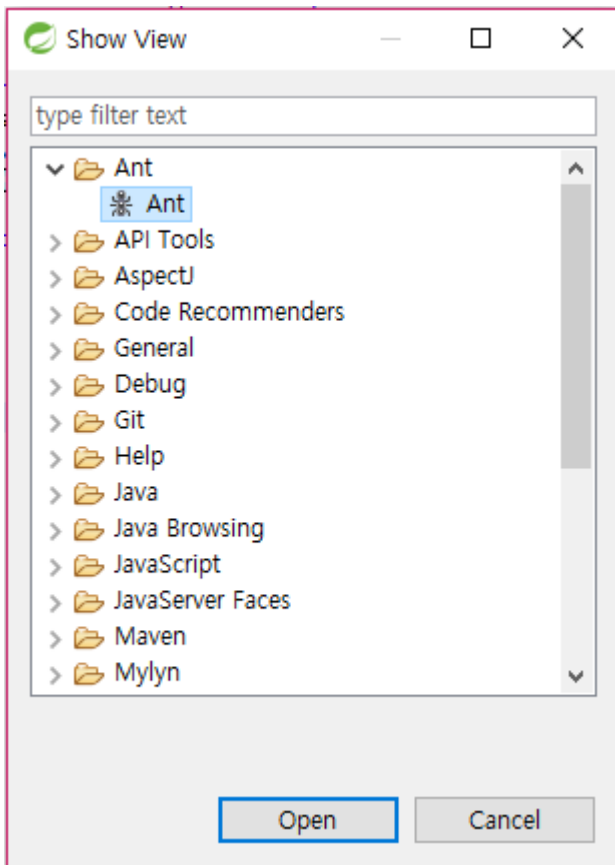


빌드

2. 이클립스-ant 사용하기

✓ ant파일 불러오기

다음 창이 뜨면 Ant->Ant를 선택하고 Ok

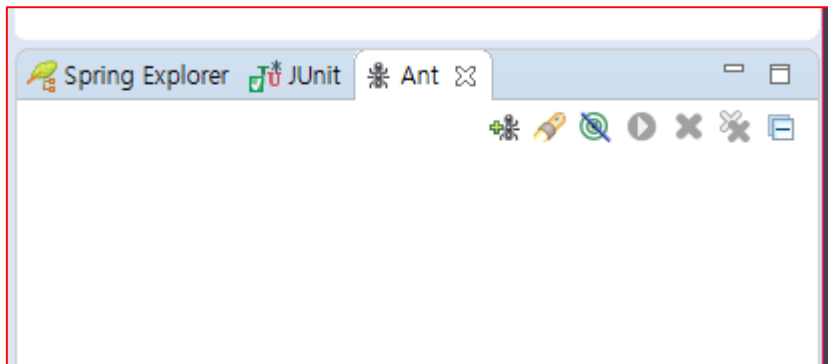


빌드

2. 이클립스-ant 사용하기

✓ ant파일 불러오기

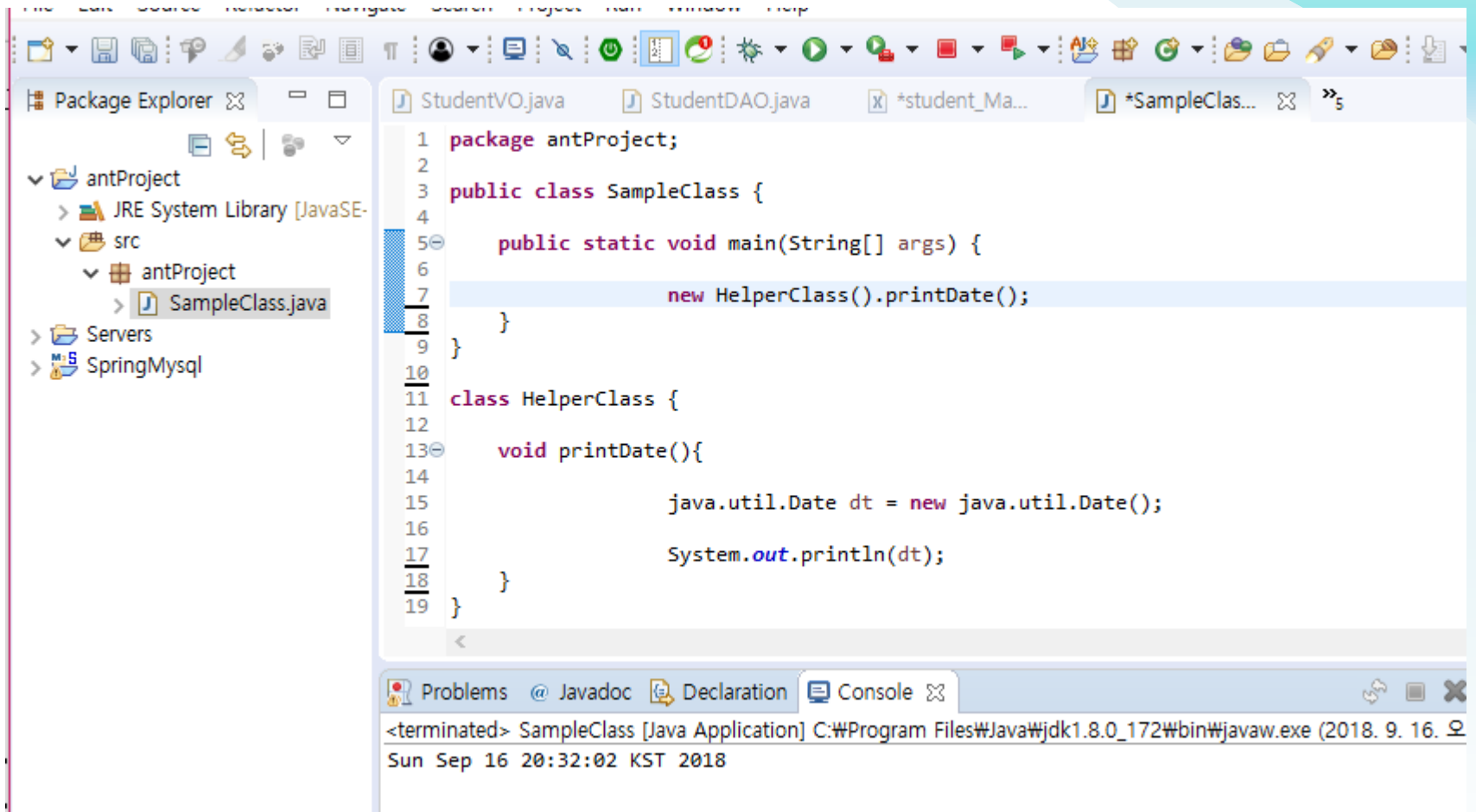
아래쪽에 Ant항목이 추가



빌드

