

# Other Features

by Paul Hammant

## 1. Features

There are various features of AltRMI that are either different to RMI or are configurable.

### 1.1. Choice of location of generated Proxy class.

Classes providing client side implementation of the transported interface(s) can be either on the client side or the server side (and duly transported) at the time of lookup. On the server side, the proxy classes can be generated in advance (Ant Task) or at runtime - javac is invoked.

### 1.2. Choice of castability of generated proxy class.

To suit remote facilities that are happy with reflection and do not need to cast to an interface to use a bean (I am thinking of BeanShell) the proxy class can be generated without specifying that it implements the interface(s).

### 1.3. Suspendable/Resumable service.

The Server supports suspend() and resume(). With the current implementation this replies in a timely fashion to the client that the client should try later. The client waits for the notified amount of time and seamlessly tries the request again. A server could cycle through suspended and back to resumed will not affect the client except for the a delay.

### 1.4. Recovering transport

AltRMI tries to recover its transports should they fail. The recovery is pluggable in that the developer can choose when/how/if the connection handler tries to recover the connection. Any inprogress, but disconnected method invocation will attempt to be recoved and just return as normal, albeit after a longer than normal delay.

## **1.5. Event API**

For suspensions, abnormal ends of connection etc, there is a listener that can be set that will allow actions to be taken. Abnormally terminated connections will by default try to be reconnected, the listener can decide if, how many, and how often the retries occur.

## **1.6. Pluggable Transport independent keep-alive concept.**

On the client side, there is an API for a pluggable pinger. While the server may support timeouts for client connections, the client could choose to overcome that default policy with a ping concept.

## **1.7. Unpublishable and republishable API**

The server is able to unpublish a service. In conjunction with suspend()/resume() a service can be republished, upgraded etc whilst in use, or just offlined.

## **1.8. Startable API for Server**

The server implements and acts upon start() and stop() methods.

## **1.9. Not just pass by value.**

AltRMI started life as 'pass by value' only. It now supports return types and parameters wrapped in another AltRMI Facade.

## **1.10. No duplicate instances.**

For Facades, if you call `Person p = getPerson("Fred")` twice you will get the same instance on the client side if it is the same instance on the server side.

## **1.11. Load balancing server grouping**

In some configurations, the client side can have multiple servers it will forward calls to (TODO).

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