

Developers

Table of contents

1 Overview.....	2
2 Getting Started.....	2
3 Specification Language.....	2
4 JAppGen Directory Structure.....	2
5 Application Directory Structure.....	3

1. Overview

2. Getting Started

You need to begin by reading the [Users Guide](#), acquiring and installing the necessary basic tools (a JDK, Tomcat, MySQL, and whatever other database systems you intend to use), setting up appropriate [environmental variables](#), and downloading and installing **both** the binary and source JAppGen distributions.

3. Specification Language

JAppGen applications are generated from a specification written in an extended version of [Relax NG](#), an XML schema language. There are two variants of Relax NG: a more verbose **XML syntax** and the more concise **compact syntax**. Tools for automatically converting between the two forms and between them and XML 1.0 DTDs and W3C XML Schema. [trang](#) is one such tool. It will convert any of these to any other, with the exception of XML Schema, which is output-only.

You can see the current version of the specification language by clicking on [jags](#) on the submenu on the left of this page. An older version is at [old-jags.html](#).

4. JAppGen Directory Structure

Note:

This description is accurate for UNIX/Linux platforms, where development is currently active. As testing on Windows progresses, the description will be updated.

JAppGen's directory structure is more or less conventional for an Ant-managed build.

```
+${JAPPGEN_HOME}
+eg                // described below
+lib
  +ant
    ant-1.5.4.jar
    LICENSE
    optional-1.5.4.jar
  +jappgen
    jappgen-${VERSION}.jar
    LICENSE
  ...
+jappgen
  build.properties // editable configuration file
  build.xml        // Ant script
```

```

+bin
  jappgen.sh           // generates applications
  jappgen.bat         // Windows version
+dist
  jappgen-bin-${VERSION}.zip // created by dist target
  jappgen-src-${VERSION}.zip
+src
  +java
    +org
      +jappgen
        **/*.java
  +test // unit tests, same dir struc as
src/java/
+target // created by compile target
  jappgen-${VERSION}.jar // created by jar target
+classes
  +org
    +jappgen
      **/*.class
+test-classes // compiled unit tests, same dir struc

```

The library directory `lib/` contains external jars organized by source organization together with applicable licenses.

The Ant script `build.xml` controls the build process. Source code `src/java/org/jappgen/**/*.java` is compiled into the corresponding place under `target/org/jappgen/**/*.class` (where `***` represents zero or more intervening subdirectories).

Unit tests in `src/test/org/jappgen/**/*.java` are similarly compiled into `target/org/jappgen/**/*.class`.

When tests are successful, a JAppGen jar file is created and written to `target/jappgen-${VERSION}.jar`. The jar file includes class files under `target/classes` but excludes unit test classes.

Periodically a release is created as two zip files.

The binary release is a subset of the full directory tree, including `bin/`, `eg/`, `lib/`, and documentation but excluding `src/` and the directories generated by JAppGen builds, `dist` and `target`. The jar file for the release is in `lib/` as `lib/jappgen/jappgen-${VERSION}.jar`.

The source release contains only the Java source code, `jappgen-${VERSION}/jappgen/src/` and the files below that.

Detarring the source distribution creates part of the directory structure above. Running the Ant script `build.xml` with an appropriate target creates most of the rest.

By default the `$JAPPGEN_HOME` directory is named `jappgen-${VERSION}`.

5. Application Directory Structure

The section describes the directory structure of an application generated by JAppGen. By default an application is created in a subdirectory of `$JAPPGEN_HOME/eg/`, but other target directories can be specified.

```
+etc
  ${APP_NAME}.jags           // application description
+${APP_NAME}
  build.properties         // editable configuration file
  build.xml
+bin
+dist                      // created by application's dist target
  ${APP_NAME}-bin-${VERSION}.zip
  ${APP_NAME}-src-${VERSION}.zip
+lib
  +ant
    ant-1.5.4.jar
    LICENSE
    optional-1.5.4.jar
    ...                      // subset of jars, those needed by app
+src
  +java
    +**                     // organization path, eg com/example/
      +db                   // Hibernate - persistence layer
        **/*.java
        **/*.hbm.xml       //   Hibernate mapping files
      +biz                   // Spring Framework: middle tier
        **/*.java
      +front                 // JavaServer Faces MVC
        +beans/**/*.*.java //   backing beans
        **/*.*.java        //
      +test                  // same dir struc as src/
+target                    // created by app's compile target
  jappgen-${VERSION}.jar
+classes
  +classes
  +test-classes           // unit tests, same dir struc as
+classes
  +web
    *.jsp                  // wrapper files for web/*.jsp
  +tiles
    *.jsp
  WEB-INF
    applicationContext.xml // Spring/Hibernate control file
    faces-config.xml       // JavaServer Faces control
    tiles.xml              // Struts Tiles control
    ${APP_NAME}.tld        // application-specific tile defs
    web.xml                 // application control file
```