

RMI Tools

by SuperBonBon

1. How can UPNP# help with RMI ?

Imagine a remote object stub listening on port 9999 and a remote registry listening on port 1099. All the objects are hosted on a computer behind a nice NAT router. If you want to expose these object to the WAN probably behind the router, you'll have to open the ports of the router. When you are lucky, the object listening on port 9999 is set to always listen on this port, if you aren't, you'll probably have to make a netstat everytime you restart your "closed source unconfigurable" application to open the good port on the router/firewall. UPNP# IGD devices can help to do this job automatically.

2. What are the UPNP# integration costs ?

Well basically you'll need to do only one modification to your code : Instead of extending the *java.rmi.server.UnicastRemoteObject* for your remote object implementation class you'll have to extend the *net.sbbi.upnp.remote.UnicastRemoteObject* class. Here is an example :

```
package net.sbbi.upnp.samples;

import java.net.InetAddress;
import java.rmi.RemoteException;

import net.sbbi.upnp.remote.UnicastRemoteObject;

public class HelloWorld extends UnicastRemoteObject
    implements HelloWorldInterface {

    public HelloWorld() throws RemoteException {
    }

    public String say( String myName ) throws RemoteException {
        String hostName = "localhost";
        try {
            hostName = InetAddress.getLocalHost().getHostName();
        } catch ( Exception ex ) {
        }
        return "Hello world to " + myName + " from computer " + hostName;
    }
}
```

Finally to make everything work you'll need an UPNP# IGD device on your remote object network. Every remote object creation will automatically open the port on the router and create a shutdown hook to close the port during a JVM shutdown. If you decide to destroy a remote object you can call the `net.sbbi.upnp.remote.UnicastRemoteObject.closePort()` method on your remote object to explicitly close the server port.

3. Samples

A small working sample is provided with the *HelloWorldClientSample* and *HelloWorldServerSample* scripts in the distribution bin directory.