

**IBM WebSphere Commerce - Express
Developer Edition**



迁移指南

版本 5.5

**IBM WebSphere Commerce - Express
Developer Edition**



迁移指南

版本 5.5

注意:

在使用本资料及其支持的产品之前，请务必阅读第 61 页的『声明』中的一般信息。

第一版 (2003 年 9 月)

本版本适用于 IBM WebSphere Commerce - Express Developer Edition for Windows 2000 V5.5 (产品号 5724-A18) 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。确认您正在使用本产品级别的正确版本。

通过您当地的 IBM 代表或 IBM 分部可订购出版物。

IBM 欢迎您提出宝贵意见。您可以使用在线 IBM WebSphere Commerce 文档反馈表单发送您的意见，该表单在以下 URL 提供:

<http://www.ibm.com/software/webservers/commerce/rcf.html>

当您发送信息给 IBM 后，即授予 IBM 非专有权，IBM 可以它认为合适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 1996, 2003. All rights reserved.

目录

| | | | |
|---|-----------|--|-----------|
| 前言 | v | 应用 IBM WebSphere Commerce 5.5.0.2 修订包 | 22 |
| 本书的更新 | v | 迁移 WebSphere Commerce 开发环境数据库模式 | 22 |
| 本书中使用的约定 | v | 迁移 WebSphere Commerce 实例 | 23 |
| 何处可找到更多信息 | vi | 转换定制的或扩展的代码 | 27 |
| 第 1 部分 迁移前 | 1 | VisualAge for Java 和 WebSphere Studio Application Developer 之间的不同 | 27 |
| 第 1 章 迁移 WebSphere Commerce 开发环境之前 | 3 | 转换扩展的或修改的 WebSphere Commerce 业务逻辑和代码 | 28 |
| 迁移的先决条件 | 3 | 迁移支付实例和数据库 | 37 |
| 必备硬件 | 3 | 第 3 部分 迁移后 | 39 |
| 必备软件 | 4 | 第 4 章 迁移后操作 | 41 |
| 迁移环境 | 4 | 特定于环境的迁移后操作 | 42 |
| 环境 A - 完全开发环境 | 6 | 环境 B | 42 |
| 环境 B - 开发环境与 WebSphere Commerce 共存但不共享有用资源 | 6 | 第 4 部分 转换代码的教程 | 45 |
| 第 2 章 影响 WebSphere Commerce 开发环境迁移的代码更改 | 7 | 第 5 章 教程的迁移流程 | 47 |
| WebSphere Commerce 5.4 和 WebSphere Commerce Studio 5.4 安装环境 | 7 | 第 6 章 教程 A: 迁移扩展或定制的控制器命令 | 49 |
| 报价请求 (RFQ) | 7 | 迁移企业 bean | 49 |
| 协作工作空间 | 8 | 从 VisualAge for Java 导出企业 bean | 49 |
| 数据库模式的差异 | 12 | 删除 | |
| API 区别 | 12 | WebSphereCommerceServerExtensionData 文件夹中的 ExtensionJDBCHelperBean | 49 |
| 企业 bean 从 1.0 迁移至 1.1 规范 | 13 | 将企业 bean 导入到 WebSphere Studio Application Developer 中 | 50 |
| 检查 MSGTYPES 表中的独特索引 | 14 | 编辑方法以避免 java.rmi.RemoteException 错误 | 50 |
| WebSphere 应用程序命令框架 | 14 | 更改访问隔离级别 | 50 |
| J2EE 连接器体系结构 | 15 | 生成部署代码 | 51 |
| 第 2 部分 迁移流程 | 17 | 迁移定制的 JSP 文件 | 51 |
| 第 3 章 迁移 WebSphere Commerce 开发环境 | 19 | 迁移扩展的控制命令 | 51 |
| 备份 WebSphere Commerce 和 WebSphere Commerce 开发环境有用资源 | 19 | 在“时代先锋”样本商店中测试新逻辑 | 52 |
| 升级 DB2 产品级别 | 19 | 第 7 章 教程 B: 迁移扩展或定制实体 bean 和任务命令 | 53 |
| 升级 WebSphere Commerce 开发环境 | 20 | | |
| 将任务命令、控制器命令和数据 bean 导出到 JAR 文件 | 21 | | |
| 将企业 bean 导出至 JAR 文件 | 21 | | |

| | |
|----------------------------------|----|
| 迁移扩展的任务命令 | 53 |
| 迁移扩展的数据 bean | 54 |
| 迁移企业 bean | 55 |
| 将 bonusPoint 字段迁移至用户实体 | 55 |
| 更新模式和表映射以包含 BONUS 表 | 55 |
| 生成部署代码. | 57 |
| 迁移定制的 JSP 文件 | 57 |

| | |
|-----------------------------|----|
| 在“时代先锋”样本商店中测试新逻辑 | 57 |
|-----------------------------|----|

第 5 部分 附属资料 **59**

| | |
|--------------|----|
| 声明 | 61 |
| 商标. | 63 |

前言

本书描述如何将 WebSphere® Commerce 开发环境从 WebSphere Commerce Studio V5.4 迁移至 WebSphere Commerce - Express Developer Edition V5.5。

如果您已经应用了 WebSphere Commerce V5.4 的任一修订包或 Commerce Enhancement Pack 并已经对这些版本的 WebSphere Commerce 使用了 WebSphere Commerce Studio V5.4，则也支持从那些代码级别的迁移。

本书的更新

本迁移指南及本迁移指南的任何更新版本将在 WebSphere Commerce 技术库中提供。

要了解该产品的最新更改，请参阅更新的产品 README 文件，还可从上面的 Web 站点获取。如果在没有安装 WebSphere Commerce Studio V5.4 及支持的产品的机器上安装 WebSphere Commerce 开发环境，请参阅对应于您的数据库的《*WebSphere Commerce Studio 安装指南*》。

本书中使用的约定

本书使用以下突出显示的约定：

- **粗体字**表示命令或者诸如字段名、图标或菜单选项之类的图形用户界面（GUI）控件。
- 等宽字表示完全按显示原样输入的文本示例、文件名以及目录路径和名称。
- 斜体字用于强调词语。斜体还表示必须用相应系统值替代的名称。当您看到以下任何名称时，请按如下所述使用系统值予以替换：

host_name

WebSphere Commerce 开发环境机器的全限定主机名（例如，ibm.com 是全限定的）。

drive 代表安装了正讨论的产品或组件的驱动器的盘符（例如，C:）。

WC_installdir

WebSphere Commerce - Express 5.5 的安装路径。缺省安装路径为 \Program Files\WebSphere\CommerceServer55。例如，c:\Program Files\WebSphere\CommerceServer55。

WCDE_installdir

WebSphere Commerce 开发环境的安装路径。缺省安装路径为 \WebSphere\CommerceDev55。例如，c:\WebSphere\CommerceDev55。

WSAD_installdir

WebSphere Studio Application Developer V5.1 的安装路径。缺省安装路径是 \WebSphereStudio。例如，c:\WebSphereStudio。

WSAD_workspace

WebSphere Studio Application Developer V5.1 工作空间的安装路径。缺省安装路径是 \WebSphere\workspace_db2。例如，c:\WebSphere\workspace_db2。



此图标用于标记“技巧” - 可帮助您完成任务的附加信息。

何处可找到更多信息

有关 WebSphere Commerce 开发环境的信息，请参考 WebSphere Commerce 开发环境站点。

关于 WebSphere Commerce 的信息，请参阅以下 Web 站点：

- WebSphere Commerce 技术库，它包括关于下载更新文档（例如指南和联机帮助）的区域。
- WebSphere Commerce 支持
- WebSphere Commerce - Express 概述

第 1 部分 迁移前

本部分讨论在您迁移至 WebSphere Commerce 开发环境之前应当复查的重要信息。这一部分包括关于硬件及先决条件、设置和可能的迁移环境以及重要的代码更改的信息。

第 1 章 迁移 WebSphere Commerce 开发环境之前

本章提供关于您从 WebSphere Commerce Studio V5.4 迁移至 WebSphere Commerce - Express Developer Edition V5.5 之前应注意的事项和执行的的任务的信息。

迁移的先决条件

为了使用本指南中的迁移指示信息，您的 WebSphere Commerce Studio V5.4 机器必须满足在以下部分中详细描述硬件和软件需求。

必备硬件

- 您需要满足以下条件的专用奔腾® III 733 MHz IBM 兼容个人计算机：
 - 随机存取存储器 (RAM) 最少 512 MB。
 - 以下是选择要安装的每个选项所需的磁盘空间大小：
 - 用于商店前台有用资源开发：
 - WebSphere Studio - 300 MB
 - 商店归档文件工具 - 25 MB
 - Blaze Advisor 和 Blaze Innovator Workbench - 55 MB
 - 用于后端开发：
 - WebSphere Application Developer for Java™ Enterprise - 1.2 GB
 - IBM® DB2® 通用数据库 - 300 MB
 - WebSphere Commerce 开发环境 - 300 MB
 - Blaze Advisor™ Server - 45 MB
 - IBM Distributed Debugger - 160 MB
 - Applet Designer - 15 MB
 - Page Detailer - 10 MB

在 C: 驱动器上还需要额外的 50 MB。如果机器格式化为 FAT 分区且分区超过 1.024 GB，则需要两倍的可用磁盘空间。安装程序将检查是否有足够的可用磁盘空间，如果没有足够空间，安装程序将向您提示警告消息。

- CD-ROM 驱动器。
- 颜色深度至少为 256 色且屏幕分辨率设置为 800x600 或更高的图形监视器。
- 鼠标或其它定位设备。

- TCP/IP 协议支持的局域网 (LAN) 适配器。

必备软件

迁移至 WebSphere Commerce - Express Developer Edition V5.5 之前，确保满足以下最小软件需求：

- 确保安装了 Windows® 2000 Server Edition 或 Advanced Server Edition (应用了 Service Pack 3)。您可以从 <http://www.microsoft.com> 获取更新。

重要信息

如果正在将 WebSphere Commerce Studio V5.4 与 Windows NT® 一起使用，则在迁移至 WebSphere Commerce - Express Developer Edition V5.5 之前，必须将操作系统升级为 Windows 2000。Windows NT 不支持此版本的 WebSphere Commerce 开发环境。

- 在您将用于访问 WebSphere Commerce Studio 的每台机器上安装带有最新关键安全性更新的 Internet Explorer 5.5 Service Pack 1。可以在 Microsoft® Web 站点 <http://www.microsoft.com> 下载 Microsoft Internet Explorer 的副本。
- 在安装 WebSphere Commerce 开发环境之后，您需要安装 IBM WebSphere Commerce 5.5.0.2 修订包。该修订包包括用于迁移过程的修订和增强功能。该修订包可从 WebSphere Commerce 支持站点获取。单击对应于您正在使用的修订版的链接，并查看软件下载部分。下载该修订包，并遵循所提供的用于安装的指示信息。（如果您计划从 WebSphere Studio Application Developer V5.1 升级至 WebSphere Studio Application Developer V5.1，指示信息也在修订包文档中提供。）在没有安装该修订包的情况下，不应继续进行本指南中的迁移步骤（诸如迁移实例或数据库）。

迁移环境

当从 WebSphere Commerce Studio V5.4 迁移至 WebSphere Commerce - Express Developer Edition V5.5 时，可以选择迁移至与您完成 WebSphere Commerce 开发环境新安装后相当的环境。即，WebSphere Commerce 开发环境迁移完成后，可以从机器上除去不需要的 WebSphere Commerce 有用资源。

如果您希望继续有一个 WebSphere Commerce 的本地实例，则不必除去这些有用资源。在此情况下，建议将您的 WebSphere Commerce 本地实例迁移至 WebSphere Commerce 5.5 级别。

支持以下环境作为迁移 WebSphere Commerce 开发环境的最终结果：

- 环境 A - 完全开发环境
- 环境 B - 开发环境和 WebSphere Commerce 共存但并不共享有用资源。

这些方案的迁移路径会在以下部分中描述。在每种情况中，环境可以使用远程 WebSphere Commerce 数据库或本地（独立配置）数据库。此外，每个环境都可用于开发商店前台有用资源（如商店的外观）、商店后台办公室逻辑（如定制命令和企业 bean）或商店前台有用资源和后台办公室逻辑的组合。

使用环境 A 或 B 迁移至 WebSphere Commerce - Express Developer Edition V5.5 的高级步骤如下：

1. 请确保您已经复查并理解了第 3 页的『迁移的先决条件』和第 4 页的『迁移环境』中所描述的信息。
2. 如第 19 页的『备份 WebSphere Commerce 和 WebSphere Commerce 开发环境有用资源』中所概述，备份现有的 WebSphere Commerce Studio V5.4 和 WebSphere Commerce 5.4 有用资源。
3. 如第 19 页的『升级 DB2 产品级别』中所概述，升级到 DB2 - Express V8.1。
4. 将 WebSphere Commerce Studio V5.4 升级至 WebSphere Commerce - Express Developer Edition V5.5。详细的安装和升级步骤如第 20 页的『升级 WebSphere Commerce 开发环境』中所述。这包括安装 WebSphere Studio Application Developer V5.1。
5. 安装 IBM WebSphere Commerce 5.5.0.2 修订包。该修订包包括用于迁移过程的修订和增强功能。该修订包可从 WebSphere Commerce 支持站点获取。单击对应于您正在使用的修订版的链接，并查看软件下载部分。下载该修订包，并遵循所提供的用于安装的指示信息。在没有安装该修订包的情况下，不应继续进行本指南中的迁移步骤（诸如迁移实例或数据库）。
6. 如第 22 页的『迁移 WebSphere Commerce 开发环境数据库模式』中所述，将现有的 WebSphere Commerce Studio V5.4 数据库模式迁移至 WebSphere Commerce - Express Developer Edition V5.5 级别。
7. 如第 23 页的『迁移 WebSphere Commerce 实例』中所述，将现有 WebSphere Commerce 实例升级至 WebSphere Commerce 5.5 级别。

