

Stack Overflow is a question and answer site for professional and enthusiast programmers. It's 100% free, no registration required.

Take the 2-minute tour ×

Unable to find a javac compiler



The world's first flash-optimized,
in-memory NoSQL Database.
Now Open Source.

Learn More

I am trying to package my web application into war file using Ant.

When I build, I am getting the following error:

```
C:\Documents and Settings\Administrator\workspace\Assignment7\build.xml:67: Unable to find
a javac compiler;
com.sun.tools.javac.Main is not on the classpath.
Perhaps JAVA_HOME does not point to the JDK.
It is currently set to "C:\Program Files\Java\jre6"
```

The following is my build.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<project name="Generate War" default="install" basedir=".">

    <property name="build.dir" value="build"/>
    <property name="dir.name" value="Assignment7"/>
    <property name="package.name" value="${dir.name}.war"/>

    <property name="content" value="webContent"/>
    <property name="web-inf" value="${content}\WEB-INF"/>
    <property name="meta-inf" value="${content}\META-INF"/>
    <property name="jsp.dir.name" value="${content}"/>

    <property name="lib" value="${web-inf}\lib"/>
    <property name="src" value="src"/>
    <property name="dest.dir" value="target"/>

    <!-- Setting path to the server webapp folder -->

    <property name="webapp.dir" value="C:\Program Files\Apache Software Foundation\Tomcat
6.0\webapps"/>

    <!-- Using temp folder for convinence -->

    <property name="temp.dir" value="temp"/>
    <property name="temp.dir.web-inf" value="${temp.dir}\WEB-INF"/>
    <property name="temp.dir.meta-inf" value="${temp.dir}\META-INF"/>
    <property name="temp.dir.lib" value="${temp.dir.web-inf}\lib"/>
    <property name="temp.dir.classes" value="${temp.dir.web-inf}\classes"/>
    <!--<property name="temp.dir.classes.dir" value="${temp.dir.classes}\**"/>-->

    <property name="package.file" value="${dest.dir}\${package.name}"/>

    <path id="build.class.path">
        <fileset dir="${lib}">
            <include name="**/*.jar"/>
        </fileset>
    </path>
    <path id="build.class.path1">
        <fileset dir="C:\program files\java\jre6\lib">
            <include name="**/*.jar"/>
        </fileset>
    </path>

    <target name="clean">
        <delete>
            <fileset dir="${dest.dir}" includes="**/*"/>
        </delete>
        <delete dir="${temp.dir}"/>
        <delete dir="${temp.dir.classes}"/>
        <delete dir="${temp.dir.web-inf}"/>
        <delete dir="${temp.dir.meta-inf}"/>
    </target>

    <target name="prepare" depends="clean">
```

```

<mkdir dir="${dest.dir}"/>
<mkdir dir="${temp.dir}"/>
<mkdir dir="${temp.dir.web-inf}"/>
<mkdir dir="${temp.dir.lib}"/>
<mkdir dir="${temp.dir.classes}"/>
<mkdir dir="${temp.dir.meta-inf}"/>
</target>

<target name="compile" depends="prepare">
  <echo>=== "compie" ===</echo>
  <javac srcdir="${src}" destdir="${temp.dir.classes}" debug="on">
    <classpath refid="build.class.path"></classpath>
  </javac>
  <!-- copydir src="${build.dir}\classes" dest="${temp.dir.classes}"/>-->
</target>

<target name="package" depends="compile">
  <echo>"PACKAGING THE FILES"</echo>
  <copy file="${meta-inf}\MANIFEST.MF" tofile="${temp.dir.meta-inf}\MANIFEST.MF"
  overwrite="true"/>
  <copy file="${web-inf}\web.xml" tofile="${temp.dir.web-inf}\web.xml"
  overwrite="true"/>
  <copy file="${web-inf}\tiles.xml" tofile="${temp.dir.web-inf}\tiles.xml"
  overwrite="true"/>
  <copy todir="${temp.dir.classes}">
    <fileset dir="${src}">
      <include name="**/*.xml"/>
    </fileset>
  </copy>

  <war destfile="${package.file}" webxml="${temp.dir.web-inf}\web.xml"
  basedir="${temp.dir}">
    <fileset dir="${jsp.dir.name}"/>
    <lib dir="${lib}"></lib>
    <classes dir="${temp.dir.classes}"></classes>
  </war>
</target>