2. 이클립스-ant 사용하기

- ✓ Java Project를 하나 생성하고 이어서 클래스를 하나 만들고 아래와 같은 코드를 입력하여 실행



빌드

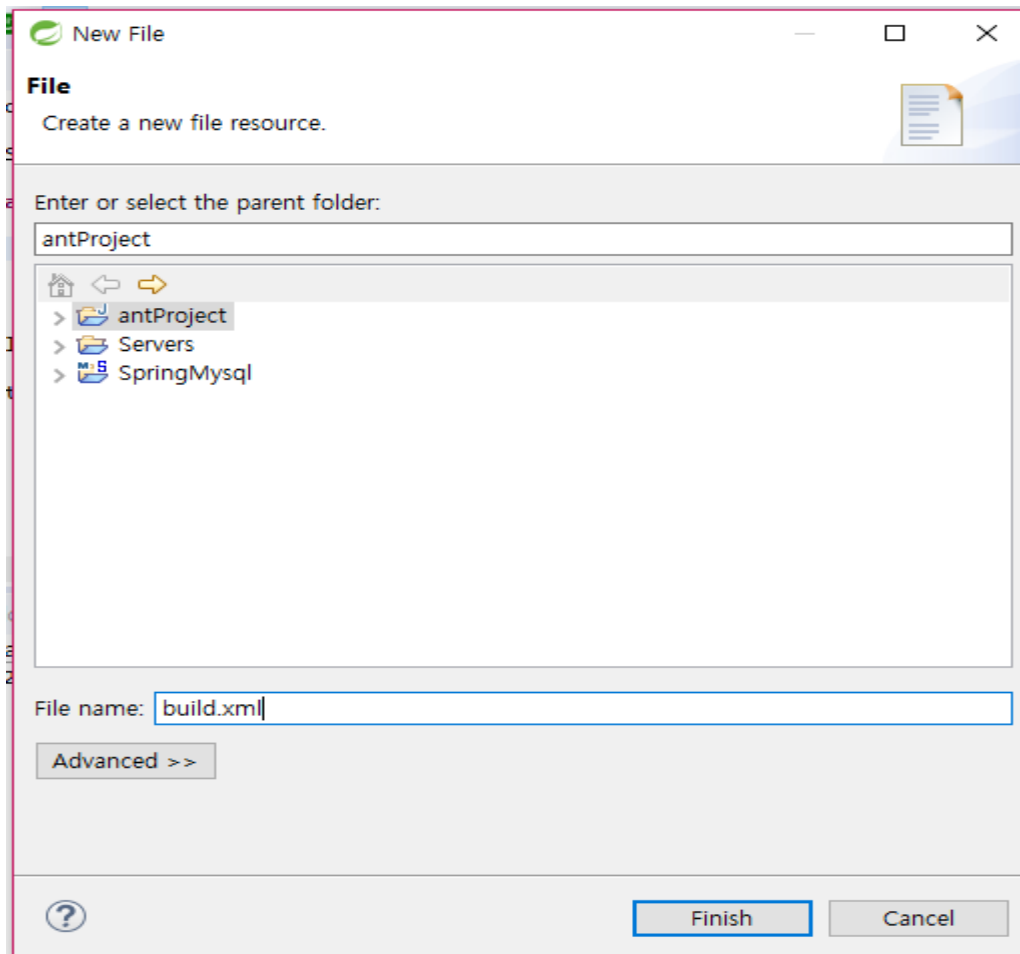
2. 이클립스-ant 사용하기

- ✓ Ant를 이용하여 특정 폴더를 생성하고 그 안에 이 소스를 컴파일한 후, 지정된 폴더에 JAR포맷으로 압축하여 실행하는 절차

빌드

2. 이클립스-ant 사용하기