注：实例迁移涉及自动将商店前台或 Web 有用资源（即，JSP 文件、HTML 页面或图形）从 VisualAge[®] for Java 复制至 WebSphere Studio Application Developer。但是，您也应确保您已经考虑到在 WebSphere Commerce Studio V5.4 中为商店前台创建的所有有用资源。

将现有的 WebSphere Commerce 实例迁移至 WebSphere Commerce 5.5 级别包括以下步骤：

- 迁移 instance.xml 文件。
 - 将商店前台或 Web 有用资源从 VisualAge for Java 复制至 WebSphere Studio Application Developer Web 项目中。这包括您用于商店前台的任何 JSP 文件、HTML 页面或图形。
 - 将包含非企业 bean 定制代码的 JAR 文件抽取到称为 WebSphereCommerceServerExtensionsLogic 的 WebSphere Studio Application Developer Java 项目中。
 - 迁移在运行实例迁移脚本时复制的 JSP 文件，以符合 JSP 1.2 规范。
- ___ 8. 将 VisualAge for Java 中包含的 WebSphere Commerce 定制代码转换到 WebSphere Studio Application Developer V5.1。详细的步骤如第 27 页的『转换定制的或扩展的代码』中所述。
- ___ 9. 如第 37 页的『迁移支付实例和数据库』中所概述，迁移 WebSphere Commerce 支付数据库和实例。对于 WebSphere Commerce 5.4，安装支付数据库的机器要不同于包含 WebSphere Commerce 的机器。对于 WebSphere Commerce 5.5，可以选择将支付数据库本地安装到与 WebSphere Commerce 开发环境相同的机器上。
- ___ 10. 完成第 41 页的第 4 章，『迁移后操作』中所概述的迁移后步骤。这包括卸载一些 WebSphere Commerce 组件。

环境 A - 完全开发环境

在完全开发环境中迁移的情况下，WebSphere Commerce 开发环境用于开发商店有用资源和逻辑；它不用于主管生产商店。这样迁移的最终结果是与 WebSphere Commerce 开发环境的新安装类似的结构。即，在完全开发环境中，在迁移之后 WebSphere Commerce 并不存在于机器上而只有 WebSphere Commerce 开发环境存在于机器上。在此情况下，将迁移您现有的开发商店并安装“时尚潮流”样本商店。

环境 B - 开发环境与 WebSphere Commerce 共存但不共享有用资源

在开发环境与 WebSphere Commerce 共存（但不共享有用资源）的情况下，最终结果应该是一台包含开发环境的机器完全独立于 WebSphere Commerce 5.5 的本地实例。即，在此环境中，WebSphere Commerce 和 WebSphere Commerce 开发环境存在于一台机器上。开发环境和 WebSphere Commerce 都使用数据库；数据库内容相同但并不共享。在迁移完成后，每个数据库都应继续相同，XML 文件也相同。请注意，您不必迁移数据库两次。而在迁移 WebSphere Commerce 开发环境之后，应该备份已迁移的数据库，然后创建新的数据库（例如，Mall2）并恢复到新的数据库。关于此情况的详细信息，请参阅步骤第 42 页的 3 这一步。

第 2 章 影响 WebSphere Commerce 开发环境迁移的代码更改

一旦您理解了迁移的先决条件和可能的迁移环境，您需要复查影响迁移的 WebSphere Commerce Studio V5.4 的代码更改。本章讨论在迁移 WebSphere Commerce 开发环境之前应注意的代码更改。

WebSphere Commerce 5.4 和 WebSphere Commerce Studio 5.4 安装环境

对于 WebSphere Commerce Studio V5.4，您可以选择在同一机器上安装 WebSphere Commerce 开发环境和 WebSphere Commerce 以使它们共存。还可以选择在安装 VisualAge for Java 时安装在 WebSphere Test Environment 中使用的样本商店。如果选择了样本商店，则会创建以下项：

- 数据库；例如，缺省的 VAJ_DEMO 数据库。安装期间，您可以选择更改此项。
- 商店目录结构。
- 实例 XML 文件；例如，缺省的 vaj_demo.xml 实例 XML 文件。安装期间，您可以选择更改此项。

这些有用资源由 WebSphere Test Environment 使用。WebSphere Commerce Studio V5.4 的可用样本商店为“流行时尚”商店且 WebSphere Commerce 开发环境安装程序使用主 XML 文件将样本商店填充数据库。

或者，对于 WebSphere Commerce Studio V5.4，您可以在独立的机器上维护 WebSphere Commerce 开发环境和 WebSphere Commerce。

报价请求 (RFQ)

报价请求 (RFQ) 是买方组织向卖方组织请求报价以获取给定产品或产品集的合适价格的过程。买方组织中的买方可在卖方组织产品目录中找不到他们想要的产品。在 WebSphere Commerce 5.5 中，RFQ 请求工具已更改为允许买方面对“订制”商品创建 RFQ。要完全指定“订制”商品的属性，个性化属性将同任何其它 RFQ 商品一样使用。关于 RFQ 迁移的详细信息，包括新的或已更改的（命令或数据库表）有用资源以及如何迁移 RFQ 请求工具，请参阅《WebSphere Commerce 迁移指南》中的『报价请求 (RFQ) 迁移』部分。

此外，关于支持的、不推荐的、保留的和已撤销的 API 的信息，请参阅 WebSphere Commerce 5.5 开发联机帮助。

协作工作空间

在 WebSphere Commerce 5.4 中，协作工作空间功能是由 CollabManagerBean 企业 bean 提供的。在 WebSphere Commerce 5.5 中，此企业 bean 已由 CollabManager 类替换。如果您已定制了使用 CollabManagerBean 企业 bean 的代码，则请执行以下操作来迁移代码，以与新的 CollabManager 类一起工作：

1. 编辑类或软件包导入语句以引用新的 CollabManager 类，如下：
 - a. 除去您的代码中引用 CollabManagerBean 类的 import 语句（例如，`import com.ibm.commerce.collaboration.objects.CollabManagerBean;`）。
 - b. 插入引用 CollabManager 类的新的 import 语句（例如，`import com.ibm.commerce.collaboration.manager.*;`）。
2. 编辑声明和实例语句，以使用 CollabManager 类型。
3. 编辑对象根据以下 API 映射表调用的方法。可能要求您根据映射新 API 的方式添加额外的代码。

注：等同 API 的参数或返回类型可能与 CollabManagerBean 中的不同。关于详细信息，请检查 CollabManager 类 API 引用。

下表举例说明 CollabManagerBean 类与 CollabManager 类之间的 API 映射（请注意，表中没有列出企业 bean 规范所要求的接口）：

表 1. CollabManagerBean 类与 CollabManager 类之间的 API 映射

| CollabManagerBean 中的方法 | CollabManager 类中的等同方法 |
|---|---|
| addAuthorGroupMember (CollabSpaceBean csbean) | 以下各项中的任一项： <ul style="list-style-type: none">• addCWMembers(CollabWorkspaceInfo csbean)• addCWMembers(String collabWorkspaceId, Vector memberDNs, String role)• addCWMember(String collabWorkspaceId, String memberDN, String role) |
| addManagerGroupMember (CollabSpaceBean csbean) | 以下各项中的任一项： <ul style="list-style-type: none">• addCWMembers(CollabWorkspaceInfo csbean)• addCWMembers(String collabWorkspaceId, Vector memberDNs, String role)• addCWMember(String collabWorkspaceId, String memberDN, String role) |

表 1. *CollabManagerBean* 类与 *CollabManager* 类之间的 API 映射 (续)

| | |
|--|---|
| addReaderGroupMember (CollabSpaceBean csbean) | 以下各项中的任一项: <ul style="list-style-type: none"> • addCWMembers(CollabWorkspaceInfo csbean) • addCWMembers(String collabWorkspaceId, Vector memberDNs, String role) • addCWMember(String collabWorkspaceId, String memberDN, String role) |
| createCollabSpace (CollabSpaceBean cspaceBean, java.lang.String creatorDN) | createCollabWorkspace(CollabWorkspaceInfo cspaceBean) |
| deleteAuthorGroupMember (CollabSpaceBean csbean) | 以下各项中的任一项: <ul style="list-style-type: none"> • removeCWMembers(CollabWorkspaceInfo csbean) • removeCWMembers(String cwId, Vector memberDNs) • removeCWMember(String cwId, String memberDN) |
| deleteCollabSpace (java.lang.String collabSpaceID) | removeCollabWorkpace(String collabSpaceID) |
| deleteManagerGroupMember (CollabSpaceBean csbean) | 以下各项中的任一项: <ul style="list-style-type: none"> • removeCWMembers(CollabWorkspaceInfo csbean) • removeCWMembers(String cwId, Vector memberDNs) • removeCWMember(String cwId, String memberDN) |
| deleteReaderGroupMember (CollabSpaceBean csbean) | 以下各项中的任一项: <ul style="list-style-type: none"> • removeCWMembers(CollabWorkspaceInfo csbean) • removeCWMembers(String cwId, Vector memberDNs) • removeCWMember(String cwId, String memberDN) |
| deleteCollabSpace (java.lang.String collabSpaceID) | getCollabWorkspaceDetails(String collabSpaceID) |
| isUserDNAuthorGroupMember (java.lang.String userDN, java.lang.String collabSpaceID) | getCWMemberRole(String memberDN, String cwId) |
| getCWMemberRole (String memberDN, String cwId) | getCWMemberRole(String memberDN, String cwId) |

表 1. *CollabManagerBean* 类与 *CollabManager* 类之间的 API 映射 (续)

| | |
|--|---|
| isUserDNReaderGroupMember (java.lang.String userDN, java.lang.String collabSpaceID) | getCWMemberRole(String memberDN, String cwId) |
| listAllCollabSpaces() | listCollabWorkspaces() |
| listAuthorGroupMembers (java.lang.String collabSpaceID) | listCWMembers(String collabSpaceID, String role) |
| listCollabSpaceForUserDN (java.lang.String userDN) | listCollabWorkspaces(String userDN) |
| listCollabSpaceTemplates() | listCWTemplates() |
| listCWTemplates() | listCWTemplates() |
| listReaderGroupMembers (java.lang.String collabSpaceID) | listCWMembers(String collabSpaceID, String role) |
| modifyAuthorGroupMembers (CollabSpaceBean csbean) | setCWMemberRole(String memberDN, String cwId, String role) |
| modifyCollabSpaceDescription (java.lang.String collabSpaceID, java.lang.String description) | changeCWDescription(String collabSpaceID, String description) |
| changeCWDescription (String collabSpaceID, String description) | changeCWDescription(String collabSpaceID, String description) |
| changeCWDescription (String collabSpaceID, String description) | setCWMemberRole(String memberDN, String cwId, String role) |
| TESTaddAuthorGroupMembers (java.lang.String csID) | 不适用; 不等同 |
| TESTaddAuthorGroupMembers (java.lang.String csID, java.lang.String userDN) TESTaddManagerGroupMembers (java.lang.String csID) | 不适用; 不等同 |
| TESTaddManagerGroupMembers (java.lang.String csID, java.lang.String userDN) | 不适用; 不等同 |
| TESTaddReaderGroupMembers (java.lang.String csID) TESTaddReaderGroupMembers (java.lang.String csID, java.lang.String userDN) | 不适用; 不等同 |

表 1. *CollabManagerBean* 类与 *CollabManager* 类之间的 API 映射 (续)

| | |
|--|----------|
| TESTcreateCollabSpace (java.lang.String bProcType, java.lang.String bProcID) | 不适用; 不等同 |
| TESTdeleteAuthorGroupMembers (java.lang.String csID) | 不适用; 不等同 |
| TESTdeleteAuthorGroupMembers (java.lang.String csID, java.lang.String userDN) | 不适用; 不等同 |
| TESTdeleteManagerGroupMembers (java.lang.String csID) | 不适用; 不等同 |
| TESTdeleteManagerGroupMembers (java.lang.String csID, java.lang.String userDN) | 不适用; 不等同 |
| TESTdeleteReaderGroupMembers (java.lang.String csID) | 不适用; 不等同 |
| TESTdeleteReaderGroupMembers (java.lang.String csID, java.lang.String userDN) | 不适用; 不等同 |
| TESTmodifyAuthorGroupMembers() | 不适用; 不等同 |
| TESTmodifyManagerGroupMembers() | 不适用; 不等同 |
| TESTmodifyReaderGroupMembers() | 不适用; 不等同 |

下表列出了协作工作空间的其它各种 API 映射:

表 2. 协作工作空间的其它 API 映射

| CollabManagerBean 中的方法 | 等同的实现 |
|-------------------------------|--|
| getCollabAppServerIP() | WcsApp.configProperties.getValue (ECConstants.EC_COLLAB_QP_HOST) |
| getCollabAppServerPort() | WcsApp.configProperties.getValue (ECConstants.EC_COLLAB_QP_HTTP_PORT) |

