

Creating an Activatable Object

```
import java.rmi.*;
import java.rmi.activation.*;

public class HelloImpl extends Activatable
    implements Hello {

    public HelloImpl(ActivationID id, MarshalledObject data)
        throws RemoteException
    {
        super(id, 0);
    }

    public String sayHello() throws RemoteException
    {
        return "Hello";
    }

} //HelloImpl
```

The "setup" class

The job of the "setup" class is to create all the information necessary for the activatable class, without necessarily creating an instance of the remote object

The setup class passes the information about the activatable class to `rmid`, registers a remote reference (an instance of the activatable class's stub class) and an identifier (name) with the `rmiregistry`, and then the setup class may exit

```
import java.rmi.*;
import java.rmi.activation.*;
import java.util.Properties;

public class Setup {

    public static void main(String[] args) throws Exception {

        System.setSecurityManager(new RMISecurityManager());
        Properties props = new Properties();
        props.put("java.security.policy", "/home/rmi_tutorial/activation/policy");

        ActivationGroupDesc.CommandEnvironment ace = null;
        ActivationGroupDesc exampleGroup = new ActivationGroupDesc(props, ace);
        ActivationGroupID agi = ActivationGroup.getSystem().registerGroup(exampleGroup);
```

```
String location = "file:/home/rmi_tutorial/activation/";

MarshaledObject data = null;
ActivationDesc desc = new ActivationDesc (agi,
    "examples.activation.ActivableImplementation",
    location, data);

Hello mri = (MyRemoteInterface)Activatable.register(desc);

Naming.rebind("HelloService", mri);
System.exit(0);
} //main
} //Setup

% rmid -J-Djava.security.policy=rmid.policy &

% java -Djava.security.policy=/home/rmi_tutorial/activation/policy
-Djava.rmi.server.codebase=file:/home/rmi_tutorial/activation/
SetupClass
```

Activating an object

The implementation class

```
import java.rmi.*;
import java.rmi.activation.*;

public class HelloImpl implements Hello {

    public HelloImpl(ActivationID id, MarshalledObject data) throws RemoteException {
        Activatable.exportObject(this, id, 0);
    }

    public String sayHello() throws RemoteException {...}
}
```