

INSTRUCTIONS FOR RUNNING

In these .docx i'll give you some detail advices how to run this client-server application.

I. First we need to start the server (remote service), for what:

1. Generate the stubs (client helper) and skeletons (server helper) using `rmic` - these are the client and server "helpers". You don't have to create these classes or ever look at the source code that generates them. It's all handled automatically when you run the `rmic` tool (that ships with your JDK) in terminal.

Note that `rmic` should run with .class file, not with .java, but without .class extension like you'll see below:

```
$ rmic com.company.ServiceServerImpl
Warning: generation and use of skeletons and static stubs for JRMP
is deprecated. Skeletons are unnecessary, and static stubs have
been superseded by dynamically generated stubs. Users are
encouraged to migrate away from using rmic to generate skeletons and static
stubs. See the documentation for java.rmi.server.UnicastRemoteObject.
```

In my case «com.company» is the name of the package.

2. Start the RMI registry in terminal. The `rmiregistry` is like the white pages of a phone book. It's where the user goes to get the proxy (the client

```
$ rmiregistry
```

stub/helper object). The `rmiregistry` should work in the background.

3. Open next one terminal and start the remote service (`ServiceServerImpl.java`). The remote service should work in the background.

II. And now we'll start the client (`ServiceBrowser.java`) and having fun.