4. 由于 *CollabManager* 不是企业 bean, 请除去先前用于企业 bean 编码处理企业 bean 所引起的异常的所有 try 和 catch 条目。除去 try 和 catch 条目可避免编译错误。
5. 编译代码并修正由 IDE 引起的任何编译错误。

数据库模式的差异

有关 WebSphere Commerce 5.5 和 WebSphere Commerce 开发环境的数据库模式信息的详细信息，请参阅 WebSphere Commerce 开发联机帮助。一旦启动了联机帮助，请选择 **WebSphere Commerce 开发信息 > 参考 > 数据 > 数据库模式**。从此处选择**数据库表**以查看所有数据库模式信息的依字母顺序的列表。选择**本发行版中的数据库更改**以查看 V5.5 的模式更改。联机帮助还提供数据模型信息。



确保您访问 WebSphere Commerce 技术库

(<http://www.ibm.com/software/commerce/library/>) 以获取最新版本的 WebSphere Commerce 文档，包括对联机帮助中数据库模式信息的任何更新。

API 区别

关于 WebSphere Commerce 5.5 和 WebSphere Commerce 开发环境的 API 信息的详细信息，请参阅 WebSphere Commerce 开发联机帮助。一旦启动了联机帮助，请选择 **WebSphere Commerce 开发环境 > 参考 > API 信息**。此联机帮助还提供关于以下问题的 API 信息：

- V5.5 支持哪些 API。
- 为 V5.5 更改了哪些 API。如果您使用或定制了这些 API，请确保复查已更改的版本。有了 WebSphere Commerce 5.5 和 WebSphere Commerce 开发环境，您将迁移至企业 bean 级别 1.1。关于这种实现以及企业 bean 查找程序助手实现的更多信息，请参阅第 13 页的『企业 bean 从 1.0 迁移至 1.1 规范』和第 27 页的『VisualAge for Java 和 WebSphere Studio Application Developer 之间的不同』。
- 仅为 IBM 内部使用和出于引用目的而保留哪些 API。
- 对 V5.5 不推荐使用哪些 API。如果您使用了任何不推荐的 API，则请参阅联机帮助中关于替换 API 以用于 V5.5 的信息。
- 已从 V5.5 撤销且不再支持哪些 API。
- 哪些 API 不再与 V5.5 兼容，但 V5.1 和 V5.4 仍支持。

还包括数据 bean、企业 bean、数据库表的交叉引用信息，以及命令、任务和数据库表。



确保您访问 WebSphere Commerce 技术库

(<http://www.ibm.com/software/commerce/library/>) 以获取最新版本的 WebSphere Commerce 文档，包括对联机帮助中 API 信息的任何更新。

企业 bean 从 1.0 迁移至 1.1 规范

从 1.0 转换到 1.1 规范不会影响访问 bean 层或您的代码；您可以在 WebSphere Commerce 开发环境中使用先前使用过的代码。从 1.0 转换到 1.1 会影响企业 bean。下表举例说明对 WebSphere Commerce 开发环境所作的企业 bean 更改，从而将影响从 1.0 到 1.1 规范的迁移。

表 3. 企业 bean 更改以及对 WebSphere Commerce 开发环境的影响

| 对 WebSphere Commerce 开发环境所作的更改 | 更改如何影响企业 bean |
|--|--|
| 将企业 bean “impl” 重命名为 “bean” 基本类 | 没有影响，因为这种更改涉及命名和打包约定。 |
| 从 BeanBase 类中除去 “实现 java.io.Serializable” | 虽然此更改并不直接与企业 bean 1.1 规范相关，您将需要对企业 bean 作此更改，以保证在迁移后正常工作。 |
| 除去序列版本 UID | 虽然此更改并不直接与企业 bean 1.1 规范相关，您将需要对企业 bean 作此更改，以保证在迁移后正常工作。 |
| 确保 ejbCreate() 返回主键对象 | 您必须作此更改才能符合企业 bean 1.1 规范。 |
| 从 ejbCreate()、ejbPostCreate()、ejbActivate()、ejbPassivate()、ejbStore()、ejbRemove() 和所有远程方法中除去 RemoteException | 您必须作此更改才能符合企业 bean 1.1 规范。 |
| 从 bean 或 bean 基本类中除去 getEntityContext()、setEntityContext()、unsetEntityContext() | 没有影响，因为这是内部清除修正。 |
| 将所有 ejbCreate() 和 ejbPostCreate() 代码移至 bean 基本类 | 没有影响，因为这是内部清除修正。 |
| 匹配 ejbPostCreate() 和 ejbCreate() 方法 | 您必须作此更改才能符合企业 bean 1.1 规范。 |
| 添加 WCSecurity 角色 | 虽然此更改并不直接与企业 bean 1.1 规范相关，您将需要对企业 bean 作此更改，以保证在迁移后正常工作。 如果在 WebSphere Studio Application Developer 中打开了安全性，则企业 bean 安全角色是需要的。 |

表 3. 企业 bean 更改以及对 WebSphere Commerce 开发环境的影响 (续)

| | |
|---|--|
| <p>将企业 bean 事务设置指定为“必需”</p> | <p>虽然此更改并不直接与企业 bean 1.1 规范相关，您将需要对企业 bean 作此更改，以保证在迁移后正常工作。</p> <p>此企业 bean 事务设置是必需的。该值取决于企业 bean 的用法。</p> |
| <p>为每个企业 bean 将企业 bean 隔离级别指定为“可重复读取”</p> | <p>虽然此更改并不直接与企业 bean 1.1 规范相关，您将需要对企业 bean 作此更改，以保证在迁移后正常工作。</p> <p>此企业 bean 隔离级别是必需的。该值取决于使用的数据库类型。</p> |
| <p>更改 JNDI 名称</p> | <p>没有影响，因为这种更改涉及命名和打包约定。</p> |
| <p>删除 xyzBeanFinderHelper 接口</p> | <p>虽然此更改并不直接与企业 bean 1.1 规范相关，您将需要对企业 bean 作此更改，以保证在迁移后正常工作。</p> <p>BeanFinderHelper 接口已废弃。所有查找程序均在 ibm-ejb-jar-ext.xmi 中定义。</p> |
| <p>将 xyzBeanFinderObjectImpl 重命名为 xyzBeanFinderObjectBase</p> | <p>虽然此更改并不直接与企业 bean 1.1 规范相关，您将需要对企业 bean 作此更改，以保证在迁移后正常工作。</p> <p>BeanFinderHelper 接口已废弃。所有查找程序均在 ibm-ejb-jar-ext.xmi 中定义。</p> |

检查 MSGTYPES 表中的独特索引

MSGTYPE_ID 和 NAME 列是 MSGTYPES 表中新增的。MSGTYPE_ID 列是主键，而 NAME 列是独特索引。如果您已创建您自己的消息类型，则请确保它们不与这些新列冲突。如果发生冲突，您可以修改 MSGTYPE_ID 的值（通过向数字加 10,000），或修改 NAME 的值。您将需要重新编译那些引用定制消息类型的命令。否则，您从 WebSphere Commerce 5.4 迁移数据时可能会遇到问题。

WebSphere 应用程序命令框架

对于 WebSphere Commerce 5.4，使用 San Francisco 命令框架。但 WebSphere Commerce 5.5 已更改为使用 WebSphere 应用程序命令框架。因此，在迁移定制或扩展代码时，注意以下几点是很重要的：

- 为了使新的 WebSphere Application Server V5 运行时库生效，从 WebSphere Commerce 5.4 控制器命令或任务命令扩展而来的所有客户命令都需要重新编译。
- 要求控制器命令和任务命令的调用者设置命令上下文。

J2EE 连接器体系结构

对于 WebSphere Commerce Suite 5.1 和 WebSphere Commerce 5.4，使用公共连接器框架。但是对于 WebSphere Commerce 5.5，则更改为使用 J2EE 连接器体系结构。如果您使用 WebSphere Commerce Suite 5.1 或 WebSphere Commerce 5.4 的连接器，则将需要重新编写它。关于 J2EE 框架和 J2EE 连接器体系结构的详细信息，请参阅 Sun Microsystems 的以下 Web 站点：

<http://java.sun.com/j2ee/connector>。

此外，WebSphere Commerce 开发环境还提供一个称为“样本适配器”的示例连接器，它是一个样本 J2EE 连接器体系结构适配器（连接器）。

第 2 部分 迁移流程

本部分描述从 WebSphere Commerce Studio V5.4 迁移至 WebSphere Commerce - Express Developer Edition V5.5 的典型迁移流程。迁移过程包括一系列步骤。请参阅此部分中的以下各章，获取关于完成每个迁移步骤的指示信息。

第 3 章 迁移 WebSphere Commerce 开发环境

本章描述将 WebSphere Commerce 开发环境升级至 WebSphere Commerce - Express Developer Edition V5.5 的详细步骤。

注：在升级 WebSphere Commerce 开发环境之前，请确保已复查第 1 页的第 1 部分，『迁移前』中概述的信息（包括所概述的硬件和软件需求）。

备份 WebSphere Commerce 和 WebSphere Commerce 开发环境有用资源

在升级任何软件或将任何定制代码转换至 WebSphere Commerce 开发环境级别之前，请确保 WebSphere Commerce 和 WebSphere Commerce 开发环境有用资源都已备份，如下所示：

- 备份以下 WebSphere Commerce 有用资源：
 - WebSphere Commerce 目录树
 - WebSphere Commerce 数据库

有关备份 WebSphere Commerce 目录树和数据库的步骤，请参阅对应于 WebSphere Commerce 版本（您要从此版本迁移）的《*WebSphere Commerce 迁移指南*》中的『备份 WebSphere Commerce 5.4』部分。

- 备份 WebSphere Commerce 开发环境有用资源。

升级 DB2 产品级别

如果使用 DB2 作为数据库管理系统，则在您已经备份了 WebSphere Commerce 和 WebSphere Commerce Studio 有用资源之后，必须在升级至 WebSphere Commerce 开发环境之前先升级至 DB2 - Express V8.1 并对其进行配置。要将 DB2 V8.1 安装和配置为 WebSphere Commerce Studio 数据库，请遵循以下高级步骤：

1. 记录 DB2 迁移之前数据库的配置设置。
2. 更改诊断错误级别。
3. 使 DB2 服务器变为脱机，以便进行 DB2 迁移。
4. 验证数据库已就绪，可进行 DB2 迁移。
5. 备份数据库。
6. 可选：如果您将使用复制，则必须将所有 DB2 日志文件归档。
7. 安装 DB2 企业服务器版版本 8.1。
8. 迁移数据库。

9. 可选: 迁移 DB2 Explain 表。

有关以上步骤的详细信息, 请参阅 *Quick Beginnings for DB2 Servers* 文档中的第 2 部分 “Migrating your DB2 server”。该文档可从 DB2 技术支持库 (<http://www.ibm.com/software/data/db2/library/>) 中获取; 您将需要浏览至 DB2 Technical Support Version 8 Information Center。

升级 WebSphere Commerce 开发环境

一旦备份了 WebSphere Commerce 和 WebSphere Commerce 开发环境有用资源, 就可以升级 WebSphere Commerce 开发环境。这包括安装 WebSphere Studio Application Developer V5.1, 并且安装过程将把系统升级至 DB2 通用数据库™ 企业版版本 8.1。

注: 重要的是不要卸载 VisualAge for Java 或较早版本的 WebSphere Studio Application Developer (用于 WebSphere Commerce Studio V5.4), 因为您将需要将有用资源和项目从这些应用程序导出到 WebSphere Studio Application Developer V5.1 中。但为了避免冲突, 请确保将 WebSphere Commerce - Express Developer Edition V5.5 安装在不同于 VisualAge for Java 或较早版本的 WebSphere Studio Application Developer 的目录中。还要注意, VisualAge for Java 项目将不会在 WebSphere Studio Application Developer V5.1 工作区中工作。

WebSphere Commerce - Express Developer Edition V5.5 安装程序完成以下任务:

- 安装 WebSphere Commerce 开发环境代码 (包括 WebSphere Studio Application Developer V5.1), 并且如果您使用 DB2 作为数据库管理系统, 则安装过程将把系统升级至 DB2 通用数据库企业版版本 8.1。
- 创建带有以下子目录的名为 \WebSphere\CommerceDev55 的新目录结构:
 - \properties
此目录包含 WebSphere Commerce 5.5 级别代码的属性文件。
 - \bin
此目录包含数据库迁移脚本。
- 将原始 wcs_instances 文件复制到新的 \WebSphere\CommerceDev55\instances 目录结构中, 并更新文件以反映新的结构。

要安装 WebSphere Commerce 开发环境, 请执行以下操作 (关于详细的安装说明, 请参阅《*WebSphere Commerce Studio 安装指南*》):

- 1. 将 WebSphere Commerce - Express Developer Edition CD 1 插入 CD-ROM 驱动器中。如果已经启用了自动播放，则安装过程开始。如果未启用，请运行 setup.exe。
- 2. 从系统环境路径除去 \WebSphere\CommerceDev\bin 目录路径。要除去此目录，请执行以下操作：
 - a. 访问系统特性窗口。
 - b. 选择环境变量。在系统变量下，选择路径。
 - c. 编辑此路径并除去旧的 Commerce Studio 开发路径（例如，WebSphere\CommerceDev\bin）。
 - d. 单击确定。
- 3. 遵循安装向导中的提示。
- 4. 一旦 WebSphere Commerce 开发环境安装完成，并且已经在 VisualAge for Java 中创建了定制代码，则定制代码需要迁移至 WebSphere Studio Application Developer V5.1。

