**Gradle Build from Ant Build: Importing Ant Build File**

To demonstrate how easy it is to [import an Ant build](http://blog.toddcostella.com/2013/01/gradle-migration-part-1-importing-build-xml/) in a Gradle build, I first provide the code listing for a simplified Ant build.

Simply use

**ant.importBuild 'build.xml'**

**Ant Build File: build.xml**

<?xml version="1.0" encoding="UTF-8"?>

<project name="JavaArrays" default="all" basedir=".">

<description>Java Array Utility Functions</description>

<property name="javac.debug" value="true" />

<property name="src.dir" value="src" />

<property name="dist.dir" value="dist" />

<property name="classes.dir" value="classes" />

<property name="javadoc.dir" value="${dist.dir}/javadoc" />

<property name="jar.name" value="javaArrays.jar" />

<property name="jar.filesonly" value="true" />

<path id="classpath">

</path>

<target name="-init">

<mkdir dir="${classes.dir}" />

<mkdir dir="${dist.dir}" />

</target>

<target name="compile"

description="Compile the Java code."

depends="-init">

<javac srcdir="${src.dir}"

destdir="${classes.dir}"

classpathref="classpath"

debug="${javac.debug}"

includeantruntime="false" />

</target>

<target name="jar"

description="Package compiled classes into JAR file"

depends="compile">

<jar destfile="${dist.dir}/${jar.name}"

basedir="${classes.dir}"

filesonly="${jar.filesonly}">

</jar>

</target>

<target name="all"

description="Compile Java source, assemble JAR, and generate documentation"

depends="jar, javadoc" />

<target name="javadoc" description="Generate Javadoc-based documentation">

<mkdir dir="${javadoc.dir}" />

<javadoc doctitle="Examples of Java Array Utility Functions"

destdir="${javadoc.dir}"

sourcepath="${src.dir}"

classpathref="classpath"

private="true"

author="Dustin" />

</target>

<target name="clean" description="Remove generated artifacts.">

<delete dir="${classes.dir}" />

<delete dir="${dist.dir}" />

</target>

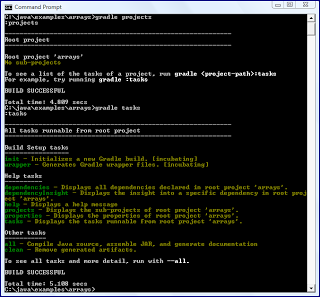
</project>

The above Ant build file has some fairly typical targets with names like "compile", "jar", "javadoc", and "clean". All of this functionality can be imported into a Gradle build file. The next code listing is the complete Gradle build file that does this.

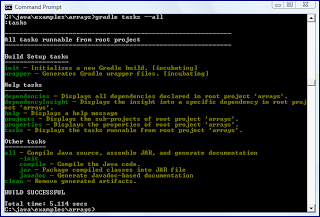
**Gradle build.gradle that imports Ant build.xml**

ant.importBuild 'build.xml'

The one-line Gradle build file shown above imports the Ant build file shown earlier. The effects of this can be easily seen in the following screen snapshots. The initial screen snapshot shows that the single line Gradle build file makes the "arrays" project available to the Gradle build as well as "other tasks" of "all" and "clean" with the descriptions associated with those Ant targets.



One can use gradle tasks --all to see all Ant targets, including the dependent targets such as "compile", "jar", and "javadoc". This is demonstrated in the next screen snapshot.



The next screen snapshot demonstrates running the default "all" target in the Ant build from the Gradle build.

[](http://3.bp.blogspot.com/-4e3PYJ9MBjk/UszqyAZrAtI/AAAAAAAAFDM/wMxAKW0-VIU/s1600/gradleImportedAntProjectAllTargetsRun.png)

As the build listings and images have demonstrated, importing an existing Ant build in a Gradle build is a straightforward process.