

IBM WebSphere Business Integration  
Adapters



# 适配器开发工具箱用户指南

版本 2.4.0



IBM WebSphere Business Integration  
Adapters



# 适配器开发工具箱用户指南

版本 2.4.0

**注意！**

在使用本资料及其支持的产品之前，请阅读第 11 页的『声明』中的信息。

**2003 年 12 月 19 日**

本文档的此版本适用于 IBM WebSphere Business Integration Adapter Development Kit V2.4.0 及所有后续发行版和修订版，直到在新版本中另有声明为止。

要对 IBM 发送有关 IBM WebSphere Business Integration 文档的意见，请将电子邮件发送至 [ctscrcf@cn.ibm.com](mailto:ctscrcf@cn.ibm.com)。我们欢迎您提出宝贵意见。

当您发送信息给 IBM 后，即授予 IBM 非专有权，IBM 可以它认为适当的任何方式使用或分发此信息，而无须对您承担任何责任。

**© Copyright International Business Machines Corporation 2003, 2004. All rights reserved.**

---

# 目录

关于本文档 . . . . .	v
读者 . . . . .	v
相关文档 . . . . .	v
排版约定 . . . . .	v
<b>本发行版中的新增内容 . . . . .</b>	<b>vii</b>
<b>第 1 章 TwineBall . . . . .</b>	<b>1</b>
读者 . . . . .	1
适配器体系结构 . . . . .	1
安装目录 . . . . .	3
Java 源包 . . . . .	3
准备和运行 . . . . .	3
<b>第 2 章 Java 本机接口 (JNI) . . . . .</b>	<b>7</b>
与 API 交互作用 . . . . .	7
编译和运行 TwineBall JNI 实现 . . . . .	9
构建 JNI 适配器 . . . . .	9
<b>附录. 声明 . . . . .</b>	<b>11</b>
编程接口信息 . . . . .	12
商标和服务标记 . . . . .	12



---

## 关于本文档

IBM<sup>®</sup> WebSphere<sup>®</sup> Business Integration Adapters 产品服务组合为领先的电子商务技术、企业应用程序以及旧的系统和大型机系统提供集成连接。产品集包括用于定制、创建和管理业务流程集成组件的工具和模板。

本文档描述如何配置和运行 IBM WebSphere Business Integration Adapters 产品中所包括的适配器开发工具箱 (ADK) 的样本。

---

## 读者

本文档适合于在客户站点上使用适配器开发工具箱的顾问、开发者和系统管理员。

---

## 相关文档

随本产品提供的全套文档描述所有 WebSphere Business Integration Adapters 安装的公共功能部件和组件，并包括关于特定组件的参考资料。

随本产品提供的全套文档描述所有 WebSphere Business Integration Adapters 安装的公共功能部件和组件，并包括关于特定组件的参考资料。

可从以下站点中安装相关文档：

- 有关一般适配器信息、将适配器与 WebSphere 消息代理 (WebSphere MQ Integrator、WebSphere MQ Integrator Broker 和 WebSphere Business Integration Message Broker) 配合使用的信息以及将适配器与 WebSphere Application Server 配合使用的信息，请参阅 IBM WebSphere Business Integration Adapters 信息中心：  
<http://www.ibm.com/websphere/integration/wbiadapters/infocenter>
- 有关将适配器与 WebSphere InterChange Server 配合使用的信息，请参阅 IBM WebSphere InterChange Server 信息中心：  
<http://www.ibm.com/websphere/integration/wicserver/infocenter>  
<http://www.ibm.com/websphere/integration/wbicollaborations/infocenter>
- 有关 WebSphere 消息代理的更多信息：  
<http://www.ibm.com/software/integration/mqfamily/library/manualsa/>
- 有关 WebSphere Application Server 的更多信息：  
<http://www.ibm.com/software/webservers/appserv/library.html>

这些站点包含关于下载、安装和查看文档的简单指示。

---

## 排版约定

本文档使用以下约定：

<code>courier</code> 字体	指示文字值，如命令名、文件名、您输入的信息或系统在屏幕上打印的信息。
斜体	指示第一次出现的新术语、变量名或交叉引用。

