

RMI - Java远程方法调用的实现 (一) 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\\_EF\\_BC\\_8DJava\\_c104\\_145599.htm](https://www.100test.com/kao_ti2020/145/2021_2022_RMI_EF_BC_8DJava_c104_145599.htm) 一、RMI概述 Java Remote Method Invocation (RMI -- Java远程方法调用)为采用Java对象的分布式计算提供了简单而直接的途径。这些对象可以是新的Java对象，也可以是围绕现有API的简单的Java包装程序。Java体现了“编写一次就能在任何地方运行的模式。而RMI可将Java模式进行扩展，使之可在任何地方运行”。RMI可利用标准Java本机方法接口JNI与现有的和原有的系统相连接。RMI还可利用标准JDBC包与现有的关系数据库连接。RMI/JNI和RMI/JDBC相结合，可帮助您利用RMI与目前使用非Java语言的现有服务器进行通信，而且在您需要时可扩展Java在这些服务器上的使用。RMI可帮助您在扩展使用时充分利用Java的强大功能。

二、RMI的优点 从最基本的角度看，RMI是Java的远程过程调用(RPC)机制。与传统的RPC系统相比，RMI具有若干优点，因为它是Java面向对象方法的一部分。传统的RPC系统采用中性语言，所以是最普通的系统--它们不能提供所有可能的目标平台所具有的功能。RMI以Java为核心，可与采用本机方法与现有系统相连接。这就是说，RMI可采用自然、直接和功能全面的方式为您提供分布式计算技术，而这种技术可帮助您以不断递增和无缝的方式为整个系统添加Java功能。其主要优点如下：

- 1、面向对象：RMI可将完整的对象作为参数和返回值进行传递，而不仅仅是预定义的数据类型。也就是说，您可以将类似Java哈希表这样的复杂类型作为一个参数进行传递。而在目前的RPC

系统中，您只能依靠客户机将此类对象分解成基本数据类型，然后传递这些数据类型，最后在服务器端重新创建哈希表。RMI则不需额外的客户程序代码(将对象分解成基本数据类型)，直接跨网传递对象。

2、可移动属性：RMI可将属性(类实现程序)从客户机移动到服务器，或者从服务器移到客户机。例如，您可以定义一个检查雇员开支报告的接口，以便察看雇员是否遵守了公司目前实行的政策。在开支报告创建后，客户机就会从服务器端获得实现该接口的对象。如果政策发生变化，服务器端就会开始返回使用了新政策的该接口的另一个实现程序。您不必在用户系统上安装任何新的软件就能在客户端检查限制条件--从而向用户提供更快的反馈，并降低服务器的工作量。这样就能具备最大的灵活性，因为政策改变时只需要您编写一个新的Java类，并将其在服务器主机上安装一次即可。

3、设计方式：对象传递功能使您可以在分布式计算中充分利用面向对象技术的强大功能，如二层和三层结构系统。如果您能够传递属性，那么您就可以在您的解决方案中使用面向对象的设计方式。所有面向对象的设计方式无不依靠不同的属性来发挥功能，如果不能传递完整的对象--包括实现和类型--就会失去设计方式上所提供的优点。

4、安全：RMI使用Java内置的安全机制保证下载执行程序时用户系统的安全。RMI使用专门为保护系统免遭恶意小应用程序侵害而设计的安全管理程序，可保护您的系统和网络免遭潜在的恶意下载程序的破坏。在情况严重时，服务器可拒绝下载任何执行程序。

5、便于编写和使用：RMI使得Java远程服务程序和访问这些服务程序的Java客户程序的编写工作变得轻松、简单。远程接口实际上就是Java接口。服

务程序大约用三行指令宣布本身是服务程序，其它方面则与任何其它Java对象类似。这种简单方法便于快速编写完整的分布式对象系统的服务程序，并快速地制做软件的原型和早期版本，以便于进行测试和评估。因为RMI程序编写简单，所以维护也简单。

6、可连接现有/原有的系统：RMI可通过Java的本机方法接口JNI与现有系统进行交互。利用RMI和JNI，您就能用Java语言编写客户端程序，还能使用现有的服务器端程序。在使用RMI/JNI与现有服务器连接时，您可以有选择地用Java重新编写服务程序的任何部分，并使新的程序充分发挥Java的功能。类似地，RMI可利用JDBC、在不修改使用数据库的现有非Java源代码的前提下与现有关系数据库进行交互。

7、编写一次，到处运行：RMI是Java“编写一次，到处运行”方法的一部分。任何基于RMI的系统均可100%地移植到任何Java虚拟机上，RMI/JDBC系统也不例外。如果使用RMI/JNI与现有系统进行交互工作，则采用JNI编写的代码可与任何Java虚拟机进行编译、运行。

8、分布式垃圾收集：RMI采用其分布式垃圾收集功能收集不再被网络中任何客户程序所引用的远程服务对象。与Java虚拟机内部的垃圾收集类似，分布式垃圾收集功能允许用户根据自己的需要定义服务器对象，并且明确这些对象在不再被客户机引用时会被删除。

9、并行计算：RMI采用多线程处理方法，可使您的服务器利用这些Java线程更好地并行处理客户端的请求。

Java分布式计算解决方案：RMI从JDK 1.1开始就是Java平台的核心部分，因此，它存在于任何一台1.1 Java虚拟机中。所有RMI系统均采用相同的公开协议，所以，所有Java系统均可直接相互对话，而不必事先对协议进行转换

。 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)