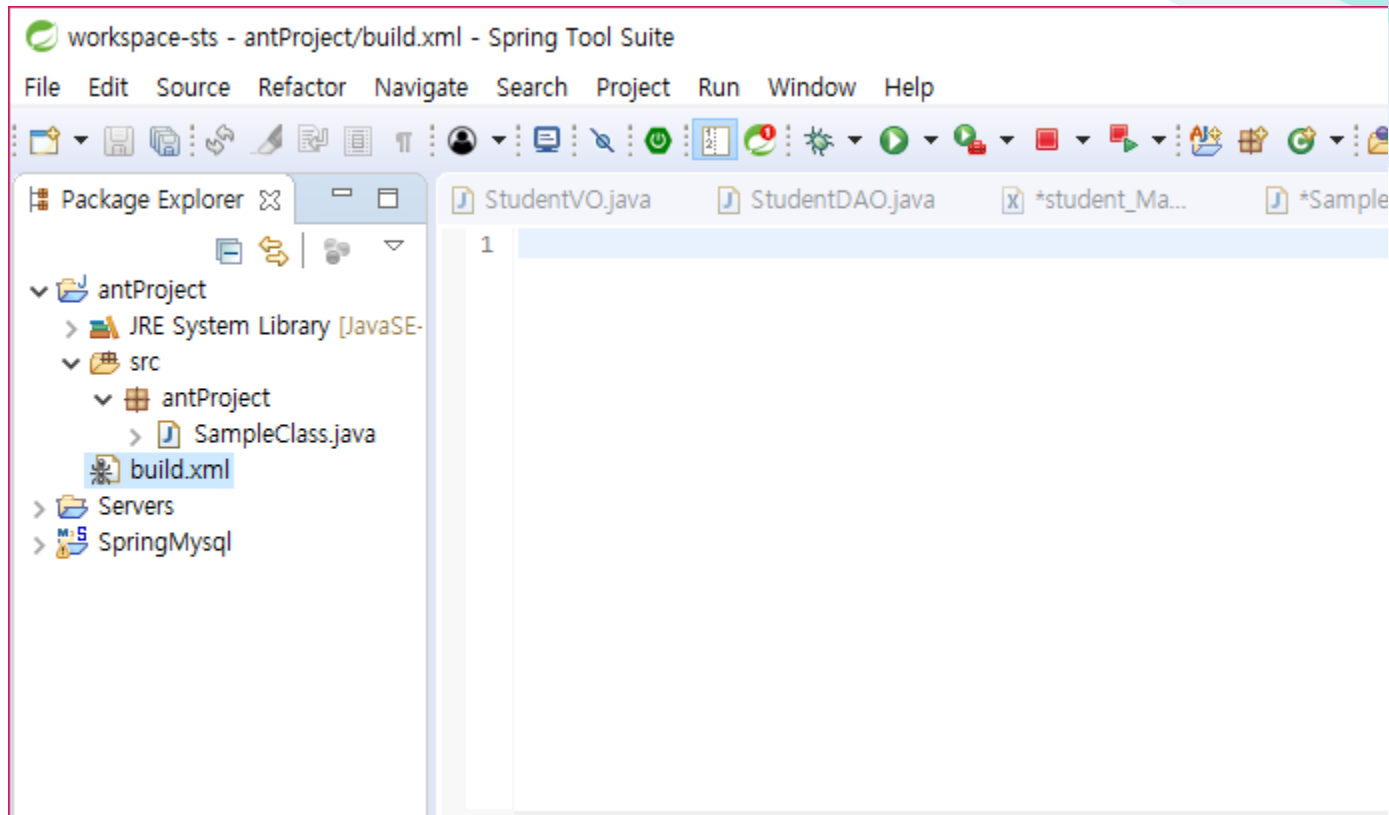
- 1) 먼저 Package Explorer 에서 적용하고자 하는 프로젝트 위에 마우스 우측을 눌러 New->File을 선택한다.
- ✓ Ant가 자동으로 인식하는 build.xml 파일을 생성하려고 한다.



빌드

2. 이클립스-ant 사용하기

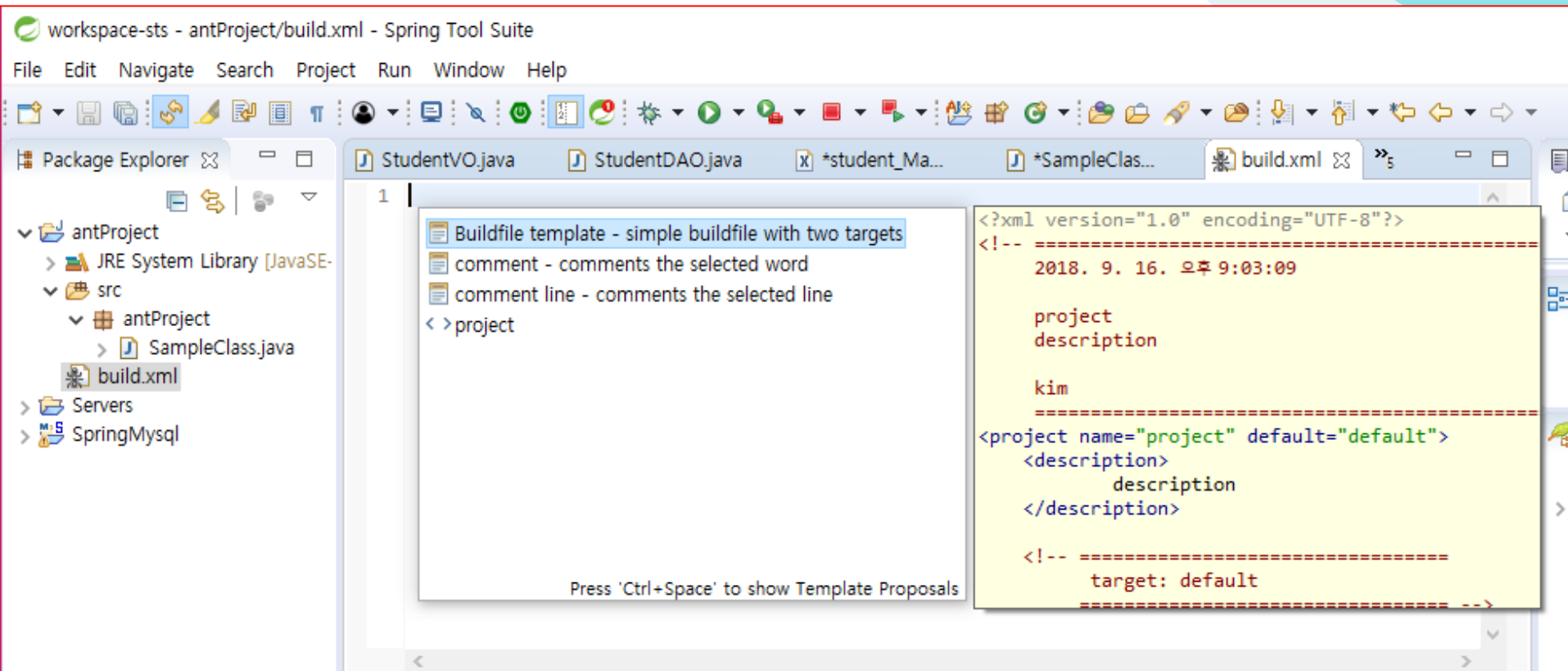
2) Eclipse의 에디터 화면에 내용이 없는 build.xml 파일이 나타난다.



