**소개**

1. JAVA 기반 빌드 툴로서 JAVA에 적합하게 만들어져 있고 자동화에 유리함.
2. 자바 소스 컴파일, 파일이나 폴더의 이동 및 복사, 삭제, jar, war, ear, zip 파일의 생성. DB 스크립트 실행, 외부 프로그램 실행 등 언급한 것 외에도 여러가지 작업이 가능.

**선수조건**

1. Ant를 지원하는 툴의 존재
2. 빌드할 소스코드의 존재

**필요도구**

1. Exria or Eclipse (Ant 탑재되어있음)
2. Tomcat

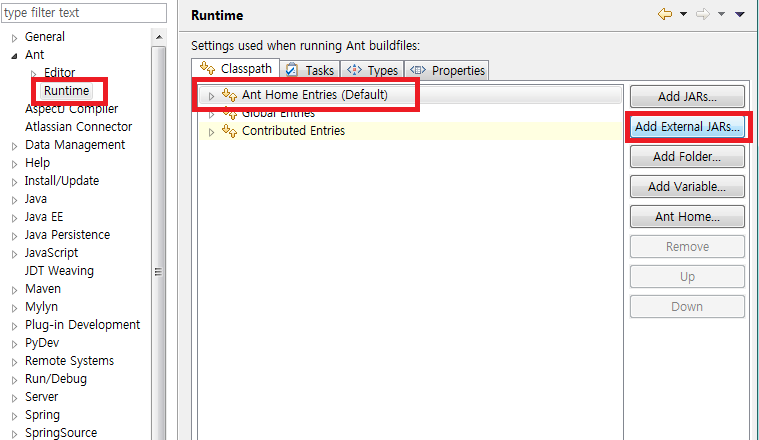
**목적**

1. **Ant도구를 이용하여 웹프로젝트를 패키지 구조에 맞게 war파일로 생성한다.**
2. **생성한 war파일을 원격지로 FTP를 이용하여 upload 한다.**

**실행방법**



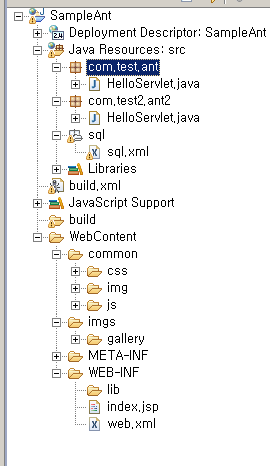
**ftp전송시 필요한 jar파일을 다운받는다.**

****

Window -> Preferences -> Ant -> Runtime -> Classpath항목에서

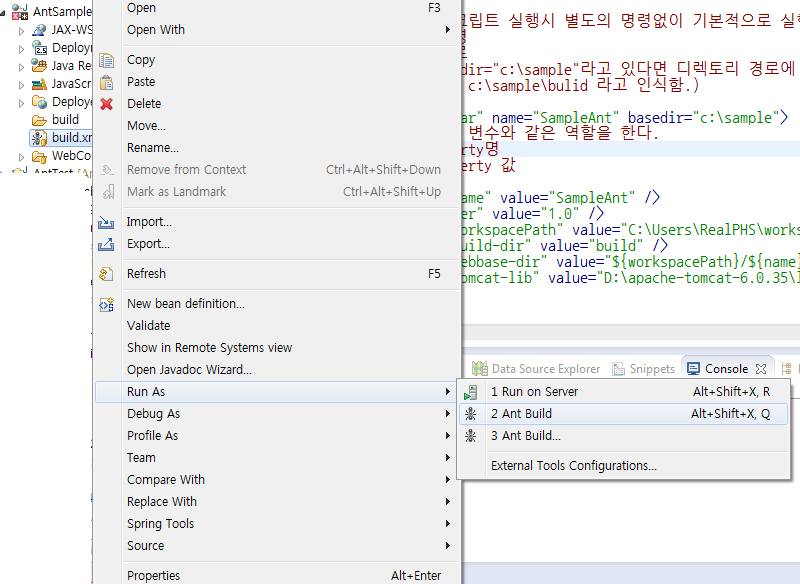
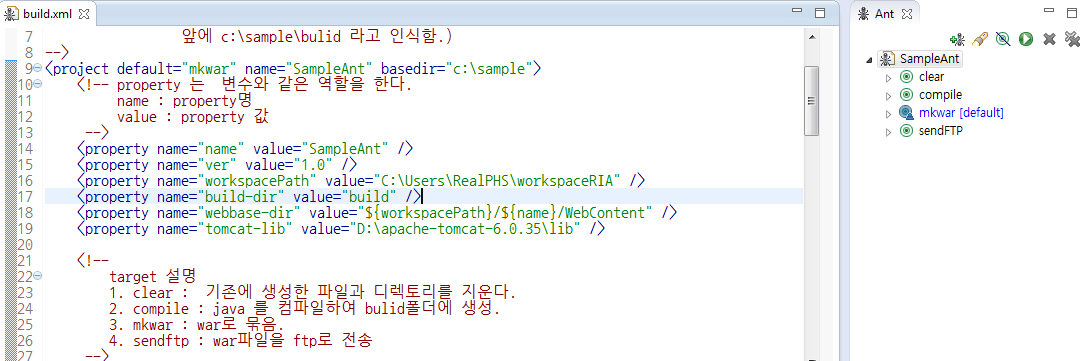
Ant Home Entries 클릭 후 Add External JARs... 클릭하여 commons-net-X.X.jar 를 추가

**예제**



1. 그림과 같이 샘플 프로젝트 구조는 위의 그림과 같다.
2. 그림과 같이build.xml파일을 만든다.
3. 아래는 build.xml 내용이다.