---

---

蓝色文本	仅当您联机查看手册时才能看到蓝色文本，它指示交叉引用超链接。单击任何蓝色文本可以跳至引用的对象。
{ }	在语法行中，花括号括起一组选项，您必须从这些选项中选择一个且只能选择一个。
	在语法行中，管道用来隔离一组选项，您必须从这些选项中选择一个且只能选择一个。
[ ]	在语法行中，方括号将一个可选参数括在其中。
...	在语法行中，省略号指示重复先前的参数。例如， <code>option[,...]</code> 意味着您可以输入多个用逗号隔开的选项。
< >	尖括号括起名称的单个元素，以将它们互相区分开，如 <code>&lt;server_name&gt;&lt;connector_name&gt;tmp.log</code> 。
/, \	在本文中，反斜杠 (\) 用作目录路径的约定。对于 UNIX 安装，用斜杠 (/) 替代反斜杠。
%text% 和 \$text	在百分比 (%) 符号内的文本指示 Windows <code>text</code> 系统变量或用户变量的值。UNIX 环境中的等同符号表示法是 <code>\$text</code> ，指示 <code>text</code> UNIX 环境变量的值。

---



---

## 本发行版中的新增内容

《适配器开发工具箱用户指南》是 WebSphere Business Integration Adapter Framework V2.4 发行版中的新增内容。



---

## 第 1 章 TwineBall

TwineBall 样本由连接器、ODA 和应用程序组成，用来演示适配器开发的一些基本原理。该应用程序称为 TwineBall，它提供以下各项：

- 为面向数据的基本操作（如创建、检索、更新和删除）提供 RMI 接口
- 对所包括的 Object Discovery Agent 提供实体定义信息

---

### 读者

本文档适合于考虑开发定制适配器的有经验的 Java 开发者。在开始开发适配器之前，您还应该学习“信息中心”文档中的 *IBM WebSphere Connector Development Guide for Java* 和 *IBM WebSphere Business Object Development Guide*。

有关适配器的一般概述，请参阅“信息中心”文档中的 *Technical Introduction to WebSphere Business Integration Adapters*。