注： 仅当您要迁移后端有用资源（如命令或数据 bean）时才需要此步骤。

为了使此过程更为方便，现在启动 VisualAge for Java 并将以下项目和组导出到 JAR 文件，如『将任务命令、控制器命令和数据 bean 导出到 JAR 文件』和『将企业 bean 导出至 JAR 文件』中所概述。（关于转换定制代码和逻辑的详细信息在第 27 页的『VisualAge for Java 和 WebSphere Studio Application Developer 之间的不同』中有描述。）

将任务命令、控制器命令和数据 bean 导出到 JAR 文件

启动 VisualAge for Java 并导出包含新建的或扩展的任务命令、控制器命令或数据 bean 的定制代码的 VisualAge for Java 项目。在此情况下，必须如下将包含此代码的项目导出到 JAR 文件：

1. 在“VisualAge for Java 工作台”窗口中，选择并右键单击项目名，然后单击**导出**。
2. 选择 **Jar 文件** 单选按钮并单击**下一步**。
3. 指定 JAR 文件的名称。
4. 选择 **.java** 复选框以导出 Java 文件。
5. 按需要填写其它字段。请参阅 VisualAge for Java 联机帮助，获取关于如何执行此任务的更多信息。

将企业 bean 导出至 JAR 文件

启动 VisualAge for Java 并如下将任何定制企业 bean 都导出至 JAR 文件（您必须将包含定制企业 bean 的所有企业 bean 组都导出到单个 JAR 文件）：

如果拥有最少的定制企业 bean

如果您拥有最小数目的定制企业 bean（即，约为十个或十个以下），则建议您使用 WebSphere Studio Application Developer V5.1 重新创建企业 bean。有关如何实现此任务的详细信息，请参阅 *WebSphere Studio Application Developer V5.1 Migration Guide*。

如果拥有许多定制企业 bean

启动 VisualAge for Java，并通过使用 Export Tool for Enterprise Java Beans 1.1 来导出任何定制企业 bean。您必须将包含定制企业 bean 1.1 的所有企业 bean 组都导出到单个 JAR 文件，如下：



如果在导出过程中目标数据库类型选择框被禁用，则表示您已丢失企业模式映射。

1. 在 VisualAge for Java 中，向工作空间添加 **Export Tool for Enterprise Java Beans 1.1** 功能。
2. 在工作台的 EJB 页面上，右键单击一个或多个企业 bean 组并单击导出 > **EJB 1.1 JAR**。
3. 按需要填充字段。确保已选择 **.java** 复选框。
4. 确保选择了数据库。
5. 单击**完成**。
6. 将 JAR 文件导入到 WebSphere Studio Application Developer V5.1 中。

应用 IBM WebSphere Commerce 5.5.0.2 修订包

在安装 WebSphere Commerce 开发环境之后，您需要安装 IBM WebSphere Commerce 5.5.0.2 修订包。该修订包包括用于迁移过程的修订和增强功能。该修订包可从 WebSphere Commerce 支持站点获取。单击对应于您正在使用的修订版的链接，并查看软件下载部分。下载该修订包，并遵循所提供的用于安装的指示信息。在没有安装该修订包的情况下，不应继续进行本指南中的迁移步骤（诸如迁移实例或数据库）。

迁移 WebSphere Commerce 开发环境数据库模式

一旦升级了 WebSphere Commerce 开发环境，则下一步就是将 WebSphere Commerce 开发环境数据库模式迁移至 WebSphere Commerce - Express Developer Edition V5.5 级别。

要迁移 WebSphere Commerce 开发环境数据库模式，请如下运行名为 migratedb54.bat 的数据库模式迁移脚本：

- ___ 1. 运行数据库模式迁移脚本之前，请确保 setenv.bat 文件中的 JAVA_HOME 环境变量指向机器上现有版本的 IBM Developer Kit, Java Technology Edition。setenv.bat 文件通常位于 WCDE_installdir\Commerce\bin 目录中。
例如，如果 IBM Developer Kit, Java Technology Edition 安装在机器上的 \eclipse\jre 目录中，则将 JAVA_HOME 设置为
JAVA_HOME=WCDE_installdir\eclipse\jre。缺省情况下，JAVA_HOME 指向 WebSphere Studio 5.1 安装目录（例如
JAVA_HOME=C:\WebSphere\Studio5\runtimes\base_v5\java\jre）。
- ___ 2. 运行数据库模式迁移脚本之前，请确保已删除所有定制的外键引用。这将防止在迁移过程中可能遇到的各种数据库错误。数据库模式迁移之后，您应手工重新构建这些外键引用。
- ___ 3. 要运行数据库迁移脚本，请参阅《WebSphere Commerce 迁移指南》中的『迁移数据库』部分。

注：

- a. 在 WebSphere Commerce 开发环境中，数据库模式迁移脚本可在 WCDE_installdir\commerce\bin 目录中获取。
- b. 您可以运行 premigratedb 脚本，如《WebSphere Commerce 迁移指南》中所述，您可以在命令提示符下完成此操作；但 migratedb 脚本应从 DB2 命令行处理器窗口运行。

迁移 WebSphere Commerce 实例

一旦已将 WebSphere Commerce 开发环境数据库模式迁移至 WebSphere Commerce - Express Developer Edition V5.5 级别，则下一步就是迁移 WebSphere Commerce 实例。

在 WebSphere Commerce 实例迁移过程期间，将完成以下任务：

- 迁移 instance.xml 文件。
- 将商店前台或 Web 有用资源从 VisualAge for Java 复制到 WebSphere Studio Application Developer Web 项目。这包括您用于商店前台的任何 JSP 文件、HTML 页面或图形。
- 将包含非企业 bean 定制代码的 JAR 文件抽取到称为 WebSphereCommerceServerExtensionsLogic 的 WebSphere Studio Application Developer Java 项目中。
- 迁移运行数据库迁移脚本时复制的 JSP 文件，以符合 JSP 1.2 规范。

注: 实例迁移涉及自动将商店前台或 Web 有用资源（即，JSP 文件、HTML 页面或图形）从 VisualAge for Java 复制至 WebSphere Studio Application Developer。但是，您也应确保您已经考虑到在 WebSphere Commerce Studio V5.4 中为商店前台创建的所有有用资源。

要迁移实例，请如下运行两个实例迁移脚本 `wstudioimenv.bat` 和 `wstudioim.bat`：

1. 如果要迁移商店前台有用资源，请启动 WebSphere Studio Application Developer V5.1，为商店有用资源创建 Web 项目并注意该项目的名称。您将需要在运行 `wstudioim.bat` 实例迁移脚本之前创建此项目，因为您需要在运行此脚本时指定 Web 项目名称作为参数值。

注: 如果仅要迁移后端有用资源，则可跳过此步骤。

2. 请确保为 `WC_installdir\xml` 目录中找到的 `product.xml` 文件中的迁移路径正确设置了所有值。`product.xml` 的值由 WebSphere Commerce 5.5 安装程序填写，但建议您检查这些值，以确保它们对于您的环境是正确的。特别地，请确保 `<migrationFrom>` 元素和此元素的所有子元素均包含值。如果它们不包含值，则必须手工插入值。这些值可以从用于 WebSphere Commerce 5.4 的 `product.xml` 文件中获取；请参考 `<CommerceServer>` 标记。

`<migrationFrom>` 元素应该具有以下格式：

```
<migrationFrom>
  <edition>
    <name>name</name>
  </edition>
  <version>5</version>
  <release>rel</release>
  <modification>mod</modification>
  <fixpak>fixpak</fixpak>
  <path>path</path>
  <altpath>long_path</altpath>
</migrationFrom>
```

其中：

name 您正迁移的 WebSphere Commerce 的起始版本。名称可以是以下几项中的任何一个：

- **Pro** - 代表从 WebSphere Commerce Professional Edition 迁移
- **Business** - 代表从 WebSphere Commerce Business Edition 迁移

rel 您正迁移的 WebSphere Commerce 的起始发行版。对于 WebSphere Commerce 5.4，指定 **4**。

mod 您正迁移的 WebSphere Commerce 的起始修订级别。修订级别的值总是 **0**。例如，要从 WebSphere Commerce 5.4.0.1 进行迁移，则 `product.xml` 文件中的修改级别为 **0**。

fixpak 您正迁移的 WebSphere Commerce 的起始修订包级别。例如，要从 WebSphere Commerce 5.4.0.4 迁移，则 product.xml 文件中的修订包级别是 **4**。

path 您正迁移的 WebSphere Commerce 的起始版本的安装路径。这通常与 *long_path* 值相同。例如，要从 WebSphere Commerce 5.4 迁移，则路径是 *WC54_installdir*，如 *c:\WebSphere\WebSphere Commerce*。

long_path

您正迁移的 WebSphere Commerce 的起始版本的完整安装路径。这通常与 *path* 值相同。例如，要从 WebSphere Commerce 5.4 迁移，则路径是 *WC54_installdir*，如 *c:\WebSphere\WebSphere Commerce*。

为了您参考起见，如果您正从 WebSphere Commerce 5.4.0.1 Business Edition 迁移，则您的 product.xml 文件需要包含以下示例内容：

```
<migrationFrom>
  <edition>
    <name>Business</name>
  </edition>
  <version>5</version>
  <release>4</release>
  <modification>0</modification>
  <fixpak>1</fixpak>
  <path>c:\WebSphere\WebSphere Commerce</path>
  <altpath>c:\WebSphere\WebSphere Commerce</altpath>
</migrationFrom>
```

3. 打开 *WCDE_installdir\Commerce\bin\setenv.bat* 文件并确保以下设置的值正确：
 - *WAS_HOME* 设置为 *WAS_installdir\runtimes\base_v5*
 - *JAVA_HOME* 设置为 WebSphere Commerce 开发环境机器上的 *\eclipse\jre* 目录；例如，*WCDE_installdir\runtimes\base_v5\java\jre* 或 *C:\WebSphere\Studio5\runtimes\base_v5\java\jre*。
 - *WCS_HOME* 设置为 *WC_installdir*；即，您安装 WebSphere Commerce 5.5 的主目录。例如，*C:\Program Files\WebSphere\CommerceServer55*
4. 编辑 *WC_installdir\bin\wstudioimenv.bat* 文件并确保以下设置的值正确：
 - *SET™ WCIM_MIGRATE_FROM* 应为 *SET WCIM_MIGRATE_FROM=54*
 - *SET WC_PATH* 设置为 *WC_installdir*；即，您安装 WebSphere Commerce 5.5 的目录。例如，*C:\Program Files\WebSphere\CommerceServer55*
 - *SET ANT_PATH* 设置为 *WSAD_installdir\runtimes\base_v5\lib*
 - *SET WORK_DIR* 设置为 *WC_installdir* 的临时工作目录；例如，*WC_installdir\temp*
 - *SET LOG_FILE* 设置为记录关于实例迁移的信息的文件的名称。

- SET INSTANCE 是 WebSphere Commerce 5.5 实例的名称。

5. 从命令提示符切换至 *WCDE_installdir*\bin 目录并通过输入以下内容运行第一个脚本:

```
wstudioimenv.bat
```

6. 从命令提示符, 切换至 *WCDE_installdir*\bin 目录并通过输入以下一行内容来运行第二个脚本:

```
wstudioim.bat WSAD_workspacedir VAJ_installdir  
custom_code_JAR_file store_properties_dir
```

其中:

WSAD_workspacedir

WebSphere Studio Application Developer V5.1 工作区的路径, 在安装 WebSphere Commerce 开发环境时指定。例如, c:\WebSphere\workspace_db2。

VAJ_installdir

VisualAge for Java 的安装路径。缺省安装路径是 \Visual Age for Java。例如, d:\Visual Age for Java。

custom_code_JAR_file

包含诸如定制企业 bean、命令或数据 bean 之类定制代码的 JAR 文件。例如, d:\myCustomCode.jar。

store_properties_dir

先前版本的 WebSphere Commerce 的商店属性文件的路径。示例路径为 c:\WebSphere\CommerceServerDev\wc.ear\wcstores.war \WEB-INF\classes。

注: 如果仅要迁移商店前台有用资源, 则对于 *custom_code_JAR_file* 参数, 指定空值 (即, 使用双引号指示空值)。例如, 在此情况下, 可以执行实例迁移脚本, 如下所示:

```
wstudioim.bat WSAD_workspacedir VAJ_installdir ""  
store_properties_dir
```

7. 查看日志文件 *WCDE_installdir*/temp/logs/wstudioim.log 以查看程序是否成功运行。
8. 编辑迁移的实例 XML 文件如下:
 - a. 搜索 `WorkspacePath=""`
 - b. 在引号之间添加 WebSphere Commerce 开发环境工作空间路径 (例如, `WorkspacePath="D:\WEBSPH~1\WORKSP~1"`)。

