

IBM WebSphere Commerce



管理指南

版本 5.5

IBM WebSphere Commerce



管理指南

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 163 页的『声明』中的信息。

第一版，第二修订版（2003 年 7 月）

本版本适用于 IBMWebSphere Commerce 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。您可使用以下 URL 上提供的联机 IBM WebSphere Commerce 文档反馈表发送您的意见：

<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 2003. All rights reserved.

目录

关于本书	v	查看日志文件	60
何处可找到信息	v	配置 LDAP	61
WebSphere Commerce 出版物	v	配置规则服务	61
WebSphere Commerce 联机帮助	v	配置登台服务器	62
Web 上的更多信息	vi	配置连接	66
+ 更改摘要	vi	配置 WebSphere Commerce Payments	66
开始以前	vii		
知识要求	vii	第 4 章 主管商店管理	69
本书中使用的约定	viii	管理转销商	69
路径变量	viii	管理经销商	69
		站点管理	70
第 1 章 WebSphere Commerce 管理简介	1	第 5 章 动态高速缓存	71
管理工具	2	WebSphere Commerce servlet 或 JSP 结果高速缓存	71
管理控制台	2	WebSphere Application Server 动态高速缓存	71
配置管理器	2	动态高速缓存	71
WebSphere Commerce Payments	3	启用动态高速缓存服务和 servlet 高速缓存	72
组织管理控制台	9	启用动态高速缓存监视器	72
WebSphere 贸易加速器	9	WebSphere 动态高速缓存监视器相关性标识的树形视图	73
WebSphere Application Server	9	配置可高速缓存对象	73
数据库	10	高速缓存缺省值	74
		满页和分段高速缓存	75
第 2 章 基本管理任务	11	满页高速缓存	75
仅一次性的管理任务	11	分段高速缓存	76
安装和配置	11	高速缓存 servlet 过滤	76
常规管理任务	11	商店页面高速缓存策略	78
启动和停止 WebSphere Commerce	11	WebSphere Commerce 中的高速缓存失效机制	78
启动 WebSphere Commerce 配置管理器	12	在 cachespec.xml 中定义失效策略	78
创建、配置和删除实例	14	通过高速缓存监视器除去高速缓存条目	81
打开管理控制台	16	调用动态高速缓存失效 API	81
打开组织管理控制台	17	为基于命令的失效写入命令	81
启用和禁用 WebSphere Commerce 组件	17	在 WebSphere Commerce 中使用命令失效的示例	82
启动和停止 WebSphere Commerce Payments	18	高速缓存失效示例	82
安装指南中的管理任务	19	高速缓存失效 API (DynaCacheInvalidation)	85
		CACHEIVL 表	85
第 3 章 配置	21	外部高速缓存中的失效	86
配置服务	21	Web 服务器插件高速缓存	86
体系结构配置	21	简单文件 servlet	87
消息传递服务	21	问题确定	87
调度服务	21		
日志记录服务	22	第 6 章 系统维护	89
WebSphere Commerce Payments 问题确定	24	数据库清理实用程序	89
规则服务	25	数据库清理实用程序对象	90
LDAP - 轻量级目录访问协议	25	数据库清理命令	95
登台服务器	32	清除数据库	98
连接服务	54	为 WebSphere Commerce 实用程序设置 PATH 环境变量	100
配置任务	57	删除对象	100
配置消息传递服务	57	删除帐户对象	100
配置调度服务	57	删除地址对象	101
配置日志记录	57		

删除可承诺库存对象	101
删除附件对象	102
删除拍卖对象	102
删除拍卖日志对象	103
删除自动投标代理日志对象	103
删除基本商品对象	104
删除投标日志对象	104
删除计算代码对象	105
删除竞销对象	105
删除竞销统计信息对象	106
删除产品目录条目对象	106
删除产品目录组对象	107
删除合同对象	108
删除赠券对象	108
删除预期库存记录对象	109
删除预期库存记录详细信息对象	109
删除论坛消息对象	110
删除供货中心对象	110
删除库存代码调整对象	111
删除库存调整对象	111
删除指定商品信息对象	112
删除成员消息关系对象	112
删除消息对象	113
删除订单对象	113
删除组织对象	114
删除产品顾问统计信息对象	114
删除产品比较统计信息对象	115
删除产品探测统计信息对象	115
删除策略对象	116
删除产品集对象	116
删除报价请求对象	117
删除退货商品对象	117
删除退货原因对象	118
删除销售辅助统计信息对象	118
删除已登台对象	119
删除商店对象	119
删除用户对象	120
删除用户通信量日志对象	121
删除供应商对象	121
添加新配置至数据库清理实用程序	122
删除对象的示例	123
第 7 章 性能	125
使用 WebSphere Commerce PMI 模块进行性能监视	126
设置 PMI 模块	126
启动 Tivoli 性能查看器	127
WebSphere Commerce PMI 模块报告值	128
其它性能工具	129
维护调度程序	129
LDAP 的复制	130
测试登台服务器上的站点	130

配置登台服务器	130
为定制表创建触发器	130
配置远程数据库	130
复制数据至登台数据库	131
运行登台检查命令	131
传播数据至生产数据库	132
复制文件至生产服务器	132
删除已登台对象	133
WebSphere Commerce Payments 性能调整参数	133
管理 iSeries 上的 Payments 实例数据库	135

第 8 章 疑难解答 137

CrossWorlds® 适配器	137
WebSphere Commerce Payments	138
调整以获得高性能环境	138

附录 A. LDAP 方案 141

LDAP 方案: 作为成员资源库的 LDAP 服务器	141
----------------------------	-----

附录 B. LDAP 文件 143

ldapmap.dtd	143
ldapentry.xml	143

附录 C. WebSphere Commerce Payments 教程 149

步骤 1: 访问 WebSphere Commerce Payments 用户界面	149
步骤 2: 创建 WebSphere Commerce Payments 商家并授权卡匣	150
步骤 3: 定义 WebSphere Commerce Payments 用户	150
步骤 4: 指定用户角色	151
以商家管理员登录	152
步骤 5: 创建帐户	152
指定帐户设置	153
步骤 6: 创建品牌	154
步骤 7: 使用样本结帐创建订单	154
步骤 8: 核准订单	155
从订单页面核准订单	155
使用核收功能核准订单	156
步骤 9: 圈出支付金额	156
步骤 10: 结算批量	157
步骤 11: 发出返款	158
步骤 12: 查看批量总量	159

附录 D. WebSphere Commerce Payments 的本地语言支持 (NLS) 信息 161

NLS 提示和技巧	161
受支持的语言和语言环境	162

声明 163

关于本书

本书面向管理员并描述了如何使用管理控制台以及其它管理工具管理 WebSphere® Commerce 组件。


何处可找到信息


WebSphere Commerce 具有描述完整的电子交易解决方案的联机 and 硬拷贝信息。此外，与 WebSphere Commerce 捆绑销售的软件产品提供了更多信息，描述了软件的具体特征和功能。本节提供了何处找到各类信息的快速概述。

WebSphere Commerce 出版物

- 《WebSphere Commerce 基础》，V5.5
- 《WebSphere Commerce 编程指南与教程》，V5.5
- 《WebSphere Commerce 安装指南》，V5.5
- 《WebSphere Commerce 快速入门》，V5.5
- 《WebSphere Commerce 附加软件指南》，V5.5
- 《WebSphere Commerce 迁移指南》，V5.5
- 《WebSphere Commerce 商店开发指南》，V5.5
- 《WebSphere Commerce 样本商店指南》，V5.5
- 《WebSphere Commerce 管理指南》，V5.5
- 《WebSphere Commerce 安全性指南》，V5.5
- *WebSphere Commerce Payments Programming Guide and Reference*, V5.5
- *WebSphere Commerce Payments OfflineCard Cassette Supplement*, V5.5
- *WebSphere Commerce Payments CustomOffline Cassette Supplement*, V5.5

关于这些出版物的更新，请访问以下 Web 地址：

 Business Edition
(http://www.ibm.com/software/webservers/commerce/wc_be/)

 Professional Edition
(http://www.ibm.com/software/webservers/commerce/wc_pe/)

WebSphere Commerce 联机帮助

WebSphere Commerce 联机帮助由可以使用 Web 浏览器查看的联机信息组成。

可以使用以下地址从在 Internet Explorer V5.5 或更高版本上运行的 Web 浏览器访问联机帮助：

`https://host_name:8000/wchelp`

其中 `host_name` 是您的 WebSphere Commerce 机器名称。

Web 上的更多信息

WebSphere Commerce 支持

要查找支持信息，包括新闻组、FAQ、技术注解、疑难解答信息和下载内容，请访问以下 Web 地址：

WebSphere Commerce 支持 Web 页面
(<http://www.ibm.com/software/commerce/support/>)

WebSphere Commerce 技术库 Web 页面
(<http://www.ibm.com/software/commerce/library/>)

软件合作伙伴

有许多软件合作伙伴提供产品和服务来增强 WebSphere Commerce 的功能。关于这些合作伙伴的信息，请访问 Commerce Zone
(<http://www.ibm.com/software/wsdd/zones/commerce/>) 并单击 Software Developers 链接。

Redbooks™

要查找更多高级技术信息，请访问 Redbooks Web 站点
(<http://www.ibm.com/redbooks>) 并搜索 WebSphere Commerce。

+ 更改摘要

+ 本指南及其任何更新版本都可以在 WebSphere Commerce 技术库 Web 页面
+ (<http://www.ibm.com/software/commerce/library/>) 获取。关于 WebSphere Commerce
+ 版本的其它信息，请参阅概述页面：

- + • Business Edition
+ (http://www.ibm.com/software/webservers/commerce/wc_be/)
- + • Professional Edition
+ (http://www.ibm.com/software/webservers/commerce/wc_pe/)

+ 关于其它支持信息，请参阅 WebSphere Commerce 支持站点
+ (<http://www.ibm.com/software/commerce/support/>)。

+ 要了解该产品的最新更改，请参阅更新后产品的自述文件，该文件也可从上述 Web 站
+ 点获取。关于如何安装 WebSphere Commerce 及其支持的产品的说明，请参阅
+ 《WebSphere Commerce 安装指南》。

+ 本书上一版本之后所做的更新通过页边空白中包含的修订字符标识。本书对修订字符
+ 使用以下约定：

- + • “+”（加）字符标识本书当前版本中已做的更新。

+ 下表显示对本书做出的主要更改。

更改	受影响的章节或页面
更正了 iSeries™ 平台的 dbclean 示例。	第 89 页的第 6 章，『系统维护』

+
+
+
+
+
+
+

更改	受影响的章节或页面
更正了 iSeries 平台的 dbclean 和 stagingcopy、stagingcheck 以及 stagingprop 示例。	第 125 页的第 7 章, 『性能』 和第 32 页的 『登台服务器』
“为单机工具记录” 一节不再使用, 并已因此除去。	第 57 页的 『配置日志记录』
对关于动态高速缓存的信息的澄清以及杂项更新。	第 73 页的 『配置可高速缓存对象』

开始以前

《IBM® WebSphere Commerce 管理指南》提供了站点管理员或系统管理员必须在 WebSphere Commerce 站点中执行的的任务的相关信息。本指南中把这两种角色之中任一角色称为站点管理员角色。

重要事项

您将在本书通篇看到特定于平台的信息。请确保遵循特定于您的平台的表示信息。

本书分成以下几章:

WebSphere Commerce 管理简介

基本管理任务

配置

主管商店管理

动态高速缓存

系统维护

性能

疑难解答

附录 A: LDAP 方案

附录 B: LDAP 文件

附录 C: WebSphere Commerce Payments 教程

附录 D: WebSphere Commerce Payments 本地语言支持 (NLS) 信息

知识要求

本书应当由需要了解如何设置和维护 WebSphere Commerce 站点的站点管理员或系统管理员阅读。管理员必须具备以下方面的知识:

- 电子交易解决方案体系结构
- 网络接口组件（例如防火墙、DFS™、DNS、路由器和网络集线器）的安装和配置
- 数据库技术。
- XML
- 商店业务过程。

本书中使用的约定

本书使用以下约定：

黑体字表示图形用户界面（GUI）控件，例如字段名称、按钮或菜单选项。

等宽字表示您按照所示内容原样输入的文本的示例以及目录路径。

斜体字用于强调以及您用自己的值替代的变量。



表示可以帮助您完成任务的附加信息。

400 表示特定于 WebSphere Commerce for IBM Eserver iSeries 400®（以前称为 AS/400®）的信息

AIX 表示特定于 WebSphere Commerce for AIX® 的信息。

Linux 表示特定于 WebSphere Commerce for Linux® 的信息。

Solaris 表示特定于 WebSphere Commerce for Solaris Operating Environment 软件的信息。

Windows

表示特定于 WebSphere Commerce for Windows® 的信息。

Professional 表示特定于 WebSphere Commerce Professional Edition 的信息。

Business 表示特定于 WebSphere Commerce Business Edition 的信息。

路径变量

本指南使用以下变量来表示目录路径：

WC_installdir

这是 WebSphere Commerce 的安装目录。以下是 WebSphere Commerce 在不同操作系统上的缺省安装目录：

AIX /usr/WebSphere/CommerceServer55

400 /QIBM/ProdData/CommerceServer55

Linux /opt/WebSphere/CommerceServer55

▶ Solaris /opt/WebSphere/CommerceServer55

▶ Windows C:\Program Files\WebSphere\CommerceServer55

▶ 400 WC_userdir

这是 WebSphere Commerce（它可以被修改或需要由用户配置）使用的所有数据的目录。WebSphere Commerce 实例信息就是此类数据的一个示例。此目录对 OS/400® 是唯一的。

WC_userdir 变量表示以下目录: /QIBM/UserData/CommerceServer55

WAS_installdir

这是 WebSphere Application Server 的安装目录。以下是 WebSphere Application Server 在不同操作系统上的缺省安装目录:

▶ AIX /usr/WebSphere/AppServer

▶ 400 /QIBM/ProdData/WebAS5

▶ Linux /opt/WebSphere/AppServer

▶ Solaris /opt/WebSphere/AppServer

▶ Windows C:\Program Files\WebSphere\AppServer

WAS_userdir

这是 WebSphere Application Server（它可以被修改或需要由用户配置）使用的所有数据的目录。

▶ 400 /QIBM/UserData/WebAS5/Base/WAS_instance_name

第 1 章 WebSphere Commerce 管理简介

本书着重说明了通常由站点管理员执行的任务。本书中讨论的站点管理员角色与许多公司里存在的系统管理员角色同义，以下称作站点管理员。

站点管理员是 WebSphere Commerce 附带的已定义缺省角色之一。通常，站点管理员安装、配置和维护 WebSphere Commerce 以及关联的软件和硬件。站点管理员响应系统警告、警报和错误，并诊断和解决系统问题。此角色通常控制访问和授权（创建成员并将成员指定给相应的角色）、管理 Web 站点、监视性能和管理负载均衡任务。站点管理员还可负责对不同的开发阶段（例如，测试、登台和生产）建立和维护若干服务器配置。此角色还处理关键系统备份和解决性能问题。

关于其它 WebSphere Commerce 角色的更多信息，请参阅 WebSphere Commerce 产品联机帮助。站点管理员执行以下方面的任务：

访问管理

- 用户
- 组织
- 角色
- 访问组
- 策略
- 资源组
- 操作组

安全性

- 帐户策略
- 密码策略
- 帐户锁定策略
- 安全性检查程序

性能 监视系统性能

配置

- 传送
- 消息类型
- 消息记录
- 跟踪
- 组件配置
- 调度程序
- 注册表
- 关于本产品

WebSphere Commerce Payments

- 用户
- 商家设置
- WebSphere Commerce Payments 设置
- 卡匣

- 跟踪
- 访问管理
- 定义站点的消息传递系统
- 将规则服务传播至生产服务器
- 配置 WebSphere Commerce Payments
- 配置和维护高速缓存
- 配置调度程序
- 配置记录
- 维护数据库
- 管理 WebSphere Application Server
- 管理 Web 服务器
- 监视系统性能
- 确保站点安全性
- 疑难解答

管理工具

有几种工具可以使站点管理员的任务执行起来更容易。本节中描述主要工具。后续章节讨论其它工具和可以使用其它工具执行的任务。

管理控制台

管理控制台允许站点管理员执行与站点配置和商店配置有关的任务。关于使用管理控制台的更多详细信息，请参阅第 16 页的『打开管理控制台』。站点管理员使用管理控制台执行的任务包括以下内容：

- 管理用户、组织、角色和成员组
- 管理访问控制
- 定义站点的传送和消息类型
- 指定 WebSphere Commerce Payments 设置
- 启用和禁用 WebSphere Commerce 组件
- 调度要为站点运行的作业
- 更新注册表组件
- 发布商店归档文件

商店发布实用程序

商店发布实用程序包含在管理控制台中并允许您基于随 WebSphere Commerce 提供的样本快速地创建商店归档文件。此实用程序还允许您通过启用或禁用商店中的某些功能（例如，协作）来配置已发布的商店。关于发布商店的更多信息，请参阅 WebSphere Commerce 联机帮助中的“发布商店归档文件”。

配置管理器

配置管理器建立和更改用于部署和运行 WebSphere Commerce 所必需的基础结构。建立基础结构涉及获取关于数据库、Web 服务器和 WebSphere Commerce Payments 位置的信息，以及配置这些应用程序以便用于 WebSphere Commerce 所需的所有必要信息。它还涉及确定机器的设置信息以及初始化 WebSphere Commerce 应用程序本身所需的任何信息。关于使用配置管理器的更多详细信息，请参阅第 12 页的『启动 WebSphere Commerce 配置管理器』或参阅《WebSphere Commerce 安装指南》。

配置管理器允许站点管理员执行管理任务和配置任务，而不必处理语法敏感的配置文件。站点管理员可以执行的一些功能如下：

- 创建或删除 WebSphere Commerce 实例
- 更改 WebSphere Commerce 实例配置设置
- 配置 Web 服务器
- 配置数据库来充当登台服务器
- 输入 WebSphere Commerce Payments 的参数

WebSphere Commerce Payments

WebSphere Commerce Payments 是 WebSphere Commerce 的组件，为因特网商家提供了安全电子支付处理。以基于开放标准的技术为基础，WebSphere Commerce Payments 使用支付卡匣来支持多个支付协议，它们包括：

- VisaNet，全球远程通信数据和支付处理供应商，具有授权和结算支付能力。
- BankServACH，与自动清算中心网络（ACH）交互以支持在线电子支票支付的支付网关。
- Paymentech，直接链接到 Visa 和 MasterCard 的在线信用卡支付的商家处理器。
- 为 WebSphere Commerce Payments 编写的第三方支付卡匣。

简言之，WebSphere Commerce Payments 与 WebSphere Commerce 集成并提供类似现金收银机的功能来管理支付处理。由于 WebSphere Commerce Payments 隐藏于因特网商家店面之后，通过银行和其它金融机构接收支付和处理这些支付，所以客户与 WebSphere Commerce Payments 从不直接交互。

注：IBM WebSphere Commerce Payments（后面称之为 WebSphere Commerce Payments）以前称为 Payment Manager。从 V3.1.3 开始，支付应用程序重命名为 WebSphere Commerce Payments，且本文档中通篇更改了对该产品的引用。

适应多种支付方式

WebSphere Commerce Payments 实现了多支付框架体系结构，该体系结构提供了一种灵活而可扩展的方式，以适应因特网上需要接受多支付方式的商家的需求。多支付框架使支付管理（框架）从特定的支付方式（软件卡匣）中分离出来，使它们每一个都能独立发展。

WebSphere Commerce Payments 提供了一个插件体系结构，每种支付方式的软件卡匣借此连接到支付框架。该框架提供了使用任何支付方式产生和接收支付时所需的类属基础结构功能。

支付卡匣是一些符合 WebSphere Commerce Payments 框架的数据流和控制约定的软件应用程序。每个支付卡匣都包含特定支付方法和协议的实现。

卡匣可以由 IBM 或第三方支付系统实现者编写。IBM 支持卡匣开发，并为有兴趣编写自己的支付卡匣的开发者提供了详细的表示信息。关于卡匣开发的更多信息，请参阅 *WebSphere Commerce Payments Cassette Kit Programming Guide* 并从以下地址下载：<http://www.software.ibm.com/commerce/payments/download.html>

WebSphere Commerce Payments 用户界面

您可通过 WebSphere Commerce UI 或独立 UI 访问 Payments 功能。两者都可被 Payments 管理员和使用主管 WebSphere Commerce Payments 的各个商家用来:

- 配置 WebSphere Commerce Payments
- 以支付中立方式执行以下日常支付处理任务:
 - 核准订单
 - 圈出支付金额
 - 结算批量
 - 发出返款
 - 查看每日批量总量

WebSphere Commerce Payments 用户界面是基于浏览器的, 您可以使用 Web 浏览器的安全套接字层 (SSL) 功能以远程方式安全地访问此界面。

WebSphere Commerce Payments 角色

WebSphere Commerce Payments 实施了角色, 这样可根据用户角色为每个用户显示不同的视图 (例如, Payments 管理员的视图和商家管理员的视图)。在商家组织内部, WebSphere Commerce Payments 启用了不同角色的概念, 这样商家就可以监视自己的用户。例如, 职员仅限于执行诸如核准订单的操作, 而商家或 Payments 管理员能修改与金融机构的关系。

Payments 框架中有四种角色, 它们具有到 WebSphere Commerce 中对应角色的相关映射。如果您是创建用户和分配角色的管理员, 知道这些是很有用的。当您在 WebSphere Commerce 组织管理控制台中创建用户时, 您必须首先为这些用户分配下面列出的 WebSphere Commerce 角色。然后用户将显示在 Payments UI 中, 从中您可为他们分配相应的 Payments 角色。下表将 Payments 用户角色映射为对应的 WebSphere Commerce 角色:

表 1. 角色映射

Payments 角色	WebSphere Commerce 角色
Payments 管理员	站点管理员
商家管理员	站点管理员
主管	运作或销售经理
职员	客户服务主管

Payments 管理员和商家管理员都可管理 WebSphere Commerce Payments。主管和职员是财务角色。虽然他们不管理 WebSphere Commerce Payments, 但是他们管理支付处理功能。下表描述了每种 Payments 角色的职责:

