Ant Installation

Prepared by:	Akhil Garg
Last Revised Date:	July. 27, 2017
Revision:	1.0.1

The purpose of this document is to give the overview about the Ant Installation.

Revision History

Date	Version	Description	Author
27-June-2017	1.0.1	Initial Version	Akhil

1. Prerequisites:

Ant requires the installation of the Java Development Kit (JDK).

2. Installing Apache Ant:

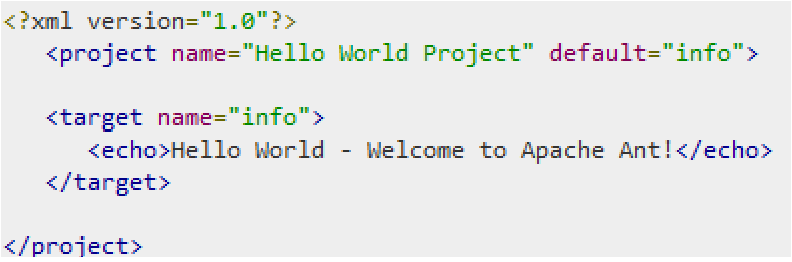
It is assumed that you have already downloaded and installed Java Development Kit (JDK) on your computer. If not, please follow the instructions here.

- Ensure that the JAVA_HOME environment variable is set to the folder where your JDK is installed.
- Download the binaries from http://ant.apache.org.
- Unzip the zip file to a convenient location c:\folder. using Winzip, winRAR, 7-zip or similar tools.
- Create a new environment variable called ANT_HOME that points to the Ant installation folder, in this case c:\apache-ant-1.8.2-bin folder.
- Append the path to the Apache Ant batch file to the PATH environment variable. In our case this would be the c:\apache-ant-1.8.2-bin\bin folder.

3. Build File:

Typically, Ant's build file, called build.xml should reside in the base directory of the project. However there is no restriction on the file name or its location. You are free to use other file names or save the build file in some other location.

For this exercise, create a file called build.xml anywhere in your computer with the following contents in it:



```
<?xml version="1.0"?>
   <project name="Hello World Project" default="info">

   <target name="info">
      <echo>Hello World - Welcome to Apache Ant!</echo>
   </target>

</project>
```

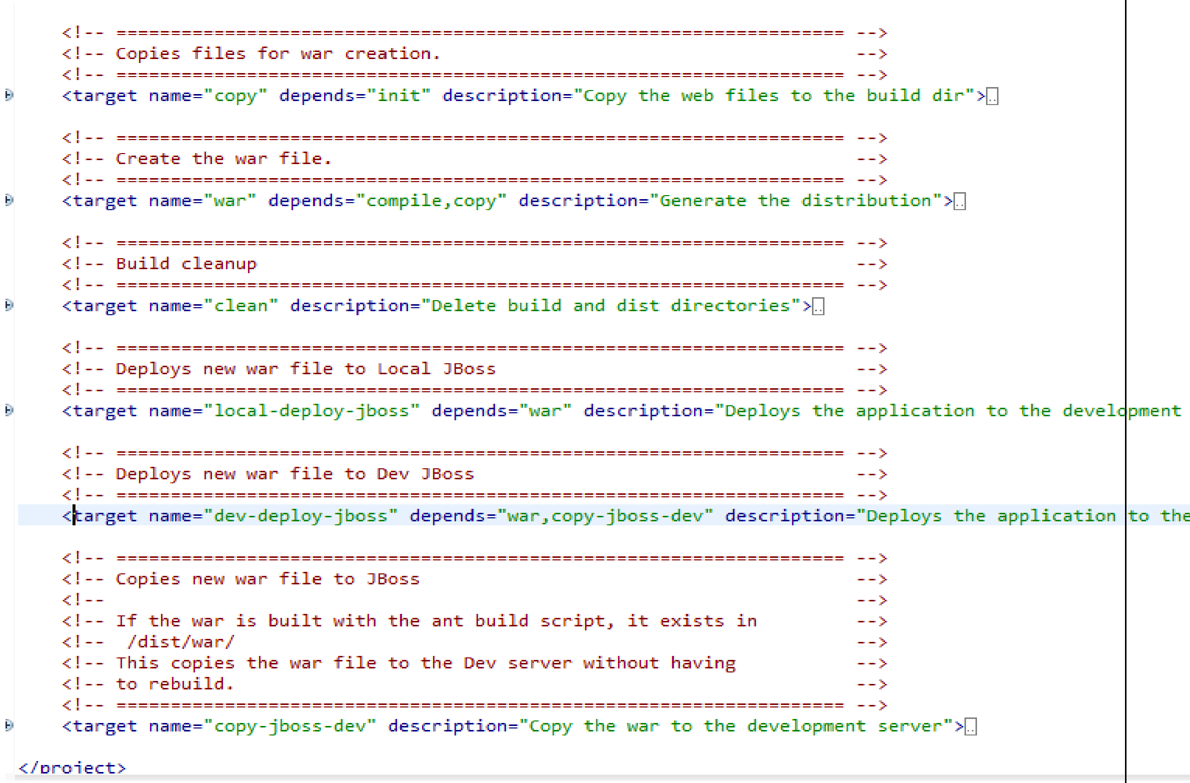
Note that there should be no blank line(s) or whitespace(s) before the xml declaration. If you allow them, the following error message occurs while executing the ant build -

The processing instruction target matching "[xX][mM][lL]" is not allowed. All build files require the project element and at least one target element.