빌드

2. 이클립스-ant 사용하기

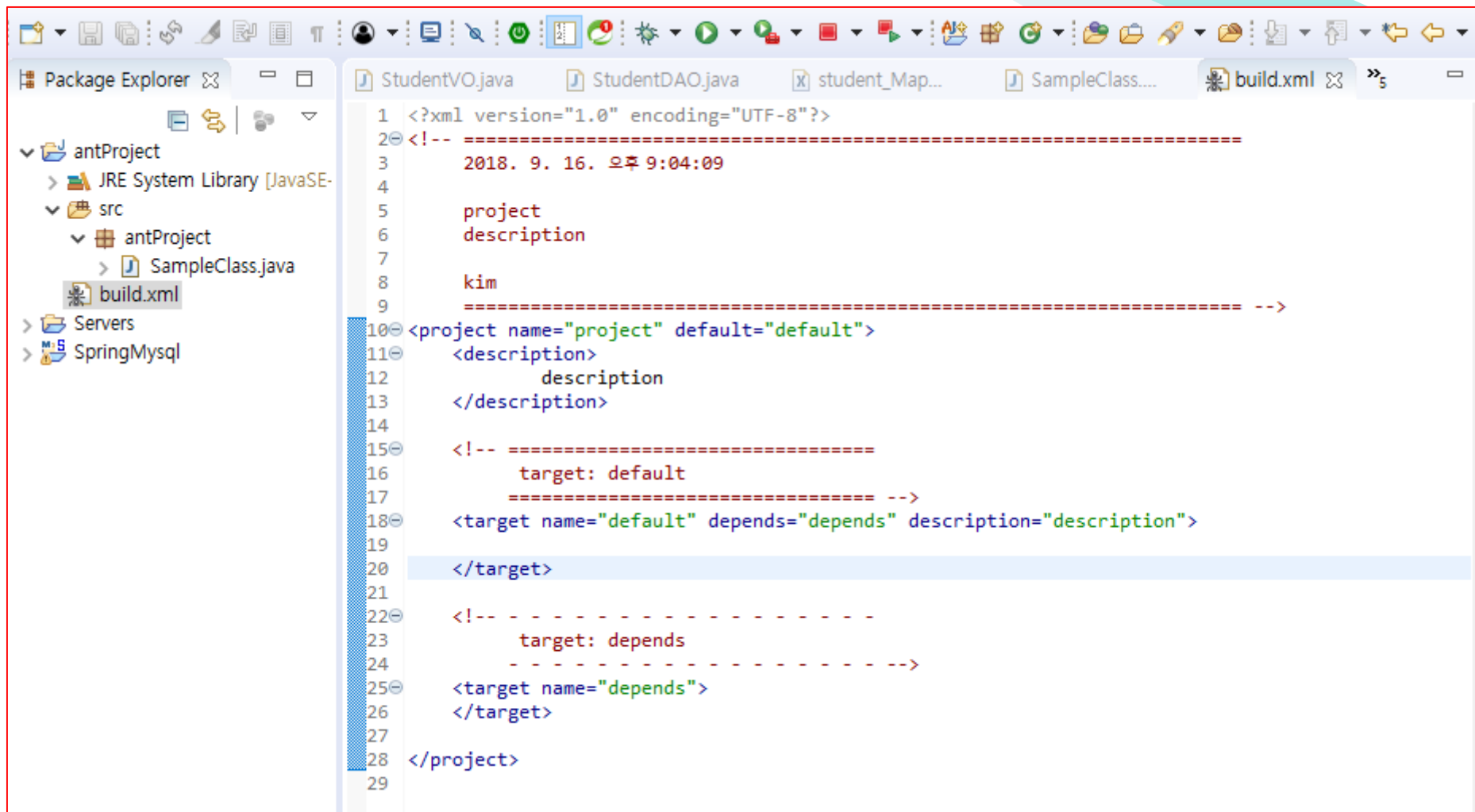
3) build.xml 화면을 마우스로 클릭한 후에 Ctrl + Space를 누르면 4가지 메뉴가 나타난다. 여기서 맨 위의 항목을 더블 클릭하면 Eclipse는 간단한 build.xml 파일을 완성하여 보여준다



빌드

2. 이클립스-ant 사용하기

4) 간단한 build.xml 의 내용은 다음과 같다.