表 2. 角色职责

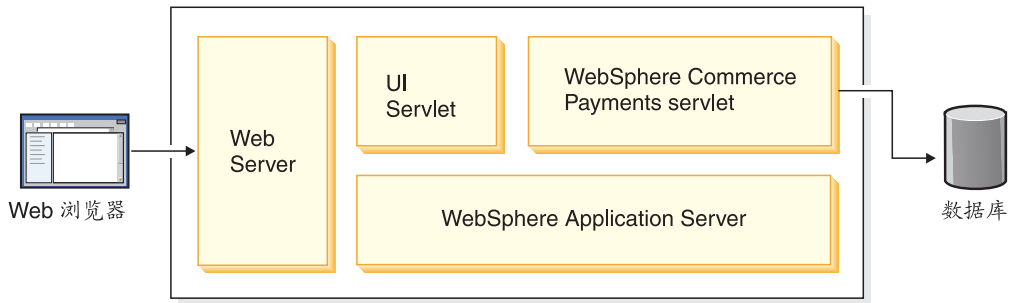
角色	职责
Payments 管理员	<ul style="list-style-type: none"> • 定义商家管理员、主管和职员 • 配置商家及其卡匣 • 标识 Payments 主机名和状态 • 配置任何已安装的卡匣 • 添加、删除和更新事件侦听器 • 结算支付 • 核准或核收订单 • 发出返款和逆向返款 • 圈出订单金额 • 搜索订单和批量 • 查看每日批量总量
商家管理员	<ul style="list-style-type: none"> • 定义商家管理员、主管和职员 • 配置商家及其卡匣 • 添加、删除和更新事件侦听器
主管	<ul style="list-style-type: none"> • 结算支付 • 核准或核收订单 • 发出返款和逆向返款 • 圈出订单金额 • 搜索订单和批量 • 查看每日批量总量
职员	<ul style="list-style-type: none"> • 结算支付 • 核准或核收订单 • 圈出订单金额 • 搜索订单和批量 • 查看每日批量总量

WebSphere Commerce Payments 组件

要理解 WebSphere Commerce Payments, 查看以下 WebSphere Commerce Payments 并定义其组件是有用的:

- 配置了 WebSphere Application Server 的 Web 服务器
- WebSphere Commerce Payments Servlet
- 用户界面 servlet
- 数据库

下图说明了 WebSphere Commerce Payments 的内部组件:



WebSphere Commerce Payments Servlet 是 WebSphere Commerce Payments 的主要组件。该 Servlet 设计用来与 WebSphere Application Server (为所有平台提供公共 servlet 环境的产品) 协同工作。

本部分其余内容进一步定义 WebSphere Commerce Payments Servlet 及其附属支持产品:

- Web 服务器
- WebSphere Application Server
- 数据库

Web 服务器: 为什么要配置 Web 服务器? 在本章前面部分已经提及, WebSphere Commerce Payments 会处理进入的 HTTP 请求。HTTP 请求首先是由配置了 WebSphere Application Server 的 Web 服务器答复的。然后, 这些请求再被转送到其它 WebSphere Commerce Payments 组件。

IBM WebSphere Application Server: 为什么用 WebSphere Application Server 配置 Web 服务器? 原因在于两个方面:

1. WebSphere Application Server 是一个可与多种平台上的各种 Web 服务器一起工作的产品。
2. WebSphere Application Server 提供可在其中执行 WebSphere Commerce Payments servlet 的环境。

WebSphere Commerce Payments Servlet: WebSphere Commerce Payments Servlet (也称为 Payment Servlet) 是 WebSphere Commerce Payments 的核心。它接收来自用户界面或其它商家应用程序的请求, 并在执行安全性检查之后, 调用支付卡匣以处理任何必需的支付处理过程 (例如与金融网络的通信), 并且最后将响应发送回调用应用程序。这是可以连接到 WebSphere Commerce Payments 数据库并访问存储在其上的安全数据的唯一组件。

WebSphere Commerce Payments 和电子交易实体

在本部分中您将看到 WebSphere Commerce Payments 是如何支持商家购物软件的。三个方案演示了 WebSphere Commerce Payments 如何与商家软件和类属支付协议一起协同工作, 与以下电子交易实体进行交互:

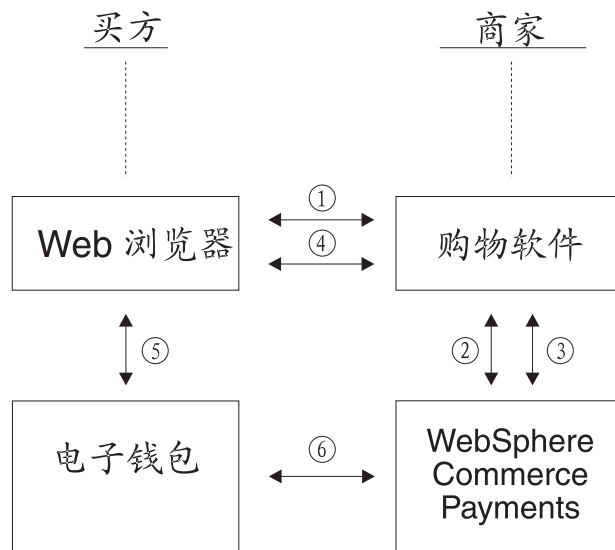
- 商家
- 买方
- 金融机构

随后的方案说明了使用 WebSphere Commerce Payments 的电子交易。请注意这些方案是一般化的，并不代表特定于特定支付协议的任何交互，这一点很重要。更确切地说，它们探讨的是许多电子交易业务方案通用的基本交易流程。

方案 1: 现金和电子钱包: 第一个方案使用虚构的现金协议并涉及两个实体: 买方和商家。资金将从买方转移到商家。在此交易中，买方使用特定于协议的电子钱包应用程序。以下方案图中说明了如下交易步骤:

1. 买方和商家购物软件之间的交互以买方向商家发出购买请求而结束。
2. 为响应购买请求，商家软件将支付启动命令（即接收支付）发送到 WebSphere Commerce Payments。
3. WebSphere Commerce Payments 用支付启动消息进行响应，此消息包含买方电子钱包程序的特定于协议的信息。
4. 商家软件把此支付启动消息转发到买方的浏览器。
5. 买方浏览器接收的支付启动消息启动了买方的电子钱包程序。
6. 买方通过与 WebSphere Commerce Payments 和电子钱包软件交互完成交易。

方案 1: WebSphere Commerce Payments 与现金和电子钱包的交互:



此方案独立于随后两个方案的原因是此方案演示的电子交易仅涉及两个实体:

- 买方（用电子钱包代表）
 - 使用 WebSphere Commerce Payments 的商家
- 没有涉及金融机构。

方案 2: 信用卡和电子钱包: 第二个方案说明了使用虚构信用卡协议的购买，它涉及三个实体:

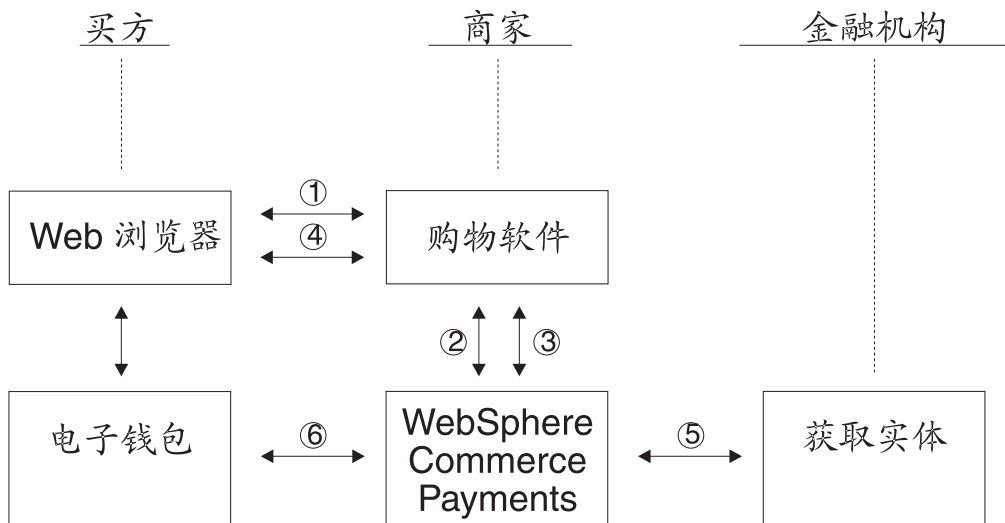
- 商家
- 买方（也使用电子钱包）
- 金融机构

此方案与方案 1 类似，只有以下不同:

- 当买方的电子钱包完成购买请求时，WebSphere Commerce Payments 向金融机构的收款实体发出核准请求，以确保持卡人有足够的资金。
- 当 WebSphere Commerce Payments 发出一个表示核准状态的确认响应到买方的电子钱包时，交易结束。

下面的方案图在第 5 步和第 6 步中说明了这一差异：

方案 2： WebSphere Commerce Payments 与信用卡和电子钱包的交互：

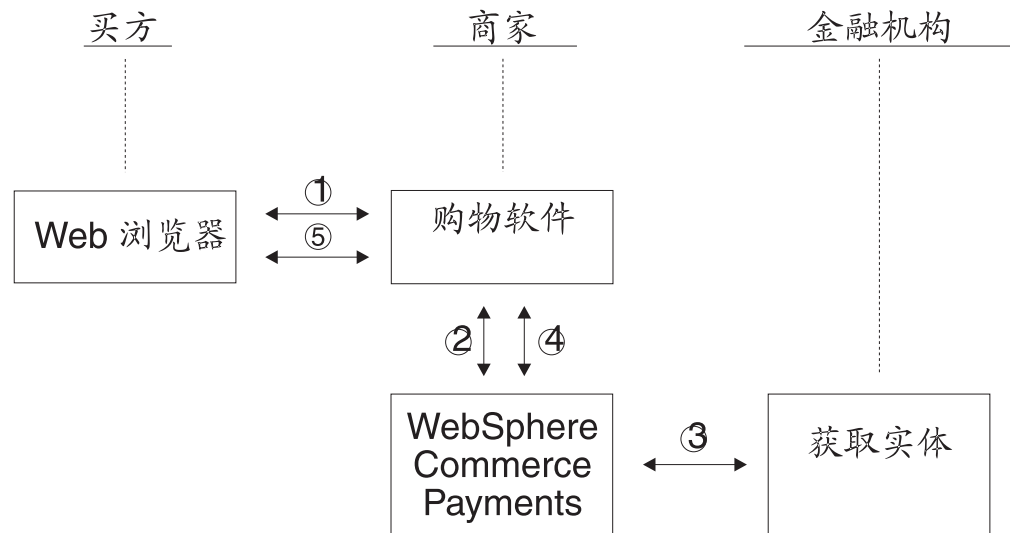


随着客户的电子钱包程序启动及交易完成，金融机构的收款实体确认持卡人的资金足够，同时 WebSphere Commerce Payments 向买方确认支付资金已圈出。

方案 3：信用卡和 SSL： 最后的方案说明了一个信用卡协议，其中买方没有电子钱包或特定于消费者的软件。所有消费者信息都以安全方式（例如以 SSL 套接字方式）传送到商家软件。以下方案图中说明了如下交易步骤：

1. 买方和商家购物软件之间的交互以买方向商家发出购买请求而结束。
2. 为响应购买请求，商家软件将支付启动命令发送到 WebSphere Commerce Payments。
3. WebSphere Commerce Payments 试图核准交易（通过收款实体的金融机构确保持卡人有足够的资金）。
4. WebSphere Commerce Payments 将结果（例如，资金已被圈出）返回给商家软件。
5. 最后，通知买方购买请求成功或者失败。

方案 3: WebSphere Commerce Payments 与信用卡和 SSL 交互:



前面的图说明了可能的电子交易方案，并演示了 WebSphere Commerce Payments 在这些方案中的角色。许多其它业务流程也可能发生，而在这些交互中 WebSphere Commerce Payments 将采用类似的支付功能。

组织管理控制台

组织管理控制台允许站点管理员控制访问站点或商店的组织。您有权执行的各项任务通过各个菜单显示在组织管理控制台主页上。这些任务是基于用户的角色和权限级别的。关于更多信息，请参阅 WebSphere Commerce 联机帮助中的组织管理控制台信息。

WebSphere 贸易加速器

WebSphere 贸易加速器使您可以通过完成各种商店操作（从管理商店外观到创建和维护订单再到跟踪商店活动）来维护网上商店。您有权以您的角色执行的任务显示在 WebSphere 贸易加速器主页菜单上。这些任务是基于用户角色和权限级别的，而用户角色和权限级别由站点管理员使用管理控制台定义。关于更多信息，请参阅 WebSphere Commerce 联机帮助中的 WebSphere 贸易加速器信息。

注：由站点管理员使用组织管理控制台或 WebSphere 贸易加速器执行的任务在联机帮助中描述。

WebSphere Application Server

WebSphere Application Server 是用于构建、部署和管理因特网和内部网 Web 应用程序的基于 Java 的应用程序环境。WebSphere Application Server 还为 Java™ Server Page 文件提供运行时支持。

WebSphere Application Server 中的管理员控制台提供各种工具和功能来允许管理员执行以下操作：

- 停止和启动 WebSphere Commerce Server
- 停止和启动 IBM WebSphere Commerce Payments
- 安装和配置各种资源（例如，servlet 和企业 bean）
- 对资源指定安全性

- 确保所有应用程序可用
- 授予或撤销用户访问权（例如，为新员工创建新帐户以及向特定用户组添加成员）
- 监视 Commerce Server 性能
- 克隆应用程序组件以获得改进的性能
- 提供运行中的 WebSphere 应用程序的跟踪和调试信息

WebSphere Application Server 信息可以从 WebSphere Application Server Web 站点获得：<http://www.ibm.com/software/webservers/appserv>。关于 WebSphere Application Server 系统管理的更多详细信息，请参阅以下文档：

- *Systems Administration*
- *Tuning and Troubleshooting*

数据库

您可以使用 DB2[®] 或 Oracle 作为数据库管理系统。关于更多信息，请参阅每个系统的《*WebSphere Commerce 安装指南*》。

第 2 章 基本管理任务

本章讨论需要您定期或仅一次性执行的基本管理任务。根据业务过程的不同，您可能不需要执行每一个任务。

仅一次性的管理任务

以下任务通常只是一次性执行的。

安装和配置

您作为站点管理员的任务之一是安装和配置 WebSphere Commerce 站点。通常，这是需要您执行的第一步。安装通常涉及根据所使用的操作系统安装 WebSphere Commerce 软件。安装步骤可在随产品提供的安装指南中找到。有关安装和配置的更多信息，请参阅《WebSphere Commerce 安装指南》。进一步的配置信息可在 WebSphere Commerce 联机帮助中找到。

常规管理任务

以下任务是定期执行的：

- 启动和停止 WebSphere Commerce
- 打开配置管理器
- 创建、配置和删除 WebSphere Commerce 实例
- 打开管理控制台
- 打开组织管理控制台
- 启用和禁用 WebSphere Commerce 组件
- 启动和停止 WebSphere Commerce Payments

启动和停止 WebSphere Commerce


WebSphere Commerce 的每个实例都有它自己的 WebSphere Commerce Server。WebSphere Commerce Server 是基于 Java 的贸易服务器，用来控制 WebSphere Commerce 系统中的信息流。关于 WebSphere Commerce Server 中每个组件的更多信息，请参阅 WebSphere Commerce 产品和开发联机帮助中的“WebSphere Commerce Server”。

将随名为 *WC_instance_name* 的应用程序服务器启动和停止 WebSphere Commerce 实例，其中 *instance_name* 是要启动或停止的 WebSphere Commerce 实例的名称。例如，要启动名为 *demo* 的 WebSphere Commerce 实例，您将需要启动名为 *WC_demo* 的应用程序服务器。

要启动或停止 WebSphere Commerce 实例，请执行以下操作：

1. 确定要启动或停止哪个 WebSphere Commerce 实例。
2. 遵循用于您平台的停止和启动应用程序服务器的表示信息。关于这些表示信息的更多信息，请参阅 WebSphere Commerce 产品和开发联机帮助中的“在您的平台上启动和停止应用程序服务器”。

如果 WebSphere Commerce Payments 还没有启动，您可能希望在启动 WebSphere Commerce 后启动它。关于启动 WebSphere Commerce Payments 的表示信息，请参阅第 18 页的『启动和停止 WebSphere Commerce Payments』。

注:  关于在 WebSphere Commerce Studio 中启动 WebSphere Commerce 的表示信息，请参阅 WebSphere Commerce 产品联机帮助中的“在 WebSphere Commerce Studio 中启动和停止 WebSphere Commerce”。

启动 WebSphere Commerce 配置管理器

配置管理器使您能够启用和配置各种不同的 WebSphere Commerce 组件。应尽可能地使用配置管理器而不是通过修改 WebSphere Commerce 配置文件来启用组件。要创建或配置 WebSphere Commerce 实例，应当使用配置管理器。

在 AIX、Linux 和 Solaris 软件上启动 WebSphere Commerce 配置管理器

要在 AIX、Linux 和 Solaris 软件上启动 WebSphere Commerce 配置管理器，请执行以下操作：

1. 执行以下操作启动服务器：

- a. 打开终端窗口。
- b. 以作为安装 WebSphere Commerce 的一部分运行 `postinstall` 脚本时创建的 WebSphere Application Server 用户登录。缺省 WebSphere Application Server 用户名为 `wasuser`。
- c. 发出以下命令：

```
export DISPLAY=host_name:0.0
xhost +host_name
cd /usr/WebSphere/CommerceServer55/bin
./config_server.sh
```



```
export DISPLAY=host_name:0.0
xhost +host_name
cd /opt/WebSphere/CommerceServer55/bin
./config_server.sh
```

其中 `host_name` 是机器的全限定主机名，您希望从此机器访问配置管理器。

注:

- 1) 不要关闭在其中输入 `config_server.sh` 命令的终端窗口，否则配置管理器服务器将停止。
 - 2) 不要将配置管理器服务器运行作为后台进程 - 这是潜在的安全风险。
2. 执行以下操作启动客户机：
- a. 打开另一个终端窗口。
 - b. 以作为安装 WebSphere Commerce 的一部分运行 `postinstall` 脚本时创建的 WebSphere Application Server 用户登录。缺省 WebSphere Application Server 用户名为 `wasuser`。

- c. 发出以下命令:

▶ AIX ▶ Solaris

```
export DISPLAY=host_name:0.0
xhost +host_name
cd /usr/WebSphere/CommerceServer55/bin
./config_client.sh &
```

▶ Linux

```
export DISPLAY=host_name:0.0
xhost +host_name
cd /opt/WebSphere/CommerceServer55/bin
./config_client.sh &
```

其中 *host_name* 是机器的全限定主机名, 您希望从此机器访问配置管理器。

在 iSeries 上启动 WebSphere Commerce 配置管理器

要在 iSeries 上启动 WebSphere Commerce 配置管理器, 请执行以下操作:

1. 执行以下操作启动服务器:

- a. 登录 iSeries, 确保概要文件具有 *SECOFR 用户类, 并以英语或选为实例缺省语言的语言的特定于语言的设置对它作了设置。
- b. 输入以下命令启动 QShell 会话:

```
STRQSH
```

并在 QShell 会话中执行以下操作:

- 1) 发出以下命令切换至 WebSphere Commerce 服务器 bin 目录:

```
cd /qibm/proddata/commerceserver55/bin
```

- 2) 发出以下命令启动配置管理器服务器程序:

```
config_server.sh -port server_port_number
```

其中 *server_port_number* 是 iSeries 服务器上的端口号, 配置管理器将侦听该端口号。此参数是可选的, 缺省值是 1099。该值必须在 1024 和 65535 之间, 并且当前不在使用中。

注: 如果正在使用的系统中主语言与创建实例时使用的语言不同, 则必须将 QSYSlanguage_feature_number 库添加到用户概要文件的库列表中。否则概要文件将试图在 QSYS 下找到它。要添加语言特征库, 请使用 EDTLIBL 命令。

- c. 如果这是配置管理器第一次在系统上运行, 您将看到以下消息:

```
正在将 Java 程序连接到 /QIBM/ProdData/WebCommerce55/lib/wcsconfig.JAR。
正在将 Java 程序连接到 /QIBM/ProdData/WebCommerce55/lib/wcsruntime.JAR。
正在将 Java 程序连接到 /QIBM/ProdData/WebCommerce55/wc.ear/lib/wcslogging.JAR。
正在将 Java 程序连接到 /QIBM/ProdData/WebCommerce55/lib/xml4j.JAR。
正在将 Java 程序连接到 /QIBM/ProdData/WebCommerce55/lib/sslite.ZIP。
```

- d. 接收到以下消息后, 继续进行到下一步:

```
已创建注册表。CMServer 绑定在注册表中。
```

2. 执行以下操作启动客户机:

- a. 在客户机上使用命令提示符, 切换至 WCS400 目录。
- b. 通过运行以下命令来配置客户机:

```
config_client.bat iSeries_Host_name Server_port_number
```

其中

iSeries_Host_name

是服务器全限定主机名。

Server_port_number

是 iSeries 服务器上的端口号，配置管理器将侦听该端口号。

在 Windows 上启动 WebSphere Commerce 配置管理器

要在 Windows 上启动 WebSphere Commerce 配置管理器，请执行以下操作：

1. 通过选择开始 → 设置 → 控制面板 → 管理工具 → 服务并检查 IBM WC 配置管理器服务具有“已启动”状态，确保 IBM WC 配置管理器服务器进程正在运行。

重要事项

保持 IBM WC 配置管理器服务器服务一直运行可能会潜在地引起安全性问题。当您不在使用配置管理器时请停止 WC 配置管理器服务器服务。

要防止潜在的安全性问题，还应确保将 IBM WC 配置管理器服务器设为手动启动，而非自动。

2. 选择开始 → IBM WebSphere Commerce → 配置。

创建、配置和删除实例

要创建、配置或删除 WebSphere Commerce 实例，应当使用配置管理器。设计此工具的目的是允许您配置实例，同时又防止您输入无效的值。

应当使用配置管理器修改大多数 WebSphere Commerce 配置参数。不过，某些参数必须通过 WebSphere Commerce 配置文件 (*instance_name.xml*) 进行更改。请参阅联机帮助中个别组件的文档，来确定是应当通过配置管理器，还是通过 WebSphere Commerce 配置文件 (*instance_name.xml*) 进行配置。

创建和配置实例

要创建实例，请执行以下操作：

1. 打开配置管理器。关于打开配置管理器的指示信息，请参阅第 12 页的『启动 WebSphere Commerce 配置管理器』。
2. 展开带机器名的节点，选择**实例列表**，并从“操作”菜单中选择**创建实例**。
3. 实例创建向导将引导您逐步完成实例创建过程。如果需要关于任何一个面板的帮助，请选择**帮助**。


要配置实例，请执行以下操作：

1. 打开配置管理器。
2. 展开带机器名的节点，展开**实例列表**，并选择希望修改的实例。
3. 选择想要修改的组件
4. 如果此组件已经存在，可以选择它，这样将打开相应的面板。然后可以根据需要修改组件。

5. 如果组件不存在，可以用鼠标右键单击组件，并选择创建组件。向导将引导您逐步完成组件创建过程。




删除实例

要删除 WebSphere Commerce 实例，请执行以下操作：

1. 备份 WebSphere Commerce 实例（建议）。
2. 切换至 `WC_installdir\instances\instance_name\xml` 目录， `WC_userdir/instances/instance_name/xml`，并将 `instance_name.xml` 文件复制到备份目录中。
3. 启动 WebSphere Application Server。
4. 通过执行以下操作除去 WebSphere Commerce Server:

-  从 QShell 会话命令行输入：

```
WC_installdir/bin/rmCommerceServer.sh instance_name
```

   在命令提示符下，输入以下命令：

```
WC_userdir/bin  
./rmCommerceServer.sh instance_name
```

 在命令提示符下输入：


```
WC_installdir/bin  
./rmCommerceServer instance_name
```

5. 在 WebSphere Commerce 配置管理器中，用鼠标右键单击实例并选择删除。
6. 删除 WebSphere Commerce Business Edition 数据库：

