

Java Remote Method Invocation (RMI) Lab

Java RMI is a Java API which allows remote method invocation between java processes running on different jvms. This is an example of the Object-Oriented Model of Inter-Process Communication. RMI is the object oriented equivalent of RPC (Remote Procedure Call). RMI only works between Java processes. CORBA (which we will look at later) allows Object-Oriented IPC between processes which are not written in the same language.

More details on RMI: <https://docs.oracle.com/javase/tutorial/rmi/index.html>

General instructions:

First download dsJavaRMI_Lab.zip from Moodle, and unzip the folder. Within the dsJavaRMI_Lab folder you should have 2 subfolders, beginning with the letters a and b. The instructions to compile and run the code within each subfolder are listed under the headings below.

Part A instructions:

1. Open and inspect the .java files in ../dsJavaRMI_Lab/a_rmiMessageServiceLab
2. Open a command prompt window and change directory to
../dsJavaRMI_Lab/a_rmiMessageServiceLab
3. Run javac gmit/*.java to compile the .java files within the gmit package
4. Run java gmit.MessageServer; the server program will create an instance of MessageServiceImpl, and bind the instance to the name howdayService
5. Run java gmit.MessageClient in a new command prompt window to start the client; the client remotely invokes the getMessage() method on the instance of the MessageServiceImpl that was created by the MessageServer
6. Close all command prompt windows before starting part b

Part B instructions:

1. Open and inspect the .java files in ../dsJavaRMI_Lab/b_rmiMessageServerPassByReference
2. Open a command prompt window and change directory to
../dsJavaRMI_Lab/b_rmiMessageServerPassByReference
3. Run javac gmit/*.java to compile the .java files within the gmit package
4. Run java gmit.MessageServer; the server program will create an instance of MessageServiceImpl, and bind the instance to the name howdayService
5. Run java gmit.MessageClient in a new command prompt window to start the client; to start the client. This time the client remotely invokes the getMessage() method on a reference to an instance of RemoteMessageImpl on the server.