The screenshot shows the Eclipse IDE interface. On the left, the Package Explorer displays a project named 'antProject' with a sub-project 'src' containing 'SampleClass.java' and 'build.xml'. The main editor window shows the content of 'build.xml', which is an Ant build file. The file contains XML tags for project configuration, including a project name, description, and two targets: 'default' and 'depends'.

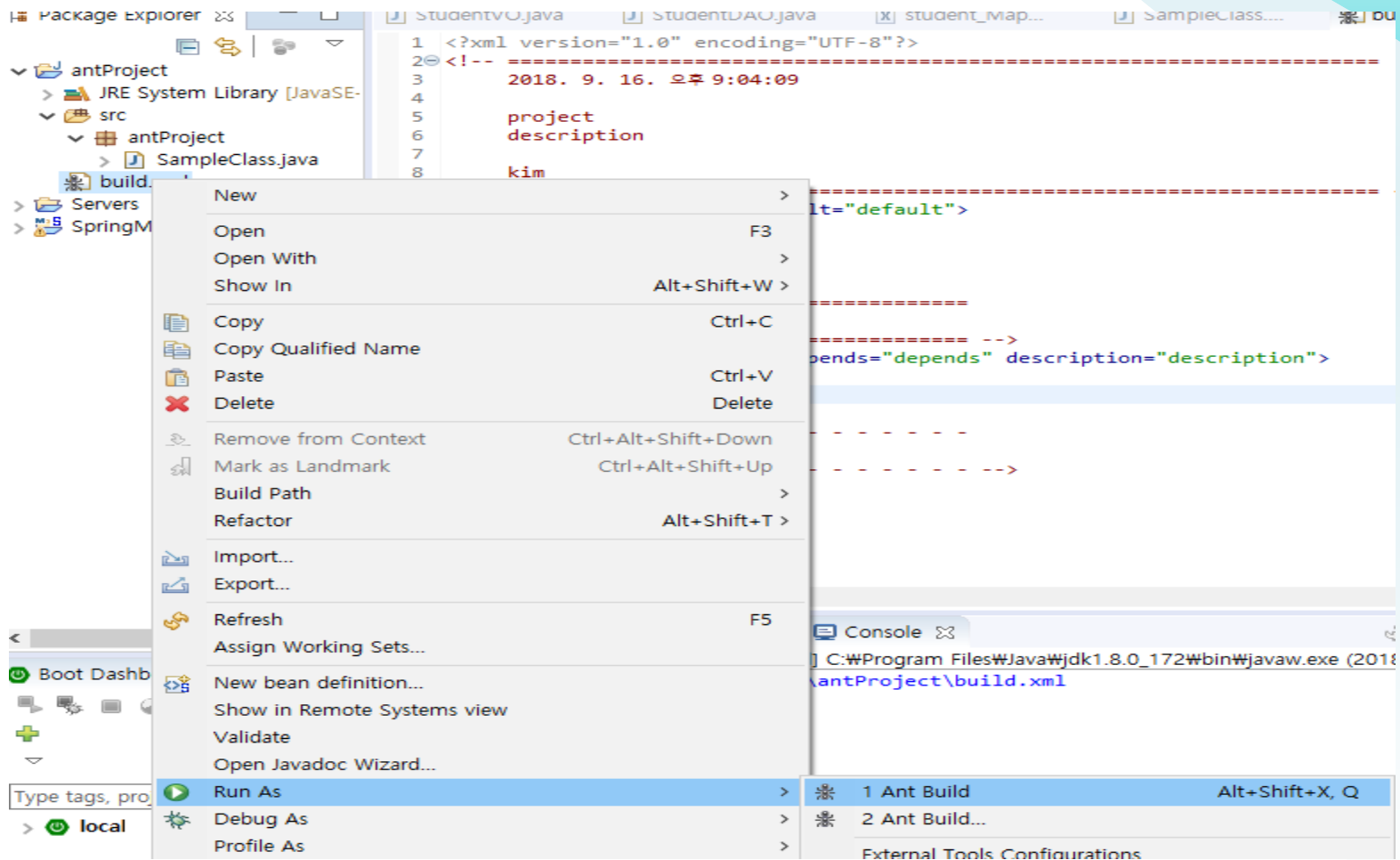
```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!-- =====
3     2018. 9. 16. 오후 9:04:09
4
5     project
6     description
7
8     kim
9     ===== -->
10 <project name="project" default="default">
11   <description>
12     description
13   </description>
14
15   <!-- =====
16       target: default
17       ===== -->
18   <target name="default" depends="depends" description="description">
19
20   </target>
21
22   <!-- - - - - -
23       target: depends
24       - - - - - -->
25   <target name="depends">
26   </target>
27
28 </project>
29
```

빌드

2. 이클립스-ant 사용하기

5) 이제 완성된 build.xml 파일을 Ant 로 하여금 실행하도록 해 보자.