---

### 适配器体系结构

TwineBall 服务器使用 DB2 中创建的数据库来存储数据。此服务器可处理许多不同实体定义，但最常用的表是：

- Customer
- Address
- Order
- OrderLine

第 2 页的图 1 显示 TwineBall 体系结构。

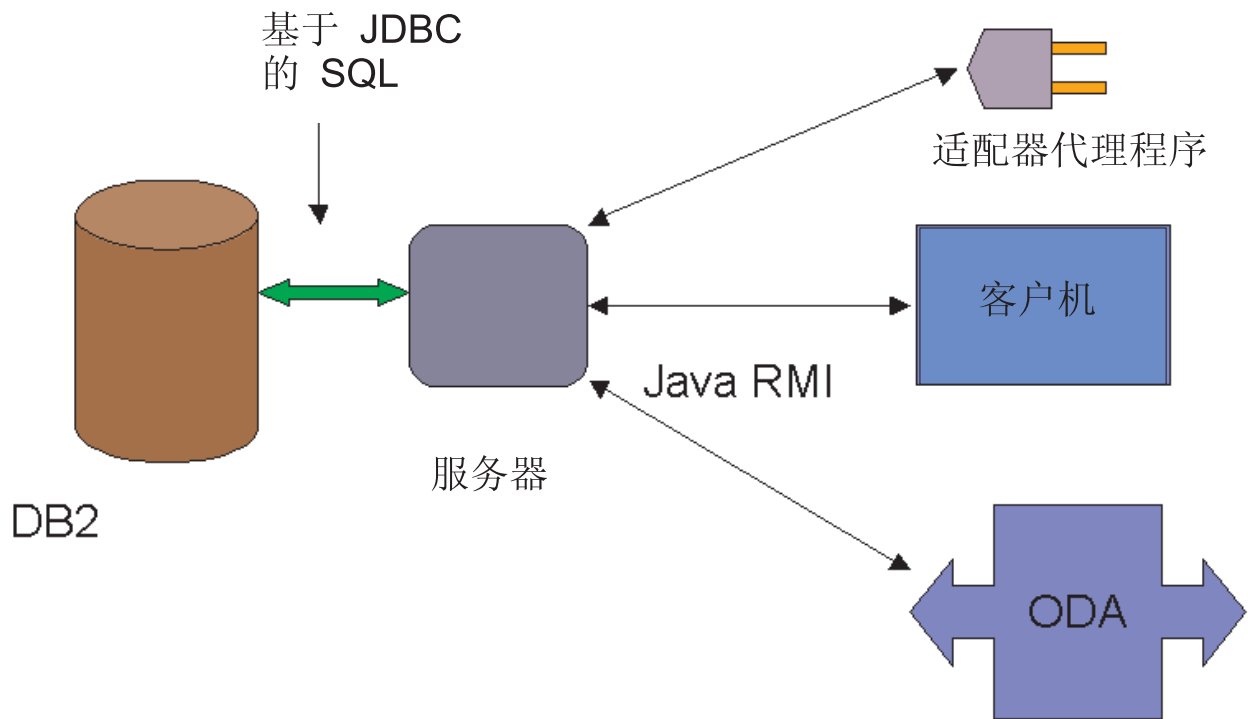


图 1. 原始 TwineBall 体系结构

事件传递通过轮询机制来提供。为了使轮询能起作用，用户必须在 DB2 中创建事件和归档表，还必须将触发器放在适当位置。

适配器和 ODA 均通过 Java RMI 连接至此服务器。适配器和 ODA 均采用表示服务器名称、端口和服务名称的服务器 URL。

标准服务器 URL 是 `RMI://localhost:2112/TwineBall`。

或者，TwineBall 适配器可独立于服务器运行。在这种情况下，它调用 C++ 编写的 TwineBall API 的备用实现。该适配器将装入 C++ 共享库和调用，而不是进行远程 RMI 调用。有关更多信息，参阅第 5 页的『使用 JNI』。

## TwineBall 服务器

TwineBall 服务器组件列示如下。

- **TwineBallConnection**: 包含大部分数据库逻辑。
- **TwineBallServer**: 提供与 RMI 客户机的连接。
- **TwineBallInterface**: 为要调用的适配器、ODA 或其它客户机提供 API。
- **TwineBallConnectionPool**: 共享 TwineBall 连接（视为 API 的一部分）。

## 适配器

适配器组件列示如下。

- **AdapterAgent**: 通过提供初始化和终止逻辑来扩展 CWConnectorAgent。
- **AdapterBOHandler**: 扩展 CWConnectorBOHandler: 处理服务调用请求。

- **AdapterEventStore**: 通过提供事件传递逻辑来扩展 CWConnectorEventStore。

## Object Discovery Agent

ODA 组件列示如下。

- **ObjectDiscoveryAgent**: 通过提供初始化来扩展 ODKAgentBase, 并调用 ObjectFinder 和 ObjectAnalyzer。
- **ObjectFinder**: 实现 getTreeNodes 的逻辑。
- **ObjectAnalyzer**: 实现 generateDefs 的逻辑。

---

## 安装目录

安装 TwineBall 样本时生成若干个目录。它们列示如下。

表 1. 文件目录

目录	用途
/dependencies	运行 TwineBall 服务器所需要的属性和策略文件。
/documentation	包含有关如何使用样本的自述文件。
/JNI_supplement	包含演示如何使用 JNI 在适配器中调用 C++ API 的代码。
/Repository	包含适配器和 ODA 的消息文件。
/Samples	包含适配器的业务对象。
/setup	包含设置以及运行服务器、适配器和 Object Discovery Agent 的脚本。

