

J2EE：用RMI和CORBA进行分布式Java编程 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/289/2021\\_2022\\_J2EE\\_EF\\_BC\\_9A\\_E7\\_94\\_A8\\_c104\\_289952.htm](https://www.100test.com/kao_ti2020/289/2021_2022_J2EE_EF_BC_9A_E7_94_A8_c104_289952.htm) Java远程方法调用（RMI）机制和公用对象请求代理体系（CORBA）是最重要和使用最广泛的两种分布式对象系统。每个系统都有其特点和短处。它们在行业中被用于从电子交易到保健医疗的各个领域。一个项目如果要从这两种分布式机制中选用一个，往往难以抉择。本文概括地介绍了RMI和CORBA，更重要的是，它将介绍如何开发一个有用的应用程序，用于从远程主机下载文件。然后它将：  
简要介绍分布式对象系统 简要介绍RMI和CORBA 让你对在RMI和CORBA中开发应用程序所涉及的工作有个初步印象 演示如何使用RMI和CORBA，从远程主机传送文件 对RMI和CORBA进行简单比较 客户机/服务器模型 客户机/服务器模型是分布式计算的一种形式，在这种形式中，一个程序（客户机）与另一个程序（服务器）通讯以便交换信息。在这种模型中，客户机和服务器通常都说同样的语言 - - 也就是说客户机和服务器能理解同一个协议 - - 这样它们才能通讯。虽然客户机/服务器模型的实现方式多种多样，但典型做法是使用底层套接字。使用套接字开发客户机/服务器系统意味着，我们必须设计一个协议，也就是客户机和服务器都认识的一组命令集，通过这些命令它们就能通讯了。举例来说，HTTP协议中提供了一个名为GET的方法，所有Web服务器都必须实现这个方法，所有Web客户机（浏览器）都必须使用这个方法，才能获取文档。分布式对象模型 基于分布式对象的系统是一组对象的集合，这些对象以一种明确定义封

装的接口把服务的请求者（客户机）和服务的提供者（服务器）分隔开。换言之，客户机从服务的实现中分离出来，变成数据的呈现和可执行代码。这就是基于分布式对象的模型与纯粹的客户机/服务器模型的主要区别之一。在基于分布式对象的模型中，客户机向对象发送消息，然后对象解释该消息以便决定要执行什么服务。这项服务，也就是方法，可以选择是让对象还是让代理来执行。Java远程方法调用（RMI）和公用对象请求代理体系（CORBA）就是这种模型的例子。

RMI RMI是一个分布式对象系统，它使你能够轻松地开发出分布式Java应用程序。在RMI中开发分布式应用程序比用套接字开发要简单，因为不需要做设计协议这种很容易出错的工作。在RMI中，开发者会有一种错觉，似乎是从本地类文件调用的本地方法，其实参数传送给远程目标，目标解释参数后再把结果发回给调用方。

RMI应用程序初步使用RMI开发分布式应用程序包括以下步骤：定义一个远程接口 实现这个远程接口 开发服务器 开发客户机 生成存根和基干，启动RMI注册表、服务器和客户机 下面我们将通过开发一个文件传输程序来实践这些步骤。范例：文件传输程序 这个应用程序允许客户机从远程主机上传送（即下载）任何类型的文件（纯文本或二进制文件）。第一步是定义一个远程接口，这个接口规定了服务器所提供方法的信号，客户机将调用这些方法。定义一个远程接口用于文件下载应用程序的远程接口如代码范例1所示。接口FileInterface提供了一个方法downloadFile，这个方法接受String参数（文件名），将文件的数据以字节数组的形式返回。代码范例1 1:

```
FileInterface.java import java.rmi.Remote. import
```

java.rmi.RemoteException. public interface FileInterface extends Remote { public byte[] downloadFile(String fileName) throws RemoteException. } 请注意FileInterface的以下特征: 它必须声明为public, 这样客户机才能加载实现远程接口的远程对象。它必须扩展为Remote接口, 以满足使该对象成为远程对象的要求。这个接口中的每种方法都必须投出一个java.rmi.RemoteException。 100Test 下载频道开通, 各类考试题目直接下载。详细请访问 [www.100test.com](http://www.100test.com)