JAppGen

Table of contents

1. Overview

JAppGen is a **Java Application Generator**, more specifically a tool for generating a J2EE application from a set of configuration files.

As we see it now, such an application has three primary layers:

- a presentation layer, what the user sees on the browser
- the application object domain, tree-structured sets of instances of the application's classes
- the relational model of persistent data, what is written to and retrieved from the disk

These are usually three very different ways of looking at the data. The other two layers bind the three together.

- the presentation/object model mapping, the **P/O mapping**, is responsible for mapping between internal and external representations of data, including supplying defaults and validating input
- the object/relational mapping, the **O/R mapping**, similarly handles transformations between the object model and the flat files of the relational DBMS model

As it is envisioned now, JAppGen will generate a directory structure and a set of files, mostly JSP, Java, and XML but also including Windows and UNIX/Linux command files.

2. Motivation

The two major motivating factors behind the JAppGen project are repetition fatigue on the one hand and the availability of very high quality free software tools on the other.

Anyone who has worked with J2EE applications is familiar with the repetition involved in coding up simple files. A quick spot check, for example, showed me that I just used a Bash script of about 40 lines to generate over 400 lines of Java -- and if you stripped out the comments and unnecessary line breaks in the script you were down to six lines of text generating six Java classes as 422 lines of code. True, the Bash script invokes a generously commented 250-line Perl script, but we only had to write that once.

Conservatively the Java-to-Bash ratio is over ten to one. However it would not be unreasonable to say that the Bash script says as much in 6 lines of code as Java does in 422: that's 70x as concise.

This is made worse by the other languages that are necessary. A typical project involves an Ant build file of several hundred lines, also highly repetitious; innumerable equally verbose JSP and Javascript files; and countless configuration files of various types.

The flip side is the number of very high quality tools available, including for example Apache Tomcat, Struts Tiles, JavaServer Faces, the Spring Framework, Hibernate, and various relational DBMSs. Most of these are available for nothing, at least for non-commercial use.

That is, they are free.

These tools make it possible to put together fairly complex applications very quickly and at reasonably low cost. However, they require a peculiar wide range of knowledge and skills to use, and the resulting applications are tedious to maintain, because information is scattered widely among those innumerable configuration files.

The primary objective of the JAppGen project is to sharply reduce the repetition in writing J2EE applications. We intend to do this by making it possible to generate an application's J2EE code and configuration files from a few JAppGen configuration files, sometimes only one.

3. The Tools

At this time we expect to build most of JAppGen applications using other open source tools, specifically:

- **Tiles**, a component of Apache Struts
- **Java Server Faces**
- the Apache **Spring Framework**
- **Hibernate**
- a number of **SQL** databases, including
    - the Hypersonic SQL database, **HSQLDB**
    - **MySQL**
    - **Oracle**
    - **PostgreSQL**
    - **Sybase**

At least initially JAppGen applications will be set up to run on **Apache Tomcat** (the latest stable 5.5.x release) using the most recent version of Sun's **Java 1.5**.

Where possible jars for dependencies will be included in the JAppGen non-commercial distribution. It will obviously not be possible for the Oracle and Sybase databases or the Sun SDKs. However we will provide information on obtaining Oracle's and Sybase's software on a no-fee basis as well as links to sites for downloading source code for the other dependencies.