---

## Java 源包

包含 Java 资源的包的结构如下。

表 2. Java 资源

资源	内容
com.ibm.wbia.TwineBall.Adapter	TwineBall 适配器
com.ibm.wbia.TwineBall.oda	TwineBall Object Discovery Agent
com.ibm.wbia.TwineBall.Server	TwineBall 服务器和 API
com.ibm.wbia.TwineBall.jnibridge	JNI 补充 (存根库和向量网桥)

---

## 准备和运行

仅对 Windows 测试了 TwineBall 服务器、适配器和 ODA。它们可能在其它平台上工作, 但对它们进行测试是一项留给用户的练习。

要运行 TwineBall 样本, 必须已安装 JDK (最好是 1.3.1)。

## 设置数据库

为了演示创建、复原、更新和删除（CRUD）操作和真实的元数据驱动的设计，TwineBall 服务器使用实际的数据库来定义其实体和存储其数据。这需要进行某些附加设置。提供了一些脚本使设置变得容易，但是这些脚本将需要稍微修改以与您的系统相匹配。

要设置数据库，请遵循以下步骤。

1. 安装 WebSphere Business Integration 适配器框架或 WebSphere InterChange Server 及其所有先决条件。
2. 安装 IBM DB2 版本 8 修订包 2。
3. 在操作系统中创建下列用户标识：  
TwineBall 和 TwineBallAdapter。  
为它们提供密码  
sample42
4. 创建 Twine 数据库。在 setup/SQL 目录中，您将找到 Twine-database.bat 和 Twine-database.sql。  
编辑 Twine-database.sql 以将密码与 db2admin 标识相匹配。
5. 运行 Twine-database.bat 脚本。您应该不会接收到任何错误消息。如果接收到错误消息，则解决它们后再继续。
6. 创建 TwineBall 服务器所需的表。在 setup/SQL 目录中，您将找到 TwineBall-Schema.bat 和 TwineBall-Schema.SQL。
7. 运行 TwineBall-Schema.bat 脚本。解决任何错误后再继续。
8. 设置事件和归档表。在 setup/SQL 目录中，您将找到 WBIA-Events.sql 和 WBIA-Events.bat。
9. 运行 WBIA-Events.bat 脚本。解决任何错误后再继续。

## 运行服务器

要运行服务器，请遵循以下步骤。

1. 通过将所有 Java 源导入 IDE 的项目中，编译 /src 和 JNI\_Supplement 目录中的整个 Java 源树。  
配置 IDE 以将 InterChange Server 或“适配器框架”的 /lib 目录中的所有 JAR 文件放在 CLASSPATH 中。
2. 在“适配器框架”或 InterChange Server 安装的 connectors 目录下创建称为 TwineBall 的目录。
3. 使用正常 Java 包结构（即，com.ibm.sample.MyClass 存储为 com/ibm/sample/Myclass.class）将所有已编译的类文件导出至此目录。
4. 而且，将 setup\scripts 目录中的所有文件复制到此文件夹中。应用程序通常将在与适配器不同的目录中，但用此方法构造目录将简化安装。
5. 将新创建的 TwineBall 目录添加至系统 CLASSPATH 中。这将允许 Java 运行时查找服务器类。
6. 确保 Java Development Kit bin 目录位于 CLASSPATH 中。运行 runrmic.bat 文件。这将产生 RMI 通信所需要的存根和框架。
7. 在 TwineBall 目录内创建称为 Properties 的新目录。

8. 将 TwineBall.policy 和 TwineBall.db2.properties 文件从 dependencies 目录移至此目录并将 TwineBall.db2.properties 重命名为 TwineBall.properties。
9. 编辑 TwineBall.properties 文件以将 server.name 更改为您的系统名称。例如，如果您的机器称为 JEFFB，则 server.name 将为 //JEFFB:2112/TwineBallServer。
10. 通过执行 runserver 脚本启动 TwineBall 服务器。当它运行时，将出现消息“Server ... has been registered”。
11. 保留服务器在窗口中运行。ODA 和适配器将通过 RMI 连接至它。

