

RMI规范--第二章Java分布式对象模型 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/145/2021_2022_RMI_E8_A7_84_E8_8C_83-_c104_145024.htm 2.1 分布式对象应用程序 RMI 应用程序通常包括两个独立的程序：服务器程序和客户机程序。典型的服务器应用程序将创建多个远程对象，使这些远程对象能够被引用，然后等待客户机调用那些远程对象上的方法。而典型的客户机程序则从服务器中得到一个或多个远程对象的引用，然后调用远程对象的方法。RMI 为服务器和客户机进行通讯和信息传递提供了一种机制。这样的应用程序有时被称为分布式对象应用程序。分布式对象应用程序需要：定位远程对象 应用程序可使用两种机制中的一种得到对远程对象的引用。它既可用 RMI 的简单命名工具 `rmiregistry` 来注册它的远程对象；也可将远程对象引用作为常规操作的一部分来进行传递和返回。与远程对象通讯 远程对象间通讯的细节由 RMI 处理；对于程序员来说，远程通讯看起来就象标准的 Java 方法调用。给作为参数或返回值传递的对象加载类字节码因为 RMI 允许调用程序将纯 Java 对象传给远程对象，所以 RMI 将提供必要的机制，既可以加载对象的代码又可以传输对象的数据。服务器调用注册服务程序以使名字与远程对象相关联。客户机在服务器注册服务程序中用远程对象的名字查找该远程对象，然后调用它的方法。RMI 能用 Java 系统支持的任何 URL 协议（例如 HTTP、FTP、file 等）加载类字节码。2.2 术语的定义 在 Java 分布式对象模型中，`remote object` 是这样一种对象：它的方法可以从其它 Java 虚拟机（可能不同的主机上）中调用。该类型的对象由一种或多种

remote interfaces (它是声明远程对象方法的 Java 接口) 描述。远程方法调用 (RMI) 就是调用远程对象上远程接口的方法的动作。更为重要的是, 远程对象的方法调用与本地对象的方法调用语法相同。

2.3 分布式和非分布式模型比较

Java 分布式对象模型在以下几方面与 Java 对象模型相似: 远程对象的引用在任一种方法调用中(本地或远程)都能以参数形式传递或以结果形式返回。远程对象可以被强制转换成任何远程界面, 只要该界面为使用内置 Java 语法进行强制类型转换的实现所支持。内置 Java 操作符 instanceof 可用来测试远程对象所支持的远程接口。Java 分布式对象模型在以下几方面与 Java 对象模型不同: 远程对象的客户机与远程接口发生交互, 而从不与这些接口的实现类交互。远程方法的非远程参数和返回结果是通过复制而非引用的方式传递的。这是因为对象的引用只在单个虚拟机中才有用。远程对象以引用的方式进行传递, 而不是复制实际的远程实现。某些 java.lang.Object 类定义的方法的语义专用于远程对象。因为调用远程对象的失败模式本来就比调用本地对象的失败模式复杂, 所以客户机必须处理远程方法调用期间发生的额外异常。

2.4 RMI 接口和类概述

2.4.1 java.rmi.Remote 接口

在 RMI 中, 远程接口是声明了可从远程 Java 虚拟机中调用的方法集。远程接口必须满足下列要求: 远程接口至少必须直接或间接扩展 java.rmi.Remote 接口。远程接口中的方法声明必须满足下列远程方法声明的要求: 远程方法声明在其 throws 子句中除了要包含与应用程序有关的异常(注意与应用程序有关的异常无需扩展 java.rmi.RemoteException)之外, 还必须包括 java.rmi.RemoteException 异常(或它的超类, 例如

java.io.IOException 或 java.lang.Exception)。远程方法声明中，作为参数或返回值声明的（在参数表中直接声明或嵌入到参数的非远程对象中）远程对象必须声明为远程接口，而非该接口的实现类。java.rmi.Remote 接口是一个不定义方法的标记接口：`public interface Remote` 远程接口必须至少扩展 `java.rmi.Remote` 接口（或其它扩展 `java.rmi.Remote` 的远程接口）。然而，远程接口在下列情况中可以扩展非远程接口：远程接口也可扩展其它非远程接口，只要被扩展接口的所有方法（如果有）满足远程方法声明的要求。例如，下面的接口 `BankAccount` 即为访问银行帐户定义了一个远程接口。它包含往帐户存款、使帐户收支平衡和从帐户取款的远程方法：
`public interface BankAccount extends java.rmi.Remote { public void deposit(float amount) throws java.rmi.RemoteException. public void withdraw(float amount) throws OverdrawnException, java.rmi.RemoteException. public float getBalance() throws java.rmi.RemoteException. }` 下例说明了有效的远程接口 `Beta`。它扩展非远程接口 `Alpha`（有远程方法）和接口 `java.rmi.Remote`：
`public interface Alpha { public final String okay = "constants are okay too". public Object foo(Object obj) throws java.rmi.RemoteException. public void bar() throws java.io.IOException. public int baz() throws java.lang.Exception. }`
`public interface Beta extends Alpha, java.rmi.Remote { public void ping() throws 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