-  如果正在使用 DB2 数据库，请执行以下操作：
 - a. 打开 DB2 命令窗口并输入以下命令：

```
db2 drop db db_name  
db2 uncatalog node node_name
```

其中 `db_name` 是您在配置管理器中输入的 WebSphere Commerce 数据库的名称，`node_name` 是您为这些数据库编目的节点名称。缺省节点名为机器主机名的简短版本（即非全限定）。

-  如果正在使用 Oracle 数据库，请执行以下操作：
 - a. 启动 SQL*Plus 会话并输入以下命令：


```
sqlplus system/manager@wc_SID
```

其中 `wc_SID` 是 WebSphere Commerce 数据库的 Oracle 系统标识（SID）。






- b. 在 SQL*Plus 会话中发出以下命令：

```
DROP TABLESPACE wc_tablespace_name  
INCLUDING CONTENTS;  
DROP USER wc_oracle_user_ID  
CASCADE;  
EXIT;
```

注：还必须除去实际的表空间文件。表空间文件名为 `wc_SID.ora`，其中 `wc_SID` 是 WebSphere Commerce 数据库的 Oracle 系统标识（SID）。

 通过运行以下 SQL 语句删除实例数据库资料库: `drop schema instance_name`。如果您尝试删除模式并获得消息,表示您在其被连接时不能删除该接收器,请尝试以下命令:
`ENDJRNPF FILE(*ALL) JRN(instance_name/QSQJRN)`。

7. 要完全删除 WebSphere Commerce 实例,请删除以下目录:

-     `WC_installdir/instances/instance_name`
-  `WC_userdir/instances/instance_name`

8.  使用 `DLTUSRPRF` 命令删除实例用户概要文件。


打开管理控制台

无论什么时候您想执行管理任务,都可能需要在管理控制台中完成它。管理控制台允许您通过完成管理操作和配置任务来控制站点或商店。有关管理控制台的更多信息,请参阅第 2 页的『管理控制台』。

1. 通过执行以下操作之一,启动管理控制台:

- 在浏览器中访问以下 URL:
`https://host_name:8002/adminconsole`

其中 `host_name` 是您的 WebSphere Commerce Server 的全限定 TCP/IP 名称。

-  如果您已安装 WebSphere Commerce Studio 并希望从此机器打开管理控制台,请执行以下操作:
 - a. 确保已在 WebSphere Commerce Studio 中启动了 WebSphere Commerce。关于在 WebSphere Commerce Studio 中启动 WebSphere Commerce 的表示信息,请参阅 WebSphere Commerce 产品联机帮助中的“在 WebSphere Commerce Studio 中启动和停止 WebSphere Commerce”。
 - b. 在浏览器中访问以下 Web 地址:

`https://host_name/webapp/wcs/tools/servlet/ToolsLogon?XMLFile=adminconsole.AdminConsoleLogon`

其中 `host_name` 是 WebSphere Commerce Studio 机器的全限定 HTTP 主机名

2. 通过执行以下操作,登录管理控制台:

- a. 在“登录”页面中,输入管理控制台登录用户名和密码,并单击**登录**。如果您有权使用多个商店或多种语言,则将显示“管理控制台站点/商店选择”页面。如果您有权使用单个商店和语言,则显示管理控制台主页。您有权执行的任务显示在管理控制台主页上。

3. 在“管理控制台站点/商店选择”页面上,选择**站点**或**商店**单选按钮,并单击**确定**。

- 如果您选择了**站点**,则将显示站点管理控制台主页。
- 如果您选择了**商店**,则可从**名称**和**语言**下拉框中选择商店名称和语言。将显示管理控制台主页。

打开组织管理控制台

通过执行以下操作之一，登录组织管理控制台：

- 通过执行以下操作之一，启动组织管理控制台：

- 在浏览器中访问以下 URL: `https://host_name:8004/orgadminconsole`

其中 *host_name* 是您的 WebSphere Commerce Server 的全限定 TCP/IP 名称。

- 如果您已安装 WebSphere Commerce Studio 并希望从此机器打开组织管理控制台，请执行以下操作：

1. 确保已在 WebSphere Commerce Studio 中启动了 WebSphere Commerce。关于更多信息，请参阅 WebSphere Commerce 联机帮助。

2. 在浏览器中访问以下 Web 地址：

```
https://host_name/webapp/wcs/tools/servlet/ToolsLogon?XMLFile=
buyerconsole.BuyAdminConsoleLogon
```

其中 *host_name* 是 WebSphere Commerce Studio 机器的全限定 HTTP 主机名。

- 要登录组织管理控制台，请执行以下操作：

1. 在“登录”页面中，输入组织管理控制台登录用户名和密码，并单击“登录”。如果您有权处理多个商店或语言，则将显示组织管理控制台主页。您有权执行的任务显示在组织管理控制台主页上。

- 要登录并更改组织管理控制台密码，请执行以下操作：

1. 在“登录”页面中，输入组织管理控制台登录用户名和密码。
2. 选择**更改密码**复选框，并单击**登录**。将显示“更改密码”页面。
3. 在**旧密码**字段中，输入当前的组织管理控制台登录密码。此字段最多可以接受 128 个字母数字字符。
4. 在**新密码**字段中，输入新的登录密码。此字段最多可以接受 128 个字母数字字符。
5. 在**新密码确认**字段中，重新输入新的登录密码。
6. 单击**更改**保存新密码。将显示组织管理控制台主页。您有权执行的任务显示在组织管理控制台主页上。

启用和禁用 WebSphere Commerce 组件

作为管理员，您可能希望 WebSphere Commerce 的某些组件只在某些时间可用。由于不同的客户需求，WebSphere Commerce 允许您在需要时启用或禁用 WebSphere Commerce 组件，方法是使用管理控制台中的**配置**菜单。

要启用或禁用 WebSphere Commerce 组件，请执行以下操作：

1. 打开管理控制台并以站点管理员身份登录。
2. 从**配置**菜单中，单击**组件配置**。将显示一个页面，上面列出所有对站点可用的组件。
3. 要启用组件，请执行以下操作：
 - a. 从**可用的组件**列表中，选择想要启用的那些组件并单击**添加**
 - b. 验证想要启用的所有组件是否都在**选定的组件**列表中，并单击**确定**。

注：在启用任何组件之前，请参阅关于该组件的文档以确保其正确配置。

4. 要禁用组件，请执行以下操作：

- a. 请从选定的组件列表中选择想要禁用的组件，并单击**除去**。
- b. 验证想要禁用的所有组件是否都在**可用的组件**列表中，并单击**确定**。

启动和停止 WebSphere Commerce Payments

WebSphere Commerce Payments 是用于在线商家的、不依赖于协议的支付事务处理服务器。它为站点提供类似现金收银机的功能，支持使用特定于协议的卡匣的多种支付方式。这些卡匣是软件组件，可以附加到 Payments 框架来将一般支付和管理命令解释为特定于支付协议的请求，这些请求随后将转发至相应的接收方，如“收款方”机构的支付网关。最后结果类似于在传统商店的结帐柜台处收银员用支付卡划款。Payments 代表商家（或一组商家，例如商店集团）处理因特网支付中的所有后台详细信息，并提供图形界面来简化交易管理。Payments 执行以下功能：

- 在启动并依赖于卡匣的情况下，它验证是否具备所有必需的“收款方”证书。如果缺少一个证书，“收款方”或“支付网关”将自动要求提供该证书。
- 它会永久运行，侦听专用端口并服务于所有来自电子钱包和商家服务器的进入请求。
- 它依据卡匣使用 TCP/IP 协议或其它协议通过因特网与“收款方”通信。
- 它使用数据库表来维护关于交易状态、核准请求以及存款请求的信息。这些表中的记录为工作目的及跟踪和记录保存而保留。数据库还包含一些配置表，它们保留关于商家、支付卡品牌，以及为商家提供卡的授权和支付划款服务的“收款方”的信息。

商家服务器使用 Payments API 与 Payments 交互作用。这些 API 被设计为具有足够的通用性，以支持安全电子交易（SET™）以及其它支付技术。

作为管理员，您将需要启动和停止 Payments 服务。启动和停止支付引擎需要在 Payments 的安装目录中具有 .payment 文件。.payment 文件是在 Payments 安装期间创建的。


启动 WebSphere Commerce Payments

要启动 WebSphere Commerce Payments，请执行以下操作：


1. 启动 WebSphere Application Server 管理控制台并确保启动了 WebSphere Payments 应用程序服务器。

2. 如下登录：



    以您的数据库实例标识登录。

 以具有管理员组特权的用户身份登录。

3. 转至 WebSphere Commerce Payments 的安装目录。缺省安装位置如下：

 /usr/lpp/WebSphere/CommerceServer55/payments

 /QIBM/ProdData/CommercePayments/V55

  /opt/WebSphere/CommerceServer55/payments

 drive:\Program Files\IBM\WebSphere Commerce Payments

4. 输入以下命令：

   ./IBMPayServer

 IBMPayServer

 运行此命令后，您必须指定密码。

注: 如果配置 WebSphere Commerce Payments 时您已选择**需要密码**, 则为了完全初始化 WebSphere Commerce Payments 应用程序, 您必须调用 IBMPayServer 命令。IBMPayServer 命令用来指定 WebSphere Commerce Payments 密码, 该密码用来解密数据库中存储的任何敏感数据。关于此命令的更多信息, 请参阅 WebSphere Commerce 产品和开发联机帮助中的“发出 IBMPayServer 命令”。关于配置 WebSphere Commerce Payments 的更多信息, 请参阅第 66 页的『配置 WebSphere Commerce Payments』。关于配置 WebSphere Commerce Payments 的详细信息, 请参阅第 149 页的附录 C, 『WebSphere Commerce Payments 教程』。

停止 WebSphere Commerce Payments

要停止 WebSphere Commerce Payments, 需要执行以下操作:

- 停止 WebSphere Commerce Payments。
- 停止 WebSphere Application Server 下的 Payment Manager 应用程序服务器。

要停止 WebSphere Commerce Payments, 请完成第 18 页的『启动 WebSphere Commerce Payments』中的步骤 1–3, 并对步骤 4 使用以下命令:

```
 AIX  Solaris  Linux ./StopIBMPayServer  
 Windows StopIBMPayServer
```

注: 关于通过配置管理器或 WebSphere Commerce Studio 停止 Commerce Payments 的信息, 请参阅 WebSphere Commerce 产品联机帮助中的“在 WebSphere Commerce Studio 中启动和停止 WebSphere Commerce Payments”以及“从配置管理器启动和停止 WebSphere Commerce Payments”。

安装指南中的管理任务

在《WebSphere Commerce 安装指南》中可找到以下管理任务:

- 修改 WebSphere Commerce 实例或 Commerce Payments 实例
- 删除 WebSphere Commerce 实例
- 其它命令行配置任务
- 将远程 DB2 数据库编目
- 生成 WebSphere Commerce 加密密码
- 生成 WebSphere Commerce Payments 加密密码

第 3 章 配置

这一部分描述了站点管理员用来确保 WebSphere Commerce 站点最佳运行的配置工具和任务。

配置服务

体系结构配置

站点管理员负责与体系结构设计者一起工作来实现 WebSphere Commerce 将要在其中运行的配置。例如，WebSphere Commerce 和必需的堆栈（Web 服务器、WAS、数据库和 Commerce Payments）可能全部驻留在相同的物理机器或节点上，从而导致单一节点体系结构，或者 WebSphere Commerce 的各层（软件产品）可能驻留在不同的物理节点（机器）上，从而导致多节点体系结构。取决于解决方案的复杂性，实际上可能存在一种群集的环境，在该环境的不同机器或克隆之间实现了负载共享。站点管理员也可能负责对不同的开发阶段（例如，测试、登台和生产）建立若干服务器配置。关于 WebSphere Commerce 可用的不同配置的更多信息，请参阅《WebSphere Commerce 安装指南》。

消息传递服务

管理控制台中的消息传递服务界面允许站点管理员设置与管理站点和商店的消息传递。WebSphere Commerce 消息传递系统允许您全面管理对在 WebSphere Commerce 中生成的消息的定义和发送。消息传递系统可以使用各种传送方法（例如，电子邮件和明码文件）发送消息。对于电子邮件来说，受支持的出站协议是 SMTP；消息编码依赖于指定的语言。明码文件消息使用 UTF-8 编码标准。您也可选择配置消息传递系统使用 WebSphere MQ 和 IBM CrossWorlds[®] 将消息发送到后端系统。还可配置消息传递服务发送出站消息并花费一段时间等待回复消息。要配置出站消息传递系统，请使用管理控制台中的“配置”菜单。

WebSphere Commerce 使用实现了 Java 2 Enterprise Edition Connector Architecture (J2EE/CA) 的插件模型来提供系统与各种传送方法之间的公共接口。将公共接口用于外部传送方法实现了使传送方法的实现详细信息独立于消息传递系统的运作。该体系结构使得插入符合 J2EE/CA 接口的附加传送方法成为可能。这样，您可以轻松定制适合您的业务需要和环境的解决方案。

调度服务

调度程序是 Commerc Server 的一个组件，主要用来调度和启动基于时序方案的作业。每个已调度作业作为单独的线程运行。您可调度多个作业同时运行，然而，建议分散作业以控制工作负载并避免较长的空闲时段。作业是已调度在指定时间或时间间隔运行的 WebSphere Commerce 命令。要指定作业的时序，请使用 AddJob 命令的命令开始和间隔时间参数。有关 AddJob 命令的更多信息，请参阅 WebSphere Commerce 产品和开发联机帮助中的“AddJob 命令”。

调度程序必须在以下情况下运行：

- IBM WebSphere Commerce Payments 正在运行。

- 拍卖正在运行。
- 规则服务正传播到生产环境
- 站点管理员正在更改系统并希望与所有的克隆进行更改内容方面的通信，而不重新启动每个群集实例。
- 使用可承诺（ATP）功能。
- 站点发送已处理的消息。

如果不适用这些情况中的任何一个，站点管理员可以选择禁用调度程序。

日志记录服务

在 WebSphere Commerce 服务器中日志记录消息的目的是当 WebSphere Commerce 应用程序中发生意外错误或异常情况时通知管理员。消息日志和跟踪是重要的诊断工具，它协助站点管理员确定问题根源。跟踪是一种问题确定机制。开发阶段期间跟踪帮助开发者调试代码，跟踪还可帮助技术支持小组解决客户问题。因为 WebSphere Commerce 的日志记录使用 WebSphere Application Server 的日志记录工具，所以可使用日志分析程序。日志分析程序是一个图形实用程序，可帮助查看并分析日志文件。关于日志分析程序的更多信息，请参阅 WebSphere Application Server 信息中心。

WebSphere Commerce 提供了用于日志记录的工具。对于现有客户，我们继续支持 ECTrace 和 ECMessage。对于新的实现，我们建议使用 WebSphere Application Server 提供的 JRAS（WebSphere Commerce 应用程序可使用它）。

WebSphere Commerce 日志 API 为：

- ECTrace 跟踪数据流。将跟踪条目捕获到文件中以用于调试目的。
- ECMessage 记录诊断消息。消息是区分语言环境的，且存储在文件系统的日志文件中。如果启用错误通知，则技术支持会接收到警告通知。诊断日志用于问题确定。缺省情况下，日志文件名为 `activity.log`。
- JRas 是独立日志记录工具箱，提供消息日志记录和诊断跟踪原语。

缺省输出文件的位置是 `WAS_installdir/logs/WC_<instance name>/`。
`WAS_userdir/logs/WC_instance_name/`。缺省输出文件是：

400

- `native_stderr.log` 是进程日志，包含写入标准错误（stderr）流的文本。
- `native_stdout.log` 是进程日志，包含写入标准输出（stdout）流的文本。
- `startServer.log` 是启动服务器时的日志。
- `stopServer.log` 是停止服务器时的日志。
- `SystemErr.log` 在服务器运行时记录所有系统错误。
- `SystemOut.log` 在服务器运行时记录系统输出文件。
- `activity.log` 记录持续活动。此日志位于 `WAS_installdir/logs` 目录中。

关于 WebSphere Application Server 日志 API 的信息，请参阅 WebSphere Application Server 信息中心中关于日志记录和跟踪的部分。

ECMessage

诊断日志用于问题确定。ECMessageLog 记录诊断消息且 ECMessages 是本地化的。ECMessage 分为以下类别：

- 系统消息

- 用户消息

系统和用户消息: 系统消息出现在日志中, 它们有助于调试问题。系统消息为站点管理员提供诊断信息。这些消息可跟在系统故障之后, 或表示某些其它重要事件。

系统消息被指定了特定于产品的消息标识:

CMNnnnns

其中

nnnn: 是用来标识哪个产品受到影响的关键数字。

s: 消息的严重性。

用户消息频繁显示在浏览器上, 以帮助访问站点的客户。用户消息将给出关于问题的详细信息, 例如, 它们将表示指定参数是否有效, 这表示客户当他们重新提交请求时要修正哪个值。站点管理员可使用消息标识来查找与该消息关联的更多详细信息; 客户可将消息报告给支持人员以进行疑难解答。

日志记录级别: WebSphere Commerce 日志记录系统中有五种日志记录级别或严重性。系统消息严重性为: 错误、状态、警告、信息和调试。

- 缺省情况下, 错误消息会一直记录下来。错误消息暴露出可能导致系统故障的错误条件。错误消息可以作为电子邮件和 WebSphere MQ 消息或通过另一种形式的通知发送给已在消息传递注册的站点管理员。
- 状态消息表示 WebSphere Commerce 所达到的某些状态。例如, 每个组件装入时或消息进入到某特定状态时, 都应将消息保存起来。状态消息将帮助技术支持了解组件和应用程序的状态。
- 警告消息显示潜在问题。
- WebSphere Commerce 系统中发生事件时, 会随之出现提示性消息。提示性消息与触发系统状态更改的事件相关。例如, 提交订单后会保存提示性消息。
-

调试消息与跟踪组件相重叠。然而, 调试消息使站点管理员不需要技术支持就能自己调查问题。

因为 WebSphere Commerce 使用 WebSphere Application Server 日志记录工具且 WebSphere Application Server 仅具有三种类型的日志记录级别, 将把 WebSphere Commerce 日志记录级别映射到 WebSphere Application Server, 如下所示:

表 3.

WebSphere Commerce 中的日志记录级别	WebSphere Application Server 中的日志记录级别
ERROR/ERR	TYPE_ERROR/TYPE_ERR
INFORMATION/INFO	TYPE_INFORMATION/TYPE_INFO
DEBUG	TYPE_INFORMATION/TYPE_INFO
WARNING/WARN	TYPE_WARNING/TYPE_WARN
STATUS	TYPE_INFORMATION/TYPE_INFO

WebSphere Application GUI 用于控制日志记录级别或严重性类型的启用。站点管理员可在 WebSphere Application Server 管理控制台中指定要记录哪些日志记录严重性。关于启用日志记录级别的更多信息, 请参阅 WebSphere Application Server 信息中心中的信息。

ECTrace

跟踪用于问题确定。跟踪辅助开发者在开发阶段调试代码，并辅助技术支持小组解决客户的问题。

跟踪数据保存在跟踪文件中以供今后参考。数据结构包含上下文信息，例如，类名、方法名和文本消息。多个数据结构描述了软件应用程序中的数据流。通过分析数据结构序列，开发者可以了解所执行的路径，这样可以帮助确定导致故障的原因。

JRas

JRas 由多个 java 软件包组成，这些软件包提供消息日志记录和诊断跟踪原语。这些原语没有绑定到任何特定产品或平台。JRas 基本上由以下几个组件组成：

- 记录程序：记录程序是用户代码与之交互的主要对象。存在两类记录程序：消息记录程序和跟踪记录程序。消息记录程序仅创建消息记录，而跟踪记录程序创建跟踪记录。记录程序包含一个或多个处理程序，它将事件转发到处理程序以进一步处理。
- 处理程序：处理程序从记录程序接收事件，并在输出设备或事件消费者之上提供抽象。一个示例是文件处理程序，它知道如何将事件写入文件。
- 格式化程序：使用格式化程序配置处理程序，格式化程序知道如何格式化某些类型的事件。
- 事件类型 - 消息和跟踪具有它们自己的预定义事件类型。
- 事件类 - 独立 JRas 日志记录工具箱定义了消息和跟踪事件类。

注：关于 JRas 日志记录工具箱的更多信息，请参阅 WebSphere Application Server 信息中心中的 WebSphere Application Server JRas 文档。

WebSphere JRas 扩展：为了集成到 WebSphere Application Server 运行时之中或为了在 J2EE 环境中使用，WebSphere 提供了一组扩展类。WebSphere Commerce 的日志记录使用 WebSphere Application Server 日志记录工具，并且这些扩展类提供从不同 WebSphere 产品生成的消息和跟踪的更好的相关性。这组扩展类称为 WebSphere JRas 扩展。WebSphere JRas 扩展提供了适当的记录程序实现类。从 WebSphere 管理器类直接且独有地获得这些消息和跟踪记录程序类的实例，该管理器类位于 `com.ibm.websphere.ras` 数据包中。其它组件（例如 Payments 和 JCA 消息传递框架）也使用 WebSphere JRas 扩展。关于与 WebSphere Commerce 和 Payments 跟踪组件关联的 WebSphere JRas 扩展的列表，请参阅第 57 页的『启用跟踪组件』。

注：WebSphere Commerce 为调用 WebSphere JRas 扩展 API 的 `ECMessage` 和 `ECTrace` 提供了包装器，但建议站点管理员直接调用 JRas API。

WebSphere Commerce Payments 问题确定

当您开始使用 WebSphere Commerce Payments 时，您可能需要诊断实际使用应用程序时遇到的问题。WebSphere 日志分析程序提供了一个界面，通过该界面您可分析错误并跟踪日志以决定操作过程。除了以下消息，通过日志分析程序可使用所有 Payments 错误和跟踪日志：

- 第三方卡匣
- 独立 Payments UI
- 样本结帐
- 卡匣柜

规则服务

WebSphere Commerce 使用规则服务与 Advisor 规则服务器交互。规则服务充当便于两个应用程序之间进行通信的接口。规则服务也提供方便的方法以更新您站点基于规则的部分而不必停止整个 WebSphere Commerce。规则服务与竞销存在一对一的关系。使用 WebSphere 贸易加速器发布的每一竞销都必须具有相应的规则服务。这种设计提供了很大灵活性，使每种市场营销服务，乃至竞销，都可以独立于每种其它的服务而启动、停止和刷新。无论何时使用 WebSphere 贸易加速器更新客户概要文件、竞销或规则服务，需要将这些文件传送到生产服务器上的合适位置并刷新规则服务。每一应用程序克隆或 Java 虚拟机都具有规则服务器的本地实例。虽然跨应用程序克隆边界的通信需要是很小的，但在一种情况下是必要的：更新规则服务的状态。为了接收每一应用程序克隆的状态报告，系统使用调度程序来广播更新规则服务状态的请求。当用户按下“查看状态”按钮时，将收集并显示这些结果。