9. 将定制企业 bean 导入到 WebSphere Studio Application Developer V5.1 中的 WebSphereCommerceServerExtensionsData 企业 bean 项目。如下启动 WebSphere Studio Application Developer V5.1 并导入在步骤 第 21 页的『将企业 bean 导出至 JAR 文件』中从 VisualAge for Java 导出的企业 bean JAR 文件:
 - a. 在 WebSphere Studio Application Developer 中, 从 J2EE 透视图选择文件 > 导入 > **EJB JAR 文件**。单击下一步。
 - b. 指定包含企业 bean 的 JAR 文件。
 - c. 通过选择称为 **WebsphereCommerceServerExtensionsData** 的现有项目, 指定将导入企业 bean 的企业 bean 项目。
 - d. 单击**下一步**并从 JAR 文件选择所有企业 bean 文件。
 - e. 单击**完成**。如果遇到任何错误(错误将列在“任务”视图中), 请参阅第 32 页的『将企业 bean 迁移至 WebSphere Studio Application Developer 的故障诊断技巧』以进行故障诊断。在已经导入企业 bean 之后, 请遵循第 32 页的『定位企业 bean 信息』中的指示信息以确定在何处查找企业 bean 和方法属性、模式及映射等。

转换定制的或扩展的代码

一旦迁移了 WebSphere Commerce 实例, 则下一步就是将包含在 VisualAge for Java 工作空间中的已扩展或已修改的业务逻辑和代码转换到 WebSphere Studio Application Developer V5.1 级别。为了帮助完成转换定制或扩展代码的复杂任务, 本指南中提供了两个教程。请参阅第 45 页的第 4 部分, 『转换代码的教程』, 了解更多关于迁移定制的或扩展的代码的信息。

VisualAge for Java 和 WebSphere Studio Application Developer 之间的不同

在您将代码从 VisualAge for Java 工作区转换至 WebSphere Studio Application Developer 之前, 了解这两个软件应用程序之间的不同之处非常重要。以下突出显示 VisualAge for Java 和 WebSphere Studio Application Developer 之间的主要区别:

- Enterprise Java Bean (EJB) 规范级别已由 1.0 更改为 1.1。即, 1.1 规范用于使用 WebSphere Studio Application Developer V5.1 的 WebSphere Commerce 开发环境。
- 对于 Web 应用程序, JSP 级别从 1.1 更改至 1.2。
- 对于 Web 应用程序, Servlet 级别保留为 2.3。
- 受支持的 Java 2 平台的级别已由 1.2 更改为 1.3 (编译器可将目标定为生成 1.4 级代码, 但 WebSphere Application Server 运行时环境仍是 1.3。)

- Visual Composition Editor 已经由 Visual Editor for Java 代替。
- VisualAge for Java 版本控制和专有源代码资源库已经由软件配置管理（SCM）支持插件代替。VisualAge for Java 工具 API 已经由 WebSphere Studio Workbench 插件体系结构代替。
- VisualAge for Java XML 工具已经由 WebSphere Studio Application Developer XML 工具代替。
- VisualAge for Java 项目概念已经由多种 WebSphere Studio Application Developer 项目代替。
- VisualAge for Java EJB AccessBeans（非数据 AccessBeans）中的转换器在 WebSphere Studio Application Developer EJB AccessBeans 中未提供。而在底层企业 bean 中使用企业 bean 转换器和编辑器代替。关于企业 bean 转换器和编辑器的更多信息，请参阅联机帮助文档。

转换扩展的或修改的 **WebSphere Commerce** 业务逻辑和代码

在从 WebSphere Commerce Studio V5.4 迁移至 WebSphere Commerce - Express Developer Edition V5.5 期间，将执行许多与您的 VisualAge for Java 环境相关的步骤。这些步骤如下：

- 从 VisualAge for Java 导出 Java 文件和项目资源文件。
- 启动 WebSphere Studio Application Developer 并创建新项目以包含代码。
- 将 Java 和项目资源文件导入 WebSphere Studio Application Developer。
- 将企业 bean 迁移至 WebSphere Studio Application Developer 中。
- 将定制的 WebSphere Commerce Server 企业 bean 迁移至 WebSphere Studio Application Developer 中。
- 设置测试环境并测试迁移的应用程序。

注：

1. 对于 WebSphere Commerce 5.4，使用公共连接器框架。但 WebSphere Commerce 5.5 已更改为使用 WebSphere 应用程序命令框架。因此，在迁移定制或扩展代码时，注意以下几点是很重要的：
 - 为了使新的 WebSphere Application Server V5 运行时库生效，从 WebSphere Commerce 5.4 控制器命令或任务命令扩展而来的所有客户命令都需要重新编译。
 - 要求控制器命令和任务命令的调用者设置命令上下文。
2. 关于支持的、不推荐的、保留的和已撤销的 API 的信息，请参阅 WebSphere Commerce 5.5 开发联机帮助。

从 VisualAge for Java 导出 Java 文件和项目资源文件

将代码转换到 WebSphere Studio Application Developer V5.1 级别的第一步涉及从 VisualAge for Java 导出 Java 文件和项目资源文件。

注:

1. 不支持从 VisualAge for Java 资源库成批迁移已进行版本控制的项目和资源。但是，您可以迁移 VisualAge for Java 工作区中的项目和资源。如果要将项目或资源的已进行版本控制的副本迁移到 WebSphere Studio Application Developer 中，就必须将其放入 VisualAge for Java 工作区，然后再进行迁移。
2. 如果项目包含多种数据（例如，企业 bean 和 Java 源代码文件），则应当根据数据的类型将数据分到不同的 JAR 中。

要将项目导出到 JAR 文件，请遵循以下步骤:

1. 如果您要导出的项目当前不在 VisualAge for Java 工作区中，请将它们添加到工作区。
2. 在“VisualAge for Java 工作台”窗口中，选择项目、右键单击并单击**导出**。
3. 选择 **Jar 文件** 单选按钮并单击**下一步**。
4. 输入 JAR 文件的名称。
5. 选中 **.java** 复选框以导出 Java 文件，并选中**资源**复选框以导出资源文件。
6. 按需要填写其它字段。请参阅 VisualAge for Java 联机帮助，获取关于如何完成字段填写的更多信息。

启动 WebSphere Studio Application Developer 并创建新项目以包含代码

在从 VisualAge for Java 导出 Java 文件和项目资源文件之后，将代码转换到 WebSphere Studio Application Developer 的下一步就是启动 WebSphere Studio Application Developer 并创建相应的项目。以下为一组常规迁移准则，帮助您决定应该将文件导入何种类型的 WebSphere Studio Application Developer 项目。强烈建议在创建任何项目或导入任何代码之前阅读 WebSphere Studio Application Developer 联机帮助并熟悉不同种类的 WebSphere Studio Application Developer 项目。

- 如果代码为 Web 应用程序的一部分，则应该如下将代码导入 Web 项目：
 - 将所有 Java 文件（例如控制器命令或任务命令）导入 Web 项目源目录（基于软件包语句的正确层次结构将由 WebSphere Studio Application Developer 自动创建）。
 - 将所有的资源文件（如 JSP 页面）导入 Web 项目的 Web content 目录。
- 如果代码是纯粹的 Java，则应将代码导入 Java 项目。
- 如果代码为企业 bean，则应该将代码导入企业 bean 项目。

将 Java 和项目资源文件导入 WebSphere Studio Application Developer

启动 WebSphere Studio Application Developer 并创建新项目以包含代码之后，将代码转换到 WebSphere Studio Application Developer 的下一步就是将 Java 和项目资源文件导入 WebSphere Studio Application Developer。

注： 当将文件导入 WebSphere Studio Application Developer 时，请确保将这些文件导入到了适当的目录。建议您在导入代码之前阅读 WebSphere Studio Application Developer 联机帮助并熟悉不同类型的 WebSphere Studio Application Developer 项目。这将帮助您确定哪些文件夹应该包含哪种代码。

要将 Java 和项目资源文件导入 WebSphere Studio Application Developer，请遵循这些步骤：

1. 打开 WebSphere Studio Application Developer 并切换至“资源”透视图。
2. 选择文件 > 导入 > **Zip 文件**。单击下一步。
3. 浏览至适当的 JAR 文件。
4. 选择要导入的文件以及要包含文件的项目或文件夹。

将企业 bean 迁移至 WebSphere Studio Application Developer

在将 Java 和项目资源文件导入 WebSphere Studio Application Developer 之后，将代码转换到 WebSphere Studio Application Developer 的下一步就是将企业 bean 迁移至 WebSphere Studio Application Developer。此任务包括三个子任务：从 VisualAge for Java 导出企业 bean、将企业 bean 导入 WebSphere Studio Application Developer 以及生成部署代码。

从 VisualAge for Java 导出企业 bean：

1. 在 VisualAge for Java 中，如果您尚未向工作空间添加 **Export Tool for Enterprise Java Beans 1.1** 功能部件，则添加之。
2. 在工作台的 EJB 页面上，右键单击企业 bean 组并单击导出 > **EJB 1.1 JAR**。
3. 按需要填充字段。确保已选择 **.java** 复选框。
4. 确保选择了数据库。
5. 单击完成。

将企业 bean 导入 WebSphere Studio Application Developer：

1. 在 WebSphere Studio Application Developer 中，创建新的 EJB 项目和新的企业应用程序归档项目。自动将您切换至 J2EE 透视图。
2. 选择文件 > 导入 > **EJB JAR 文件**。单击下一步。
3. 选择 JAR 文件、EJB 项目和 EAR 项目。
4. 单击下一步并从 JAR 文件选择所有 EJB 文件。

5. 单击**下一步**并选择所有相关项目。如果不确定有哪些相关项目，请选择全部。
6. 单击**完成**。如果遇到任何错误（错误将列在“任务”视图中），请参阅第 32 页的『将企业 bean 迁移至 WebSphere Studio Application Developer 的故障诊断技巧』以进行故障诊断。在已经导入企业 bean 之后，请遵循第 32 页的『定位企业 bean 信息』中的指示信息以确定在何处查找企业 bean 和方法属性、模式及映射等。

编辑方法以避免 `java.rmi.RemoteException` 错误:

1. 在 WebSphere Studio Application Developer 中，切换至 J2EE 视图。
2. 从“项目导航器”视图，打开 `WebSphereCommerceServerExtensionData\ejbModule\com.ibm.commerce.sample.objects\enterprise_bean`
其中 `enterprise_bean` 是导入到 WebSphere Studio Application Developer 的企业 bean 名称；例如，`MyEnterpriseBean.java`。
3. 修改以下方法，这样它们不会抛出 `java.rmi.RemoteException` 错误:

```
ejbLoad()  
ejbStore()  
unsetEntityContext()
```

例如，将 `public void ejbLoad() throws java.rmi.RemoteException` 更改为 `public void ejbLoad()`

4. 保存更改。

更改访问隔离级别:

1. 在 WebSphere Studio Application Developer 中，切换至 J2EE 视图。
2. 从“项目导航器”视图，打开 `\WebSphereCommerceServerExtensionData\ejbModule\META-INF` 下的 `ejb-jar.xml`。
3. 单击**访问**。
4. 选择隔离级别下的**已提交**节点，并单击**除去**。
5. 单击**添加**，选择**可重复读取**，并单击**下一步**。
6. 选择企业 bean 的名称，并单击**下一步**。
7. 展开企业 bean 的名称并选中所有 Home 方法的复选框（即，企业 bean 名称下的第一个图标）。
8. 单击**完成**。
9. 验证**重复**节点现在显示在**隔离级别**下。
10. 保存更改。

生成部署代码:

1. 在 WebSphere Studio Application Developer 中，请转至“J2EE 层次结构”视图、展开“EJB 模块”文件夹并选择新导入的 EJB JAR 文件。
2. 从 **EJB JAR** 弹出菜单中，单击生成 > 部署和 RMIC 代码。选择该选项为所有企业 bean 生成代码。

将企业 bean 迁移至 WebSphere Studio Application Developer 的故障诊断技巧:

以下是一些故障诊断技巧，这些技巧可能在将企业 bean 迁移至 WebSphere Studio Application Developer 时提供帮助:

- 当迁移方法查找程序助手时，查找程序助手界面会消失，因为它们已经移至 XML 描述文件中。由于查找程序助手对象通常实现此界面，所以此迁移会产生问题；必须除去该实现。
- 如果在企业 bean 中使用继承且构建您自己定制的查找程序，则它们可能会损坏，因为生成代码中字段的映射在 WebSphere Studio Application Developer 中有所不同。要对此进行修正，请转至 EJCMPxxxxxx 生成的类，查找为 findByPrimaryKey 生成的选择语句并将其用作模板。
- 如果在 WebSphere Studio Application Developer 内设置“首选项”页面，为了防止每次保存更改都执行自动构建，就必须执行以下步骤以生成企业 bean 部署的代码和 RMIC 存根:
 1. 选择企业 bean 项目，右键单击并选择生成 > 部署和 RMIC 代码。
 2. 由于 RMIC 编译器仅能看到已编译的代码并且由于生成的 Java 类仍未编译，您将接收到错误。接收到此错误后，请切换至 Java 视图并构建项目。
 3. 重复步骤 1。这次就不会接收到任何错误了。
 4. 重复步骤 2 编译新生成的存根，这样在部署企业 bean 时不会接收到运行时错误。

定位企业 bean 信息： 下表包含企业 bean 项的列表以及在将它们迁移至 WebSphere Studio Application Developer 后到哪里查找企业 bean:

表 4. 迁移至 WebSphere Studio Application Developer V5.1 后定位企业 bean 信息

| 项 | 位置 |
|------------------------------|--|
| 企业 bean (字段、类) | 在“J2EE 层次结构”视图中展开“EJB 模块”文件夹。 |
| 企业 bean (查找程序类、访问 bean、生成的类) | 在“导航器”视图中展开“EJB 模块”文件夹，并转至 com 目录 (或您的套装方案结构)。 |

