Real's HowTo

... <- More

Java HowTo

Use ANT to Build a JAR with version/build number

- Download ANT from http://ant.apache.org/bindownload.cgi
- Unzip into the root of C: (Windows), rename the folder to \Ant
- Create a file call *antenv.cmd*. Before using ANT, you must execute this file to set up the environment. You may need to adjust the values according to your installation!

```
set ANT_HOME=c:\ant
set JAVA_HOME=C:\Progra~1\Java\jdk1.5.0
set PATH=%ANT_HOME%\bin;%JAVA_HOME%\bin;%path%
```

- Create a simple Hello class as shown below (you need have a package).

```
package howto;
public class Hello {
   public static void main( String[] args ){
        System.out.println( "Hello World" );
    }
}
```

- Execute your *antenv.cmd* to set the environment
- Compile your Hello class and make a jar to check if your environment is ok.

```
> javac howto/Hello.java
> jar -cvf hello.jar howto/Hello.class
added manifest
adding: howto/hello.class(in = 415) (out= 284) (deflated 31%)
> java -cp hello.jar howto.Hello
Hello World
```

- Create a *build.xml* which is the default ant build file ...
- ... and load it into your favorite editor. Type the following ant script and save it.

```
<project default="buildHello">
  <target name="compile">
    <javac srcdir="." />
  </target>

  <target name="buildHello" depends="compile" />
</project>
```

Go back to the DOS shell and launch Ant with

```
> ant compile
Buildfile: build.xml
```

```
compile:
    [javac] Compiling 1 source file
```

- Go back to the editor and add the jar building command. The target *compile* is also executed since the target *jar* depends on it.

```
<project default="buildHello">
  <target name="compile">
    <javac srcdir="." />
  </target>

  <target name="jar" depends="compile">
    <jar destfile="hello.jar"
         basedir="."
         includes="**/*.class"
         />
  </target>

  <target name="buildHello" depends="compile,jar" />
</project>
```

- Execute your Ant script :

```
> ant jar
Buildfile: build.xml

compile:

jar:
      [jar] Building jar: /Dev/hello.jar

BUILD SUCCESSFUL
Total time: 2 seconds
$ jar -tvf hello.jar
jar -tvf hello.jar
     0 Wed May 03 17:06:32 EST 2006 META-INF/
    55 Wed May 03 17:06:32 EST 2006 META-INF/MANIFEST.MF
    55 Wed May 03 17:06:32 EST 2006 howto/
   335 Wed May 03 16:36:16 EST 2006 howto/Hello.class
```

- Try your new Jar file

```
> java -cp Hello.jar howto.Hello
Hello World
```

- Make your Jar *auto-executable* by specifying the Main class into a MANIFEST.MF file to be included in the Jar. Add the following lines to build.xml

```
<project default="buildHello">
  <target name="compile">
    <javac srcdir="." />
  </target>

  <target name="jar" depends="compile">
     <delete file="hello.jar"/>
     <delete file="MANIFEST.MF"/>
     <manifest file="MANIFEST.MF">
        <attribute name="Built-By" value="${user.name}"/>
        <attribute name="Main-Class" value="howto.Hello"/>
    </manifest>

      <jar destfile="hello.jar"
           basedir="."
           includes="**/*.class"
           manifest="MANIFEST.MF"
           />
  </target>

  <target name="buildHello" depends="compile,jar" />
</project>
```

- Now you can launch Hello with only

```
> ant jar
...
> java -jar hello.jar
Hello World
```

- Add a build number with the Ant task *buildnumber* . Ant will create (if necessary) and increment a property in a file called *build.num* (so you need to keep it!). Then a property, *build.number*, is available in your Ant script. The version number is also a property. Here it is hard-coded in the script but you can read it from a file.

```
<project default="buildHello">
  <target name="compile">
    <javac srcdir="." />
  </target>

  <target name="jar" depends="compile">
     <delete file="hello.jar"/>
     <delete file="MANIFEST.MF"/>
     <property name="version.num" value="1.00"/>
     <buildnumber file="build.num"/>

     <manifest file="MANIFEST.MF">
        <attribute name="Built-By" value="${user.name}"/>
        <attribute name="Main-Class" value="howto.Hello"/>
        <attribute name="Implementation-Version"
                     value="${version.num}-b${build.number}"/>
    </manifest>

      <jar destfile="hello.jar"
           basedir="."
           includes="**/*.class"
           manifest="MANIFEST.MF"
           />
  </target>

  <target name="buildHello" depends="compile,jar" />
</project>
```

- If you examine the MANIFEST.MF, you now see a new entry

```
Implementation-Version: 1.00-b1
```

- and after the next build operation, the entry will be

```
Implementation-Version: 1.00-b2
```

- Modify the Hello class to read the *Implementation-Version* information and display it.

```
package howto;

public class Hello {
   public static void main( String[] args ){
       System.out.println( "Hello World ");
       System.out.println("version : " +
         Hello.class.getPackage().getImplementationVersion() );
    }
}
```

- Finally we add a target to cleanup and the main target to build everything to the *build.xml* file.

```
<project default="buildHello">
  <target name="compile">
    <javac srcdir="." />
  </target>

  <target name="jar">
     <delete file="hello.jar"/>
     <property name="version.num" value="1.00"/>
     <buildnumber file="build.num"/>

     <manifest file="MANIFEST.MF">
        <attribute name="Built-By" value="${user.name}"/>
        <attribute name="Main-Class" value="howto.Hello"/>
        <attribute name="Implementation-Version"
                 value="${version.num}-b${build.number}"/>
    </manifest>

      <jar destfile="hello.jar"
           basedir="."
           includes="**/*.class"
           manifest="MANIFEST.MF"
           />
  </target>

  <target name="cleanup">
       <delete>
         <fileset dir="." includes="**/*.class"/>
         <fileset file="MANIFEST.MF"/>
       </delete>
  </target>

  <target name="buildHello" depends="compile,jar,cleanup" />
</project>
```

- Build and launch the Hello class

```
> ant buildHello
...
> java -jar hello.jar
```

```
Hello World
version : 1.00-b3
```

b3 == build #3 (if it's your third build !)

Build number is great but a Built date is useful too!

```
...
 <target name="jar">
   <delete file="hello.jar"/>
   <property name="version.num" value="1.00"/>
   <buildnumber file="build.num"/>
   <tstamp>
     <format property="TODAY" pattern="yyyy-MM-dd HH:mm:ss" />
   </tstamp>

   <manifest file="MANIFEST.MF">
      <attribute name="Built-By" value="${user.name}"/>
      <attribute name="Main-Class" value="howto.Hello"/>
      <attribute name="Implementation-Version"
             value="${version.num}-b${build.number}"/>
      <attribute name="Built-Date" value="${TODAY}"/>
  </manifest>

  <jar destfile="hello.jar"
       basedir="."
       includes="**/*.class"
       manifest="MANIFEST.MF"
       />
</target>
```

See also this Howto.

Comments (2)

Want to contribute?
Join or Sign In

That's a really cool guid. Thanks!
andi 5/11

Brilliant guide, thanks! Exactly what I wanted and easy to understand.
kat 3/26

Translate »

If you find this article useful, consider making a small donation
to show your support for this Web site and its content. PayPal DONATE

[home]