Advisor 规则服务器

WebSphere Commerce 使用 Advisor 规则服务器处理规则和提供个性化市场营销内容，包括广告和推荐销售技术。此服务器合并到 WebSphere Commerce Server 中。规则服务器由 WebSphere Commerce Server 调用，后者可基于当前购物环境传递信息。规则服务器对照由商家或商务人员创建的规则集合处理此信息，并为特定环境编译相应的输出。该输出基于使用 WebSphere 贸易加速器定义的条件。规则服务器由“许可证信息”手册中包含的某些限制加以控制。您需要从 HNC Inc, 获取独立许可证以超越这些限制。

LDAP – 轻量级目录访问协议

LDAP 是一个客户机 / 服务器协议，用于访问目录服务。它最初用作 X.500 的前端，但也可与独立的以及其它类型的目录服务器一起使用。LDAP 用作集中信息仓库，以支持不同应用程序之间的信息共享。

LDAP 信息模型建立在包含有关某个对象（例如，人）信息的条目上。条目由属性组成，这些属性具有一个类型及一个或多个值。每个属性都具有语法，以确定在属性中允许何种值以及那些值在目录操作过程中如何运用。属性语法的示例包括：IA5（ASCII）字符串、JPEG 图像、u-law 编码的声音、URL 和 PGP 密钥。通常，一个条目由其专有名称或 DN 唯一标识。

在 WebSphere Commerce 中，注册用户具有一个唯一标识，该标识是一个与专有名称具有相同格式的字符串。如果 WebSphere Commerce 数据库用作成员资源库，DN 的格式将是“uid=logonIDvalue”，随后是用户所属的父组织实体的 DN。所以，如果一个用户具有登录标识“john”并属于 IBM 组织中的 SWG 组织单位，则其唯一标识将是“uid=john, ou=SWG, o=IBM, o=Root Organization”。如果目录服务器用作成员资源库，注册用户的唯一标识将是他在目录服务器中的 DN，在此情况下，“o=Root Organization”部分可能不是他的 DN 的一部分，因为 WebSphere Commerce 支持目录服务器中现有的用户条目，这些条目可能不在“根组织”下。

WebSphere Commerce 还支持在目录服务器中存储组织实体，每个组织实体具有和 DN 相同格式的唯一标识。如果 WebSphere Commerce 数据库用作成员资源库，则唯一标识的格式将是“ou=organizationalEntityName”，随后是父组织实体的 DN。所以如果称为市场营销的组织单位位于 IBM 组织中的 SWG 组织单位中，则唯一标识将是“ou=Marketing,ou=SWG,o=IBM,o=Root Organization”。如果目录服务器用作成员资源库，组织实体的唯一标识将是它在目录服务器中的 DN，在此情况下，“o=Root

Organization”部分可能不是它的 DN 的一部分，因为 WebSphere Commerce 支持目录服务器中现有的组织条目，这些条目可能不在“根组织”下。还要注意在 DN 中使用“o”还是“ou”取决于组织实体是组织还是组织单位。

如果目录服务器用作成员资源库，则用户和组织实体可以存储在目录服务器中，您可以配置哪个 LDAP 属性应该用作 RDN 属性以及哪个 WebSphere Commerce 属性为 RDN 属性提供值。一般的 DN 如下：

```
uid=jsmith, l=Toronto, st=Ontario, c=CA
```

其中

uid 这是用户的唯一标识。以上示例中的 uid 属性称为 RDN 属性。它唯一标识父条目下 DN 为 *l=Toronto, st=Ontario, c=CA* 的 LDAP 条目。在 WebSphere Commerce 中，如果认证提问是登录标识和密码，WebSphere Commerce 将在 LDAP 服务器上给定的搜索空间下搜索“RDN = 登录标识”的节点。如果 X.509 证书是提问类型的，WebSphere Commerce 将使用证书的主题名称作为 RDN 属性值。

l 用户位置，或所在市/县/区。

st 用户所在省或直辖市。

c 用户的国家或地区。

以下 IBM(R) Redbooks 包含关于 LDAP 的附加信息：

- *LDAP Implementation Cookbook*
- *Understanding LDAP*

可从以下站点获得 Redbooks: <http://www.redbooks.ibm.com>。

要在 WebSphere Commerce Server 和 LDAP 服务器之间使用 SSL，请参阅 LDAP 服务器提供的表示信息以执行适当的设置（例如，创建密钥环文件）和通过使用 *instancename.xml* 文件中的 *JNDIEnvPropNameX* 和 *JNDIEnvPropValueX* 属性指定 WebSphere Commerce Server 所需的任何环境属性。

使用 LDAP 认证

用户概要文件信息可以存储在 WebSphere Commerce 数据库中或目录服务器上。认证选项有：

- **LDAP**
用户可以用其 DN 或 RDN 值以及密码登录。如果使用 RDN，则使用在 *ldapentry.xml* 文件中配置的搜索库来搜索用户。将对照 LDAP 服务器来认证用户。LDAP 服务器上的用户信息复制到 WebSphere Commerce 数据库中，供运行时操作使用。
- **数据库**
使用用户提供的登录标识和密码对照 WebSphere Commerce 数据库对用户进行认证。
- **其它**
使用第三方接口进行认证，将概要文件数据存储在 WebSphere Commerce 数据库中

要指定认证方式，请使用“配置管理器”中的“成员子系统”页面。LDAP 或数据库认证都可以使用 X.509 证书。如果使用 X.509 证书，则 Web 服务器将认证用户。在这种情况下，认证方式设置将确定概要文件数据的存储位置，是存储在 LDAP 中还是存储在 WebSphere Commerce 数据库中。

LDAP 注册表

在 WebSphere Commerce 中，您可以使用 LDAP 或 WebSphere Commerce 数据库作为成员资源库。用户和组织实体可以存储在 LDAP 服务器中。成员组当前只能存储在 WebSphere Commerce 数据库中。如果使用 LDAP 作为注册表，则这将与 *instancename.xml* 文件中的以下设置相对应：

AuthenticationMode=LDAP, ProfileDataStorage=LDAP。

数据在 LDAP 服务器和 WebSphere Commerce 数据库之间复制。您可以用 *ldapentry.xml* 文件配置要复制哪些数据。

下表描述了 WebSphere Commerce 中提供的 LDAP 服务的级别：

表 4. 用户注册

	LDAP 作为注册表	不支持 LDAP
通过 WebSphere Commerce 的新用户注册或注册用户信息更新	在 WebSphere Commerce 数据库和 LDAP 服务器中都创建或更新用户信息。	在 WebSphere Commerce 数据库中创建或更新用户信息。
LDAP 用户从其它 LDAP 应用程序上注册或更新信息	仅当用户登录到 WebSphere Commerce Server 时，或在 WebSphere Commerce Server 中的逻辑需要用户信息时，用户信息才复制到 WebSphere Commerce 数据库中。	N/A
LDAP 服务器的可用性	LDAP 服务器必须一直可用。	N/A
LDAP 连接失败	向用户显示错误页面。 错误消息记录在系统日志文件中。	N/A

表 5. 组织实体注册

	LDAP 作为注册表	不支持 LDAP
通过 WebSphere Commerce 的新组织实体注册或组织实体信息更新	在 WebSphere Commerce 数据库和 LDAP 服务器中都创建或更新组织实体信息。	在 WebSphere Commerce 数据库中创建或更新组织实体信息。
从其它 LDAP 应用程序创建或更新组织实体信息	仅当 WebSphere Commerce Server 中的逻辑需要组织实体信息时，才在 WebSphere Commerce 数据库中创建或更新组织实体。	不适用。

其注册仍正在审批核准的用户也存储在目录服务器中。缺省情况下，上下文相关的属性不存储在目录服务器中。

LDAP 中只永久保存用户注册地址。而不永久保存 LDAP 用户通讯录中的好友、亲属或商业伙伴的地址。

注意： 如果已选择 LDAP 服务器作为成员资源库，则不得使用装入程序软件包以将用户和组织实体信息批量装入 WebSphere Commerce 数据库中。在 WebSphere Commerce 5.5 中已引入了业务模型，它将把数据大量装入数据库。关于该业务模型的更多信息，请参阅《WebSphere Commerce 商店开发指南》。

LDAP 配置参数

安装了您希望使用的 LDAP 系统后，您需要完成配置管理器“成员子系统”页面中的相应字段，或 *instancename.xml* 文件（如果您正在手工配置 LDAP 支持）。此文件描述 *instancename.xml* 文件中 MemberSubSystem 部分的参数。注意 WebSphere Commerce 的每个实例包含一个 *instancename.xml* 文件。要避免可能的冲突，请确保对于每个实例，LDAP 配置信息都是相同的。

注意： 以下为示例代码；值是虚构的且仅用于此示例目的。对于您自己的文件，您必须对参数提供有效值。


```

<MemberSubSystem name="Member SubSystem"
  ProfileDataStorage="LDAP"
  AuthenticationMode="LDAP">
<Directory LdapPort="636"
  LdapType="SECUREWAY"
  LdapAuthenticationMode="SIMPLE"
  EntryFileName="d:/WebSphere/CommerceServer55/
xml/ldap/ldapentry.xml"
  LdapAdminPW="EaDPFd9VAf0="
  LdapVersion="3"
  LdapHost="basswood.torolab.ibm.com"
  SingleSignOn="0"
  LdapAdminDN="cn=root"
  display="false"
  MigrateUsersFromWCdb="ON"
  JNDIEnvPropName1="java.naming.referral"
  JNDIEnvPropValue1="ignore"
  JNDIEnvPropName2="java.naming.security.
protocol"
  JNDIEnvPropValue2="ssl"
  ...
  JNDIEnvPropNameN="java.naming.language"
  JNDIEnvPropValueN="en-US"
  LdapTimeOut="0" />
</MemberSubSystem>

```

参数描述

ProfileDataStorage

此属性的值确定用户和组织实体的概要文件数据存储在哪里。注意即使值为 LDAP，还是会复制一些概要文件数据到 WebSphere Commerce 数据库中。

AuthenticationMode

此属性的值确定认证用户的系统。有效值是：

- LDAP（缺省）
- DB
- 其它（用于第三方机制）

LdapAdminDN

管理员的 DN，用于执行 LDAP 上的操作。必须在 LDAP 上设置此 DN 并赋予其 LDAP 上的许可权。

LdapAuthenticationMode

此属性的值取决于您正在使用哪个目录服务器。

Simple 和 None 对 IBM SecureWay® 和 Netscape iPlanet 都是有效的。

- Simple (缺省)
- None

LdapTimeOut

在 LDAP 搜索超时前的秒数。也可以在 LDAP 服务器上指定超时值。以这两个值中的小者为准。如果 LdapTimeOut 的值是 0，那么搜索超时仅由 LDAP 服务器配置确定。缺省值是 0。

LdapVersion

LDAP 协议版本号。唯一的有效值是 3。

EntryFileName

包含了关于哪个 WebSphere Commerce 属性映射为哪个 LDAP 属性的信息的 XML 文件。

缺省值为:

 Solaris Linux Windows

`WC_installdir/
xml/ldap/ldapentry.xml`

 AIX 400

`WC_installdir/xml/ldap/ldapentry.xml`

LdapPort

LDAP 服务器使用的端口。缺省值为 389。

SingleSignOn

值为“0”表示关闭单一注册。“1”表示打开。

LdapAdminPW

无缺省值 - 空白

管理员的加密密码。如果手工修改了该 XML，则使用 `wcs_encrypt.bat` 或 `wcs_encrypt.sh` 生成它。

LdapHost

LDAP 服务器的主机名。缺省值是当前主机名。

MigrateUsersFromWCdb

应打开此选项以确保将引导用户和由商店发布进程创建的任何组织推到 LDAP。否则，应当关闭此选项。缺省值为 OFF (关闭)。

LdapType

您正在使用的 LDAP 目录服务器，例如

 Windows AIX Solaris Linux IBM SecureWay V5.1 (缺省)

 400

- OS400 - DIRECTORY SERVICES
- IBM SecureWay V3.2.2

JNDIEnvPropNameX 和 JNDIEnvPropValueX

使用这些属性指定您需要 WebSphere Commerce Server 设置的任何 JNDI 环境属性。这些属性必须在 `instancename.xml` 文件中手工指定。

注意 WebSphere Commerce 5.1 的以下更改:

- MappingFileName 更改为 EntryFileName
- ldapmap.xml 更改为 ldapentry.xml

LdapPersonRDN、LdapPersonSearchRoot、LdapPersonDefaultBase、LdapPersonOCS、LdapOrgOCS 和 LdapOrgUnitOCS 现在是 ldapentry.xml 文件的一部分，且它们的名称已更改为如下:

- LdapPersonRDN 现在是 rdnName
 - LdapPersonSearchRoot 现在是 searchBase
 - LdapPersonDefaultBase 现在是 defaultBase
 - LdapPersonOCS 现在是 objClass
 - LdapOrgOCS 现在是 objClass
 - LdapOrgUnitOCS 现在是 objClass
- 仅支持 LDAP 协议版本 3

关于 ldapentry.xml 和 ldapmap.dtd 文件的信息，请参阅第 143 页的附录 B，『LDAP 文件』。

LDAP 设计特征

以下信息突出描述了 WebSphere Commerce 中 LDAP 实现的一些设计特性。

- 多个 WebSphere Commerce 属性可以复制为带指定分隔符的单个 LDAP 属性。
- 可以从多个 LDAP 属性复制为单个 WebSphere Commerce 属性。要执行此操作，请在 ldapentry.xml 文件中使用多个 `<map>` 部分；但应该知道有可能发生覆盖。
- 单个 WebSphere Commerce 属性可复制为多个 LDAP 属性。
- 单个 LDAP 属性可复制为多个 WebSphere Commerce 属性。
- 只有用户的缺省地址保存至 LDAP。将缺省的注册地址定义为具有 `selfaddress=1` 和 `logonid=nickname` 的用户通讯录中的地址。其中登录标识是 USEREG 表中的 `logonid`，而昵称是 ADDRESS 表中的 `NICKNAME`。用户通讯录中的其它地址将不存储在 LDAP 上。
- 只有 WebSphere Commerce 注册用户而非临时用户才保存到 LDAP。
- WebSphere Commerce 允许在 LDAP 上搜索用户条目时指定多个搜索库。然而，如果找到满足搜索条件的多个用户条目，则将生成一个错误。

下表描述 WebSphere Commerce 中 LDAP 支持的一些限制。

限制	描述
LDAP 中的多值属性	WebSphere Commerce 不存储或检索多值的 LDAP 属性，但在检索到第一个值的情况下也会容忍 LDAP 中的多值属性。在更新期间，如果 LDAP 中已存在新值，则不执行更新；否则将更新多值 LDAP 属性的第一个值。

限制	描述
LDAP 参照和引用	WebSphere Commerce 不支持 LDAP 参照和引用。
多组件 RDN	只能将一个 LDAP 属性指定为 RDN 属性，而不能将 LDAP 属性的组合指定为 RDN 属性。

登台服务器

多数网上商店都每天 24 小时，一年 365 天营业，使得对系统执行维护或测试更改非常困难。WebSphere Commerce 登台服务器允许站点管理员将其生产数据库复制到登台数据库，以便在不影响客户的情况下测试更新。这有助于测试产品目录的更新，而这对测试新的购物处理命令同样也是重要的。

登台服务器由以下组件构成：

WebSphere Commerce 实例

测试和修改数据。

数据库模式脚本

为登台数据库创建登台表和触发器。登台数据库包含与生产数据库相同的模式和表，以及一组触发器来记录登台数据库中所作更改。登台数据库模式脚本将触发器添加到数据库。

使用数据库触发器将更改记录到 STAGLOG 表（一个登台表）中。无论何时更改登台数据库中的数据库表记录，STAGLOG 表都会记录此更改。

“登台复制”实用程序

允许管理员将数据从生产数据库复制到登台数据库。您可以将数据复制到与站点相关的表、与商家相关的表或个别表中。

“登台传播”实用程序

允许管理员将更改从生产数据库传播到生产数据库。STAGLOG 表中的信息标识登台数据库中必须在生产数据库中插入、更新或删除的那些记录。然后已标识的记录在生产数据库中更新。处理过的记录在 STAGLOG 表的 STGPROCESSED 列中以 1 表示。

“登台检查”实用程序

允许管理员检查登台服务器和生产服务器上两个表之间潜在的唯一索引键冲突。

登台服务器限制

在使用登台服务器之前，应该了解以下限制：

- 您无法用买方组织自我管理功能使用登台服务器。
- 所有登台表（除了 MEMBER、MBRREL、MBRROLE 和 MBRATTRVAL）的 member_id 列必须是组织或成员组，而非用户。
- 对于所有站点表，member_id 必须是 -2001 或 0。对于所有同时包含站点和商家数据的表，与站点数据相关的行的 member_id 必须是 0 或 -2001。

- 如果正在生产系统上使用 RFQ 功能，则不能使用“登台复制”命令。您必须使用推送方式，而对推送方式未提供“登台复制”命令。在启动生产站点之前，请创建登台服务器并设置登台数据库。在登台服务器上部署和测试数据，然后使用“登台传播”命令推送到生产服务器上。
- 您不能在登台服务器上创建或更新 RFQ。

键分割： 在转销商市场中，IBM 和站点管理员在登台服务器上更新数据，同时转销商在生产服务器上更新数据。这很可能导致主键冲突。

WebSphere Commerce 使用键管理器为表生成主键，键范围在 KEYS 表中定义。如果生产服务器和登台服务器使用相同的键范围，则键管理器可能在生产服务器和登台服务器上为同一个表分配相同的主键值，从而导致主键冲突。因此，必须在创建登台实例和生产实例后立即分割键范围。这将确保登台服务器和生产服务器使用不同的键范围。

目前，WebSphere Commerce 使用以下 SQL 在登台服务器和生产服务器上分割键范围。SQL 语句将当前键范围分割成三个相等的部分。

第一部分

```
update keys set upperbound=(upperbound-lowerbound)/3 + lowerbound
where tablename in (select tablename from stgmertab) or tablename
in (select tablename from stgsitetab)
```

第二部分

```
db2 update keys set upperbound = (upperbound-lowerbound)/3*2 + lowerbound,
lowerbound = (upperbound-lowerbound)/3 +lowerbound+1, counter =
counter+(upperbound-lowerbound)/3 +1 where tablename in
(select tablename from stgmertab)
or tablename in (select tablename from stgsitetab)
```

第三部分

```
db2 update keys set lowerbound = (upperbound-lowerbound)/3*2 + lowerbound +1
, counter = counter +(upperbound-lowerbound)/3*2 +1 where tablename in
(select tablename from stgmertab) or tablename in (select tablename from stgsitetab)
```

第一个 SQL 语句（第一部分）必须在生产服务器上运行。这确保生产服务器将占用整个键范围的三分之一。第二个 SQL 语句（第二部分）必须在登台服务器上运行。这意味着登台服务器将占用整个键范围的第二个三分之一。第三个 SQL 语句（第三部分）可以为将来的另一个可能的登台服务器保留。

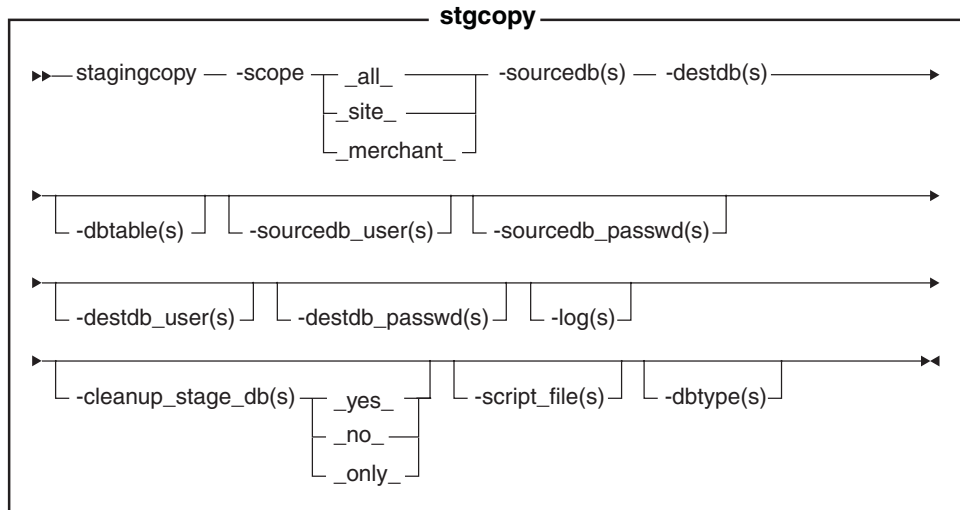
注： 上述键分割 SQL 已与实例配置过程集成在一起。您不必手工运行它们。

登台服务器命令

“登台复制”实用程序命令 (Windows 2000、AIX、Linux、iSeries 和 Solaris)：

“登台复制”实用程序将数据从生产数据库复制到登台数据库。请注意，如果 RFQ 在您的生产系统上，则不能使用此命令。关于详细信息，请参阅第 32 页的『登台服务器限制』。

要运行“登台复制”实用程序，在一台可以同时连接到登台服务器和生产服务器数据库的机器的命令行中输入以下命令。在一行中输入整个命令。这里以多行显示仅为了演示的目的。



注:

1. **Oracle** 必须在命令中包含可选参数、登录用户标识和密码，即使您当前正在用相同的用户标识运行此实用程序也是如此。
2. **400** iSeries 现在使用 stagingCopy.sh。要在 iSeries 上运行外壳程序脚本，请执行以下操作：
 - a. 作为 CCSID 不是 65535 的用户概要文件登录。
 - b. 在 OS/400 命令行上输入以下命令打开 QSHELL 命令窗口：STRQSH。
 - c. 运行如下命令：

/QIBM/ProdData/CommerceServer55/bin/stagingCopy.sh (parameters . . .)