4. Target Elements:

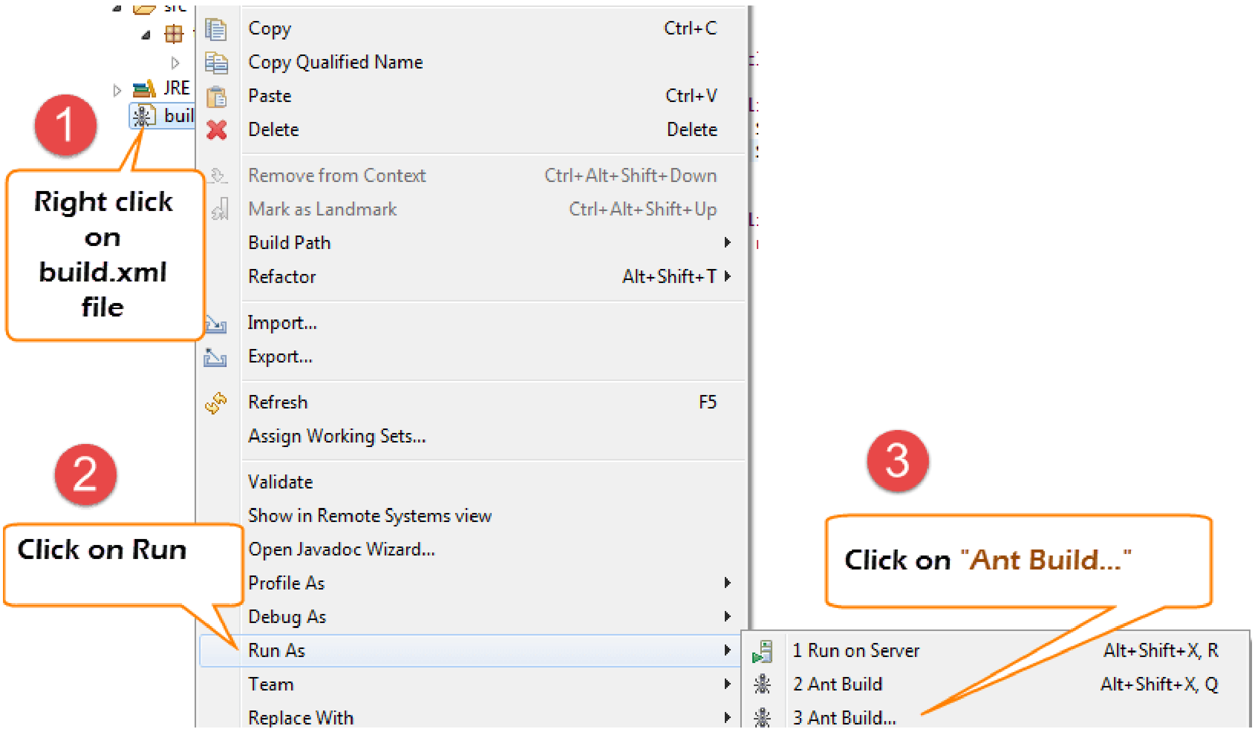
A target is a collection of tasks that you want to run as one unit. In our example, we have a simple target to provide an informational message to the user.