<target name="jps">
  <copy todir="${webapp.dir}\${dir.name}">
    <fileset dir="${content}">
      <include name="**/*.jsp"/>
      <include name="**/*.html"/>
      <include name="**/*.css"/>
      <include name="**/*.gif"/>
      <include name="**/*.jpg"/>
      <include name="**/*.png"/>
      <include name="**/*.js"/>
    </fileset>
  </copy>
</target>

<target name="install" depends="package">
  <copy file="${package.file}" todir="${webapp.dir}" overwrite="true"/>
</target>

</project>

```

How can I compile a class file that does not have a main class?

How do I set the class path?

For reference, I am using Eclipse.

java ant war

edited Dec 6 '11 at 20:27



Beau Grantham

2,520 5 15 30

asked Dec 6 '11 at 13:26



Sarin Jacob Sunny

1,044 2 13 45

5 Answers

All your ant stuff will work fine except the javac task which needs the tools.jar, located in the /lib directory from the JDK, JRE is not sufficient in that case. Therefore the hint from ant : "Unable to find a javac compiler;..."

When working with Eclipse the default setting points to your JRE installation.

So, one of your first steps after starting Eclipse for the first time should be :

Window > Preferences > Installed JREs and change the settings from JRE to JDK.

Alternatively use :

Window > Preferences > Ant > Runtime > Classpath > Global Entries

and add the tools.jar from your JDK/lib folder

edited Feb 13 '13 at 19:23

answered Dec 6 '11 at 20:14

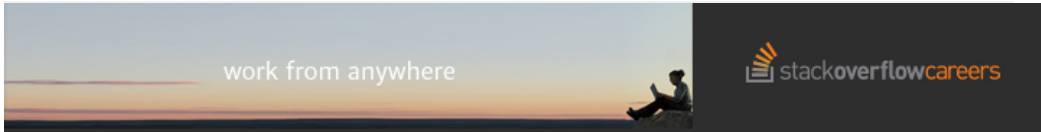


Rebse

6,104 1 14 29

The comp isn't lying ;) How do you start your ant script ?= – Rebse Jan 19 at 18:49

No, I forgot where I commented I was so flustered. I actually had the wrong JDK version in my path variable since the "clean install" required me remove outdated Java. Bit of a typo hunt but I wholly approve your answer now. – John Jan 19 at 19:31



Currently you have set JAVA_HOME to your installed jre instead of the jdk. Install the current JDK then set JAVA_HOME to your JDK Directory.

You'll find the current jdk here

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

edited Apr 17 '12 at 15:00

answered Dec 6 '11 at 13:52



sandrozbinden

523 4 17

I had the same problem. In my case, I fixed it by pointing JAVA_HOME to the JDK folder, rather than the JDK\bin folder.

answered Jan 1 '14 at 13:49



Umar Farooq Khawaja

2,500 15 33

I made the mistake of installing the JDK and the JRE to the same location. Installing the JDK automatically installs a JRE and I inadvertently selected the same location as the JDK.

This caused considerable confusion in eclipse.

Install the JDK and the JRE to *different* locations.

answered Mar 22 '13 at 18:01



rainecc

677 4 5

I had this problem and commented on <http://stackoverflow.com/a/8406221/866333> saying the fix was no good. (Actually it was reporting that JAVA_HOME was something else so I suggested a script was changing it).

As their reply stated the compiler doesn't lie.

So true, I eventually spotted my old JAVA_HOME variable was something like JDK_v7.0.7 when I'd just removed that (from a 2012 restore DVD) and updated to like JDK_v7.7.1. Why it saw fit to fallback to the JRE, suppose it was a convenience.

eg

```
JAVA_HOME = c:\Java\JDK_v7.0.7
```

@runs script...@

Perhaps JAVA_HOME does not point to the JDK. It is currently set to "C:\Program Files\Java\jre7"

@check at cmd prompt@

```
set  
JAVA_HOME = c:\Java\JDK_v7.0.7
```

so it is kinda lieing isn't it?

edited Feb 25 at 9:24

answered Jan 19 at 19:34



John

2,118

1

14

39

protected by [Community](#) ♦ Apr 26 '14 at 5:03

Thank you for your interest in this question. Because it has attracted low-quality answers, posting an answer now requires 10 [reputation](#) on this site.

Would you like to answer one of these [unanswered questions](#) instead?