

# Facade Design

by Paul Hammant

## 1. Introduction

You can choose to generate your proxies in advance of use, or to defer generation until runtime. As the generation requires javac in tools.jar (multi megabyte), you have to decide whether your runtime environment or distribution can support or legally distribute it.

## 2. Generation in advance

If you are using Ant for builds, then you can use the 'altrmiproxies' taskdef to make proxies in advance of use.

```
<altrmiproxies genname="Hello" srcgendir="${build.home}/genjava"
  classgendir="${build.home}/classes" verbose="true"
  interfaces="org.apache.altrmi.test.TestInterface"
  additionalfacades="org.apache.altrmi.test.TestInterface2">
  <classpath>
    <pathelement location="${build.home}/classes"/>
  </classpath>
</altrmiproxies>
```

All you have to do after that is place them in a jar for later use. That jar could be client or server-side, thus classes can be retrieved from the server for use on the client or preexist in clients environment.

## 3. Generation at runtime

If you are using Ant for builds, then you can use the 'altrmiproxies' taskdef to make proxies in advance of use.

```
DynamicGeneratorClassRetriever dgcr = new DynamicGeneratorClassRetriever(mGenJarURL);
PublicationDescription pd = new PublicationDescription(TestInterface.class, new Class[]
dgcr.generate("Hello", pd, this.getClass().getClassLoader());
```

This by implication means server-side classes, so the above is code in the server. It also has a performance hit at time of first retrieval.

Copyright (c) @year@ The Apache Incubator Project. All rights reserved. \$Revision: 1.2 \$ \$Date: 2003/02/16 21:41:35 \$