**Define the CORBA Interface (IDL)**

Create a file called StringReverser.idl to define the interface for the string-reversing service.

idl

CopyEdit

// StringReverser.idl

module StringReverserApp {

interface StringReverser {

string reverse\_string(in string input);

};

};

Create the StringReverserServer.java file to implement the CORBA server that provides the string-reversing service.

java

CopyEdit

import StringReverserApp.\*;

import org.omg.CORBA.\*;

import org.omg.PortableServer.\*;

public class StringReverserServer {

public static void main(String[] args) {

try {

// Initialize the ORB

ORB orb = ORB.init(args, null);

// Create and initialize the root POA

POA rootPOA = POAHelper.narrow(orb.resolve\_initial\_references("RootPOA"));

rootPOA.the\_POAManager().activate();

// Create the string reverser servant

StringReverserImpl stringReverser = new StringReverserImpl();

// Activate the servant

stringReverser.setORB(orb);

org.omg.CORBA.Object ref = rootPOA.servant\_to\_reference(stringReverser);

StringReverser href = StringReverserHelper.narrow(ref);

// Bind the object reference in the naming service

org.omg.CORBA.Object objRef = orb.resolve\_initial\_references("NameService");

NamingContextExt ncRef = NamingContextExtHelper.narrow(objRef);

// NameService Binding

NameComponent[] path = ncRef.to\_name("StringReverser");

ncRef.rebind(path, href);

System.out.println("StringReverser Server is ready...");

// Wait for invocations from clients

orb.run();

} catch (Exception e) {

e.printStackTrace();

}

}

}

class StringReverserImpl extends StringReverserPOA {

private ORB orb;

// Set the ORB to use in the servant

public void setORB(ORB orb) {

this.orb = orb;

}

// Implement the reverse\_string method

public String reverse\_string(String input) {

StringBuilder reversed = new StringBuilder(input);

return reversed.reverse().toString();

}

}

**4. Implement the Client**

Create the StringReverserClient.java file to implement the CORBA client that will invoke the string-reversing method on the server.

java

CopyEdit

import StringReverserApp.\*;

import org.omg.CORBA.\*;

import org.omg.CosNaming.\*;

import org.omg.CosNaming.NamingContextPackage.\*;

public class StringReverserClient {

public static void main(String[] args) {

try {

// Initialize the ORB

ORB orb = ORB.init(args, null);

// Resolve the object reference in the naming service

org.omg.CORBA.Object objRef = orb.resolve\_initial\_references("NameService");

NamingContextExt ncRef = NamingContextExtHelper.narrow(objRef);

// Resolve the name of the StringReverser object

NameComponent[] path = ncRef.to\_name("StringReverser");

StringReverser stringReverser = StringReverserHelper.narrow(ncRef.resolve(path));

// Call the reverse\_string method

String inputString = "Hello, CORBA!";

String reversedString = stringReverser.reverse\_string(inputString);

System.out.println("Original String: " + inputString);

System.out.println("Reversed String: " + reversedString);

} catch (Exception e) {

e.printStackTrace();

}

}

Terminal 1:

idlj -fall ReverseModule.idl

javac \*.java ReverseModule/\*.java

orbd -ORBInitialPort 1056&

java ReverseServer -ORBInitialPort 1056&

Terminal 2:

java ReverseClient -ORBInitialPort 1056 -ORBInitialHost localhost