## 启动 TwineBall 连接器

要启动 TwineBall 连接器，请遵循以下步骤。

1. 在“适配器框架”或 InterChange Server 安装的 /connectors 目录内创建称为 TwineBall 的目录。
2. 如果您未运行 WebSphere InterChange Server 或“适配器框架”，则将配置文件放在该目录中。  
如果正在运行 InterChange Server，则导入该配置。
3. 创建对代理程序运行适配器所必需的 WebSphere MQ 队列。如果代理程序是 InterChange Server，则由于您可以使用 IDL 传输，因此这可能不必要。
4. 启动“连接器配置器”并将 MessageFileName 标准属性更改为“TwineBallAdapter.txt”。
5. 尝试使用启动适配器的常用语法来启动连接器。例如：

```
start_TwineBall TwineBall WMQI -c
C:\IBM\WebSphereAdapters\connectors\TwineBall\TwineBall.cfg
```

6. 要测试适配器，可在 InterChange Server 中设置传递（Pass-Through）协作，或在非 InterChange Server 方式下颠倒传递和服务调用队列。然而，这些过程不在本文档所讨论的范围内。

## 启动 TwineBall ODA

要启动 TwineBall ODA，请遵循以下步骤。

- 尝试通过运行 start\_twineballODA.bat 文件来启动 ODA。
- 尝试通过“业务对象设计器”来发现 ODA 以便使用它。
- 当选择将从中创建业务对象的节点时，请同时选择要创建的所有业务对象。除非您明确选择了子业务对象，否则将不会创建它们。

## 使用 JNI

可构建 Java 适配器并使用“Java 本机接口”（JNI）来调用 C++ API。有关详细信息，请参阅第 7 页的第 2 章，『Java 本机接口（JNI）』。





---

## 第 2 章 Java 本机接口 (JNI)

本章描述如何将“Java 本机接口”(JNI)与 Java 适配器配合使用以与 C++ 应用程序集成。

有三种基本方法可用来启用具有 C++ API 的应用程序。

- 使用 CDK 构建 C++ 适配器。
- 构建 Java 适配器并使用 JNI (Java 本机接口) 来调用 C++ API。
- 根本不用构建适配器, 而是使用技术适配器, 如 MQSeries 或 COM 的适配器。

我们将讨论第二个选项, 并查看使用此方法的样本适配器。

**注:** 此示例用来显示如何构建使用 JNI 的 Java 适配器并显示样本 JNI 适配器如何工作。具有 JNI 实现技术方面的知识是您理解和使用此方法的先决条件。

JNI 旨在允许 Java 应用程序调用 C/C++ 库。以下示例使用 Java 来调用一个 C++ API。实现该 API 以显示 TwineBall。TwineBall 是一个 Java 适配器, 可以调用 C++ API。

---

### 与 API 交互作用

TwineBall 适配器通过 TwineBall 接口执行它对应用程序服务器的所有调用。在 Java RMI 中, 对远程接口的调用由 JVM 自动编组。图 2 显示基于 Java RMI 的原始交互作用。

