

通过E-mail共享Java对象 PDF转换可能丢失图片或格式，建议  
阅读原文

[https://www.100test.com/kao\\_ti2020/467/2021\\_2022\\_\\_E9\\_80\\_9A\\_](https://www.100test.com/kao_ti2020/467/2021_2022__E9_80_9A_E8_BF_87E-ma_c67_467994.htm)

[E8\\_BF\\_87E-ma\\_c67\\_467994.htm](https://www.100test.com/kao_ti2020/467/2021_2022__E9_80_9A_E8_BF_87E-ma_c67_467994.htm) JDK1.1的新功能--序列化接口  
(Serializableinterface),简化了对象持久化(Persistence)的实现

。以下介绍如何通过SMTPE-mail将对象传送给另一个用户。

摘要：一些应用程序需要以一种非实时的方式（例如旅行指南、错误报告（bugreport）、时间表（timesheet）等）和其余用户共享对象。Java语言开发工具包（JDK）1.1版提供了一个重要的功能:java.io.Serializable接口。该技术能让你知道如何序列化一个对象,然后用e-mail传给其它用户。对象持久化和用户间对象共享是许多商业解决方案的基础。例如,一个公司可以用从本公司网址启动的Applet来完成一个时间表的制作。同样该公司也可以提供象具有开支报告、旅行指南、错误报告(bugreport)等功能的Applet。在这些情况下,从Applet的使用者获得的数据需要和负责薪水、付款、旅行房间预订的人们共享。执行这些职能的人们可能分布在不同的城市和国家,可能工作在不同的时区,不能希望每个工作人员都能象猫头鹰一样在晚上工作以填写这样的表格。相同的信息也不应该重新输入。因此,能够存储并且把这些相关对象传送到商业应用中是这些applet有别于其它applet的优势。目前已经有许多方法能实现对象的持久化,例如使用对象数据库和磁盘文件。同样的也有许多办法可以共享对象,例如将数据写入一个套接字,或者实现一个符合CORBA,SOM的模型。以上这几种方案均有自己的优点,当你设计你的商业解决方案时,需要认真地考虑这些方案。但是,还有一种开销不大但可靠的方法,它使

用Internet和Intranet用户能够获取的 技术服务在世界范围传送对象的拷贝。它就是简单邮件传输协议,SMTP。用E-mail发送Java对象 存储和保存对象的一个简单方法是将对象序列化而后用E-mail将它发送给别的用户。这种方法有以下优点:发送的计算机或NC(网络计算机)无需硬盘空间 使用现有的系统传送、排队、发送对象 允许用户使用最喜欢的邮件客户程序来接受邮件 提供简单的机制将同一对象的拷贝分发给许多人 这种方法也有不足之处:邮件的传送可能因为E-mail主机的关机而被较长时间地延迟。所有的主机都可能出现这种情况,E-mail服务器的错误恢复优先级通常比数据库服务器低。邮件的传送不能得到保证--在你的E-mail服务器通知你邮件没有发出时,你不得不重新 发送邮件。E-mail服务器和POP客户程序的功能不足以处理大量交易信息。这些不足和你使用的应用程序有关。对于很多商业解决方案,这些不足并不重要。作为一个设计人员,你工作的一部分就是在全面考虑价格、性能和需求的情况下确定系统的最佳整体结构。使用Java传送对象的四个步骤: Applet必须依次以下面所列出的四个步骤传送Java对象: 序列化有关对象 发送时选择Base64编码方式对序列化对象编码(RFC1521) 与一个SMTP服务器连接 将该对象传送到这个SMTP服务器 下面将介绍如何用E-mail发送一个假设的"臭虫"报告到公司的质量保证部门。将对象序列化

JDK1.1提供的一个奇妙的机制,java.io.Serializable接口,能够序列化并且重建对象。这个接口能使用存储对象(writeObject())和恢复对象(readObject())方法函数。在很多情况下,使用这个接口很方便,只需实现并且调用这两个方法函数。以下的代码定义了一个简单的BugReport对象,它实现了最简单的序列化

接口。 1 import java.io.\*. 2 public class BugReport implements  
Serializable { 3 private Float m\_SoftwareVersion. // version number  
from Help.About, e.g. "1.0" 4 private String m\_ErrorDescription. //  
Description of error 5 private int m\_Severity. // 1=System unusable -  
5=Minor Aesthetic defect 6 public BugReport (Float  
SoftwareVersion, String ErrorDescription, int Severity) { 7  
m\_SoftwareVersion = SoftwareVersion. 8 m\_ErrorDescription =  
ErrorDescription. 9 m\_Severity = Severity. 10 } 11 public BugReport  
( ) {} // for reconstituting serialized objects 12 public void save  
(OutputStream os) 13 throws IOException { 14 try { 15  
ObjectOutputStream o = new ObjectOutputStream(os). 16  
o.writeObject(this). 17 o.flush(). 18 } 19 catch (IOException e)  
{throw e.} 20 } 21 public BugReport restore (InputStream is) 22  
throws IOException, ClassNotFoundException { 23 BugReport  
RestoredBugReport = null. 24 try { 25 ObjectInputStream o = new  
ObjectInputStream(is). 26 RestoredBugReport =  
(BugReport)o.readObject(). 27 } 28 catch (IOException e) {throw  
e.} 29 catch (ClassNotFoundException e) {throw e.} 30 return  
RestoredBugReport. 31 } 32 } 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)