

## Java IDL: The "Hello World" Example on Two Machines

To enable the Hello World Tutorial to run on two machines, follow the steps as directed in the [tutorial](#), with the following changes. This tutorial was written for the Java (tm) 2 Platform, Standard Edition (J2Se(tm)), version 1.4. In this example, the client, stubs, and skeletons are located on the client machine, and the server and name server are located on the server machine. This scenario can be changed to meet your needs and is provided simply as an introduction to one way to distribute an application across two machines.

1. Create (as shown in the tutorial) and compile the `Hello.idl` file on the client machine:

```
idlj -fall Hello.idl
```

2. Create `HelloClient.java` on the **client** machine. Compile the `*.java` files, including the stubs and skeletons (which are in the directory `HelloApp`):

```
javac *.java HelloApp/*.java
```

3. Create `HelloServer.java` on the **server** machine. Compile the `.java` files:

```
javac *.java
```

4. Start the Java Object Request Broker Daemon, `orbd`, which includes a Naming Service, on the server machine. To do this on Unix:

```
orbd -ORBInitialPort 1050 -ORBInitialHost servermachinename&
```

To do this on Windows:

```
start orbd -ORBInitialPort 1050 -ORBInitialHost servermachinename
```

Both `-ORBInitialPort` and `-ORBInitialHost` are required arguments on the `orbd` command line. This example starts the name server on port 1050, because on Solaris you must become root to start a process on a port under 1024. If you want to use a different *nameserverport*, replace 1050 with the correct port number throughout this example.

When using `orbd`, the ORBD must be run on the same machine as the machine on which the servers will be activated. Another Naming Service, `tnameserv`, could be used if you prefer to run the Naming Service on a machine other than the machine on which the servers will be activated.

5. On the server machine, start the Hello server, as follows:

```
java HelloServer -ORBInitialPort 1050
```

If you used a different *nameserverport*, replace 1050 with the correct port number. You do not need to specify the `-ORBInitialHost` argument because the Hello server will be running on the same host as the name server in this example. If the Name Server were running on a different machine, you would specify which machine using the `-ORBInitialHost nameserverhost` argument.

6. On the client machine, run the Hello application client. From a DOS prompt or shell, type:

```
java HelloClient -ORBInitialHost nameserverhost -ORBInitialPort 1050
```

Note that *nameserverhost* is the host on which the IDL name server is running. In this case, it is the server machine.

If you used a different *nameserverport*, replace 1050 with the correct port number.

7. Kill or stop `orbd` when finished. The name server will continue to wait for invocations until it is explicitly stopped.