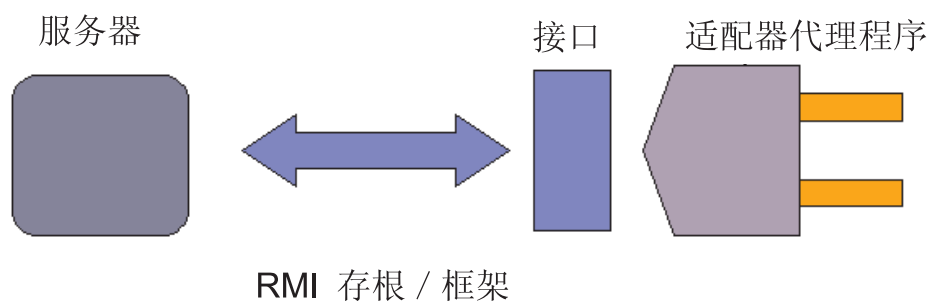


图 2. 基于 RMI 的交互作用

为了演示 JNI, 已用 C++ 共享库替换图形的服务器端。现在该图形看起来象第 8 页的图 3。

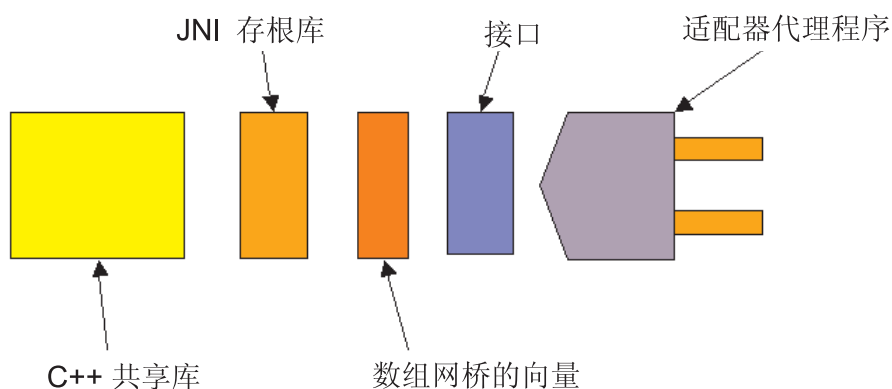


图 3. 修改后的 TwineBall 体系结构

因为定义良好的 Java 接口已经在调用应用程序，所以在适配器端几乎只需要进行很少的代码更改就能更改此接口的实现。这允许我们用相同适配器代码来演示使用 JNI 的适配器和使用 Java RMI 的适配器。仅接口后面的实现会更改。

TwineBall 接口大量使用 Java 向量。我们决定将向量转换成字符串数组，而不是尝试将这些向量引入 C++ 环境中。这就是 VectorToArrayBridge 所起的作用。如果适配器使用 Java 实用程序类更方便，则您可能想要使用类似的方法。

“JNI 存根库”声明本机方法。这就是将 C++ 中实现的方法与 Java 环境联系在一起的方法。

在本示例中，将用单个 C++ 模块来实现 C++ 共享库。然而，此共享库包括 JNI.H 头文件，从而可识别 Java 类型。基本 Java 类型和 C++ 类型之间的转换在 C++ 环境中仍经常发生。与现有的 C++ 应用程序 / API 进行交互将需要编写新的 C++ 代码和 Java 代码。

请注意，将不再涉及数据库。在本示例中，将用预先确定的返回值来实现接口。

在样本 JNI 源代码中，您将找到以下各段。

表 3. JNI 源代码的组件

文件名	功能
Makefile.PLATFORM	在给定的平台上构建 C++ 共享库。
com_ibm_wbia_TwineBall_jnibridge_JNIStubLibrary.cc	C++ 共享库的源代码。包含 C++ “粘接代码”。
com_ibm_wbia_TwineBall_jnibridge_JNIStubLibrary.h	C++ 共享库的头文件。
com.ibm.wbia.TwineBall.jnibridge.JNIStubLibrary	JNI 存根库
com.ibm.wbia.TwineBall.jnibridge.VectorToArrayBridge	数组网桥的向量
com.ibm.wbia.TwineBall.jnibridge.test.TestDriver	JNI 存根库的测试驱动程序

---

## 编译和运行 TwineBall JNI 实现

我们使用 Make 来构建共享库。确保您具有目标平台的完整构建环境。每个平台的 Makefile 包含有关对哪个编译器构建该文件的信息。如果安装了该编译器和 Make，则编译该库所应该执行的所有操作就是将这些文件抽取至目标平台并输入 `make -f Makefile.Platform`，其中 Platform 是期望的目标平台。

