



IMP Electronic Music Publishing System

Digital technologies have changed the balance of power in the music business. IDEA helps independent artists and labels capitalize on this by distributing their own music.

The Challenge

Integrating their application with the various digital services wasn't so easy, requiring substantial development and testing effort to provision a new service. Our charge was to create a system in which new services could be brought on line with minimal effort.

Publishing music to different services like iTunes and Napster involves similar steps, but with a fair amount of variation. Files must be encoded (*but in different formats*) and transferred (*but using different protocols*). Metadata must be transferred along with the music files (*but each service requires a differently formatted XML file*).

The Solution

We encapsulated this variation by creating a domain-specific language for e-music publishing. The framework handles common tasks such as error handling, workflow and provides primitive operations for file encoding, art generation, file transfer, and metadata creation. Most variability is driven parametrically, via Spring configuration files, rather than programatically, eliminating most of the code that would otherwise need to be written (and tested) to integrate with each service. Metadata is generated via an XML pipeline which includes XSLT processing. Additionally we used Hibernate (with annotations) and MySQL 5.0.

Results

Provisioning a new service is a matter of stringing together a set of already written and tested tasks via a Spring configuration file. A new service can often be brought online in less than a day.

```
<bean id="connect.Packager" class="com.idea.imp.service.common.DispatchingTask"
  <constructor-arg><ref bean="applicationSettingsBean"/></constructor-arg>
  <constructor-arg><list>
    <bean class="com.idea.imp.service.connect.ContextPreparer">
      <property name="stopProcessingOnFailure">
        <value>true</value>
      </property>
    </bean>
    <bean class="com.idea.imp.service.packager.FileSystemStructureBuilder">
      <property name="stopProcessingOnFailure">
        <value>true</value>
      </property>
    </bean>
    <bean class="com.idea.imp.service.packager.TemporaryFileDecoder">
      <constructor-arg><ref bean="flacDecoder"/></constructor-arg>
      <property name="stopProcessingOnFailure">
        <value>true</value>
      </property>
    </bean>

    <bean class="com.idea.imp.service.packager.TrackTransformerTask">
      <constructor-arg><ref bean="wmaEncoder"/></constructor-arg>
      <constructor-arg><value>-silent -input %in% -a_setting 192_44_2</value></constructor-arg>
      <constructor-arg><value>-192.wma</value></constructor-arg>
    </bean>

    <bean class="com.idea.imp.service.packager.TrackTransformerTask">
      <constructor-arg><ref bean="flacEncoder"/></constructor-arg>
      <constructor-arg><value>-132.at3</value></constructor-arg>
    </bean>
  </list>
</bean>
```