

Tutorial 4 - COS Naming CORBA

1 Tasks

1. Download the skeleton code
2. Compile the sources in the usual way, starting with the IDL file
3. We need to start the name server, such that the server can register with it and the client can retrieve object references from it

```
$ tnameserv -ORBInitialPort 1050 &
```

You may need to modify the port if 1050 is bound on your machine already.

4. Run the server

```
$ java HelloNamingServer -ORBInitialPort 1050
```

5. Run the client in another terminal

```
$ java HelloNamingClient -ORBInitialPort 1050
```

6. Check the outputs of the server and the client
7. Modify the server to add a new Context called “Hello2” to the root node
8. Add a new Object called “Object2” to this new Context and bind it to the HelloServant object as well
9. Modify the client to find this new object in the name space, invoke the object and verify output on the screen, you can modify the code below to access the objects from the remote name space, this works by resolving the references using the *NameComponent* instances.

```
org.omg.CORBA.Object objRef = orb.  
    ↪ resolve_initial_references("NameService");  
NamingContext rootCtx = NamingContextHelper.narrow(objRef);  
nc[0] = new NameComponent("Hello", "Context");  
nc[1] = new NameComponent("World", "Object");
```

```
//NameComponent path[] = {nc};  
org.omg.CORBA.Object objRefHello = rootCtx.resolve(nc);  
Hello helloRef = HelloHelper.narrow(objRefHello);  
  
String hello = helloRef.sayHello() ;  
System.out.println(hello);
```

10. Start the Name Server on port 49000. Update the Client and the Server to make them aware of this port number, in this case we will do this in a different way than before

```
Properties props = new Properties();  
props.put("org.omg.CORBA.ORBInitialPort", "49000");  
ORB orb = ORB.init(args, props);
```

As such we do not need to pass command line arguments if we set it through properties.