注: 必须始终指定用户参数。

参数值

scope 至登台服务器的复制的作用域级别。指定以下值中的一个：

- **_all_**
输入 **_all_** 同时复制与站点有关的记录以及与所有商家有关的记录。
- **_site_**
输入 **_site_** 仅复制与站点有关的记录。
- **_merchant_**
输入 **_merchant_** 仅复制与所有商家有关的记录。

sourcedb

生产服务器上的数据库名称。

400 生产服务器上的数据库名称（如关系数据库目录中所示）。

Oracle 使用 *host:port:sid*。例如，myhost:1521:mydb。

destdb 登台服务器上的数据库名称。

400 生产服务器上的数据库名称（如关系数据库目录中所示）。


Oracle 使用 *host:port:sid*。例如，myhost:1521:mydb。

dbtable

(可选) 要复制的任何特定表的名称。如果记录在由 `scope` 参数指定的作用域中, 则将复制此表中的所有记录; 否则不会复制任何记录。

sourcedb_user

(可选) 创建了源数据库模式的数据库管理员的登录标识。如果未指定, 则使用当前调用此实用程序的用户标识。


 (必需) 与该 Commerce 实例关联的用户概要文件。它与源数据库模式相同。

sourcedb_passwd

(可选) 由 `sourcedb_user` 参数指定的登录标识的密码。

destdb_user

(可选) 创建了目标数据库模式的数据库管理员的登录标识。

 (必需) 与该 Commerce 实例关联的用户概要文件。它与目的地数据库模式相同。

注: 如果未指定, 则使用调用此实用程序的用户标识。


destdb_passwd

(可选) 由 `destdb_user` 参数指定的登录标识的密码。如果未指定, 系统将提示您输入密码。

log

(可选) “登台复制”实用程序在其中记录其活动和错误的文件的路径和名称。如果没有指定此参数, 则在以下日志目录中创建称为 `stagingcopy_yyyy.mm.dd_hh.mm.ss.zzz.log` 的日志文件。

   `WC_installdir/logs`

 `WC_installdir\logs`

 `WC_userdir/instances.`

cleanup_stage_db

(可选) 使用此参数在使用“登台复制”实用程序之前清理登台表。当使用 `-cleanup_stage_db` 参数清理站点数据时, 请注意可能会因为自动删除关联而删除商家数据。您应当在清理并复制站点数据之后清理并复制商家数据。Yes 是缺省值。如果您指定 no, 将不会从登台表中删除任何数据。如果复制数据发生冲突或在主键或唯一索引上存在重复键, 复制可能会失败。要使用登台复制来只清理登台数据库, 而不需要来自生产数据库的数据副本, 请将 `-cleanup_stage_db` 指定为 `only`。

script file

(可选) 当使用导出和导入将生产数据库复制到指定作用域上的登台数据库时, 由“登台复制”实用程序命令生成的 SQL 脚本文件的名称。如果您使用缺省值或指定 `-cleanup_stage_db` 为 `yes`, 则脚本文件还生成删除语句来清除登台数据库。

注: 此脚本文件在 iSeries 上不受支持。

在运行脚本之前, 请验证是否有足够的磁盘空间保存导出的表。此脚本文件位于调用“登台复制”实用程序的“登台复制”实用程序目录中。

 使用 `db2 -vtd# -f script_file_name` 来运行脚本文件。

dbtype (可选) 数据库类型 (DB2 或 Oracle)。缺省值为 DB2。




关于复制登台服务器的更多信息，请参阅示例。



复制数据至登台数据库的示例： 以下示例说明了可以如何将表从生产数据库复制到登台数据库。记住这一点很重要：如果 RFQ 在生产系统上，则不能使用“登台复制”实用程序。关于详细信息，请参阅第 32 页的『登台服务器限制』。



注意您应当在一行中输入整条命令。此处以多行显示命令仅仅为了演示的目的。






示例 1




在清理登台数据库后，将生产数据库复制到登台数据库并将作用域设置为所有：

1. 设置 PATH 环境变量。  400 不适用。
2. 配置数据库。  400 不适用。
3. 切换至希望写入日志文件的目录。  400 不适用。对于 iSeries，日志文件缺省为
/QIBM/UserData/CommerceServer55/instances/stagingcopy_{sourcedb_user}
_{destdb_user}_{timestamp}.log
4. 输入以下命令：

-  
stagingcopy -scope _all_ -sourcedb *production_database_name* -destdb
staging_database_name

-  
stagingcopy -scope _all_ -sourcedb *production_database_name* -destdb
staging_database_name-dbtype oracle -sourcedb_user *user* -sourcedb_passwd
password -destdb_user *user* -destdb_passwd *password*



-     
+ . stagingcopy.sh -scope _all_ -sourcedb *production_database_name* -destdb
+ *staging_database_name* -sourcedb_user *user* -destdb_user *user*
+



-   
. stagingcopy.sh -scope _all_ -sourcedb *production_database_name* -destdb
staging_database_name dbtype oracle -sourcedb_user *user* -sourcedb_passwd
password -destdb_user *user* -destdb_passwd *password*

注意：对 Oracle 数据库名称使用 *host:port:sid*。例如，*myhost:1521:mydb*。









5. 检查 *stagingcopy_yyyy.mm.dd_hh.mm.ss.zzz.log* 文件以验证命令是否成功。

要指定日志文件名和路径，请使用日志文件参数：

-  
stagingcopy -scope _all_ -sourcedb *production_database_name* -destdb
staging_database_name -log *log_file_name*

-  
stagingcopy -scope _all_ -sourcedb *production_database_name* -destdb
staging_database_name -log *log_file_name* -dbtype oracle -sourcedb_user *user*
-sourcedb_passwd *password* -destdb_user *user* -destdb_passwd *password*

+
+
+
+

-      `. stagingcopy.sh -scope _all_ -sourcedb production_database_name -destdb staging_database_name -log log_file_name -sourcedb_user user -destdb_user user`
-    `. stagingcopy.sh -scope _all_ -sourcedb production_database_name -destdb staging_database_name -log log_file_name dbtype oracle -sourcedb_user user -sourcedb_passwd password -destdb_user user -destdb_passwd password`

注意: 对 Oracle 数据库名称使用 *host:port:sid*。例如, myhost:1521:mydb。













如果您正使用 DB2 并且不是以数据库管理员登录, 您需要为 `-sourcedb_user`、`-sourcedb_passwd`、`-destdb_user` 和 `-destdb_passwd` 选项提供值。

示例 2

在从登台数据库中清除商家表之后, 将与商家相关的表从生产数据库复制到登台数据库:

1. 设置 PATH 环境变量。
2. 配置数据库。
3. 切换至希望写入日志文件的目录。
4. 输入以下命令:





+
+
+






-   `stagingcopy -scope _merchant_ -sourcedb production_database_name -destdb staging_database_name`
-   `stagingcopy -scope _merchant_ -sourcedb production_database_name -destdb staging_database_name -dbtype oracle -sourcedb_user user -sourcedb_passwd password -destdb_user user -destdb_passwd password`
-      `stagingcopy.sh -scope _merchant_ -sourcedb production_database_name -destdb staging_database_name -sourcedb_user user -destdb_user user`
-    `stagingcopy.sh -scope _merchant_ -sourcedb production_database_name -destdb staging_database_name -dbtype oracle -sourcedb_user user -sourcedb_passwd password -destdb_user user -destdb_passwd password`




注意: 对 Oracle 数据库名称使用 *host:port:sid*。例如, myhost:1521:mydb。

5. 检查 `stagingcopy_yyyy.mm.dd_hh.mm.ss.zzz.log` 文件以验证命令是否成功。

如果只需要清除登台数据库, 指定参数 `-cleanup_stage_db`:



-   `stagingcopy -scope _merchant_ -sourcedb production_database_name -destdb staging_database_name -cleanup_stage_db only`
-   `stagingcopy -scope _merchant_ -sourcedb production_database_name -destdb staging_database_name -cleanup_stage_db only -dbtype oracle -sourcedb_user user -sourcedb_passwd password -destdb_user user -destdb_passwd password`



```
+
+
+
•      stagingcopy.sh -scope
  _merchant_ -sourcedb production_database_name -destdb staging_database_name
  -cleanup_stage_db only -sourcedb_user user -destdb_user user
```






```
•    stagingcopy.sh -scope _merchant_ -sourcedb
  production_database_name -destdb staging_database_name -cleanup_stage_db only
  dbtype oracle -sourcedb_user user -sourcedb_passwd password -destdb_user user
  -destdb_passwd password
```




注意: 对 Oracle 数据库名称使用 *host:port:sid*。例如, myhost:1521:mydb。

如果只需要复制数据, 指定参数 *-cleanup_stage_db no*:

```
•   stagingcopy -scope _merchant_ -sourcedb
  production_database_name -destdb staging_database_name -cleanup_stage_db no
```

```
•   stagingcopy -scope _merchant_ -sourcedb
  production_database_name -destdb staging_database_name -cleanup_stage_db no
  -dbtype oracle -sourcedb_user user -sourcedb_passwd password -destdb_user
  user -destdb_passwd password
```

```
+
+
+
•      stagingcopy.sh -scope
  _merchant_ -sourcedb production_database_name -destdb staging_database_name
  -cleanup_stage_db no -sourcedb_user user -destdb_user user
```

```
•   
  stagingcopy.sh -scope _merchant_ -sourcedb production_database_name -destdb
  staging_database_name -cleanup_stage_db no dbtype oracle -sourcedb_user user
  -sourcedb_passwd password -destdb_user user -destdb_passwd password
```

注意: 对 Oracle 数据库名称使用 *host:port:sid*。例如, myhost:1521:mydb。



重要信息: 当将作用域设置为商家而复制时, 请确保您首先已复制了站点作用域数据。否则, 您的复制将失败。






示例 3

在从登台数据库中清除站点表之后, 将站点表从生产数据库复制到登台数据库。

1. 设置 PATH 环境变量。
2. 配置数据库。
3. 切换至希望写入日志文件的目录
4. 输入以下命令:

```
•   stagingcopy -scope _site_ -sourcedb
  production_database_name -destdb staging_database_name
```

```
•   stagingcopy -scope _site_ -sourcedb
  production_database_name -destdb staging_database_name -dbtype oracle
  -sourcedb_user user -sourcedb_passwd password -destdb_user user
  -destdb_passwd password
```

```
+
+
+
•     
  stagingcopy.sh -scope _site_ -sourcedb production_database_name -destdb
  staging_database_name -sourcedb_user user -destdb_user user
```

- ▶ AIX
▶ Solaris
▶ Oracle

```
stagingcopy.sh -scope _site_ -sourcedb
  production_database_name -destdb staging_database_name - dbtype oracle
  -sourcedb_user user -sourcedb_passwd password -destdb_user user
  -destdb_passwd password
```

注意: 对 Oracle 数据库名称使用 *host:port:sid*。例如, myhost:1521:mydb。

5. 检查 stagingcopy_yyyy.mm.dd_hh.mm.ss.zzz.log 文件以验证命令是否成功。

由于自动删除关联限制, 删除站点表可能影响商家表。首先清除商家数据, 然后清理站点数据, 最后复制数据:

- ▶ Windows
▶ DB2

```
stagingcopy -scope _merchant_ -sourcedb
  production_database_name -destdb staging_database_name -cleanup_stage_db only
```

- ▶ Windows
▶ Oracle

```
stagingcopy -scope _merchant_
  -sourcedb production_database_name
  -destdb staging_database_name
  -cleanup_stage_db only -dbtype oracle -sourcedb_user
  user -sourcedb_passwd
  password -destdb_user
  user -destdb_passwd
  password
```

+
+
+

- ▶ AIX
▶ 400
▶ Solaris
▶ Linux
▶ DB2

```
stagingcopy.sh -scope
  _merchant_ -sourcedb production_database_name -destdb staging_database_name
  -cleanup_stage_db only -sourcedb_user user -destdb_user user
```

- ▶ AIX
▶ Solaris
▶ Oracle

```
stagingcopy.sh -scope _merchant_ -sourcedb
  production_database_name -destdb staging_database_name -cleanup_stage_db only
  dbtype oracle -sourcedb_user user -sourcedb_passwd password -destdb_user user
  -destdb_passwd password
```

注意: 对 Oracle 数据库名称使用 *host:port:sid*。例如, myhost:1521:mydb。

- ▶ Windows
▶ DB2

```
stagingcopy -scope _site_ -sourcedb
  production_database_name -destdb staging_database_name -cleanup_stage_db only
```

- ▶ Windows
▶ Oracle

```
stagingcopy -scope _site_ -sourcedb
  production_database_name -destdb staging_database_name -cleanup_stage_db only
  -dbtype oracle -sourcedb_user user -sourcedb_passwd password -destdb_user
  user -destdb_passwd password
```

+
+
+

- ▶ AIX
▶ 400
▶ Solaris
▶ Linux
▶ DB2

```
stagingcopy.sh -scope _site_
  -sourcedb production_database_name -destdb staging_database_name
  -cleanup_stage_db only -sourcedb_user user -destdb_user user
```











- ▶ AIX
▶ Solaris
▶ Oracle

```
stagingcopy.sh -scope _site_ -sourcedb
  production_database_name -destdb staging_database_name -cleanup_stage_db only
  dbtype oracle -sourcedb_user user -sourcedb_passwd password -destdb_user user
  -destdb_passwd password
```













注意: 对 Oracle 数据库名称使用 *host:port:sid*。例如, myhost:1521:mydb。

- ▶ Windows
▶ DB2

```
stagingcopy -scope _site_ -sourcedb
  production_database_name -destdb staging_database_name -cleanup_stage_db no
```

-   `stagingcopy -scope _site_ -sourcedb production_database_name -destdb staging_database_name -cleanup_stage_db no -dbtype oracle -sourcedb_user user -sourcedb_passwd password -destdb_user user -destdb_passwd password`
- +      `stagingcopy.sh -scope _site_ -sourcedb production_database_name -destdb staging_database_name -cleanup_stage_db no -sourcedb_user user -destdb_user user`
- +    `stagingcopy.sh -scope _site_ -sourcedb production_database_name -destdb staging_database_name -cleanup_stage_db no -dbtype oracle -sourcedb_user user -sourcedb_passwd password -destdb_user user -destdb_passwd password`

注意: 对 Oracle 数据库名称使用 `host:port:sid`。例如, `myhost:1521:mydb`。

-   `stagingcopy -scope _merchant_ -sourcedb production_database_name -destdb staging_database_name -cleanup_stage_db no`
-   `stagingcopy -scope _merchant_ -sourcedb production_database_name -destdb staging_database_name -cleanup_stage_db no -dbtype oracle -sourcedb_user user -sourcedb_passwd password -destdb_user user -destdb_passwd password`
- +      `stagingcopy.sh -scope _merchant_ -sourcedb production_database_name -destdb staging_database_name -cleanup_stage_db no -sourcedb_user user -destdb_user user`
-    `stagingcopy.sh -scope _merchant_ -sourcedb production_database_name -destdb staging_database_name -cleanup_stage_db no -dbtype oracle -sourcedb_user user -sourcedb_passwd password -destdb_user user -destdb_passwd password`





注意: 对 Oracle 数据库名称使用 `host:port:sid`。例如, `myhost:1521:mydb`。

示例 4

生成以下脚本以清除生产数据库并将其复制到登台数据库, 且作用域为所有。

 此示例不适用于 OS/400 for iSeries, 因为 `-script` 选项在 iSeries 上不受支持。

1. 设置 `PATH` 环境变量。
2. 配置数据库。
3. 切换至希望写入日志文件的目录。
4. 输入以下命令:

-   `stagingcopy -scope _all_ -sourcedb production_database_name -destdb staging_database_name -script_file stage_copy.sql`
-   `stagingcopy -scope _all_ -sourcedb production_database_name -destdb staging_database_name -script_file stage_copy.sql -dbtype oracle -sourcedb_user user -sourcedb_passwd password -destdb_user user -destdb_passwd password`

+
+
+

- ▶ **AIX** ▶ **Solaris** ▶ **Linux** ▶ **DB2** `stagingcopy.sh -scope _all_ -sourcedb production_database_name -destdb staging_database_name -script_file stage_copy.sql -sourcedb_user user -destdb_user user`
- ▶ **AIX** ▶ **Solaris** ▶ **Oracle** `stagingcopy.sh -scope _all_ -sourcedb production_database_name -destdb staging_database_name -script_file stage_copy.sql dbtype oracle -sourcedb_user user -sourcedb_passwd password -destdb_user user -destdb_passwd password`

注意：对 Oracle 数据库名称使用 *host:port:sid*。例如，*myhost:1521:mydb*。

5. 检查 `stagingcopy_yyyy.mm.dd_hh.mm.ss.zzz.log` 文件以验证命令是否成功。

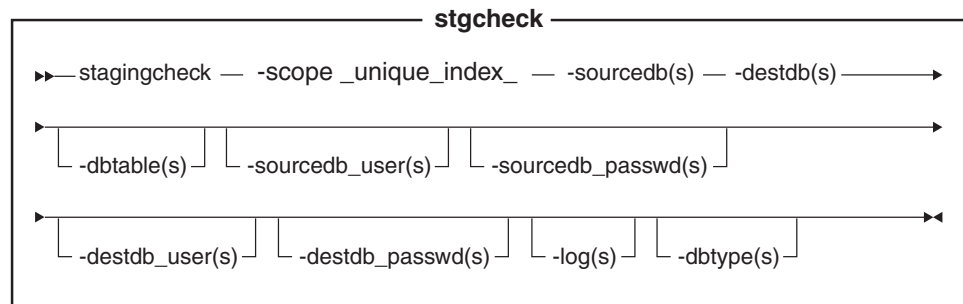
“登台复制”实用程序生成 `stage_copy.sql` 脚本来清理和复制数据库。▶ **DB2** 如果您正在使用 DB2，请运行以下脚本：

1. 以数据库管理员 (DBA) 登录。
2. 打开 DB2 命令窗口。
3. 输入：`db2 -vtd# -f stage_copy.sql`

▶ **Oracle** 如果您正在使用 Oracle，请运行以下脚本：

1. 打开 SQLPlus 窗口。
2. 作为 dba 连接。
3. 输入：`@stage_copy.sql`

“登台检查”命令 (**Windows 2000、AIX、iSeries、Linux 和 Solaris**)：“登台检查”命令确定在登台数据库和生产数据库之间是否有唯一索引键冲突。要运行“登台检查”实用程序，在登台服务器或生产服务器上的命令行中输入如下命令：在一行中输入整个命令。这里以多行显示仅为了演示的目的。



▶ **Oracle** **注意：**必须在命令中包含可选参数、登录用户标识和密码，即使您当前正在用相同的用户标识运行此实用程序也是如此。

▶ **400** iSeries 现在使用 `stagingCheck.sh`。要在 iSeries 上运行外壳程序脚本，请执行以下操作：

1. 作为 CCSID 不是 65535 的用户概要文件登录。
2. 在 OS/400 命令行上输入以下命令打开 QSHELL 命令窗口：`STRQSH`。
3. 运行如下命令：

```
/QIBM/ProdData/CommerceServer55/bin/stagingCheck.sh (parameters . . .)
```

注：必须始终指定用户参数。

参数值

scope 至登台服务器的复制的作用域级别。指定 `_unique_index_`。

sourcedb

登台服务器上的数据库名称。

 **Oracle** 使用 `host:port:sid`。例如，`myhost:1521:mydb`。

destdb 生产服务器上的数据库名称。


 **Oracle** 使用 `host:port:sid`。例如，`myhost:1521:mydb`。

dbtable

(可选) 要检查是否有唯一键冲突的特定表的名称。

sourcedb_user

(可选) 创建了登台数据库模式的数据库管理员的登录标识。如果未指定，则使用当前调用此实用程序的用户标识。


 **400** (必需) 与该 Commerce 实例关联的用户概要文件。它与源数据库模式相同。

sourcedb_passwd

(可选) 由 `sourcedb` 参数指定的登录标识的密码。

destdb_user

(可选) 创建了生产数据库模式的数据库管理员的登录标识。如果未指定，则使用调用此实用程序的用户标识。


 **400** (必需) 与该 Commerce 实例关联的用户概要文件。它与目的地数据库模式相同。

destdb_passwd

(可选) 由 `destdb_user` 参数指定的登录标识的密码。如果未指定，系统将提示您输入密码。

log (可选) “登台复制”实用程序在其中记录其活动和错误的文件的路径和名称。如果没有指定此参数，则在以下日志目录中创建称为 `stagingcopy_yyyy.mm.dd_hh.mm.ss.zzz.log` 的日志文件。

 **AIX**  **Solaris**  **Linux** `WC_installdir/logs`

 **Windows** `WC_installdir\logs`

 **400** `WC_userdir/instances`。

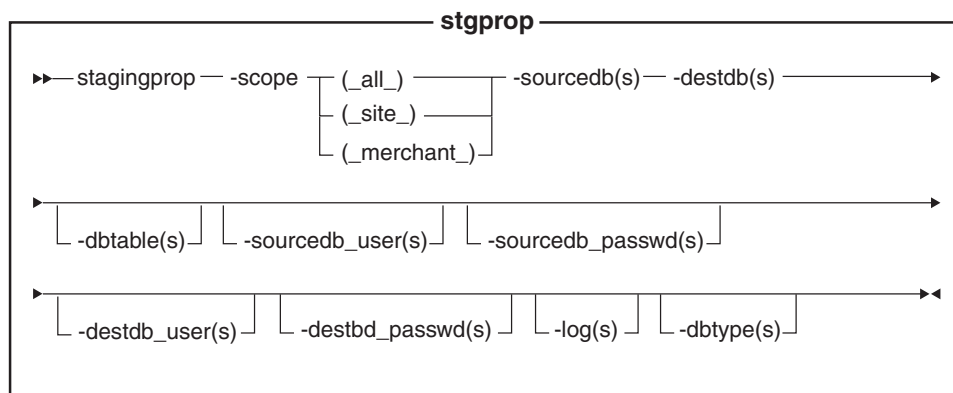
dbtype (可选) 数据库类型 (DB2 或 Oracle)。缺省值为 DB2。

关于检查唯一索引键冲突的更多信息，请参阅示例。

“登台传播”实用程序命令 (*Windows 2000*、*AIX*、*iSeries*、*Linux* 和 *Solaris*)：

“登台传播”实用程序将数据从登台数据库复制到生产数据库。请注意您不能在登台服务器上创建或更新 RFQ 对象。

要运行“登台传播”实用程序，在一台可以同时连接到登台服务器和生产服务器数据库的机器的命令行中输入以下命令。在一行中输入整个命令。这里以多行显示仅为了演示的目的。



► Oracle

注意： 必须在命令中包含可选参数、登录用户标识和密码，即使您当前正在用相同的用户标识运行此实用程序也是如此。

► 400 iSeries 现在使用 stagingProp.sh。要在 iSeries 上运行外壳程序脚本，请执行以下操作：

1. 作为 CCSID 不是 65535 的用户概要文件登录。
2. 在 OS/400 命令行上输入以下命令打开 QSHELL 命令窗口：STRQSH。
3. 运行如下命令：
`/QIBM/ProdData/CommerceServer55/bin/stagingProp.sh (parameters . . .)`

注： 必须始终指定用户参数。

参数值

scope

至生产服务器的传播的作用域级别。指定以下值中的一个：

- `_all_`
输入 `_all_` 同时传播与站点有关的记录和与所有商家有关的记录。
- `_site_`
输入 `_site_` 仅传播与站点有关的记录。
- `_merchant_`
输入 `_merchant_` 仅传播与所有商家有关的记录。

sourcedb

登台服务器上的数据库名称。 ► Oracle 使用 `host:port:sid`。例如，`myhost:1521:mydb`。

destdb 生产服务器上的数据库名称。

► Oracle 使用 `host:port:sid`。例如，`myhost:1521:mydb`。


dbtable

(可选) 要传播的任何特定表的名称。如果记录在由 `scope` 参数指定的作用域中，则此表中的所有已更改记录都将被传播；否则，不传播任何记录。

sourcedb_user


(可选) 创建了源数据库模式的数据库管理员的登录标识。如果未指定，则使

+

+ 用当前调用此实用程序的用户标识。
+  400 (必需) 与该 Commerce 实例关联的用户概要文件。它与源数据库模
+ 式相同。

sourcedb_passwd






(可选) 由 `sourcedb_user` 参数指定的登录标识的密码。

+ **destdb_user**
+ (可选) 创建了目标数据库模式的数据库管理员的登录标识。如果未指定, 则
+ 使用调用此实用程序的用户标识。当使用远程数据库时, 此参数是必填的。
+  400 (必需) 与该 Commerce 实例关联的用户概要文件。它与目的地数据
+ 库模式相同。

destdb_passwd

(可选) 由 `destdb_user` 参数指定的登录标识的密码。如果未指定, 系统将提示您输入密码。当使用远程数据库时, 此参数是必填的。

+ **log** (可选) “登台传播”实用程序在其中记录其活动和错误的文件的路径和名
+ 称。如果没有指定此参数, 则在以下日志目录中创建称为
+ `stagingprop_yyyy.mm.dd_hh.mm.ss.zzz.log` 的日志文件。