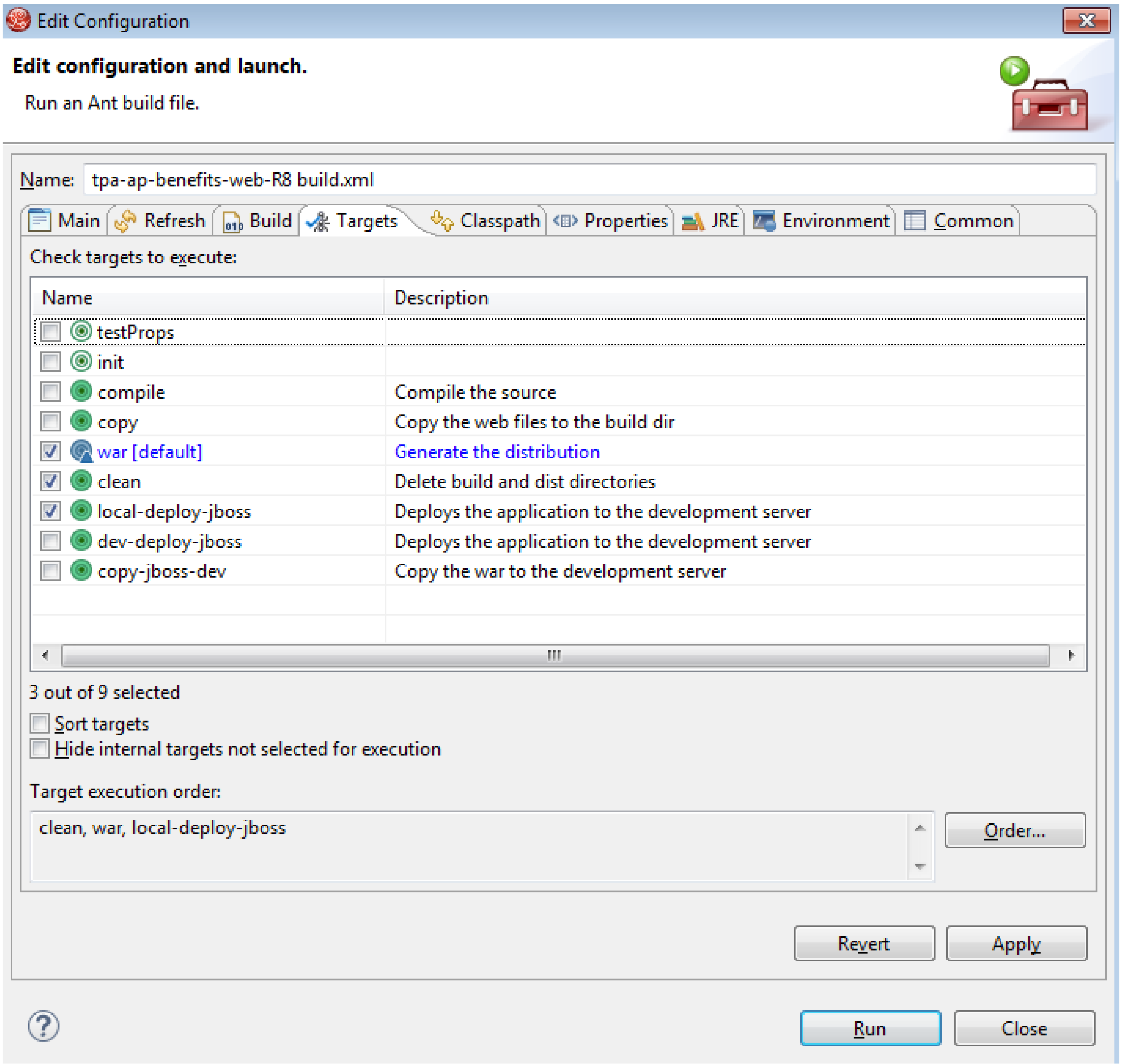
Targets can have dependencies on other targets. For example, a deploy target may have a dependency on the package target, the package target may have a dependency on the compile target and so forth. Dependencies are denoted using the depends attribute. For example:



```
    <!-- =================================================================== -->
    <!-- Copies files for war creation.                                      -->
    <!-- =================================================================== -->
    <target name="copy" depends="init" description="Copy the web files to the build dir">

    <!-- =================================================================== -->
    <!-- Create the war file.                                                -->
    <!-- =================================================================== -->
    <target name="war" depends="compile,copy" description="Generate the distribution">

    <!-- =================================================================== -->
    <!-- Build cleanup                                                       -->
    <!-- =================================================================== -->
    <target name="clean" description="Delete build and dist directories">

    <!-- =================================================================== -->
    <!-- Deploys new war file to Local JBoss                                 -->
    <!-- =================================================================== -->
    <target name="local-deploy-jboss" depends="war" description="Deploys the application to the development

    <!-- =================================================================== -->
    <!-- Deploys new war file to Dev JBoss                                   -->
    <!-- =================================================================== -->
    <target name="dev-deploy-jboss" depends="war,copy-jboss-dev" description="Deploys the application to the

    <!-- =================================================================== -->
    <!-- Copies new war file to JBoss                                        -->
    <!--                                                                     -->
    <!-- If the war is built with the ant build script, it exists in         -->
    <!--  /dist/war/                                                         -->
    <!-- This copies the war file to the Dev server without having           -->
    <!-- to rebuild.                                                         -->
    <!-- =================================================================== -->
    <target name="copy-jboss-dev" description="Copy the war to the development server">

</project>
```

To run Ant from eclipse go to build.xml file -> right click on file -> Run as… -> click Build file and get select the targets:





Edit Configuration

Edit configuration and launch.

Run an Ant build file.

Name: tpa-ap-benefits-web-R8 build.xml

Check targets to execute:

	Name	Description
□	testProps	
□	init	
□	compile	Compile the source
□	copy	Copy the web files to the build dir
☑	war [default]	Generate the distribution
☑	clean	Delete build and dist directories
☑	local-deploy-jboss	Deploys the application to the development server
□	dev-deploy-jboss	Deploys the application to the development server
□	copy-jboss-dev	Copy the war to the development server

3 out of 9 selected

□ Sort targets

□ Hide internal targets not selected for execution

Target execution order:

clean, war, local-deploy-jboss