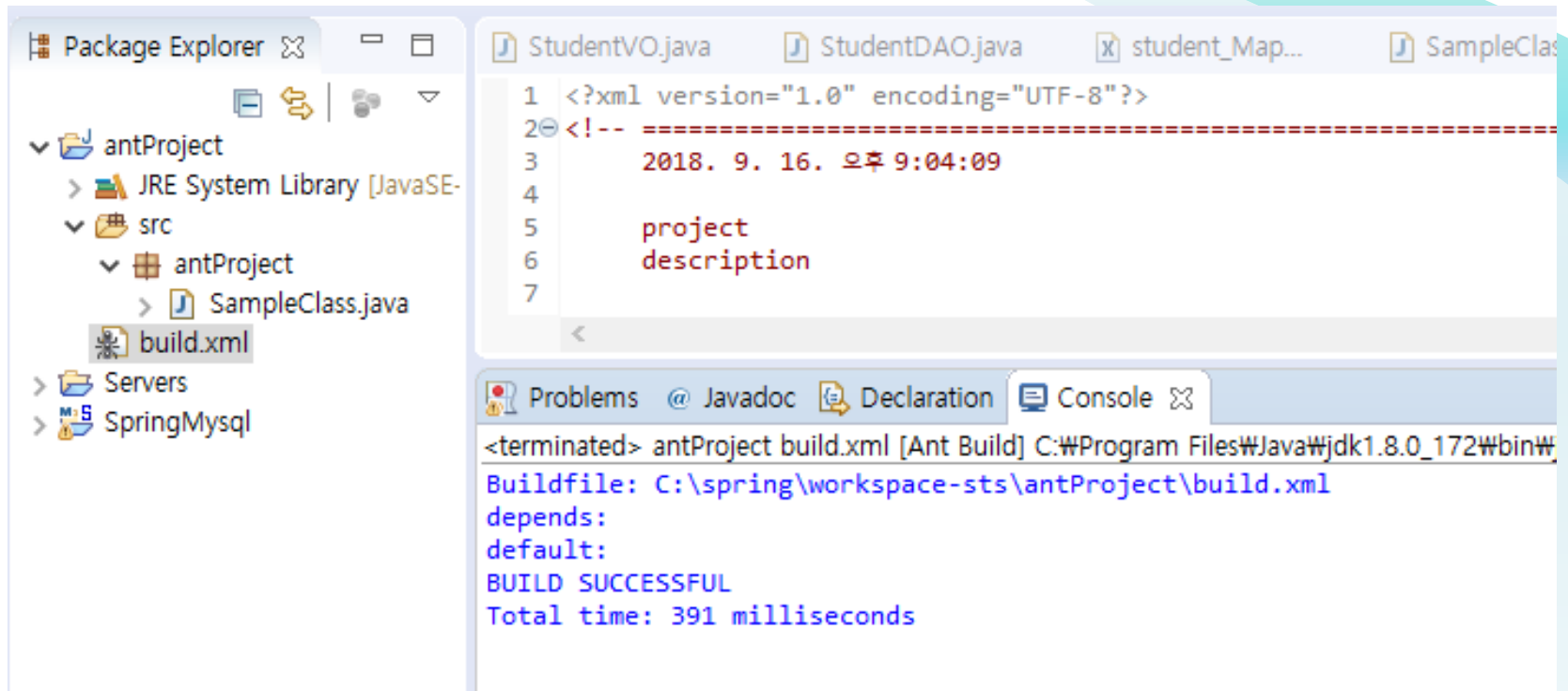
- ✓ Build.xml 에디터 화면에서 마우스 우측을 눌러 팝업메뉴에서 Run As->2 Ant Build 를 선택한다.