+  AIX  Solaris  Linux `WC_installdir/logs`
+  Windows `WC_installdir\logs`
+  400 `WC_userdir/instances`

dbtype (可选) 数据库类型 (DB2 或 Oracle)。缺省值为 DB2。

关于传播至登台服务器的更多信息, 请参阅以下示例。










传播数据至生产数据库的示例: 以下示例说明了如何将更改的记录从登台数据库传播至生产数据库。

注意您应当在一行中输入命令。此处以多行显示命令仅仅为了演示的目的。

示例 1

将所有更改从登台服务器数据库传播至生产数据库。

1. 设置 `PATH` 环境变量。
2. 配置数据库。
3. 切换至希望写入日志文件的目录
4. 输入以下命令:

-  Windows  DB2 `stagingprop -scope_all_ -sourcedb staging_database_name -destdb production_database_name`
-  Windows  Oracle `stagingprop -scope_all_ -sourcedb staging_database_name -destdb production_database_name -dbtype oracle -sourcedb_user user -sourcedb_passwd password -destdb_user user -destdb_passwd password`
-  AIX  400  Solaris  Linux  DB2 `stagingprop.sh -scope_all_ -sourcedb staging_database_name -destdb production_database_name -sourcedb_user user -destdb_user user`

- AIX
Solaris
Oracle

```
stagingprop.sh -scope _all_
-sourcedb staging_database_name -destdb production_database_name dbtype
oracle -sourcedb_user user -sourcedb_passwd password -destdb_user user
-destdb_passwd password
```

注意：对 Oracle 数据库名称使用 *host:port:sid*。例如，myhost:1521:mydb。

- 检查日志文件以验证命令成功。检查日志参数的描述以确定用于您的数据库类型和平台的此日志文件的缺省位置。

“登台传播”实用程序将首先传播所有站点数据，然后传播所有商家数据至生产数据库。如果发生错误，整个事务将回滚。

要指定日志文件名和路径，请使用日志文件参数：

- Windows
DB2

```
stagingprop -scope _all_ -sourcedb
staging_database_name-destdb production_database_name -loglog_file_name
```
- Windows
Oracle

```
stagingprop -scope _all_ -sourcedb
staging_database_name-destdb production_database_name -loglog_file_name-dbtype
oracle -sourcedb_user user -sourcedb_passwd password -destdb_user user
-destdb_passwd password
```

+
+
+

- AIX
400
Solaris
Linux
DB2

```
stagingprop.sh -scope _all_
-sourcedb staging_database_name -destdb production_database_name
-loglog_file_name -sourcedb_user user -destdb_user user
```
- AIX
Solaris
Oracle

```
stagingprop.sh -scope _all_ -sourcedb staging_database_name-destdb
production_database_name -loglog_file_name-dbtype oracle -sourcedb_user user
-sourcedb_passwd password -destdb_user user -destdb_passwd password
```

注意：对于 Oracle 数据库名称使用 *host:port:sid*，例如 myhost:1521:mydb。

如果您正使用 DB2 并且不是以数据库管理员登录，您需要为 `-sourcedb_user`、`-sourcedb_passwd`、`-destdb_user` 和 `-destdb_passwd` 选项提供值。

示例 2

将所有已修改的站点数据从登台数据库传播至生产数据库。






- 设置 PATH 环境变量。
- 配置数据库。
- 切换至希望写入日志文件的目录
- 输入以下命令：




- Windows
DB2

```
stagingprop -scope _site_ -sourcedb
staging_database_name -destdb production_database_name
```
- Windows
Oracle

```
stagingprop -scope _site_ -sourcedb
staging_database_name -destdb production_database_name -dbtype oracle
-sourcedb_user user -sourcedb_passwd password -destdb_user user
-destdb_passwd password
```

+
 +
 +
 +

-      stagingprop.sh -scope _site_ -sourcedb staging_database_name -destdb production_database_name -sourcedb_user user -destdb_user user

-    stagingprop.sh -scope _site_ -sourcedb staging_database_name -destdb production_database_name dbtype oracle -sourcedb_user user -sourcedb_passwd password -destdb_user user -destdb_passwd password



注意：对 Oracle 数据库名称使用 *host:port:sid*。例如，myhost:1521:mydb。



5. 检查日志文件以验证命令成功。检查日志参数的描述以确定用于您的数据库类型和平台的此日志文件的缺省位置。

示例 3






将所有已修改商家数据从登台数据库传播至生产数据库（在已经传播了站点数据之后）。




1. 设置 PATH 环境变量。
2. 配置数据库。
3. 切换至希望写入日志文件的目录
4. 输入以下命令：

-   stagingprop -scope _merchant_ -sourcedb staging_database_name -destdb production_database_name

-   stagingprop -scope _merchant_ -sourcedb staging_database_name -destdb production_database_name -dbtype oracle -sourcedb_user user -sourcedb_passwd password -destdb_user user -destdb_passwd password

+
 +
 +

-      stagingprop.sh -scope _merchant_ -sourcedb staging_database_name -destdb production_database_name -sourcedb_user user -destdb_user user






-    stagingprop.sh -scope _merchant_ -sourcedb staging_database_name -destdb production_database_name dbtype oracle -sourcedb_user user -sourcedb_passwd password -destdb_user user -destdb_passwd password

注意：对 Oracle 数据库名称使用 *host:port:sid*。例如，myhost:1521:mydb。

5. 检查日志文件以验证命令成功。检查日志参数的描述以确定用于您的数据库类型和平台的此日志文件的缺省位置。

定制数据库表要求

如果已经通过创建新表定制了数据库模式，则必须满足以下要求才可使用登台服务器：

- **您必须定义一个主键或唯一索引。**
登台服务器基于此键运行。要避免在 STAGLOG 表中记录多余数据，可只记录此键（主键或唯一索引）。登台实用程序将此键用于压缩以及查找要传播的数据。如果键不存在，登台实用程序将无法工作。
- **在表之间不能存在引用完整性（RI）约束循环。**
登台服务器始终在子表之前传播父表。如果存在 RI 约束循环，登台服务器将无法区分子表和父表。
- **数据库表仅包含配置数据。**
在商家到消费者的方案中，配置数据（例如，产品目录和产品目录条目）处于站点管理员控制下。如果表包含了操作数据，则客户可在站点管理员将此表复制至登台数据库后在生产数据库中更改同一个表。这将引起潜在键冲突或 RI 约束违例。
- **数据库表不能包含任何对操作表的引用。**
要进行传播的表不应当包含对操作表主键的任何外键引用。如果存在这样的引用，则如果客户在登台复制之后删除主键，数据将无法恢复至产品数据库。
-      **在生产数据库中插入两个表时不能存在插入触发器。**
对于登台服务器所包含的任意两个表（例如，R1 和 R2），当向生产数据库中的 R2 和 R1 执行插入时，不能存在向 R1 或 R2 插入行的触发器。插入触发器可创建对两个数据库的更新并产生键问题。
- **MEMBER 表不能具有唯一索引。**
- **对定制数据库表的删除限制必须小心使用。**
删除限制禁止数据库清理实用程序的执行。您还可能在清理登台数据库时遇到困难。在可清理登台数据库之前，您必须手工使用带强制选项的数据库清理实用程序命令以清理这些表。否则，清理登台数据库将失败。

要为定制表准备登台服务器，请参阅“为定制的表配置登台服务器”。

登台服务器疑难解答（Windows 2000、AIX、Linux 和 Solaris）


1. 在使用登台服务器命令 stgprop 时，您可能接收到以下消息：
SQLSTATE 54001: 语句太长或太复杂。

确保您已按照配置数据库中的建议设置了数据库的 stmheap 大小。您应当验证您是否具有足够的内存支持这种增加。

2. 如果“登台复制”实用程序未能完成处理，则可能是数据库日志太小了。从 DB2 命令窗口发出命令：

```
update database configuration for staging_server using logprimary 50
logfilsiz 1000
db2 terminate
db2stop
db2start
```

其中，*staging_server* 是登台服务器数据库的名称。如果问题仍然存在，请尝试对参数 logprimary 或 logfilsiz 使用一个更高的值。

3.  确保已经正确配置了 DB2 用户（如 db2inst1）的资源。输入以下命令将显示您的资源限制：ulimit -a
将数据段配置成 240MB，并将堆栈配置成 16MB。
关于附加信息，请参阅 *DB2 Command Reference*。

WebSphere Commerce 登台表

您可以将 WebSphere Commerce 登台表从生产服务器复制到登台服务器。按照包含的是与站点有关数据、与商家有关数据还是同时包含与两者有关的数据，对这些表分组。

每个表组列出在以下页面上：

- 站点数据作用域
- 商家数据作用域
- 站点和商家数据作用域

登台表的触发器

已经为 WebSphere Commerce 登台表定义了以下触发器。如果这些设置包含相同数据作用域及键特征，则您可以将它们应用到定制表。请参阅下面的链接，以按数据作用域查看表。

- 站点数据作用域
- 商家数据作用域
- 站点或商家数据作用域

站点数据作用域： 以下图表列出了站点数据作用域下的数据库表。

表名	唯一索引				
	整型 1	整型 2	字符型 1	整型 3	字符型 2
ACACGPDESC	是	是			
ACACTACTGP	是	是			
ACACTDESC	是	是			
ACACTGRP	是				
ACACTION	是				
ACATTR	是				
ACATTRDESC	是	是			
ACCCMDTYPE			是		
ACRELATION	是				
ACRELDDESC	是	是			
ACRESACT	是	是			
ACRESATREL	是	是			
ACRESCGRY	是				
ACRESGRP	是				
ACRESGPDES	是	是			
ACRESGPRES	是	是			
ACRESPRIM	是				
ACRESREL	是	是			
ATTACHUSG			是		
ATTRTYPE			是		
BUYERPOTYP	是				
CALUSAGE	是				
CATENTTYPE			是		

CATRELTYPE			是		
CHKARRANG	是	是			
CHKCMD	是				
COUNTCODE			是		是
COUNTRY	是		是		
DEVICEFMT	是				
FLCOMPOSE	是	是		是	
FLDOMNDESC	是	是			
FLOW	是				
FLOWDESC	是	是			
FLOWDOMAIN	是				
FLOWTYPE	是				
FLSTATEDCT	是				
FLSTATEGP	是				
FLSTATEREL	是	是			
FLSTDCTDSC	是	是			
FLSTGPDSC	是	是			
FLTRANDSC	是	是			
FLTRANSITN	是				
FLTYPEDESC	是	是			
ICDATAREG	是				
INVRSRVTYP	是				
INVRSRVDSC	是	是			
ITEMTYPE			是		
LANGUAGE	是				
LANGUAGEDS	是	是			
MASSOC			是		
MASSOCTYPE			是		
MBRATTR	是				
MBRGRPTYPE	是				
OPERATOR	是				
OPERATRDSC	是	是			
ORDCHNLTYP	是				
OUTPUTQ	是				
OUTPUTQDSC	是	是			
PARTROLE	是				
PARTROLEDS	是	是			
PLCYTYPDSC	是		是		
POLICYTYPE	是				
QTYCONVERT			是		是
QTYUNIT			是		
QTYUNITDSC	是		是		

ROLE	是				
SCHCMD	是				
SETCURR			是		
SETCURRDSC	是		是		
STATECODE			是		是
STATEPROV	是		是		
STORECGRY	是				
TAXTYPE	是				
TCSUBTYPDS	是		是		
TCSUBTYPE			是		
TCTYPE			是		
TFALGOPOL	是				
TFALGOTYPE	是				
TFALGPOLDS	是	是			
TFALGTYPDS	是	是			
TFDOMAIN	是				
TFDOMDSC	是	是			
TFSBDOMAIN	是				
TFSBDOMDSC	是	是			
TRDTYPE	是				
TRDTYPEDSC	是	是			
TXCDScheme	是				

商家数据作用域: 以下图表列出了商家数据作用域下的数据库表。

表名	唯一索引				
	整型 1	整型 2	字符型 1	整型 3	字符型 2
ACCCMDGRP	是				
ACCCUSTEXC	是	是			
ACCOUNT	是				
ACORGPOL	是	是			
ACPOLDESC	是	是			
ACPOLICY	是				
ATTACHMENT	是				
ATTRIBUTE	是	是			
ATTRVALUE	是	是			
BASEITEM	是				
BASEITEM	是	是			
BUYERPO	是				
CALCODE	是				
CALCODEDSC	是	是			
CALCODEMGP	是	是			

CALCODTXEX	是	是			
CALMETHOD	是				
CALRANGE	是				
CALRLOOKUP	是				
CALRULE	是				
CALRULEMGP	是	是			
CALSCALE	是				
CALSCALEDS	是	是			
CATALOG	是				
CATALOGDSC	是	是			
CATCNTR	是	是			
CATCONFINF	是				
CATENCALCD	是				
CATENTATTR	是				
CATENTDESC	是	是			
CATENTREL	是	是	是		
CATENTRY	是				
CATENTSHIP	是				
CATGPCALCD	是				
CATGPENREL	是	是		是	
CATGROUP	是				
CATGRPATTR	是	是			
CATGRPDESC	是	是			
CATGRPPS	是	是		是	
CATGRPREL	是	是		是	
CATGRPTPC	是	是		是	
CATTOGRP	是	是			
CHARGETYPE	是				
CHRGTYPDSC	是	是			
CNTRDISPLY	是				
CNTRNAME	是		是		
CONTRACT	是				
CREDITLINE	是				
CRULESCALE	是	是			
CURCONVERT	是				
CURCVLIST	是	是	是		是
CURFMTDESC	是	是	是		
CURFORMAT	是		是		
CURLIST	是		是		
DISPCGPREL	是				
DISPENTREL	是				
DISTARRANG	是				

FLOWADMIN	是				
INVADJCODE	是				
INVADJDESC	是	是			
ITEMSPC	是				
ITEMVERSN	是				
JURST	是				
JURSTGPREL	是	是			
JURSTGROUP	是				
LANGPAIR	是	是		是	
LISTPRICE	是		是		
MASSOCCECE	是				
MASSOCGPGP	是				
MBRATTRVAL	是				
MBRGRPUSG	是	是			
MBRREL	是	是			
MBRROLE	是	是		是	
MGPTRDPCSN	是	是			
OFFER	是				
OFFERDESC	是	是			
OFFERPRICE	是		是		
PARTICIPNT	是				
PATTRDESC	是	是			
PATTRIBUTE	是				
PATTRPROD	是	是			
PKGATTR	是				
PKGATTRVAL	是				
PKGITEMREL	是	是		是	
POLICY	是				
POLICYCMD	是		是		
POLICYDESC	是	是			
POLICYTC	是	是			
PRODSETDSC	是	是			
PRODUCTSET	是				
PRSETCEREL	是	是			
QTYFMTDESC	是	是	是		
QTYFORMAT	是		是		
RTNDNYDESC	是	是			
RTNDNYRSN	是				
RTNDSPCODE	是				
RTNDSPDESC	是	是			
RTNREASON	是				
RTNRSNDESC	是	是			

SHPARJURGP	是	是			
SHPARRANGE	是				
SHPJCRULE	是				
SHPMODEDSC	是	是			
STENCALUSG	是	是			
STORECAT	是	是			
STORECENT	是	是			
STORECGRP	是	是			
STORECNTR	是	是			
STOREDEF	是				
STOREITEM	是	是			
STORELANG	是	是			
STOREMBRGP	是	是			
STORITMFFC	是	是		是	
STORLANGDS	是	是		是	
TAXCGRY	是				
TAXCGRYDS	是	是			
TAXJCRULE	是	是		是	
TCDESC	是	是			
TDPSCNCNTR	是	是			
TERMCOND	是				
TRADEPOSCN	是				
TRADING	是				
TRDATTACH	是	是			
TRDDESC	是	是			
TXCDCLASS	是				
VENDOR	是				
VENDORDESC	是	是			
VERSIONSPC	是				
VIEWREG	是	是	是		

站点和商家数据作用域: 以下图表列出了站点和商家数据作用域下的数据库表。

表名	唯一索引				
	整型 1	整型 2	字符型 1	整型 3	字符型 2
CMDREG	是		是		
FFMCENTDS	是	是			
FFMCENTER	是				
MBRGRP	是				
MBRGRPCOND	是				
MEMBER	是				
ORGENCY	是				

SHIPMODE	是				
STADDRESS	是				
STORE	是				
STOREENT	是				
STOREENTDS	是	是			
STOREGRP	是				
URLREG	是		是		
VIEWREG	是	是	是		

连接服务

WebSphere Commerce 使用 HTTP 程序适配器、WebSphere MQ 适配器和 CrossWorlds 适配器进行到外部系统的连接。以下各节描述每个适配器以及如何 WebSphere Commerce 中使用它们。

HTTP 程序适配器

HTTP 程序适配器允许外部系统通过 HTTP 协议传递 XML 请求来与 WebSphere Commerce 通信。HTTP 程序适配器为外部系统（例如，采购系统）提供一种通过 HTTP 与 WebSphere Commerce 通信的公共方法，从而允许 WebSphere Commerce 充当这些系统的供应商以进行买方 / 供应商交易。HTTP 程序适配器通过执行以下操作处理进入的 XML 请求：

- 识别请求并验证其是否为 XML 请求。如果请求的以下三个属性受支持，则它可以识别为 XML 请求。
 - 内容类型
 - 方法
 - 字符编码

受支持的请求属性在适配器配置中指定。

- 抽取请求的输入流。
- 调用消息映射器并传递输入流的内容。
- 接收 CommandProperty 对象，此对象表示由消息映射器返回的 WebSphere Commerce 命令。
- 确定生成响应的正确设备格式。
- 执行命令。
- 发送 XML 响应消息，此消息由 JSP 创建并以所执行命令指定的查看名称和所接收到的请求的设备格式为基础。

每一请求都当作独立的会话来处理。消息凭证在消息的控制区域指定。缺省情况下，HTTP 程序适配器检查消息中的用户标识和密码参数以确定请求的可靠性。因为旧的消息不支持凭证规范，所以 HTTP 程序适配器不支持旧的消息。

HTTP 程序适配器的有效期存在于整个 WebSphere Commerce 实例中。它在实例启动时初始化（除非除去其配置参数或未启用此适配器），并且只要实例在运行，它就驻留。

注：缺省情况下禁用 HTTP 程序适配器。

关于 WebSphere Commerce 如何处理从设备接收的请求的体系结构信息，请参阅《*WebSphere Commerce 编程指南与教程*》。关于配置 HTTP 程序适配器以及其它连接服务的信息，请参阅 WebSphere Commerce 产品联机帮助。

WebSphere MQ 侦听器

WebSphere MQ 侦听器是 WebSphere Commerce 的一个组件，它通过 WebSphere MQ 处理入站消息以实现与后端系统的集成。WebSphere MQ 的侦听器既使用 Websphere MQ 适配器检索 MQ 消息，也使用程序适配器（被调用来执行这些消息）。

WebSphere MQ 侦听器具有一组预定义消息，这些消息有助于将 WebSphere Commerce 业务处理与后端或外部系统处理集成在一起。每条进入消息将激活 WebSphere Commerce 中的进程，来更新数据库表或执行其它操作。关于所提供消息的更多详细信息，请参阅 WebSphere Commerce 联机帮助中的后端集成和实现集成消息的信息。除现有的预定义消息外，WebSphere MQ 侦听器还支持消息扩展和新消息。

WebSphere MQ 作为中间件： WebSphere MQ 适配器允许您使用 WebSphere MQ 作为中间件将后端和外部系统与 WebSphere Commerce 集成在一起。WebSphere MQ 适配器允许 WebSphere Commerce 从后端系统和外部系统接收消息。支持的软件为 WebSphere MQ V5.3 或更高版本，带有 WebSphere MQ 客户机产品扩展。WebSphere MQ 客户机是 WebSphere MQ 的客户机并作为独立的下载提供。

可通过使用绑定或客户机方式将 WebSphere MQ 设置为您的中间件。当 WebSphere Commerce 安装在与 WebSphere MQ 服务器相同的机器上，且它与 WebSphere MQ 服务器通过 WebSphere MQ Java 使用 Java 本机接口相连接时应使用绑定方式。由于通信是通过直接 JNI 调用而非通过网络传送到队列管理器 API 的，因此通常情况下，绑定方式提供的性能比使用网络连接的客户机方式所提供的性能更好。当 WebSphere Commerce 安装在一台机器上，而 WebSphere MQ 服务器安装在后端系统上时，应使用客户机方式。

要验证 WebSphere MQ 连接、队列和通道，可以运行测试程序在 WebSphere MQ 队列中放入和获取消息。关于详细信息，请参阅您的 WebSphere MQ 文档以及《*WebSphere Commerce 附加软件指南*》。

CrossWorlds 适配器

CrossWorlds 适配器提供了一种新的机制来扩展 IBM WebSphere Commerce 与 IBM CrossWorlds 的业务集成。此新适配器使用 CrossWorlds 服务器访问接口，这是一个应用程序编程接口（API），它允许外部进程在 IBM CrossWorlds InterChange Server（CW ICS）中执行协作。有了此适配器，WebSphere Commerce 可通过 IBM CrossWorlds 向外部系统发送同步消息来与这些系统集成在一起。此适配器允许 WebSphere Commerce 向 CrossWorlds 服务器发送消息并等待回复。获取响应后，调用命令可继续处理其它业务逻辑。

IBM CrossWorlds 系统是软件集成产品套件，包含用于常见业务集成需求的预置业务逻辑模板（称为协作）以及各种开发和管理工具。协作定义并自动化常见的特定于行业的业务过程步骤，例如订单管理和制造物料清单管理。协作还用来协调并扩展完全不同的企业软件产品的业务过程并帮助在它们之间实现有意义的数据交换。

注：关于 CrossWorlds 适配器的更多信息，请参阅《*WebSphere Commerce 附加软件指南*》和 WebSphere Commerce 产品和开发联机帮助。

集成 WebSphere Commerce 和 IBM CrossWorlds InterChange Server:

WebSphere Commerce 被设计为与 IBM CrossWorlds InterChange Server (ICS) 进行集成。这使 WebSphere Commerce 可通过 ICS 与后端系统集成。WebSphere Commerce 与 ICS 的此项集成支持三种类型的消息流:

- 出站异步消息流允许 Websphere Commerce 向 ICS 发送消息以进行处理。
- 入站消息流允许 Websphere Commerce 从 ICS 接收消息。
- 同步消息流允许 WebSphere Commerce 向 ICS 发送请求 / 回复消息。

出站和入站流使用 Websphere Commerce 的 Websphere 业务集成 (WBI) 适配器。请求 / 回复流使用 ICS 的 Websphere Commerce SAI 适配器。

