



aicas GmbH

How to Compile Xlets using Apache Ant

Version 2.7.1dev
28 October 2016

Released under NDA

© 2016 aicas GmbH, Karlsruhe. All rights reserved.

Every effort has been made to ensure that all statements and information contained in this document are accurate. However, aicas GmbH accepts no liability for any error or omission therein.

This product includes software developed by IAIK of Graz University of Technology. This software is based in part on the work of the Independent JPEG Group.

Portions of this software are copyright ©2016 The FreeType Project (www.freetype.org). All rights reserved.

Java and all Java-based trademarks are registered trademarks of Oracle America, Inc. All other brands or product names are trademarks or registered trademarks of their respective holders.

The software included in this product contains copyrighted software that is licensed under the GNU General Public License (GPL) or GNU Lesser General Public License (LGPL). You may obtain the complete corresponding source code from us for a period of three years after our last shipment of this product. We will charge 30 EUR for the creation and shipment of a physical machine-readable copy of the source code.

Please contact us at the following address for payment instructions:

aicas GmbH
Haid-und-Neu-Straße 18
76131 Karlsruhe
Germany

Email: support@aicas.com

This offer is valid to anyone in receipt of this information.

1 Introduction

For automated builds it is often convenient to compile Xlets from the command line using an Apache Ant build script. This document shows how that goal can be achieved.

2 A Sample Apache Ant Script

Using Apache Ant for compiling Xlets is relatively straightforward. You only need to ensure that the Java compiler is using the correct (boot) class path containing only the classes that are part of the JamaicaCAR framework. JAR files containing exactly these classes are part of JamaicaCAR's SDK package.

The sample Apache Ant script in Figure 1 shows how to configure Apache Ant's *javac* task accordingly. It can be used with any JDK supporting Java 8.

```
<?xml version="1.0"?>
<project name="MyXlet" basedir="." default="jar">

  <!-- PLEASE SET THESE -->
  <!-- pointer to the SDK's target folder -->
  <property name="sdk.target.dir" value="/path/to/sdk/target/&lt;target>" />
  <!-- pointer to sources -->
  <property name="src" value="${basedir}/src" />
  <!-- pointer to resources -->
  <property name="res" value="${basedir}/res" />
  <!-- pointer to external libraries -->
  <property name="lib" value="${basedir}/lib" />

  <!-- properties that do not need to be changed -->
  <property name="bin" value="${basedir}/bin" />
  <property name="dist" value="${basedir}/dist" />

  <!-- class path containing JARs from the SDK and project-specific JARs -->
  <path id="project.class.path">
    <fileset dir="${sdk.target.dir}/lib">
      <include name="car-system-api.jar" />
      <include name="codename1.jar" />
      <include name="lwuit.jar" />
    </fileset>
    <fileset dir="${sdk.target.dir}/xletextensiondir">
      <include name="*.jar" />
    </fileset>
    <fileset dir="${lib}">
      <include name="*.jar" />
    </fileset>
  </path>

  <!-- boot class path containing JARs from the SDK -->
  <path id="project.boot.class.path">
    <fileset dir="${sdk.target.dir}/lib">
      <include name="rt.jar" />
      <include name="jce.jar" />
      <include name="java8compactprofile-addendum.jar" />
    </fileset>
  </path>

  <!-- create build directory -->
  <target name="init">
    <tstamp />
    <mkdir dir="${bin}" />
    <mkdir dir="${dist}" />
  </target>

  <!-- compile sources -->
  <target name="compile" depends="init">
    <javac includeantruntime="false"
      debug="true"
      srcdir="${src}"
      destdir="${bin}"
      bootclasspathref="project.boot.class.path"
      classpathref="project.class.path" />
  </target>

  <!-- build JAR file -->
  <target name="jar" depends="compile">
    <jar destfile="${dist}/${ant.project.name}.jar">
      <fileset dir="${bin}" />
      <fileset dir="${res}" />
    </jar>
  </target>

  <!-- clean -->
  <target name="clean">
    <delete dir="${bin}" />
    <delete dir="${dist}" />
  </target>

</project>
```

Figure 1: A sample Apache Ant script.