表 4. 迁移至 WebSphere Studio Application Developer V5.1 后定位企业 bean 信息 (续)

| 项 | 位置 |
|------------------|--|
| 关联信息 | 在“J2EE 层次结构”视图中展开“EJB 模块”文件夹并选择要处理的企业 bean JAR 文件。从其弹出菜单，选择 打开对象 > EJB 部署描述符 。在“部署描述符”中，选择 概述 选项卡。 |
| 查找程序助手描述 | 在“J2EE 层次结构”视图中展开“EJB 模块”文件夹并选择要处理的企业 bean JAR 文件。从其弹出菜单，选择 打开对象 > EJB 部署描述符 。在“部署描述符”中，选择 Bean 选项卡。 |
| 映射或模式描述 | 在“J2EE 层次结构”视图中展开“EJB 模块”文件夹并选择要处理的 EJB JAR 文件。从其弹出菜单，选择 生成 > EJB 至 RDB 映射 。 |
| 事务划分 | 在“J2EE 层次结构”视图中展开“EJB 模块”文件夹并选择要处理的企业 bean JAR 文件。从其弹出菜单选择 打开对象 > EJB 部署描述符编辑器 。在 EJB 部署描述符编辑器中选择 Bean 选项卡。 |
| 隔离级别或查找更新或只读方法标记 | 在“J2EE 层次结构”视图中展开“EJB 模块”文件夹并选择要处理的 EJB JAR 文件。从其弹出菜单，选择 打开对象 > EJB 部署描述符 。在 EJB 部署描述符中选择 访问 选项卡。 |
| EJB 环境变量 | 在“J2EE 层次结构”视图中展开“EJB 模块”文件夹并选择要处理的企业 bean JAR 文件。从其弹出菜单，选择 打开对象 > EJB 部署描述符 。在 EJB 部署描述符中选择 Bean 选项卡。 |

迁移 EJB 访问 bean: 使用 Export Tool for Enterprise Java Beans 1.1 从 VisualAge for Java Enterprise Edition V4.0 导出企业 bean 时，所有关联的 Java bean 包装器和副本助手访问 bean 的元数据也将导出。WebSphere Studio Application Developer 不支持 Rowset 访问 bean，所以这些访问 bean 的元数据不会被导出。

迁移客户查找程序助手: 在使用 Export Tool for Enterprise JavaBeans™ 1.1 从 VisualAge for Java Enterprise Edition V4.0 导出企业 bean 时，或在用 Deployment Tool for Enterprise JavaBeans (即 EJB 部署工具) 部署 1.0 JAR 文件时，查找程

序助手界面就迁移至扩展文档。如果 JAR 文件在企业 bean JAR 文件中，则元数据也会从查找程序助手界面迁移至扩展文档。但是，只有在扩展文档中找不到 JAR 文件中的查找程序描述符时才会发生到该扩展文档的查找程序助手界面迁移。如果使用 Export Tool for Enterprise Java Beans 1.1 导出企业 bean，则会从导出的 JAR 中过滤冗余的类。如果没有使用 Export Tool for Enterprise Java Beans 1.1 导出企业 bean，并且没有与冗余类一起导入企业 bean，则只需忽略这些类即可。

将定制的 WebSphere Commerce Server 企业 bean 迁移至 WebSphere Studio Application Developer

在将企业 bean 迁移至 WebSphere Studio Application Developer 之后，将代码转换到 WebSphere Studio Application Developer 的下一步骤就是将定制 WebSphere Commerce Server 企业迁移至 WebSphere Studio Application Developer。此任务包括两个子任务：将代码和元数据更改迁移至 WebSphere Studio Application Developer 中的现有 WebSphere Commerce Server 项目中，并生成部署代码和访问 bean。

将代码和元数据更改迁移至 WebSphere Studio Application Developer 中的现有 WebSphere Commerce Server 项目中： 要将代码和元数据更改迁移至 WebSphere Studio Application Developer 中的现有 WebSphere Commerce Server 项目中，请执行以下操作：

1. 在 WebSphere Studio Application Developer 中，选择要对其进行更改的 WebSphere Commerce Server 企业 bean 项目。
2. 合并 Java 代码更改。要实现此任务，请参阅 WebSphere Studio Application Developer 文档中关于如何在企业 bean 项目中进行 Java 代码更改的信息。
3. 合并表或列定义。要执行此操作，请参阅 WebSphere Studio Application Developer 文档中关于如何在企业 bean 项目中修改现有数据库表定义的内容。
4. 合并容器管理的持久性（CMP）字段。请参阅 WebSphere Studio Application Developer 文档中关于如何为企业 bean 添加或修改 CMP 字段的内容。
5. 合并查找程序方法。请参阅 WebSphere Studio Application Developer 文档中关于如何为企业 bean 项目在部署描述符中添加或修改查找程序方法的内容。
6. 为每个要更改的 WebSphere Commerce Server 企业 bean 项目重复以上步骤。

注：在代码合并过程期间，你可能遇到警告或错误。这是正常的，且在生成部署代码和访问 bean 时会修正所有的错误，如第 57 页的『生成部署代码』中所述。

生成部署代码

要重新生成部署代码，请执行以下操作：

1. 在 WebSphere Studio Application Developer 中，转至 J2EE 透视图并从 J2EE 层次结构展开“EJB 模块”。

2. 右键单击新创建的 EJB JAR 文件并选择生成 > 部署和 RMIC 代码。
3. 选择全部 EJB 并单击完成。

注：此外，请参阅以下关于将企业 bean 迁移至 WebSphere Studio Application Developer 的信息：

- 第 32 页的『将企业 bean 迁移至 WebSphere Studio Application Developer 的故障诊断技巧』
- 第 32 页的『定位企业 bean 信息』
- 第 33 页的『迁移 EJB 访问 bean』
- 第 33 页的『迁移客户查找程序助手』

设置测试环境并测试迁移的应用程序

在将企业 bean 文件迁移至 WebSphere Studio Application Developer 之后，将代码转换到 WebSphere Studio Application Developer 的最后一步就是设置测试环境并测试迁移的应用程序。

要启动服务器实例，请遵循以下步骤：

1. 在 WebSphere Studio Application Developer 中，确保关闭本地 Web 服务器或 WebSphere Application Server 服务器。否则，WebSphere Commerce 将无法成功启动。
2. 检查数据源属性页面并将缺省密码更改为您使用的数据源设置的密码。缺省情况下，缺省密码字段为空。
3. 提供数据源设置并确保为数据源设置了正确的 URL。如果您无法确保 URL 正确，您将在启动 Commerce Server 时遇到错误，并将无法成功创建数据源。对于 DB2，缺省 JDBC 数据源 URL 为 `jdbc:db2:database_name`；例如，`jdbc:db2:demo_dev`。确保数据源 URL 如此。
4. 用正确的数据库用户密码更新 WebSphere Commerce Server 数据源，如下：
 - a. 通过选择开始 > 程序 > 启动 **WebSphere Studio > Application Developer 5.0** 来启动 WebSphere Studio Application Developer。
 - b. 在 J2EE 透视图的 J2EE 层次结构视图中，展开服务器并双击 **WebSphereCommerceServer**。显示 WebSphereCommerceServer 视图。
 - c. 在 WebSphereCommerceServer 视图中，选择数据源页面。
 - d. 在“数据源”页面上，展开服务器设置。
 - e. 在 **JDBC 提供程序列表表**中，选择 **DB2 JDBC 驱动程序**。
选择 JDBC 提供程序之后，JDBC 提供程序列表下的表将更新。
 - f. 在上面选中的 **JDBC 提供程序**中定义的数据源表中，选择 **jdbc/WebSphere Commerce DB2 数据源** `instance_name`。

其中 *instance_name* 是 WebSphere Commerce Studio 实例的名称。缺省实例名称是 Demo_Dev。

- g. 单击**编辑**。打开“修改数据源”向导。
- h. 确保数据源 JDNI 名称如下：

`jdbc/WebSphere Commerce DB2 DataSource instance_name`

例如，如果 WebSphere Commerce 实例名称是 **demo**，则 JDNI 名称应当是 `jdbc/WebSphere Commerce DB2 DataSource demo`。如果数据源 JDNI 名称值并非如此，WebSphere Commerce Server 将无法启动。在这种情况下，您将需要修改数据源 JDNI 名称以匹配 `instance.xml` 文件中的值。

- i. 在“修改数据源”向导中，将**缺省用户密码**字段中的所有值替换为**缺省用户标识**字段中显示的标识的密码。
 - j. 单击**完成**以更新信息并关闭“修改数据源”向导。
 - k. 通过选择**文件 > 保存 WebSphereCommerceServer** 将更新保存至配置。
5. 转至“服务器”视图。如果在透视图找不到它，则在主菜单上选择**透视图 > 显示视图 > 服务器**。
 6. 右键单击服务器实例并选择**发布**。
 7. 当发布完成时，单击**确定**关闭弹出菜单。
 8. 右键单击服务器实例并选择**启动**。显示“控制台”视图。
 9. 遵循实例的启动过程。

现在您可以通过发布商店测试迁移的代码。遵循以下步骤使用 WebSphere Commerce 管理控制台访问并设置商店：

1. 打开浏览器窗口并输入以下 URL：
`https://host_name/webapp/wcs/admin/servlet/ToolsLogon?XMLFile=adminconsole.AdminConsoleLogonhttps`
2. 在“管理控制台登录”页面上，输入 `wcsadmin` 作为用户名、提供密码并单击**登录**。（缺省情况下，在安装期间用户名和密码都设为 `wcsadmin`，但是在第一次使用 WebSphere Commerce 管理控制台时必须更改密码。
3. 在登录到 WebSphere Commerce 管理控制台之后，选择**站点**。
4. 从“WebSphere Commerce 管理控制台”主页面内的**商店**，选择**发布**。
5. 完成此向导内的信息，单击**下一步**以继续浏览页面。
6. 在“选项”页面，单击**完成**以发布商店。

请注意，与 VisualAge for Java 中的 WebSphere Test Environment 相比，由于您刚完成了设置，您将能够执行 HTTPS 请求。

迁移支付实例和数据库

一旦已将定制代码转换至 WebSphere Studio Application Developer V5.1, 下一步就是迁移要用于 WebSphere Commerce Payment (先前的 Payment Manager) 的 WebSphere Commerce 支付实例和数据库。如果使用外部支付系统, 则不必完成此部分, 但是您必须确保支付系统能与 WebSphere Commerce 5.5 和 WebSphere Commerce 开发环境一起正常工作。

对于 WebSphere Commerce 5.4, 安装支付数据库的机器要不同于包含 WebSphere Commerce 的机器。对于 WebSphere Commerce 5.5, 您现在可以选择将支付数据库本地安装到与 WebSphere Commerce 开发环境相同的机器上。这样, 对于 V5.5, 有两个支付方案可用:

- 如果要进行远程支付, 请参阅《*WebSphere Commerce 迁移指南*》中关于如何使用 WebSphere Commerce Instance Migrator 工具迁移 Commerce Payments 实例和数据库的详细信息。
- 如果希望在包含 WebSphere Commerce 开发环境的同一机器上安装支付数据库, 则必须迁移支付数据库并确保在 WebSphere Studio Application Developer V5.1 中创建的支付实例使用迁移过的支付数据库。

第 3 部分 迁移后

本部分描述您在迁移了 WebSphere Commerce 开发环境之后应当遵循的步骤。

第 4 章 迁移后操作

完成第 17 页的第 2 部分，『迁移流程』中概括的迁移步骤之后，如果您正从 WebSphere Commerce Studio V5.4 迁移，则请确保您已完成以下步骤：

- 商店中的 JavaServer Pages 模板符合由 Sun Microsystems 创建的 JavaServer Pages 1.2 规范。需要在 WebSphere Commerce 5.4 中运行的商店中使用的 JSP 模板以支持 JSP 1.1 规范。当将商店迁移至 WebSphere Commerce 5.5 时，必须确保商店中的 JSP 模板符合由 Sun Microsystems 创建的 JavaServer Pages 1.2 规范。关于 JavaServer Pages 1.2 规范的信息，请参阅 Sun Microsystems 的 Java Web 站点：www.java.sun.com。

- JavaServer Pages 规范 1.2 声明只支持“Java”语言。因此，在 JSP 中声明的以下页面语言将失效：

```
<%@ page language="JAVA" %>
```

请注意，WebSphere Commerce Instance Migrator 工具（用于支付迁移）将代表您把所有出现的 `<%@ page language="JAVA" %>` 转换为 `<%@ page language="java" %>`。

- 如果您正使用 `AbstractAccessBean.getInitContext()` 来查找 JSP 中的初始上下文，建议您将此语句更改为 `AbstractAccessBean.getInitContext(null,null)`。
- 在 JSP 文件中除去 `<jsp:root>` 和 `</jsp:root>` 部分，这是因为它们只在 JSP 文件是 XML 文档时才有效。如果 JSP 文件不是 XML 文档（即它们不是 JavaServer Pages 格式），则您应当从这些 JSP 文件中除去 `<jsp:root>...</jsp:root>` 部分。关于此文档以及 XML 文档的更多信息，请参阅 Sun Microsystems 提供的 *JavaServer Pages Specification (Version 1.2)* 中的“JSP.5.2 JSP Documents”部分以及《WebSphere Commerce 迁移指南》中标题为『WebSphere Commerce 5.4 JSP 文件』的部分。
- 导入 `java.util.*` 软件包。在 WebSphere Application Server V5 中，使用“Vector”伪指令的 JSP 文件必须显式地包含 `java.util.Vector` 软件包。如果您在 JSP 中包含以下行，则 JSP 将无需更改即可在 WebSphere Commerce 5.5 中工作：

```
<%@ page import="java.util.*" %>
```