빌드

2. 이클립스-ant 사용하기

6) 에디터 하단에 있는 창에 Ant 의 실행결과가 출력된다

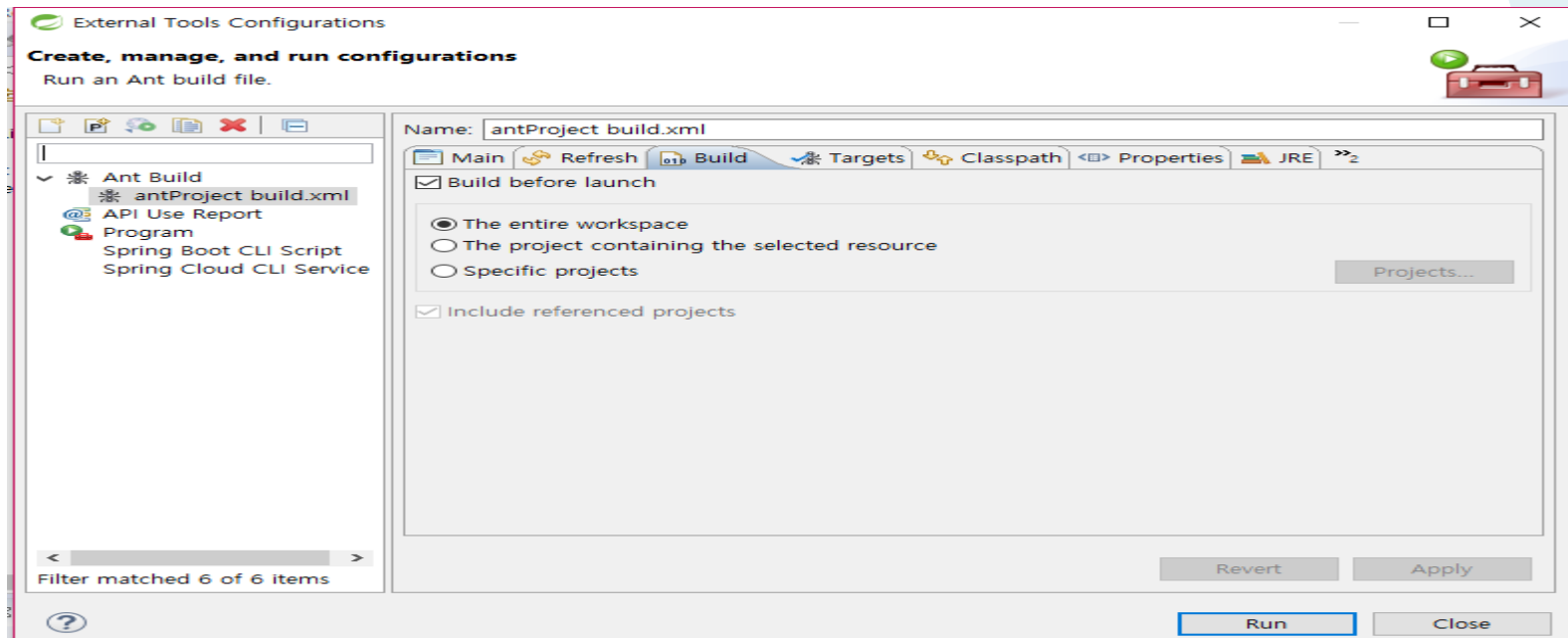
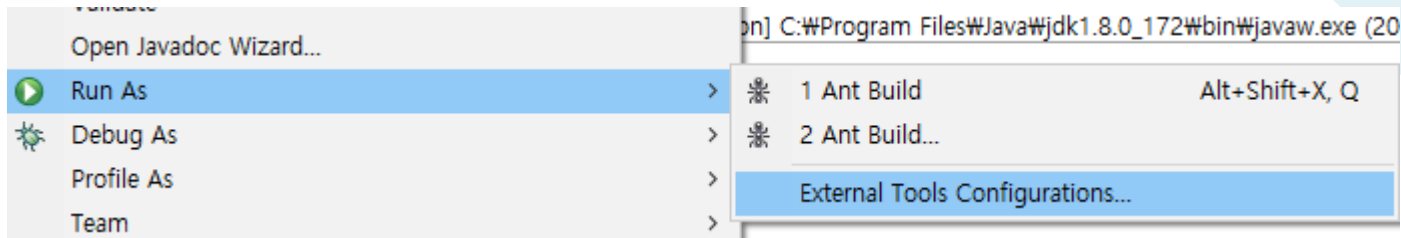


빌드

2. 이클립스-ant 사용하기

7) build.xml에 설정된 Target 는 Ant의 작업단위이다.

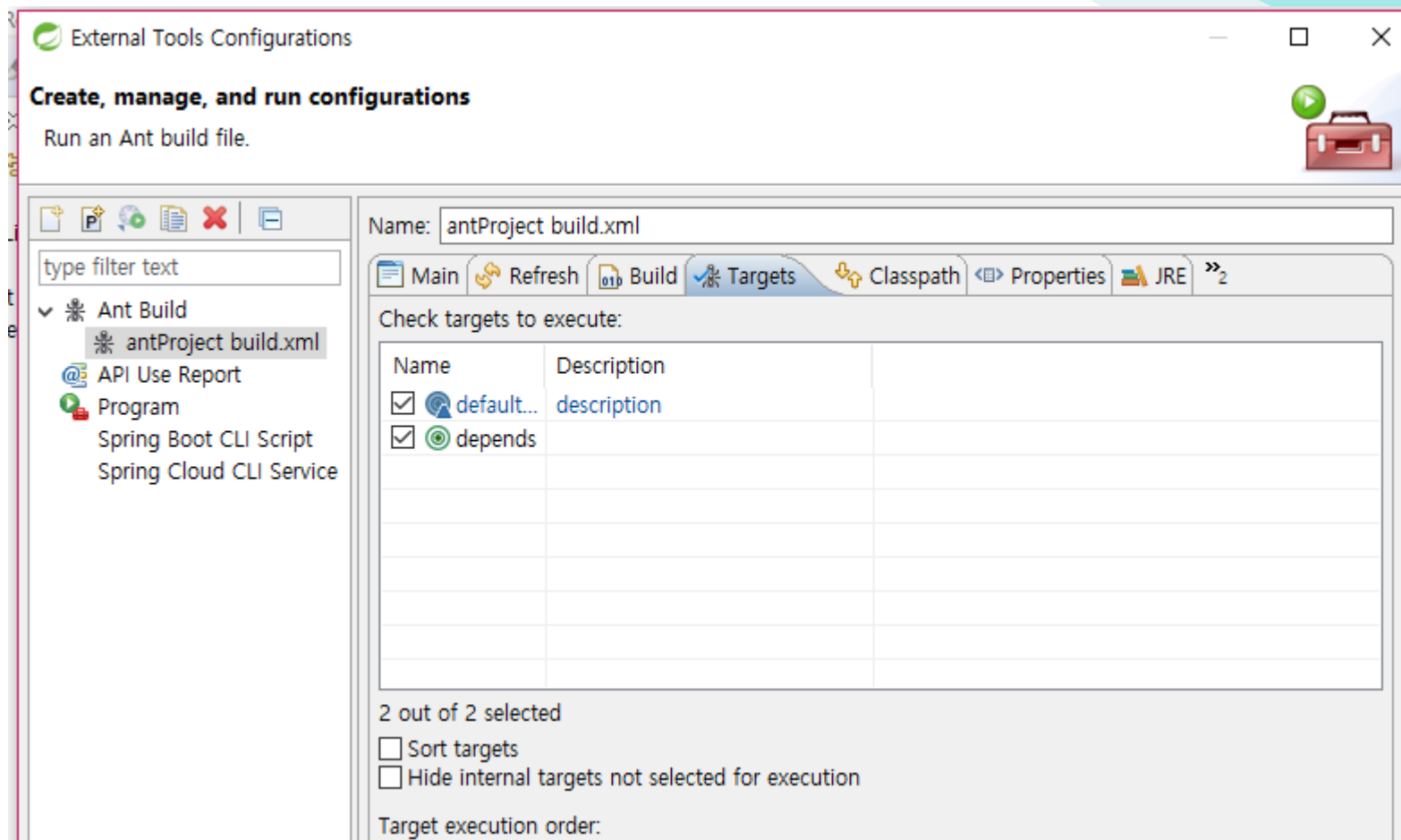
- ✓ Ant에게 Target 별로 실행을 요청하려면 팝업메뉴에서 '3 Ant Build...'를 선택하고 실행하고자 하는 Target 을 선택하면 된다.



