Installation

Table of contents

This section tells you what you need in addition to the JAppGen distribution, where to put it, how to set up your environment, and how to check that your installation is correct. For better security you might want to also consider running JAppGen under a different user ID.

1. Tools

1.1. Java and the JDK

JAppGen is being developed with Java 1.5.x, so a basic requirement is JDK 5.0 Update 6 with Java EE (or better). This can be downloaded from here.

1.2. Ant and JUnit

JAppGen itself and the `eg/` sample applications are built with Ant and tested using JUnit. Jars for both are bundled with the distribution.

Ant is the Java equivalent to make. We include an older version of Ant (1.5.4) with the distribution because the jars are smaller and recent changes are not necessary for what we do in JAppGen. If you prefer a more recent version, you can download it either in source or binary from the Ant Web site. In either case you need to set the ANT_HOME environmental variable to the directory that the Ant jars are in.

JUnit is the Java implementation of xUnit, the standard framework for unit testing. In the Java world Ant and JUnit are commonly used together. Both are free, open-source projects.

1.3. Tomcat

Tomcat is another Apache project, a J2EE servlet container. The most recent 5.5.* releases are servlet 2.4/JSP 2.0 compliant. We do not include Apache in the JAppGen distribution, but recommend that you install a binary version of Tomcat below $JAPPGEN_HOME, so that $CATALINA_HOME is for example $JAPPGEN_HOME/apache-tomcat-5.5.12. Having a separate copy of Tomcat simplifies testing (in that you don't need to worry about side-effects of testing) and may improve security.

1.4. DBMSs

We bundle the Hypersonic Java DBMS HSQLDB with the JAppGen distribution. This is a small, fast SQL engine that is particularly appropriate for unit testing, where databases can be created as a fixture, tested, and then destroyed at very little cost.

Some of our tests also assume the presence of MySQL. MySQL is one of the most-used database systems in the world and is open source and free for non-commmercial applications. Because it is not Java-based (and so is generally downloaded as a platform-specific binary) we do not bundle MySQL with the distribution, but we do strongly recommend that you get a copy and make it available to JAppGen.

At the time of writing, we intend to also test JAppGen with PostgreSQL, Oracle, and Sybase. PostgreSQL is free and open source. Both Oracle and Sybase offer free Linux versions to developers. Sybase's offer does not appear to be limited to developers.

2. Environmental Variables

JAppGen expects to have certain environmental variables set up and will not operate correctly if they aren't.

variable name	typical value	description
ANT_HOME	$JAPPGEN_HOME/lib/ant	Ant is the Java build utility; this points to the version of Ant bundled with JAppGen
CATALINA_HOME	$JAPPGEN_HOME/apache-tom	where X represents the version number of the current stable release
JAVA_HOME	/usr/local/lib/jdk1.5.0_06	wherever you have installed Java 5.0
JAPPGEN_HOME	$HOME/jappgen	wherever you unzip the JAppGen distribution to

You also need to change your path. The bash syntax is

```
        export PATH=$JAVA_HOME/bin:$ANT_HOME/bin:$JAPPGEN_HOME/bin:$PATH
```

3. lib/

The JAppGen binary distribution contains all necessary external jars in ${JAPPGEN_HOME}/lib. All of these are made available under some sort of license allowing use without payment. In order to make it easy to associate license and jar, the `lib/` directory is organized like so:

```
      +${JAPPGEN_HOME}
        +lib
          +ant
            ant-1.5.4.jar
            optional-1.5.4.jar
            LICENSE
```

That is, each group of jars is collected in a subdirectory whose name is a nickname for their supplier, the licensor. The applicable license is in the same subdirectory.

If you intend to use other external jars they should be added to `lib/` in the same way, so that

JAppGen will be able to locate them.

4. JAppGen Installation

In outline the installation process is:

- set up the environmental variables listed above
- download a JDK if you don't already have one and install it in ${JAVA_HOME}
- download Tomcat and install it in ${CATALINA_HOME}
- download and install the latest release of MySQL
- download the JAppGen binary release, `jappgen-bin-${VERSION}.zip`, and unzip it in the directory just above ${JAPPGEN_HOME}
- if you intend to modify the source code - or are just curious - you should also download the JAppGen source release, `jappgen-src-${VERSION}.zip`, and unzip it in the directory just above ${JAPPGEN_HOME}
- Edit ${JAPPGEN_HOME}/build.properties, following instructions in that file.
- Run the initial test to verify that the installation is correct

This will give you a working copy of JAppGen and the necessary supporting tools.

5. Initial Test

This is an Ant build target. After installation is complete, at the command line type

```
        cd ${JAPPGEN_HOME}
        ant initialTest
```

If the installation is correct, you will be told so. Otherwise, you will see a listing of problems detected. You will need to correct these problems before continuing.

6. Common Causes of Failures

component/tool	failure mode
Ant	not on path version before 1.5.4
JDK	not on path version before 1.5.*
Apache Tomcat	version before 5.5.* directory not writable
MySQL	not on path old version unknown user ID wrong password in config file

JAppGen	