WebSphere Commerce 出站过程: WebSphere Commerce 中的事务可触发要通过 WebSphere Commerce 消息传递系统发送到其它应用程序的消息。通过消息传递系统生成的消息可用来通过集成服务器 (例如 Websphere Business Integration Server) 与其它应用程序集成。例如, 可从 WebSphere Commerce 生成 OrderCreate XML 消息以在后端系统中创建订单。

在 WebSphere Commerce 中创建订单时, 可将它配置为向另一个系统发送与订单相关的信息以供进一步处理 (例如供货)。使用 IBM Websphere Business Integration Server 的客户可配置 Websphere Commerce 和 IBM CrossWorlds ICS 以便可以将来自 Websphere Commerce 的 XML 消息发送到 IBM CrossWorlds ICS 以进行处理。在此情况下, 订单是通过 XML 消息从 WebSphere Commerce 发送到 ICS 的。IBM CrossWorlds ICS 可处理此 XML 消息并发送到其它系统 (例如 SAP)。

在 WebSphere Commerce 中下订单时, 将生成 XML 格式的 OrderCreate 消息并置于 WebSphere MQ 输出队列中。WebSphere Commerce 代理程序的 WBI 适配器持续地轮询新消息, 它将这个消息从 WebSphere MQ 输出队列传送到连接器控制器。

控制器接收 WebSphere Commerce 特定业务对象并调用映射来生成一般业务对象 (GBO)。GBO 被传送到对应的协作, 该协作处理它们并向后端系统发送请求。关于集成 Websphere Commerce 和 IBM CrossWorlds ICS 的详细信息, 请参阅关于 Websphere Commerce 的 WBI 适配器的文档:

<http://www-3.ibm.com/software/websphere/crossworlds/library/doc/v411/welcome.html>。

WebSphere Commerce 入站过程: Websphere Commerce 被设计为与 IBM CorssWorlds ICS 集成以支持从 ICS 到 Websphere Commerce 的请求。Websphere Commerce 消息传递系统调用 Websphere Commerce 业务逻辑来支持来自 ICS 的请求。

在此方案中, 现有 WebSphere Commerce 消息用来调用由来自 IBM CrossWorlds ICS 的请求触发的业务逻辑。

WebSphere Commerce 的 WBI 适配器接收来自协作的业务对象, 使用数据处理程序将它们转换为 XML 格式消息, 然后将这些消息交付给 WebSphere MQ 队列。关于集成 Websphere Commerce 和 IBM CrossWorlds ICS 的详细信息, 请在以下地址参阅关于 CrossWorlds 适配器的文档:

<http://www-3.ibm.com/software/websphere/crossworlds/library/doc/v411/welcome.html>。

配置任务

配置消息传递服务

在系统管理期间，站点管理员可执行以下任务：

- 添加传递工具（称为“传送”）用于交付消息
- 配置传送
- 查看消息传送指定
- 配置消息类型（添加、更改或删除消息描述、严重性、传送或设备格式）
- 将传送方法添加到商店
- 激活或停用传送方法

有关配置消息传递系统的更多信息，请参阅 WebSphere Commerce 产品联机帮助。

配置调度服务

从管理控制台访问调度程序。关于使用调度程序的更多信息，请参阅 WebSphere Commerce 产品联机帮助。关于维护调度程序的信息，请参阅第 129 页的『维护调度程序』。

配置日志记录

现在所有日志记录和跟踪配置都在 WebSphere Application Server 管理控制台中完成。以下部分仅涵盖独立工具：

- 修改 ECTrace 启动选项
- 修改 ECTrace 启动选项

关于 WebSphere Application Server 日志记录和跟踪的更多信息，请参阅 WebSphere Application Server 信息中心。

启用跟踪组件

要为您的 WebSphere Commerce 系统启用跟踪，您必须使用 WebSphere Application Server 管理员控制台。

在服务器启动时启用跟踪：

1. 打开 WebSphere Application Server 管理员控制台。关于如何执行此操作的信息，请参阅 WebSphere Application Server 信息中心。
2. 单击控制台导航树中的**疑难解答 > 日志记录和跟踪**，然后单击**服务器 > 诊断跟踪**。
3. 单击**配置**。
4. 选择**启用跟踪**复选框启用跟踪，清除该复选框可禁用跟踪。
5. 通过输入适当的跟踪字符串，将跟踪规范设置为期望状态。
6. 选择将跟踪输出定向到文件还是内存中的循环缓冲区。
7. 选择生成跟踪的期望格式。
8. 保存已更改的配置。
9. 启动服务器。

注：关于这些步骤中任何一步的更多信息，请参阅 WebSphere Application Server 信息中心。

在正在运行的服务器上启用跟踪： 使用以下过程，您可修改跟踪服务状态，该状态确定正在主动跟踪运行中的服务器的哪些组件。

1. 打开 WebSphere Application Server 管理员控制台。关于如何执行此操作的信息，请参阅
2. 单击控制台导航树中的**疑难解答 > 日志记录和跟踪**，然后单击**服务器 > 诊断跟踪**。
3. 选择**运行时**选项卡。
4. 通过将跟踪规范设为期望状态，可更改现有跟踪状态。
5. （可选）如果希望更改现有跟踪输出，可配置跟踪输出。
6. 单击“应用”。

注：关于这些步骤中任何一步的更多信息，请参阅 WebSphere Application Server 信息中心。

WebSphere Commerce 跟踪组件： 以下是 WebSphere Commerce V5.5 中的跟踪组件：

表 6.

跟踪组件	WebSphere JRAS 扩展跟踪记录程序
SERVER	com.ibm.websphere.commerce.WC_SERVER
CATALOG	com.ibm.websphere.commerce.WC_CATALOG
DATASOURCE	com.ibm.websphere.commerce.WC_DATASOURCE
ORDER	com.ibm.websphere.commerce.WC_ORDER
USER	com.ibm.websphere.commerce.WC_USER
COMMAND	com.ibm.websphere.commerce.WC_COMMAND
CF	com.ibm.websphere.commerce.WC_CF
NEGOTIATION	com.ibm.websphere.commerce.WC_NEGOTIATION
RAS	com.ibm.websphere.commerce.WC_RAS
DB	com.ibm.websphere.commerce.WC_DB
METAPHOR	com.ibm.websphere.commerce.WC_METAPHOR
SCHEDULER	com.ibm.websphere.commerce.WC_SCHEDULER
DEVTOOLS	com.ibm.websphere.commerce.WC_DEVTOOLS
TOOLSFRAMEWORK	com.ibm.websphere.commerce.WC_TOOLSFRAMEWORK
RULESYSTEM	com.ibm.websphere.commerce.WC_RULESYSTEM
MERCHANDISING	com.ibm.websphere.commerce.WC_MERCHANDISING
MARKETING	com.ibm.websphere.commerce.WC_MARKETING
REPORTING	com.ibm.websphere.commerce.WC_REPORTING
TRANSPORT_ADAPTER	com.ibm.websphere.commerce.WC_TRANSPORT_ADAPTER
PERFMONITOR	com.ibm.websphere.commerce.WC_PERFMONITOR
MESSAGING	com.ibm.websphere.commerce.WC_MESSAGING
STOREOPERATIONS	com.ibm.websphere.commerce.WC_STOREOPERATIONS
CACHE	com.ibm.websphere.commerce.WC_CACHE

表 6. (续)

EVENT	com.ibm.websphere.commerce.WC_EVENT
EJB	com.ibm.websphere.commerce.WC_EJB
CURRENCY	com.ibm.websphere.commerce.WC_CURRENCY
CATALOGTOOL	com.ibm.websphere.commerce.WC_CATALOGTOOL
CONTRACT	com.ibm.websphere.commerce.WC_CONTRACT
UBF	com.ibm.websphere.commerce.WC_UBF
BI	com.ibm.websphere.commerce.WC_BI
INVENTORY	com.ibm.websphere.commerce.WC_INVENTORY
UTF	com.ibm.websphere.commerce.WC_UTF
RFQ	com.ibm.websphere.commerce.WC_RFQ
EXCHANGE	com.ibm.websphere.commerce.WC_EXCHANGE
ACCESSCONTROL	com.ibm.websphere.commerce.WC_ACCESSCONTROL
AC_UNITTEST	com.ibm.websphere.commerce.WC_AC_UNITTEST
APPROVAL	com.ibm.websphere.commerce.WC_APPROVAL
COLLABORATION	com.ibm.websphere.commerce.WC_COLLABORATION
THREAD	com.ibm.websphere.commerce.WC_THREAD
SENSITIVE_INFO	com.ibm.websphere.commerce.WC_SENSITIVE_INFO

JCA 连接器跟踪组件: 以下是 WebSphere Commerce V5.5 中的 JCA 连接器跟踪组件:

表 7.

跟踪组件	WebSphere JRAS 扩展跟踪记录程序
JCA 电子邮件连接器	com.ibm.websphere.commerce.jcaemail
JCA 文件连接器	com.ibm.websphere.commerce.jcafile
JCA JMS 连接器	com.ibm.websphere.commerce.jcajms
JCA 样本连接器	com.ibm.websphere.commerce.jcasample

Commerce Payments 跟踪组件: 以下是 WebSphere Commerce V5.5 中 Commerce Payments 跟踪组件:

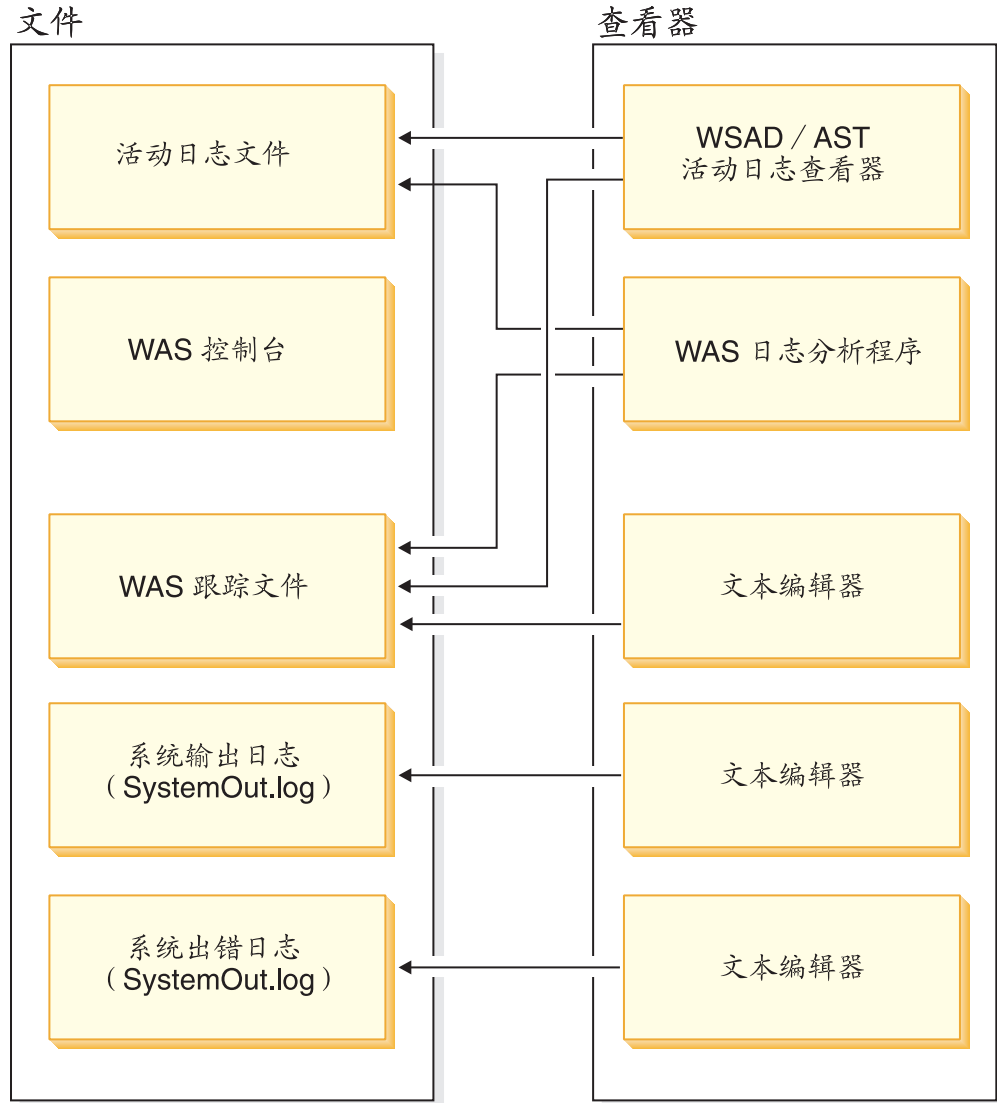
表 8.

跟踪组件	WebSphere JRAS 扩展跟踪记录程序
* (给您所有组件)	com.ibm.websphere.commerce.payments.*
MPF (Payments 框架)	com.ibm.websphere.commerce.payments.MPF
MPFUI (Payments UI)	com.ibm.websphere.commerce.payments.MPFUI
<Cassette_name> (这将启用对任何 IBM 或第三方卡匣的跟踪。WebSphere Commerce 附带的卡匣有: VisaNet、Paymentech、BankServACH、OfflineCard 和 CustomOffline。)	com.ibm.websphere.commerce.payments.<Cassette_name>

注：关于 WebSphere Commerce Payments 的更多信息，请参阅 WebSphere Commerce Payments 库。

查看日志文件

以下图表描述了查看日志文件的不同方法：



使用日志分析程序




作为 WebSphere Application Server 一部分的日志分析程序，获取一个或多个服务或活动日志，合并所有数据并显示条目。根据其故障现象数据库，该工具分析并解释日志条目中的事件或错误条件来帮助您诊断问题。日志分析程序具有特殊功能，允许它从 IBM Web 站点下载最新的故障现象数据库。要下载故障现象数据库的最新更新，请使用 WebSphere Application Server 的“文件 -> 更新数据库 -> WebSphere Application Server 故障现象数据库”选项，或日志分析程序界面中的 WebSphere Application Server 网络部署的“WebSphere Application Server 网络部署故障现象数据库”选项。

服务和活动日志： WebSphere Application Server 服务器从各种 WebSphere Application Server 组件的活动中创建服务或活动日志文件。服务或活动日志文件 activity.log 是

二进制文件，位于：installation_directory/logs/activity log。可使用日志分析程序查看服务或活动日志文件并可将服务或活动日志文件合并到一个日志文件中。不能使用文本编辑器查看服务或活动日志。日志分析程序工具被设计用来查看此文件。关于 activity.log 的更多信息，请参阅 *WebSphere Application Server System Administration Guide*。

使用日志分析程序查看服务和活动日志： 要使用日志分析程序查看服务或活动日志，请执行以下操作：

1. 切换至 `WASinstall_dir/bin` 目录。
2. 运行 `waslogbr` 脚本文件：

-  `v waslogbr.bat`。
-   `v waslogbr`。

这将启动日志分析程序界面。


3. 选择文件 > 打开。
4. 浏览至包含服务或活动日志文件的目录。
5. 选择服务或活动日志文件并单击打开。
6. 要分析记录，用鼠标右键单击左边树中的一条记录，从右键单击弹出菜单中选择 **UnitOfWorkView** 并选择分析。

现在任何旁边有一个绿色复选标记的记录都与故障现象数据库中的某记录相匹配。当选择带有复选标记的记录时，您将在右下角的窗格中看到该问题的解释。关于日志分析程序用户界面的更多信息，请参阅 *WebSphere Application Server* 信息中心。

使用文本文件

要查看日志文件，您可以使用任何文件编辑器。可以在安装了 *WebSphere Commerce Server* 的日志目录中找到日志文件。通常它在 `../appserver/logs/` 目录中（但可进行更改）。下面是可用来查找日志文件的路径示例。

    
`WAS_installdir/logs/WC_instanceName/trace.log`

注：  使用文本编辑器而非 `vi` 查看生成的 *WebSphere Commerce* 日志文件。这些行对于 `vi` 来说太长，因此不能正确查看。

配置 LDAP

一旦安装了 *WebSphere Commerce*，您可以选择将 LDAP 服务器用作成员资源库，或者开始时将此数据库用作成员资源库，以后才转换成使用 LDAP 服务器。关于配置 LDAP 的更多信息，请参阅《*WebSphere Commerce 附加软件指南*》中的“为 LDAP 配置 *WebSphere Commerce*”。

配置规则服务

要配置规则服务，请登录到管理控制台，并从规则服务菜单单击**管理**。站点管理员可使用管理控制台执行以下规则服务任务：

- 添加规则服务

- 编辑规则服务
- 检查规则服务状态
- 刷新规则服务
- 删除规则服务
- 启动规则服务
- 停止规则服务

关于用于配置规则服务的任务的更多信息，请参阅 WebSphere Commerce 联机帮助。

配置登台服务器

使用登台服务器实用程序

登台实用程序存在于登台服务器上，并使用标准 DB2/JDBC 通信连接至生产数据库和登台数据库。管理员必须确保 DB2 客户机 / 服务器环境正确设置。DB2 客户机通过 TCP/IP 端口与 DB2 服务器对话。端口号在 DB2 实例变量 SVCENAME 中设置。您必须知道此端口号并确保代理不会阻塞此端口。缺省情况下，DB2 使用 50000。关于设置 JDBC 和 DB2 客户机 / 服务器环境的更多信息，请参阅 *DB2 Administration Guide*。

一旦设置了环境且准备将数据从登台服务器复制到生产服务器时，就使用“登台传播”实用程序推送数据。可复制的表在登台表中列出。在为定制表配置登台服务器后，您所创建的用来存储定制数据的所有新表也可以复制到登台服务器中。

STAGLOG 表用作内部日志。无论何时更改登台服务器的表中的某个记录，触发器就在 STAGLOG 表中记录这个更改。对于每个修改的记录，触发器会记录修改的类型（插入、删除或更新）、记录所在表的名称，以及记录的主键或唯一索引。在完成了对登台服务器上数据库记录的更改和测试后，请使用“登台传播”实用程序将这些更改传播回生产服务器。

登台会话期间，由登台服务器所包含的数据库表不得在生产数据库上更新。当您使用“登台复制”实用程序将生产数据库复制到登台数据库时，登台会话开始。当使用“登台复制”实用程序开始另一登台会话时，此登台会话结束。在您的数据从生产数据库复制到登台数据库后，登台数据库和生产数据库根据登台服务器所包含的表实现同步。一旦表实现同步后，便不允许在生产数据库上对表进行更改。只能更新登台数据库，然后使用“登台传播”实用程序将更改传播到生产数据库。如果这两个数据库都进行了更新，则传播可能由于潜在的键冲突或引用完整性违例而失败。如果必须在登台会话期间更新生产数据库，应使用“登台复制”实用程序来使这些数据库同步，并开始新的登台会话。

要保证登台会话期间决不会在生产数据库上更新表，这些表必须只处于站点管理员的控制下。在某些情况下，在使用“登台复制”复制了数据后，生产数据库中的登台表将由个别客户或商家更新。例如，您无法禁止商家在复制到生产数据库后修改生产数据库中的 OFFER 表。在这种情况下不能使用登台服务器。不过，RFQ 对象除外。当在生产数据库上创建 RFQ 时，将在生产数据库中的贸易表中插入行。如果在登台数据库上创建合同时，也将在登台数据库中的贸易表中插入行。在这种情况下，您在登台数据库和生产数据库上更新相同的表。

注意，在使用 RFQ 对象时，对登台服务器有一些限制。您还需要运行“登台检查”实用程序查找潜在的唯一索引键冲突并改正它们，之后才能运行“登台传播”实用程序将更改传播到生产数据库。

在一般的商家到消费者站点中，可以将表分为两组：配置数据和操作数据。配置表包含诸如商店、产品目录、产品目录条目、语言、税款和折扣的数据。这些表由站点管理员控制；个别客户不能修改它们。操作表包含诸如客户信息、地址、订单和 SET 相关数据之类的数据。客户可以修改操作表。登台服务器只包含配置表。关于登台服务器所包含的表的列表，请参阅 WebSphere Commerce 登台表。

确保登台服务器所涉及的表不包含对操作表的任何外键引用也是很重要的。否则，传播可能由于从生产数据库中删除潜在的主键而导致失败。在使用登台服务器之前，应确保只有组织，而不是个别用户（例如，产品目录管理员）拥有操作数据。

在使用登台服务器之前应当了解以下内容：

- 由登台记录所引用的任何新的图像文件、HTML 文件或 JSP 文件都必须手工从登台服务器复制到生产服务器。
- 登台服务器不能复制和传播数据库模式更改、图像文件、HTML 文件或 JSP 文件。例如，如果您在登台数据库中创建了新的索引或表，必须在生产数据库中手工创建索引或表。
- “登台传播”实用程序不能传播由装入程序软件包（装入方式）或 DB2 装入实用程序装入的记录，这是由于它们两个都绕过了登台触发器。如果您使用了其中一种实用程序，则应使用“登台复制”以使数据库表重新同步，并开始新的登台会话。切勿在登台会话期间在登台数据库或生产数据库上使用装入程序软件包（装入方式）或 DB2 装入。
- 在使用“登台复制”实用程序后，必须停止并重新启动登台服务器。
- 登台服务器不支持 DB2 Text Extender。
- 不要在登台服务器上运行“数据库清理”实用程序，清理 STAGLOG 表除外。
- 有一些登台服务器限制。在使用登台服务器之前，请确保您了解登台服务器限制。关于更多信息，请参阅第 32 页的『登台服务器限制』。

“登台复制”实用程序：“登台复制”实用程序 `stagingCopy.sh for iSeries` 将数据从生产数据库复制到登台数据库。您可以复制与站点相关的表、与商家相关的表或个别表中的数据。也可以在命令语法中使用 `cleanup_stage_db` 参数在“登台复制”实用程序之前清除登台数据库。如果指定 `yes`，则“登台复制”实用程序将在复制数据之前清除所有登台表。这可能对其它带有自动删除关联的表造成影响。如果指定 `no`，则“登台复制”实用程序将不会从登台表上执行任何删除。如果复制数据发生冲突或在主键或唯一索引上存在重复键，则复制可能会失败。要将“登台复制”实用程序用于仅清除登台数据库，请指定 `only`。

“登台复制”实用程序和“登台传播”实用程序将数据库数据分为两个作用域级别：与站点相关和与商家相关。站点作用域包括对系统中所有商家公共的数据。例如，系统使用的语言和国家或地区代码。商家作用域包含与个别商家相关的数据。例如，商店信息是为个别商家定制的，而商店表中的行可能是特定于每个商家的。某些数据库表同时包含站点和商家信息。如果在“登台复制”实用程序中将 `scope` 参数指定为 `_all_`，则将复制站点数据，随后复制所有的商家数据。如果将作用域指定为 `_site_`，则仅复制站点数据。如果将作用域指定为 `_merchant_`，则仅复制商家数据。请注意您不能复制个别商家的数据，只能复制所有商家的数据。如果不将作用域设置为 `_all_`，则在复制商家数据之前复制站点数据，这是因为所有商家都使用站点数据。否则，您的复制将会由于外键和主键之间的不匹配而失败。当使用 `cleanup_stage_db` 清理站点数

