Java RMI

An Overview

Joseph Kehoe[1]

[1]Department of Computing and Networking
Institute of Technology Carlow

CDD101, 2018

Introduction

- RMI is Remote Method Invocation
- RPC for Objects
- Integrated into the Java Language
- Easy to use
- allows code to call objects in another JVM
- Runs over TCP/IP

Institiúid Teicneolaíochta Cheatharlach
INSTITUTE of TECHNOLOGY CARLOW
At the Heart of South Leinster

Java RMI vs. Corba

- Integrated directly into Java Language
- Easier to use than Corba
- Less overhead (marshalling and unmarshalling etc.)
- Corba allows different languages to talk to each other transparently
- RMI requires JVM on server - Corba is platform independent
- Corba is more powerful (services such as object migration)

Institiúid Teicneolaíochta Cheatharlach
INSTITUTE of TECHNOLOGY CARLOW
At the Heart of South Leinster

RMI-IIOP

- RMI over IIOP (Internet Inter-Orb Protocol)
- Allows Java RMI to access Corba using the Java Remote Method Protocol (JRMP) as the transport
- Simplifies access to Corba but still gets all the benefits
- Reduces complexity and footprint of Corba

Institiúid Teicneolaíochta Cheatharlach
INSTITUTE of TECHNOLOGY CARLOW
At the Heart of South Leinster

Sample Interface Code

Example

```
import java.rmi.Remote;
import java.rmi.RemoteException;
public interface RmiServerIntf extends Remote {
public String getMessage() throws RemoteException;
}
```

Institiúid Teicneolaíochta Cheatharlach
INSTITUTE of TECHNOLOGY CARLOW
At the Heart of South Leinster

Sample Client Code

Example

```
import java.rmi.Naming;
public class RmiClient {
public static void main(String args[]) throws Exception {
RmiServerIntf obj =
(RmiServerIntf)Naming.lookup("//localhost/RmiServer");
System.out.println(obj.getMessage());
}
}
```

Institiúid Teicneolaíochta Cheatharlach
INSTITUTE of TECHNOLOGY CARLOW
At the Heart of South Leinster

Sample Server Code

Example

```
import java.rmi.Naming;
import java.rmi.RemoteException;
import java.rmi.server.UnicastRemoteObject;
import java.rmi.registry.*;
public class RmiServer extends UnicastRemoteObject implements
RmiServerIntf {
public static final String MESSAGE = "Hello World";
public RmiServer() throws RemoteException {
super(0); // required to avoid the 'rmic' step, see below
}
public String getMessage() {
return MESSAGE;
}
```

Sample Server Code

Example

```
public static void main(String args[]) throws Exception {
System.out.println("RMI server started");
try { //special exception handler for registry creation
LocateRegistry.createRegistry(1099);
System.out.println("java RMI registry created.");
} catch (RemoteException e) { //do nothing, error means registry already
exists
System.out.println("java RMI registry already exists.");
}
//Instantiate RmiServer
RmiServer obj = new RmiServer();
// Bind this object instance to the name "RmiServer"
Naming.rebind("//localhost/RmiServer", obj);
System.out.println("PeerServer bound in registry");
} }
```

Sample Server Code

Try the example here!