要用 JNI 网桥运行 TwineBall 适配器：

1. 在目标平台上安装 WebSphere InterChange Server 或 WebSphere Business Integration 适配器框架。
2. 将 TwineBall 服务器和适配器的已编译类放入 `%CROSSWORLDS%/connectors/twineball` 目录中。
3. 将消息文件从“相关性”移至“消息”文件夹中。
4. 将共享库和 `start_twineball.sh` 脚本放入 `%CROSSWORLDS%/connectors/twineball` 目录中。  
在此示例中，因为适配器将调用 JNI 网桥，所以您将不需要设置 TwineBall 数据库或服务器。
5. 将 `twineball.cfg` 用作模板来配置适配器，并使用特殊字符串 `JNISTub` 作为服务器 URL。
6. 使用适配器的启动脚本以正常方式启动适配器。

适配器的 JNI 后端同时实现“TwineBallCustomer”业务对象的服务调用和事件传递。您将在“样本业务对象”文件夹中找到 TwineBallCustomer 业务对象。

---

## 构建 JNI 适配器

要构建 JNI 适配器，请遵循以下步骤。

1. 检查现有的 C++ API，并记下未在 Java 中定义的任何数据结构。您将需要在 C++ 粘接代码中转换这些数据结构。
2. 使用 Java 中与那些方法最匹配的本机声明来创建“JNI 存根”库。您将需要对适配器将使用的 API 中出现的每个函数执行此操作。
3. 运行 `javah` 来为存根库创建 C++ 头文件。示例用法：  
`javah com.ibm.wbia.TwineBall.jnibrige.JNISTubLibrary`
4. 在 C/C++ 中，在生成的头中实现每个函数。您将使用此代码来将 JNI 函数特征符粘接到 API 中。
5. 用测试驱动程序测试“JNI 存根”库中的每个函数。
6. 按编码 Java 适配器的相同方式开始编码适配器，并将“JNI 存根”库用作 Java API。

**注意事项：**

1. 当同意使用 TwineBall 时，可能适合在 JNI 层的两端进行类型转换。
2. 字符集非常重要。当在此示例中将 Java Unicode 字符串转换成 C++ 字符串时，会将它们转换为 UTF-8。如果应用程序不能处理 UTF-8，则您将需要将它们转换为特定代码页。当您执行此操作以处理所有可能的 Unicode 字符时，要特别细心，否则可能失败。

3. 在许多版本的 HP-UX 上，您将需要声明并调用共享库中的 `_main()` 函数，以便使用 C++ 执行 JNI。这会指示操作系统初始化 C++ 运行时。Win32 上的相同代码将失败，因此 `#defines` 对于此移植问题可能是必要的。

## 线程安全

注意线程技术的以下方面。

- 记住，Java 适配器在缺省情况下是多线程的。可能要求您的适配器同时在多个线程上处理多个服务调用。如果本机代码不是线程安全的，则您可能需要在启动脚本中使用 `-t` 选项来将适配器变为单线程。
- 不要将 Java 对象在 C++ 环境中从一个线程传递至另一个线程，尽管您无论如何也不太可能在适配器中执行此操作。
- 全局访问对象的同步具有正常规则。
- 您不得将 `JNIEnv*` 从一个线程传递至另一个线程，或依赖于两个不同线程中相同的指针。  
JVM 为每个线程创建不同的 JNI 环境，因此本机代码必须仅使用对当前线程有效的指针。

---

## 附录. 声明

IBM 可能并未在所有国家或地区提供本文档中讨论的产品、服务或功能特性。有关您当前所在区域的产品和服务的信息，请向您当地的 IBM 代表咨询。任何对 IBM 产品、程序或服务的引用并非意在明示或暗示只能使用 IBM 的产品、程序或服务。只要不侵犯 IBM 的知识产权，任何同等功能的产品、程序或服务，都可以代替 IBM 产品、程序或服务。但是，评估和验证任何非 IBM 产品、程序或服务，则由用户自行负责。

IBM 公司可能已拥有或正在申请与本文档内容有关的各项专利。提供本文档并未授予用户使用这些专利的任何许可。您可以用书面方式将许可查询寄往：

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
U.S.A.

