1. 最简单的RMI例子

服务器端：

IRMIImpl impl = new IRMIImpl();

Naming.rebind("rmi://210.43.109.25:1111/mytask", impl);

客户端：

Context namingContext=**new** InitialContext();

HelloService service=

(HelloService)namingContext.lookup(url+"HelloService");

Service.xxxxxxx()方法

然后就可以调用服务器端的方法了

注释：rmi框架屏蔽了底层的网络访问细节 具体的实现可以参考文件夹中的rmi实现原理代码

1. 如果客户端将自身的引用作为rmi的的参数----在上面的xxxxxxx(object)方法中，在rmi的服务端就可以调用客户端的方法
2. Rmi服务器端的延迟加载：

(3-1)实现Activatable接口

例：public HelloServiceImpl(ActivationID id, MarshalledObject data)

throws RemoteException {

super(id, 0);-----🡪>>>调用父类的方法

try {

//todo

} catch (Exception e) {

e.printStackTrace();

}

System.out.println("创建" + name);

}

(3-2)在服务端注册的时候的实现：

ActivationGroupDesc group = **new** ActivationGroupDesc(prop, **null**);

// 注册ActivationGroup

ActivationGroupID id = ActivationGroup.*getSystem*().registerGroup(

group);

String classURL = System.*getProperty*("java.rmi.server.codebase");

MarshalledObject param1 = **new** MarshalledObject("service1");

MarshalledObject param2 = **new** MarshalledObject("service2");

ActivationDesc desc1 = **new** ActivationDesc(id,

"activate.HelloServiceImpl", classURL, param1);

ActivationDesc desc2 = **new** ActivationDesc(id,

"activate.HelloServiceImpl", classURL, param2);

// 向rmid注册两个激活对象

HelloService s1 = (HelloService) Activatable.*register*(desc1);

HelloService s2 = (HelloService) Activatable.*register*(desc2);

System.*out*.println(s1.getClass().getName());

------------🡪>>从这向是重点

Context namingContext = **new** InitialContext();

// 向rmiregistry注册两个激活对象

namingContext.rebind("rmi:HelloService1", s1);

namingContext.rebind("rmi:HelloService2", s2);

1. 安全管理，什么样的方法开放

、

Properties prop = **new** Properties();

prop.put("java.security.policy",

SimpleClient.**class**.getResource("server.policy").toString());

(5)ejb具有并发性，只要编写的骨架是线程安全的