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <!--  default : ant 스크립트 실행시 별도의 명령없이 기본적으로 실행하는 작업명.  name : 프로젝트 명  basedir : 기본경로  (ex: basedir="c:\sample"라고 있다면 디렉토리 경로에 "build" 라고 입력시  앞에 c:\sample\bulid 라고 인식함.)  -->  <project default="mkwar" name="SampleAnt" basedir="c:\sample">  <!-- property 는 변수와 같은 역할을 한다.  name : property명  value : property 값  -->  <property name="name" value="SampleAnt" />  <property name="ver" value="1.0" />  <property name="workspacePath" value="C:\Users\RealPHS\workspaceRIA" />  <property name="build-dir" value="build" />  <property name="webbase-dir" value="${workspacePath}/${name}/webapps" />  <property name="tomcat-lib" value="D:\apache-tomcat-6.0.35\lib" />  <!--  target 설명  1. clear : 기존에 생성한 파일과 디렉토리를 지운다.  2. compile : java 를 컴파일하여 bulid폴더에 생성.  3. mkwar : war로 묶음.  4. sendftp : war파일을 ftp로 전송  -->  <target name="clear">  <delete file="sample.war" /> <!-- sample.war 를 삭제한다. -->  <delete dir="${build-dir}" /> <!-- c:\sample\build 경로의 폴더를 삭제한다.-->  </target>  <target name="compile" depends="clear">  <mkdir dir="${build-dir}" /> <!-- build폴더를 생성한다. -->  <!--  javac : java파일을 컴파일한다.  srcdir : 컴파일할 경로  destdir : 컴파일된 class파일이 생성될 폴더  encoding : java파일에 한글주석이 포함되어있을경우 컴파일시 한글이 깨져 에러가 나게된다  utf-8로 변경해주면 정상적으로 컴파일이 진행된다.  fork : 외부 컴파일러를 사용할지 여부.  memorymaximumsize : 메모리 최대값.  많은 파일을 컴파일을 하게 되면 java.lang.OutOfMemoryError 에러가 나게 된다.  방지하기 위해 fork="yes" memorymaximumsize="256m" 옵션을 추가해 줌.  -->  <javac srcdir="${workspacePath}/${name}/src" destdir="${build-dir}"  encoding="UTF-8" fork="yes" memorymaximumsize="256m">  <!-- classpath : 컴파일시 참조할 라이브러리들을 정의한다. -->  <classpath>  <!-- 자신의 워크스페이스 lib 경로에 있는 jar 파일들을 참조한다. -->  <fileset dir="${webbase-dir}/WEB-INF/lib">  <include name="\*.jar" />  </fileset>  <!-- 톰캣 lib 경로-->  <fileset dir="${tomcat-lib}">  <include name="\*.jar" />  </fileset>  </classpath>  </javac>  <!-- 파일복사 -->  <!-- todir : 해당 경로로 복사한다 -->  <copy todir="${build-dir}">  <fileset dir="${workspacePath}/${name}/src/java">  <include name="\*\*/\*.\*" /> <!-- 모든파일을 포함한다-->  <exclude name="\*\*/\*.java" /> <!-- java파일은 제외 -->  </fileset>  </copy>  </target>  <!-- war파일로 압축 -->  <target name="mkwar" depends="compile">  <!--  destfile : 생성할 war파일명  webxml : web.xml을 참조하여 프로젝트 구조로 묶는다.  -->  <war destfile="${name}.war" webxml="${webbase-dir}/WEB-INF/web.xml">  <!-- project이름\WebContent 하위 파일도 함께 묶는다.-->  <fileset dir="${webbase-dir}">  <include name="\*\*/\*.\*" />  </fileset>  <classes dir="${build-dir}" />  </war>  </target>  <!--  server : ftp주소  userid : ftp접속 ID  password : ftp접속 password  -->  <target name="sendFTP">  <ftp server="tomato.abc.com" userid="toto" password="1234">  <fileset file="="${name}.war" />  </ftp>  </target>    </project> |

1. 실행방법(1)
2. 
3. build.xml 파일에 마우스 오른쪽버튼눌러 나오는 메뉴에서 Run As – Ant Build 를 클릭하여 실행한다.
4. 실행방법(2)
5. 메뉴 Window-ShowView-Ant 를 클릭하게 되면 밑에 그림과 같은 메뉴가 보인다.
6. 
7. 각각의 Task Name이 뜨게되는데 task별로 더블클릭하여 실행 할수도 있다.

**결론**