本条款不适用英国或任何这样的条款与当地法律不一致的国家或地区：

International Business Machines Corporation “按现状”提供本出版物，不附有任何种类的（无论是明示的还是暗含的）保证，包括但不限于暗含的有关非侵权、适销和适用于某种特定用途的保证。某些国家或地区在某些交易中不允许免除明示或暗含的保证。因此本条款可能不适用于您。

本信息中可能包含技术方面不够准确的地方或印刷错误。此处的信息将定期更改；这些更改将编入本资料的新版本中。IBM 可以随时对本资料中描述的产品和 / 或程序进行改进和 / 或更改，而不另行通知。

本信息中对非 IBM Web 站点的任何引用都只是为了方便起见才提供的，不以任何方式充当对那些 Web 站点的保证。那些 Web 站点中的资料不是 IBM 产品资料的一部分，使用那些 Web 站点带来的风险将由您自行承担。

IBM 可以按它认为适当的任何方式使用或分发您所提供的任何信息而无须对您承担任何责任。

本程序的被许可方如果要了解有关程序的信息以达到如下目的：（i）允许在独立创建的程序和其他程序（包括本程序）之间进行信息交换，以及（ii）允许对已经交换的信息进行相互使用，请与下列地址联系：

IBM RTP Laboratory  
3039 Cornwallis Road  
P.O. BOX 12195  
Raleigh, NC 27709-2195  
U.S.A

只要遵守适当的条件和条款，包括某些情形下的一定数量的付费，都可获得这方面的信息。

本资料中描述的许可程序及其所有可用的许可资料均由 IBM 依据 IBM 客户协议、IBM 国际软件许可协议或任何同等协议中的条款提供。

此处包含的任何性能数据都是在受控环境中测得的。因此，在其他操作环境中获得的数据可能会有明显的不同。有些测量可能是在开发级的系统上进行的，因此不保证与一般可用系统上进行的测量结果相同。此外，有些测量是通过推算而估计的。实际结果可能会有差异。本文档的用户应当验证其特定环境的适用数据。

涉及非 IBM 产品的信息可从这些产品的供应商、其出版说明或其他可公开获得的资料中获取。IBM 没有对这些产品进行必要的测试，也无法确认其性能的精确性、兼容性或任何其他关于非 IBM 产品的声明。有关非 IBM 产品性能的问题应当向这些产品的供应商提出。

本资料中可能包含用于日常业务运作的数据和报表的示例。为了尽可能完整地说明这些数据和报表，这些示例可能包括了个人、公司、品牌和产品的名称。所有这些名称都是虚构的，如与实际商业企业所使用的名称和地址有雷同，纯属巧合。

所有关于 IBM 未来方向或意向的声明都可随时更改或收回，而不另行通知，它们仅仅表示了目标和意愿而已。

---

## 编程接口信息

编程接口信息（如果提供的话）用于帮助您创建使用本程序的应用软件。

通用编程接口允许您编写获取本程序工具的的服务的应用软件。

然而，此信息可能还包含诊断、修改和调整信息。提供诊断、修改和调整信息是为了帮助您调试应用软件。

**警告：**切勿使用此诊断、修改和调整信息作为编程接口，因为它随更改而变化。

---

## 商标和服务标记

下列各项是国际商业机器公司在美国和 / 或其他国家或地区的商标或注册商标：

IBM  
the IBM logo  
AIX  
CrossWorlds  
DB2  
DB2 Universal Database  
MQIntegrator  
MQSeries  
Tivoli  
WebSphere

Lotus、Domino、Lotus Notes 和 Notes Mail 是 Lotus Development Corporation 在美国和 / 或其他国家或地区的商标。Microsoft、Windows、Windows NT 和 Windows 徽标是 Microsoft Corporation 在美国和 / 或其他国家或地区的商标。

MMX、Pentium 和 ProShare 是 Intel Corporation 在美国和 / 或其他国家或地区的商标或注册商标。

Java 和所有基于 Java 的商标是 Sun Microsystems, Inc. 在美国和 / 或其他国家或地区的商标。

其他公司、产品或服务名称可能是其他公司的商标或服务标记。



IBM WebSphere Business Integration Adapter Framework V2.4.0。



中国印刷