2. 이클립스-ant 사용하기

7) 다음 화면에서 실행할 Target 에 체크하고 실행한다.

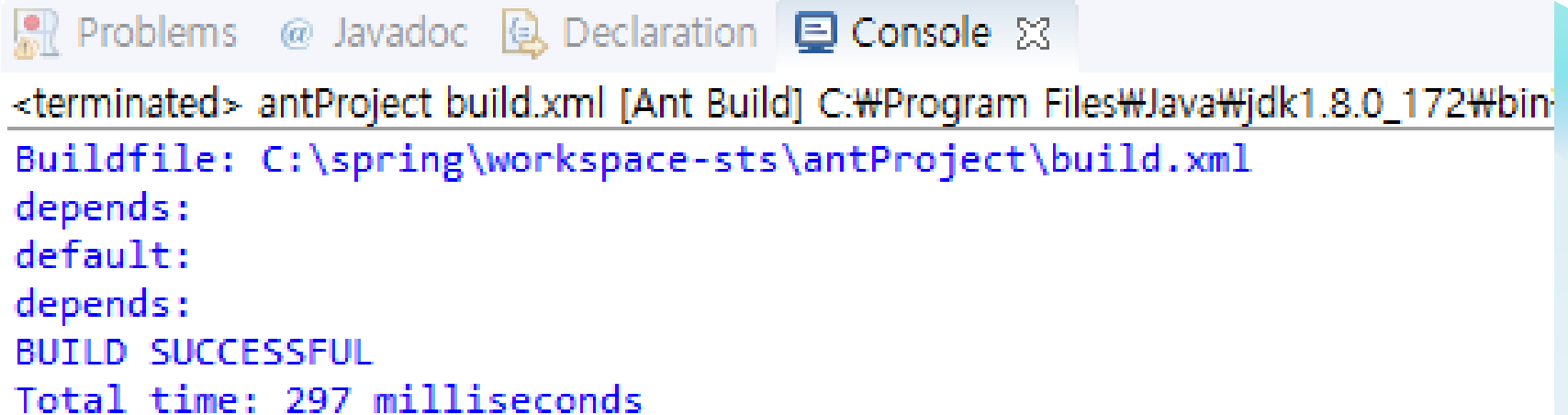
✓ 2개의 Target 모두 체크했다.



빌드

2. 이클립스-ant 사용하기

8) 2개의 타겟을 모두 실행한 결과는 다음과 같다.



The screenshot shows the Eclipse IDE's Console window. The title bar includes tabs for Problems, Javadoc, Declaration, and Console. The Console output displays the following text:

```
<terminated> antProject build.xml [Ant Build] C:\Program Files\Java\jdk1.8.0_172\bin  
Buildfile: C:\spring\workspace-sts\antProject\build.xml  
depends:  
default:  
depends:  
BUILD SUCCESSFUL  
Total time: 297 milliseconds
```

2. 이클립스-ant 사용하기

9) 간단하게 소스를 추가

- ✓ 그리고 기본 태그들에 대한 설명
- ✓ 추가한 소스는 아래와 같습니다.
- ✓ 첫번째 <target> 태그는 project 의 default 속성값을 참조하게 됩니다.
- ✓ <target> 의 name 속성값이 어디에도 없다면 <target> 아래 <echo> 는 실행이 되지 않을겁니다.
- ✓ 두번째 <target> 은 첫번째 <target> 이 실행되고 다음 실행이 되는데 이 두개의 의존관계는 depends 속성값으로 연결이 되어 있습니다

빌드

2. 이클립스-ant 사용하기

9) 간단하게 소스를 추가

```
1 <?xml version="1.0" encoding="UTF-8"?>
2
3 <project name="HelloSample" default="default">
4     <description>description</description>
5     <property name="HelloText" value="Hello"/>
6     <property name="WorldText" value="World"/>
7
8     <target name="default" depends="depends" description="description">
9         <echo>${HelloText}</echo>
10    </target>
11
12    <!-- target: depends -->
13    <target name="depends">
14        <echo>${WorldText}</echo>
15    </target>
16 </project>
```

Problems @ Javadoc Declaration Console Data Source Explorer SQL Results

<terminated> p0627 build.xml [Ant Build] C:\Program Files\Java\jre1.8.0_172\bin\javaw.exe (2018. 9. 17. 오전 9:01:09)

Buildfile: [D:\workspace\p0627\build.xml](#)

depends:

[echo] World

default:

[echo] Hello

BUILD SUCCESSFUL

Total time: 359 milliseconds