例如，如果您已迁移“多乐五金”样本商店，为了使 JSP 页面正常显示，您需要在 JSP 页面的顶部（例如在 `CatalogItemAdd.jsp` 页面的顶部）添加以下行：

```
<%@ page import="java.util.Vector"%>
```

如果您没有导入 `java.util` 软件包并且您使用软件包中的类，您将需要对 JSP 文件作以下更改。WebSphere Commerce 中使用的公共类如下：

Enumeration

要导入此特定类，请使用：

```
<%@ page import="java.util.Enumeration" %>
```

Vector

要导入此特定类，请使用：

```
<%@ page import="java.util.Vector" %>
```

ResourceBundle

要导入此特定类，请使用：

```
<%@ page import="java.util.ResourceBundle" %>
```

关于随 WebSphere Commerce 5.5 提供的类的更多信息，请参阅 WebSphere Commerce 生产和开发联机帮助。

特定于环境的迁移后操作

某些迁移后操作依赖于您的环境，如此处所述。

环境 B

对于环境 B（其中开发环境与 WebSphere Commerce 共存但并不共享有用资源），请执行以下步骤：

1. 将 `instance.xml` 文件重命名为唯一的名称（仅用于标识目的）。例如，将它从 `demo.xml` 重命名为 `demo2.xml`。
2. 修改 WebSphere Commerce 实例目录（`WebSphere\WCS\instances`）中的 `wcs_instances` 文件以指向新 XML 文件。
3. 编辑位于 `CommerceServerDev\instances\instance\xml` 目录的 `instance.xml` 文件并将数据库的 `name` 条目更新为新名称（例如，`Ma112`）。以下是示例：

```
<Database>
<DB DBMSName="DB2"
  RunDB2SG="true"
  SummaryTable="true"
  OraUserID=""
  DBName=""
  RemoteDB="false"
  DBAPwd=""
  DBServerPort=""
  DBNode=""
  DBHost=""
  DBUserID=""
```

```
DBUserPwd=""  
name="Ma112"  
active="true"  
StagingEnable="false" />  
</Database>
```

第 4 部分 转换代码的教程

以下教程举例说明如何有效地迁移定制的或扩展的代码。每个教程都是《*WebSphere Commerce 编程指南与教程*》中教程的扩展；例如，编程指南教程举例说明如何扩展控制器命令，而迁移教程解释如何将扩展的命令迁移至 WebSphere Studio Application Developer。

在您试用教程之前，请确保已经将全部的 VisualAge for Java 商店有用资源迁移至 WebSphere Studio Application Developer 开发环境中。以下是将 JSP 文件迁移至 WebSphere Studio Application Developer 中的步骤列表：

1. 打开 WebSphere Studio Application Developer 并切换至“资源”透视图。
2. 单击文件 > 导入 > 文件夹，并单击下一步。
3. 浏览至目录：`VAJ_installdir\ide\project_resources\IBM WebSphere Test Environment\hosts\default_host\default_app\web`
4. 选择全部文件进行导入，包括子文件夹。
5. 选择文件夹 `\Stores\Web Content` 以包含文件。

第 5 章 教程的迁移流程

以下是本指南中概括的教程的端到端方案或流程的高级视图，它使用三台机器的方案：机器 A 运行 WebSphere Commerce 5.4 进行生产；机器 B 是空机器，用于迁移；最后，机器 C 是运行 VisualAge for Java 的开发机。以下是此方案所需的步骤：

1. 将所需的软件堆安装至机器 B 上。
2. 运行从机器 A 到机器 B 的实例迁移。这可以是一个远程实例迁移。
3. 将数据库从机器 A 复制到机器 B。
4. 在机器 B 上运行数据库迁移。
5. 在机器 C 上安装 WebSphere Studio Application Developer V5.1。
6. 将定制代码和企业 bean 从 VisualAge for Java 导出，并导入到 WebSphere Studio Application Developer 中。
7. 迁移定制代码和企业 bean 并做单元测试。
8. 将定制代码和企业 bean 部署（汇编）到已迁移的 EAR 文件中，此 EAR 文件在运行从机器 A 到机器 B 的实例迁移步骤中生成。关于部署的详细信息，请参阅《*WebSphere Commerce 编程指南与教程*》的指示信息。
9. 部署（安装）EAR。关于部署的详细信息，请参阅《*WebSphere Commerce 编程指南与教程*》的指示信息。
10. 测试所有定制代码和企业 bean 都正常工作。
11. 停止机器 A 上的生产。
12. 将数据库从机器 A 复制到机器 B。
13. 在机器 B 上运行数据库迁移。
14. 切换配置以将机器 B 作为生产服务器使用。
15. 在机器 B 上启动 WebSphere Commerce 5.5。

第 6 章 教程 A: 迁移扩展或定制的控制器的命令

以下教程会带您完成迁移扩展的控制器的命令。此迁移包括几个子步骤: 迁移控制器的命令本身、迁移显示命令的 JSP 文件、迁移与命令关联的企业 bean 并测试所有内容已成功迁移。对应于 WebSphere Commerce 5.4 的《*WebSphere Commerce 程序员指南*》中的教程举例说明如何扩展现有的 OrderProcess 控制器的命令, 这样已通过购买而累积的总奖金点数就显示在订单确认页面上。对于此教程, 您将使用新扩展的命令 (OrderProcessCmdBonusImpl) 作为控制器的命令迁移指令的模型。

迁移企业 bean

在同一《*WebSphere Commerce 编程指南与教程*》中, 您创建了 BonusBean 企业 bean。将此企业 bean 迁移至 WebSphere Studio Application Developer 涉及几个子步骤: 将企业 bean 作为 1.1 JAR 导出、删除 WebSphereCommerceServerExtensionData 文件夹中的 ExtensionJDBCHelperBean、将企业 bean 导入 WebSphere Studio Application Developer 以及生成部署代码。

从 VisualAge for Java 导出企业 bean

1. 在 VisualAge for Java 中, 如果您尚未向工作空间添加 **Export Tool for Enterprise Java Beans 1.1** 功能, 则添加之。
2. 在工作台的 EJB 页面上, 右键单击 **WCSSamplesEntity Bean** 企业 bean 组并单击导出 > **EJB 1.1 JAR**。
3. 按需要填充字段。确保已选择 **.java** 复选框。
4. 将数据库选择为 UDB DB2。
5. 确保选择了数据库。
6. 单击完成。

删除 WebSphereCommerceServerExtensionData 文件夹中的 ExtensionJDBCHelperBean

1. 在 WebSphere Studio Application Developer 中, 切换至 J2EE 视图。
2. 从 J2EE 层次结构视图展开“**EJB 模块**”。
3. 展开 **WebSphereCommerceServerExtensionsData** 文件夹。
4. 展开会话 bean。
5. 右键单击 **ExtensionJDBCHelperBean** 并单击删除。

6. 单击**全部选中**并单击**确定**。

将企业 bean 导入到 WebSphere Studio Application Developer 中

1. 在 WebSphere Studio Application Developer 中，切换至 J2EE 视图。
2. 选择文件 > 导入 > **EJB JAR** 文件。单击下一步。
3. 选择名为 **WebSphereCommerceServerExtensionsData** 的 EJB 项目。
4. 选择名为 **WebSphereCommerceServer** 的 EAR 项目。
5. 单击下一步并从 JAR 文件选择所有 EJB 文件。
6. 单击下一步并选择所有相关项目。如果不确定有哪些相关项目，请选择全部。
7. 单击**完成**。

编辑方法以避免 java.rmi.RemoteException 错误

1. 在 WebSphere Studio Application Developer 中，切换至 J2EE 视图。
2. 从“项目导航器”视图中，打开
**WebSphereCommerceServerExtensionData\ejbModule
\com.ibm.commerce.sample.objects\BonusBean.java**
3. 修改以下方法，这样它们不会抛出 java.rmi.RemoteException 错误：

```
ejbLoad()  
ejbStore()  
unsetEntityContext()
```

例如，将 `public void ejbLoad() throws java.rmi.RemoteException` 更改为 `public void ejbLoad()`

4. 保存更改。

更改访问隔离级别

1. 在 WebSphere Studio Application Developer 中，切换至 J2EE 视图。
2. 从“项目导航器”视图中，打开
WebSphereCommerceServerExtensionData\ejbModule\META-INF 下的
ejb-jar.xml。
3. 单击**访问**。
4. 选择**隔离级别**下的**已提交**节点，并单击**除去**。
5. 单击**添加**，选择**可重复读取**，并单击下一步。
6. 选择 **Bonus** 并单击下一步。
7. 展开 **Bonus** 并选中所有 Home 方法的复选框（即，**Bonus** 下的第一个图标）。

8. 单击**完成**。
9. 验证**重复**节点现在显示在**隔离**级别下。
10. 保存更改。

生成部署代码

由于已经修改了代码，因此必须重新生成其部署代码。

要重新生成部署代码，请执行以下操作：

1. 在 WebSphere Studio Application Developer 中，切换至 J2EE 层次结构视图。
2. 展开“**EJB 模块**”文件夹。
3. 右键单击 **WebSphereCommerceServerExtensionsData** 并选择**生成 > 部署和 RMIC 代码**。
4. 选择**全部 EJB** 并单击**完成**。

迁移定制的 JSP 文件

在同一《*WebSphere Commerce 编程指南与教程*》中，修改 confirmation.jsp 模板以显示向订单过程业务逻辑添加的新业务逻辑。要将定制的 JSP 文件迁移至 WebSphere Studio Application Developer，请遵循这些步骤：

1. 打开 WebSphere Studio Application Developer 并切换至“资源”透视图。
2. 单击**文件 > 导入 > 文件夹**并单击**下一步**。
3. 浏览至目录 `VAJ_installdir\ide\project_resources\IBM WebSphere Test Environment\hosts\default_host\default_app\web\store_directory`
4. 选择 confirmation.jsp 以导入。
5. 选择文件夹 `\Stores\Web Content` 以包含文件。

迁移扩展的控制器命令

要迁移扩展的控制器命令，您必须首先将项目导出到 JAR 文件，将 Java 和资源文件导入 WebSphere Studio Application Developer，然后编译迁移的代码。

要将 _WCSamples 项目导出到 JAR 文件，请遵循以下步骤：

1. 在“VisualAge for Java 工作台”窗口中，选择 **_WCSamples** 项目、右键单击并单击**导出**。
2. 选择 **Jar 文件**单选按钮并单击**下一步**。
3. 指定 JAR 文件的名称。
4. 选择 **.java** 复选框以导出 Java 文件。

5. 按需要填写其它字段。请参阅 VisualAge for Java 联机帮助，获取关于如何执行此任务的更多信息。

要将 Java 和资源文件导入 WebSphere Studio Application Developer，请遵循这些步骤：

1. 打开 WebSphere Studio Application Developer 并切换至“资源”透视图。
2. 选择文件 > 导入 > **Zip 文件**。单击下一步。
3. 浏览至上面创建的适当 JAR 文件。
4. 在 JAR 中选择所有 .java 文件以导入。
5. 选择文件夹 \WebSphereCommerceServerExtensionsLogic\src 以包含文件。

要在 Web 项目中编译迁移的代码，请遵循以下步骤：

1. 打开 WebSphere Studio Application Developer 并切换至 J2EE 视图。
2. 从“项目导航器”视图中，右键单击 **WebSphereCommerceExtensionsLogic**。
3. 选择属性。
4. 选择 **Java 构建路径**。
5. 选择项目选项卡。
6. 选择 **WebSphereCommerceExtensionsData**。
7. 选择 **WebSphereCommerceServerExtensionsLogic** 项目。
8. 右键单击并选择构建项目。

在“时代先锋”样本商店中测试新逻辑

您必须先启动服务器才能验证新的业务逻辑。要执行此操作，从“服务器”视图中右键单击 **WebSphereCommerceServer** 并选择启动。

要验证在 WebSphere Studio Application Developer 中运行的“时代先锋”中的新业务逻辑，请执行以下操作：

1. 一旦 WebSphere Commerce Server 实例已在 WebSphere Studio Application Developer 中启动，请打开浏览器并输入商店主页的 URL。例如，输入以下 URL：
`http://hostname/webapp/wcs/stores/servlet/StoreCatalogDisplay?
storeId=store_Id&catalogId=catalog_Id &langId=-1`
其中，*store_Id* 是商店的标识，而 *catalog_Id* 是商店产品目录的标识。
2. 选择并购买产品。
3. 在已经购买了产品之后，订单确认显示该订单赚得的奖金点数。

第 7 章 教程 B: 迁移扩展或定制的实体 bean 和任务命令

以下教程会带您完成迁移扩展的实体 bean 和扩展的任务命令。此迁移包括几个子步骤: 迁移任务命令、扩展的数据 bean、与命令关联的企业 bean、显示命令的 JSP 文件并测试所有内容已成功迁移。对应于 WebSphere Commerce 5.4 的《*WebSphere Commerce 程序员指南*》中教程的后半部分举例说明如何定制现有的任务命令 `GetProductContractUnitPriceCmd` 及其相应界面。作为此教程的一部分, 您将创建新的业务逻辑以根据客户的奖金点数余额计算折扣价格。此计算由新任务命令执行。新扩展的命令、`GetNewProductContractUnitPriceCmd` 及其界面 (`GetNewProductContractUnitPriceCmdImpl`) 将用作此迁移教程的模型。

迁移扩展的任务命令

要迁移扩展的任务命令, 您必须首先将项目导出到 JAR 文件, 将 Java 和资源文件导入 WebSphere Studio Application Developer, 然后编译迁移的代码。

要将 `_WCSamples` 项目导出到 JAR 文件, 请遵循以下步骤:

1. 在“VisualAge for Java 工作台”窗口中, 选择 **_WCSamples** 项目、右键单击并单击**导出**。
2. 选择 **Jar** 文件单选按钮并单击**下一步**。
3. 指定 JAR 文件的名称。
4. 选择 **.java** 复选框以导出 Java 文件。
5. 按需要填写其它字段。请参阅 VisualAge for Java 联机帮助, 获取关于如何执行此任务的更多信息。

要将 Java 和资源文件导入 WebSphere Studio Application Developer, 请遵循这些步骤:

1. 打开 WebSphere Studio Application Developer 并切换至“资源”透视图。
2. 选择**文件 > 导入 > Zip 文件**。单击**下一步**。
3. 浏览至上面创建的适当 JAR 文件。
4. 在 JAR 中选择所有 .java 文件以导入。
5. 选择文件夹 `\WebSphereCommerceServerExtensionsLogic\src` 以包含文件。

要在 Web 项目中编译迁移的代码, 请遵循以下步骤:

1. 打开 WebSphere Studio Application Developer 并切换至 J2EE 视图。

2. 从“项目导航器”视图中，打开 **WebSphereCommerceServerExtensionsLogic\src\com\ibm\commerce\sample\commands\GetNewBaseUnitPriceCmdImpl.java**。
3. 搜索 `new BigDecimal(db1BonusPrice)` 并将它更改为 `new java.math.BigDecimal(db1BonusPrice)`。
4. 保存文件。
5. 选择 **WebSphereCommerceServerExtensionsLogic** 项目。
6. 右键单击并选择**构建项目**。

迁移扩展的数据 bean

此教程中的下一步骤是迁移扩展的数据 bean，`NewProductDataBean.java`。要迁移扩展的数据 bean，您必须首先将项目导出到 JAR 文件，将 Java 和资源文件导入 WebSphere Studio Application Developer，然后编译迁移的代码。

要将 `_WCSamples` 项目导出到 JAR 文件，请遵循以下步骤：

1. 在“VisualAge for Java 工作台”窗口中，选择 **_WCSamples** 项目、右键单击并单击**导出**。
2. 选择 **Jar 文件** 单选按钮并单击**下一步**。
3. 指定 JAR 文件的名称。
4. 选择 **.java** 复选框以导出 Java 文件。
5. 按需要填写其它字段。请参阅 VisualAge for Java 联机帮助，获取关于如何执行此任务的更多信息。

要将 Java 和资源文件导入 WebSphere Studio Application Developer，请遵循这些步骤：

1. 打开 WebSphere Studio Application Developer 并切换至“资源”透视图。
2. 选择**文件 > 导入 > Zip 文件**。单击**下一步**。
3. 浏览至上面创建的适当 JAR 文件。
4. 在 JAR 中选择所有 `.java` 文件以导入。
5. 选择文件夹 `\WebSphereCommerceServerExtensionsLogic\src` 以包含文件。

要在 Web 项目中编译迁移的代码，请遵循以下步骤：

1. 选择 **WebSphereCommerceServerExtensionsLogic** 项目。
2. 右键单击并选择**构建项目**。

迁移企业 bean

在同一《WebSphere Commerce 编程指南与教程》中，用户企业 bean 以某一方式定制，从而对应用程序而言似乎 BONUS 表的 BONUSPOINT 列实际是 USERS 表中的列。当在 USERS 表中创建了新记录时，会将相应的记录自动插入到 BONUS 表。为了创建此表连接，必须将新的容器管理的持久性（CMP）字段添加到用户实体 bean。此 CMP 字段使用次级表映射功能部件映射到 BONUS 表中的 BONUSPOINT 列。将此企业 bean 迁移至 WebSphere Studio Application Developer 包括几个子步骤：迁移 CMP 字段至用户实体 bean、更新用户数据库模式和表映射以包含 BONUS 表以及为用户实体 bean 生成部署代码和访问 bean：

将 bonusPoint 字段迁移至用户实体

在此部分中，您将使用企业 bean 工具将 CMP 字段迁移至实体 bean。该字段称为 bonusPoint 并最终映射至 BONUS 表的 BONUSPOINT 列。要重新创建用户实体 bean 的 CMP 字段，请执行以下操作：

1. 在 WebSphere Studio Application Developer 中，切换至 J2EE 视图。
2. 从 J2EE 层次结构展开“EJB 模块”。
3. 展开 **Member-MemberManagementData** “EJB 模块” > 实体 bean > 成员。
4. 右键单击用户并选择打开对象 > 部署描述符编辑器。
5. 转至 Bean 页面并选择用户 EJB。
6. 在右侧，单击添加以启动 CMP 属性向导并提供以下信息：
 - 名称: bonusPoint
 - 类型: int
 - 用 getter 和 setter 方法访问: 已启用
 - 将 getter 和 setter 方法提升到远程界面: 已禁用
7. 单击确定。
8. 保存 EJB 部署描述符更改（可以按键盘上的 **Ctrl+S**）。

bonusPoint 字段显示在“属性”窗口中。请注意，显示了警告，但是在您重新生成实体 bean 的代码时会修正这些警告，如第 34 页的『生成部署代码』中所述。

更新模式和表映射以包含 BONUS 表

现在可以用 BONUS 表更新用户模式、为新表创建外键关系并在用户实体 bean 和 BONUS 表列的字段之间创建表映射。

要创建表模式，请执行以下操作：

1. 在 WebSphere Studio Application Developer 中，切换至 J2EE 视图。
2. 从 J2EE 层次结构视图展开数据库。
3. 展开用户数据库模式，然后展开 **NULLID**。
4. 右键单击表并选择新建 > 新建表定义。打开“表定义编辑器”。
5. 输入表名 **BONUS** 并单击下一步。
6. 在“表列”页面中，创建以下两列：
 - memberid - BIGINT，键列
 - bonusPoint - INTEGER，可空
7. 单击下一步两次，转至“外键”页面。
8. 单击添加另一个，用以下信息创建新外键：
 - 外键名称: F_User_Bonus
 - 删除对象: CASCADE
 - 目标表: NULLID.USERS
9. 从源列，选择 **member_id** 并单击 > 按钮。
10. 单击完成以创建新表。

要创建 **BONUS** 表映射，请执行以下操作：

1. 在 WebSphere Studio Application Developer 中，切换至 J2EE 视图。
2. 从 J2EE 层次结构视图展开“**EJB 模块**”。
3. 右键单击 **Member-MemberManagementData** “**EJB 模块**”并选择打开对象 > 映射编辑器。
4. 从左上角窗格（“企业 bean”窗格）展开成员并选择用户 **bean**。
5. 从右上角窗格中，按住 **Ctrl** 键并选择 **MEMBER**、**USERS** 和 **BONUS** 表。所有三个表都应该突出显示。
6. 从下拉菜单中，选择映射 > 创建映射。
7. 从下拉菜单中，选择映射 > 按名称匹配。
8. 从“概述”窗口选择用户 **bean**。
9. 移至“属性”窗口并展开 **Bean 到表战略**。
10. 将 **Discriminator** 值从 User 更改为 U。
11. 保存 EJB 映射（可以按键盘上的 **Ctrl+S**）。

此刻，用户 bean 包含指示一些抽象类并未执行的错误。这些错误在生成部署代码（如第 57 页的『生成部署代码』中所述）时修正。

生成部署代码

由于已经为 User 实体 bean 修改了代码，因此必须重新生成其部署代码。

要重新生成部署代码，请执行以下操作：

1. 在 WebSphere Studio Application Developer 中，切换至 J2EE 视图。
2. 从 J2EE 层次结构视图展开“EJB 模块”。
3. 右键单击 **Member-MemberManagementData** “EJB 模块”并选择生成 > 部署和 RMIC 代码。
4. 选择全部 EJB 并单击完成。

迁移定制的 JSP 文件

在同一《WebSphere Commerce 编程指南与教程》中，修改 ProductDisplay.jsp 模板，这样客户将能够看见您向订单过程业务逻辑添加的定制价格。要将定制的 JSP 文件迁移至 WebSphere Studio Application Developer，请遵循这些步骤：

1. 打开 WebSphere Studio Application Developer 并切换至“资源”透视图。
2. 单击文件 > 导入 > 文件夹并单击下一步。
3. 浏览至目录 `VAJ_installdir\ide\project_resources\IBM WebSphere Test Environment\hosts\default_host\default_app\web\store_directory`
4. 选择 ProductDisplay.jsp 以导入。
5. 选择文件夹 \Stores\Web Content 以包含文件。

在“时代先锋”样本商店中测试新逻辑

您必须先启动服务器才能验证新的业务逻辑。要执行此操作，从“服务器”视图中右键单击 **WebSphereCommerceServer** 并选择启动。

要验证在 WebSphere Studio Application Developer 中运行的“时代先锋”中的新业务逻辑，请执行以下操作：

1. 一旦 WebSphere Commerce Server 实例已在 WebSphere Studio Application Developer 中启动，请打开浏览器并输入商店主页的 URL。例如，输入以下 URL：

```
http://host_name/webapp/wcs/stores/servlet/StoreCatalogDisplay?  
storeId=store_Id&catalogId=catalog_Id &langId=-1
```

其中：

host_name

WebSphere Commerce Studio 机器的全限定主机名

store_Id

商店的标识。

catalog_Id

商店产品目录的标识。

2. 单击**服务**标题下的**注册**链接，然后单击**新建客户**标题下的**注册**。
3. 通过使用电子邮件地址 `wctester@wc` 和密码 `wctester1` 注册新客户。
4. 用测试值填写其它字段，单击**提交**并保留浏览器为打开状态。
5. 打开 **DB2 控制中心**并执行以下操作：

- a. 单击**交互式**选项卡。
- b. 在**命令**字段中，执行以下操作：

1) 输入：

```
connect to your_database_name
```

其中 *your_database_name* 是 WebSphere Commerce 数据库的名称

2) 单击**执行**图标。

3) 输入：

```
select users_id from userreg where logonid = 'wctester@wc'
```

4) 单击**执行**图标。

- c. **查询结果**选项卡显示您在先前步骤中注册的客户的条目。记录客户的 **USERS_ID** 值。
- d. 更新新注册的客户的奖金点数余额。
- e. 单击**交互式**选项卡并在**命令**字段中输入以下命令：

```
update BONUS set BONUSPOINT = 1000 where MEMBERID = users_id
```

其中 *users_id* 是来自步骤 5c 的值。

f. 单击**执行**图标。

6. 在浏览器中，单击**男式**链接以查看商店的男式时装部分。
7. 单击有特色的特价商品的链接以查看产品页面。页面显示一般价格以及基于客户的奖金点数余额的折扣价格。

第 5 部分 附属资料

声明

本信息是为在美国提供的产品和服务编写的。

IBM 可能在其它国家或地区不提供本文中讨论的产品、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。

本出版物中任何对 IBM 许可程序的引用并非意在明示或暗示只能使用 IBM 的许可程序。只要不侵犯 IBM 的任何知识产权，任何同等功能的产品、程序或服务都可以用来代替 IBM 产品、程序或服务。在与其它产品结合使用时，除了那些由 IBM 明确指定的产品之外，其评估和验证均由用户自行负责。

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证。您可以用书面方式将许可证查询寄往：

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

关于双字节（DBCS）信息的许可证查询，请与您所在国家或地区的 IBM 知识产权部门联系，或用书面方式将查询寄往：

IBM World Trade Asia Corporation Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

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：

国际商业机器公司以“按现状”的基础提供本出版物，不附有任何形式的（无论是明示的，还是默示的）保证，包括（但不限于）对非侵权性、适销性和适用于某特定用途的默示保证。某些国家或地区在某些交易中不允许免除明示或默示的保证。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它程序（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 Canada Ltd.
Office of the Lab Director
8200 Warden Avenue
Markham, Ontario
L6G 1C7
Canada

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 国际程序许可证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，在其它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、其出版说明或其它可公开获得的资料中获取。IBM 没有对那些产品进行测试，也无法确认其性能的精确性、兼容性或任何其它关于非 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。

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。

本信息仅作为规划目的。其中的信息在可以供应所描述的产品之前可能有所更改。

此信息包含日常商业运作中所使用的数据和报告示例。为尽可能表述完整，这些示例包含人名及公司、品牌和产品的名称。所有这些人或名称均系虚构，如有实际的人名或企业名称和地址与此雷同，纯属巧合。

本产品中提供的信用卡图像、商标和贸易名称，仅供已由信用卡标记的所有者授权通过该信用卡接受支付的权限的商家使用。

商标

IBM 徽标和以下术语是国际商业机器公司在美国和 / 或其它国家或地区的商标或注册商标：

DB2

IBM

VisualAge

WebSphere

Microsoft、Windows、Windows NT 和 Windows 徽标是 Microsoft Corporation 在美国和 / 或其它国家或地区的商标或注册商标。

Java 和所有基于 Java 的商标和徽标是 Sun Microsystems, Inc. 在美国和 / 或其它国家或地区的商标或注册商标。

其它公司、产品或服务名称可能是其它公司的商标或服务标记。



中国印刷