据时，请注意可能会因为自动删除关联而删除商家数据。如果您未将作用域设置为 `_all_`，则应在清除商家数据之后清除站点数据，然后在复制站点数据后复制商家数据。

“登台复制”实用程序的另一个选项是 `script_file` 参数。通过指定脚本文件的名称，“登台复制”实用程序将生成 SQL 脚本文件，它根据指定作用域使用导出和导入将生产数据库复制到登台数据库。如果您使用缺省值或将 `cleanup_stage_db` 指定为 `yes`，则还生成删除语句来清除登台数据库。脚本文件位于启动“登台复制”实用程序的目录中。脚本文件加快了使用导出和导入复制数据库的过程。您也可以通过更改生成的脚本文件来修改“登台复制”实用程序的行为。例如，您可以更改脚本文件以使用 DB2 装入实用程序来代替导入实用程序，它将进一步加速复制过程。注意，生成的脚本将所有表导出至运行此实用程序的目录。确保您有足够的磁盘空间。

了解事务作用域非常重要。在清理登台数据库时，“登台复制”实用程序在清理完每个表之后提交事务。在复制数据时，“登台复制”实用程序在复制每个表并同步 KEYS 表后提交事务。对于生成的脚本，事务作用域会由于 DB2 导入实用程序而稍有不同。DB2 导入实用程序在完成导入后将自动提交事务。事务在同步 KEYS 表之前提交。因此，同步 KEYS 表在独立的事务中完成。

您可以使用 `dbtable` 参数指定要清除或复制的表。请注意当您指定要清除或复制的表时，此表可能不是孤立的。某些表通过引用约束而彼此相关。如果清除指定的表，还将使用自动删除关联清除其子表。如果复制指定的表，您应首先复制其父表。否则，您的清除或复制将失败。

“登台复制”实用程序是可配置且可扩展的。要处理定制表，则这些表必须满足一些条件，且必须在登台配置表中设置这些条件。在可以使用“登台复制”实用程序之前，必须遵循配置数据库的步骤。如果命令成功，则“登台复制”实用程序从 STAGLOG 表中删除所有记录。

注意：如果 RFQ 在您的系统上，则无法使用“登台复制”命令。关于更多信息，请参阅第 32 页的『登台服务器限制』。

“登台检查”实用程序：当配置和操作数据共享同一个表时，登台数据库和生产数据库之间有可能出现唯一索引键冲突。在将更改传播到生产数据库之前，请使用“登台检查”命令（或 `stagingCheck.sh` for iSeries）来确定是否有任何潜在的唯一索引冲突并在传播之前纠正这些冲突。

当您在生产数据库上使用 RFQ 和在登台数据库上创建合同时，您在两个数据库上都更新了相同的表。例如，站点管理员在登台数据库上创建合同，这将在登台数据库的 TRADING 表（和其它表）中插入一行。同时，用户在生产数据库上创建 RFQ，这将在生产数据库的 TRADING 表（和其它表）中插入一行。这两行可能在 TRADING 表中具有相同的唯一索引值。将合同从登台数据库传播到生产数据库时，将生成唯一索引键冲突，传播将失败。传播之前，请使用“登台检查”实用程序查找冲突的唯一索引键并改正它们。此后，您可以传播更改。

当使用“登台检查”实用程序时，请将 `-scope` 参数指定为 `_unique_index_` 以检查登台数据库中的增减变化引起的潜在键冲突。对于所有的插入和更新操作，它将为在 STGUINDTAB 表中指定的所有表检查潜在的索引键冲突。对于每个表，它将检查所有的唯一索引，并检查在生产数据库和登台数据库之间是否有潜在的键冲突。如果有潜在的键冲突，它将报告表名、唯一索引和冲突键值。

“登台检查”命令并不更改数据库；它报告必须解决的潜在键冲突。使用此命令时，指定 `-sourcedb` 参数为登台数据库。如果您指定生产数据库为您的源数据库，“登台检查”命令将不能正常起作用。

“登台检查”实用程序是可配置和可扩展的。可以将更多的表或定制表添加到 STGUINDTAB 表，运行登台检查命令以验证是否存在任何潜在的键冲突。

注意：请始终确保配置和操作数据并未共享同一表。

“登台传播”实用程序：在更改和测试了登台服务器上的数据库记录后，请使用“登台检查”实用程序检查潜在的唯一索引键冲突并纠正它们。现在您就准备好向生产数据库传播更改了。

“登台传播”实用程序（或 `stagingProp.sh` for iSeries）将更改从登台数据库移动到生产数据库。“登台传播”实用程序使用 STAGLOG 表识别登台数据库中更改的记录，然后在生产数据库中更新这些记录。处理过的记录在 STAGLOG 表的 STGPROCESSED 列中以 1 表示。

您可以指定 `scope` 参数以选择传播的数据类型。将其设置为 `_site_`，则所有更改了的站点数据将从登台数据库传播到生产数据库。若设置为 `_merchant_`，则将传播所有商家的更改了的数据。不能传播个别商家数据。若设置为 `_all_`，则站点和商家的数据都传播。

通过使用 `dbtable` 参数，可以传播特定的表。请确保在指定一个表之前，其父表已经传播。

“登台传播”实用程序的事务作用域与“登台复制”实用程序不同。“登台传播”实用程序命令的每次运行都计为一次事务。例如，如果指定 `scope` 为 `_site_`，则“登台传播”实用程序将对所有修改过的站点数据开始新的事务，并在成功传播后提交此事务。如果传播失败，则传播回滚，您的生产数据库的状态与从前一样。

“登台传播”实用程序是可配置且可扩展的。在传播定制表之前，它们必须符合某些条件。关于详细信息，请参阅“为定制表配置登台服务器”。在可以使用“登台传播”实用程序之前，必须遵循配置数据库的步骤。

配置数据库





在使用“登台复制”实用程序、“登台传播”实用程序或“数据库清理”实用程序之前，您也许希望执行以下操作：

注：

1. 以下步骤仅为建议，且仅在遇到“登台复制”实用程序问题时使用。

2. 以下内容不适用于 iSeries，除非由  表示

1. 设置 PATH 环境变量。

2.      如果您正在使用 DB2 数据库，

则通过发出以下命令配置登台数据库和生产数据库：





```
db2 update db config for db_name using LOGPRIMARY 80
db2 update db config for db_name using LOGBUFSZ 512
db2 update db config for db_name using DBHEAP 2048
```

```
db2 update db config for db_name using APPLHEAPSZ 2048
db2 update db config for db_name using PCKCACHESZ 8200
```

其中 *db_name* 是数据库的名称。

注:

- a. 缺省 STMTHEAP 大小为 60000。
- b. 缺省 LOCKLIST 为 2400。
- c. 缺省 STAT_HEAP_SZ 为 2048。
- d. 缺省 APP_CTL_HEAP_SZ 为 4096。

3.     增加缓冲池的大小以提高性能。根据 DB2 数据库大小和可用内存来确定最佳缓冲池大小。运行以下命令，更改缺省缓冲池大小:

```
db2 connect to db_name
db2 alter bufferpool IBMDEFAULTBP size n
db2 terminate
```

其中 *n* 是最佳的缓冲池大小。

4.  以具有 secofr 权限和非 65535 的 cssid 的用户概要文件登录。

为定制表配置登台服务器

要对定制的数据库表使用登台服务器，请执行以下配置:

1. 标识定制表作用域（站点数据、商家数据或站点和商家数据）。
2. 根据表的作用域和索引类型，使用相应的触发器示例来创建数据库表的触发器。
3. 将定制的表插入到 STGSITETAB、STGMERTAB 和 STGMRSTTAB 表中。
 - 对于站点表，仅插入到 STGSITETAB 中。
 - 对于商家表，仅插入到 STGMERTAB 中。
 - 对于同时包含站点和商家数据的表，插入到 STGSITETAB、STGMERTAB 和 STGMRSTTAB 中。

注意: 您必须确保所有的父表已经被正确地插入以及父表的 TABNBR 列小于子表的相应列。如果定制表是 WebSphere Commerce 表的父表，则还需要确保表的 TABNBR 列小于子表的相应列。

配置连接

关于配置各种适配器的信息，请参阅《WebSphere Commerce 附加软件指南》和 WebSphere Commerce 产品联机帮助。

配置 WebSphere Commerce Payments

可配置 Payments 之前，请执行以下操作:

- 确保 WebSphere Commerce Payments 是您的 WebSphere Commerce 安装的一部分。
- 创建 WebSphere Commerce Payments 实例。
- 使用配置管理器向 WebSphere Commerce Payments 实例添加卡匣。
- 启动 WebSphere Commerce Payments 实例和 WebSphere Commerce 实例。
- 创建商家及该商家的商家管理员。

要配置卡匣并使用 Payments，您必须以商家管理员登录 WebSphere Commerce Payments。WebSphere Commerce Payments 随 CustomOffline 卡匣和 OfflineCard 卡匣安装。这些卡匣支持的最小框架为 WebSphere Commerce Payments V5.5。关于更多信息，请参阅《WebSphere Commerce 安装指南》。

以下是 WebSphere Commerce 和 Payments 的缺省安装目录：

表 9. 缺省安装目录

缺省安装目录	描述
<i>WC_installdir</i>	Windows: <i>drive:\WebSphere\CommerceServer55</i> AIX: <i>/usr/WebSphere/CommerceServer55</i> Solaris、Linux: <i>/opt/WebSphere/CommerceServer55</i> 400: <i>/QIBM/ProdData/CommerceServer55</i>
<i>Payments_installdir</i>	Windows: <i>drive:\WebSphere\CommerceServer55\payments</i> AIX: <i>/usr/lpp/WebSphere/CommerceServer55/payments</i> Solaris、Linux: <i>/opt/WebSphere/CommerceServer55/payments</i> 400: <i>/QIBM/ProdData/CommercePayments/55</i>

关于配置 WebSphere Commerce Payments 的详细信息，请参阅第 149 页的附录 C，『WebSphere Commerce Payments 教程』。

第 4 章 主管商店管理

站点管理员在主管商店方案中必须执行几项任务。本章概括了站点管理员执行的任务，分类如下：

- 管理转销商
- 管理经销商
- 站点管理

管理转销商

在主管商店方案中，站点管理员负责设置转销商组织并维持与转销商的关系。此协调涉及以下任务：

- 发送新转销商组织注册的电子邮件核准。

一旦转销商注册其组织，向该转销商发送电子邮件通知其已核准其组织请求，是站点管理员的责任。在组织管理控制台中执行实际组织注册的核准。关于组织管理控制台以及可在其中执行的任务的更多信息，请参阅 *WebSphere Commerce 产品联机帮助*。

一旦转销商组织获得核准，在需求链业务模型中的转销商可以使用 *WebSphere 贸易加速器* 修改他们的商店。每个商店以关闭的状态创建，然而转销商可以使用 *WebSphere 贸易加速器* 来更新和更改此商店，并最终打开它。在主管业务模型中，每个商店以暂挂状态创建，需要由站点管理员和渠道经理恢复。在此种情况下，商家在它被恢复以前不可对此商店进行任何更改。关于该业务模型的更多信息，请参阅 *《WebSphere Commerce 基础》* 和 *《WebSphere Commerce 商店开发指南》*。有关 *WebSphere 贸易加速器* 的更多信息，请参阅 *WebSphere Commerce 产品联机帮助*。
- 发送商家主管商店核准。（仅限于使用主管业务模型的商店）

一旦设置了主管商店，站点管理员向商家发送附带主管商店请求核准的电子邮件。此电子邮件还必须包含到新主管商店的链接，以及关于如何管理它的表示信息。

注： 有关管理转销商组织的更多信息，请参阅 *WebSphere Commerce 产品联机帮助* 中的“管理转销商组织”。

管理经销商

在主管商店方案中，站点管理员负责设置经销商并创建经销商与转销商的关系。此管理涉及以下任务：

- 创建新的经销商。

通过创建经销商服务协议，站点管理员可创建新的经销商。直接联系 IBM 可获得要添加到合同中的信息。有关经销商服务协议的更多信息，请参阅 *WebSphere Commerce 产品联机帮助*。

注： 此任务也可由渠道管理器执行。

- 更改经销商设置。

有时站点管理员可能会接收到更改某些经销商设置（例如连接到经销商的超时设置）的请求。通过 SQL 语句访问数据库可更改经销商设置。有关更多信息，请参阅《*WebSphere Commerce 商店开发指南*》。

- 配置并安装 CrossWorlds 适配器

CrossWorlds 适配器是一个连接器，用来建立 WebSphere Commerce 与主管商店的外部及后端系统之间的交互。站点管理员负责配置并安装适配器以确保转销商和经销商站点之间的连接。还可创建 WebSphere Commerce 与使用 WebSphere MQ 的 SAP R/3 后端系统之间的交互，且必须由站点管理员进行设置。关于 IBM CrossWorlds 和 WebSphere MQ 的更多信息，请参阅《*WebSphere Commerce 附加软件指南*》。

站点管理

特定于主管商店的站点管理任务包括以下内容：

- 操作登台服务器。关于更多信息，请参阅第 32 页的『登台服务器』。
- 操作 DBClean。关于更多信息，请参阅第 89 页的『数据库清理实用程序』。
- 操作 WebSphere Commerce Analyzer。有关更多信息，请参阅 WebSphere Commerce 安装 CD。
- 启动和停止 WebSphere Commerce 服务器。关于更多信息，请参阅第 11 页的『启动和停止 WebSphere Commerce』。

第 5 章 动态高速缓存

本节描述了 WebSphere Commerce servlet 或 JavaServer Pages (JSP) 结果的动态高速缓存。

WebSphere Commerce servlet 或 JSP 结果高速缓存

当客户单击某链接查看产品或类别页面时，大部分时间都用于分析 HTTP 请求、访问数据库和动态创建页面。较大的站点流量和数据库中许多产品和类别条目会进一步增加装入 servlet 或 JavaServer Pages (JSP 文件) 所用时间。

服务器上的大多数 HTTP 请求都是针对产品目录信息的。这些 WebSphere Commerce 命令 (CategoryDisplay、ProductDisplay、TopCategoriesDisplay 和 StoreCatalogDisplay) 从数据库中检索信息，并将结果显示为 JSP 页面。如果自上一次查看以来，产品目录信息没有更改，则下一次客户请求时不需要重新执行 servlet 或 JSP 文件。提供已存储在高速缓存中的等价静态 servlet 或 JSP 文件将更快。通过在位于 Web 模块的 WEB-INF 目录中的 cachespec.xml 文件中定义高速缓存条目元素，可构造 servlet 或 JSP 的高速缓存。

如果与访问的页面相应的高速缓存条目不在内存中，则它会动态生成。然后此页面将存储到内存中，在其所依据的数据被修改之前此页面不必重新生成。

注：WebSphere Commerce 动态高速缓存和 URL 重写无法互操作。随着 URL 重写打开，您需要禁用 WebSphere Commerce 动态高速缓存，且您不能高速缓存您的 servlet 或 JSP 文件。关于 URL 重写的更多信息，请参阅《WebSphere Commerce 安全性指南》中的有关会话管理的章节。

WebSphere Application Server 动态高速缓存

WebSphere Commerce 利用 WebSphere Application Server 动态高速缓存服务来高速缓存 servlet 或 JSP 文件以及从 WebSphere Application Server CacheableCommand 界面扩展的命令。在创建 WebSphere Commerce 实例期间，缺省情况下启用动态高速缓存服务、servlet 高速缓存和磁盘卸载。

关于 WebSphere Application Server 动态高速缓存的更多信息，请参阅 WebSphere Application Server 信息中心 (<http://www.ibm.com/software/webservers/appserv/infocenter.html>) 中的主题“Improving performance through the dynamic cache service”。

动态高速缓存

高速缓存 servlet 或 JSP 文件结果提高了应用程序性能。WebSphere Application Server 将几项高速缓存活动 (包括 servlet、Web 服务以及 WebSphere 命令) 合并到一个称为动态高速缓存的服务中。这些高速缓存活动协同工作来提高应用程序性能并共享许多配置参数，这些配置参数在应用程序服务器的动态高速缓存服务中设置。

您可以通过从内存中的高速缓存为请求提供服务，来使用动态高速缓存提高 servlet 和 JSP 文件的性能。高速缓存条目包含 servlet 输出、servlet 执行结果以及元数据。

关于更多信息，请参阅 WebSphere Application Server 信息中心

(<http://www.ibm.com/software/webservers/appserv/infocenter.html>) 中的主题“Improving performance through the dynamic cache service”。

启用动态高速缓存服务和 **servlet** 高速缓存

要启用高速缓存，您应启用动态高速缓存服务并配置 servlet 高速缓存。关于执行这些步骤的信息，请参阅 WebSphere Application Server 信息中心

(<http://www.ibm.com/software/webservers/appserv/infocenter.html>) 中的主题“Enabling globally the dynamic cache service”和“Configuring servlet caching”。

启用动态高速缓存监视器

WebSphere Application Server 提供了动态高速缓存监视器应用程序以显示高速缓存统计信息、Edge Side 和磁盘统计信息、高速缓存条目、相关性标识以及高速缓存策略信息。为了检查 WebSphere Application Server 动态高速缓存的内容和行为，您应该安装 WebSphere Application Server 动态高速缓存监视器。要使用动态高速缓存监视器，请在使用动态高速缓存的每台应用程序服务器上安装 CacheMonitor.ear 文件，该文件位于 WAS_installdir 目录的 installableApps 子目录中。

对于 WebSphere Commerce，为安全起见，建议您对动态高速缓存监视器使用虚拟主机 VH_instance_name_admin（例如 VH_demo_admin），而不是 VH_instance_name。您可以通过 WebSphere Application Server 管理控制台添加或更改虚拟主机名称。

您可使用 Web 浏览器在以下 Web 地址访问 Web 应用程序：

`http://host_name:port/cachemonitor`

但是，为了更加安全地进行访问，建议您访问管理主机：

`https://admin_host_name:port/cachemonitor`

例如，如果为安装高速缓存监视器使用虚拟主机 VH_instance_name_admin，则如下访问该高速缓存监视器：

`https://admin_host_name:8002/cachemonitor`

注：

1. 您需要为每个 WebSphere Commerce 实例启动一个动态高速缓存监视器。
2. 无论何时如果您希望使用高速缓存监视器，且您已启用 WebSphere Application Server EJB 安全性，您将需要按照《WebSphere Commerce 安全性指南》中的“动态高速缓存监视器的安全性配置”部分执行一些附加的设置。

关于安装动态高速缓存监视器的更多信息，请参阅 WebSphere Application Server 信息中心

(<http://www.ibm.com/software/webservers/appserv/infocenter.html>) 中的主题“Displaying cache information”。


WebSphere 动态高速缓存监视器相关性标识的树形视图

WebSphere Commerce 提供了树形视图工具（高速缓存监视器的扩展），它可提供内存中相关性标识的树形视图。树形视图工具提供了相关性标识的树形视图。您可展开或折叠任何树节点以获得分层视图，并可使该节点下的高速缓存条目组无效。不会涉及到原始视图，而是将出现一个新的链接“相关性标识树形视图”。

+ 关于安装和使用树形视图工具的信息，请参阅以下目录中自述文件的“WebSphere 动态高速缓存监视器相关性标识的树形视图”：
+

`WC_installdir/features/dynacache/CacheMonitor/`

 `WC_installdir\features\dynacache\CacheMonitor\`

配置可高速缓存对象

可高速缓存对象在 `cachespec.xml` 文件中定义，可在 Web 应用程序压缩文档（WAR）WEB-INF 或企业 bean WEB-INF 目录中找到该文件。您可在应用程序服务器属性目录中放置全局 `cachespec.xml` 文件，但建议的方法是将高速缓存配置文件与部署模块放在一起。`cachespec.xml` 文件的根元素是 `<cache>`，它包含 `<cache-entry>` 元素。

要指定 servlet 或 JSP 结果高速缓存的高速缓存条目，请将以下部分添加到 `cachespec.xml` 文件：

+ `<cache-entry>`
+ `<class>servlet</class>`
+ `<name>name</name>`
+ `...`
+ `</cache-entry>`

其中 `name` 是相对 Web 路径或 servlet 或 JSP 的 servlet 映射

要指定命令高速缓存的高速缓存条目，请将以下部分添加到 `cachespec.xml` 文件：

+ `<cache-entry>`
+ `<class>command</class>`
+ `<name>name</name>`
+ `...`
+ `</cache-entry>`

其中 `name` 是到命令类的完整路径，例如

`com.ibm.commerce.dynacache.commands.MemberGroupsCacheCmdImpl`。

在 `<cache-entry>...</cache-entry>` 元素中，您可开发高速缓存标识、相关性标识以及失效规则。要高速缓存对象，WebSphere Application Server 必须能够为该对象的不同调用生成唯一标识。`<cache-id>` 元素执行此任务。每个高速缓存条目可具有多个顺序执行的高速缓存标识规则，直到某个规则返回非空高速缓存标识或没有更多要执行的规则为止。如果没有一个高速缓存标识生成规则产生有效的高速缓存标识，则不高速缓存该对象。

+ 注：对于此发行版，当指定高速缓存条目进行命令高速缓存时，只支持使用
+ `<invalidate>...</invalidate>` 的命令 `invalidation`。（不能对命令高速缓存使用
+ `<cache-id>...</cache-id>` 或 `<dependency-id>...</dependency-id>` 元素。）

```

+      <cache-entry>
+          <class>command</class>
+          <name>name</name>
+          <invalidate>
+              ...
+          </invalidate>
+      </cache-entry>

```

动态高速缓存响应 `cachespec.xml` 文件中的更改。当检测到新版本的 `cachespec.xml` 时，将替换旧的策略。将不会自动使通过旧的策略文件高速缓存的对象在高速缓存中无效。将以新的策略重新使用它们，或通过其替换算法从高速缓存中除去它们。关于 `cachespec.xml` 文件的更多信息，请参阅 WebSphere Application Server 信息中心 (<http://www.ibm.com/software/webserver/appserv/infocenter.html>) 中的主题“Cachespec.xml file”。另外，请参阅《WebSphere Commerce 商店开发指南》以获取关于开发您的商店时的附加注意事项。

高速缓存缺省值

在安装期间，WebSphere Commerce 使用以下缺省值设置高速缓存系统：

- 动态高速缓存是启用的。
- 磁盘卸载是启用的。
- Servlet 高速缓存是启用的。
- 每个业务模型的缺省高速缓存策略（作为样本提供）高速缓存以下显示命令的 JSP 文件：
 - CategoryDisplay
 - ProductDisplay
 - StoreCatalogDisplay
 - TopCategoriesDisplayView

您需要手工合并此高速缓存条目，从：

▶ AIX
▶ 400
▶ Linux
▶ Solaris

`WC55_installdir/samples/dynacache/business_model/cachespec.xml`

复制到：

`WAS_installdir/installedApps/cell_name/WC_instance_name.ear/Stores.war/WEB-INF/cachespec.xml`

▶ Windows

`WC55_installdir/samples/dynacache/business_model/cachespec.xml`

复制到：

`WAS_installdir/installedApps/cell_name/WC_instance_name.ear/Stores.war/WEB-INF/cachespec.xml`

请参阅《WebSphere Commerce 样本商店指南》或《WebSphere Commerce 商店开发指南》以获取关于启用商店中的高速缓存策略的详细信息。

满页和分段高速缓存

WebSphere Application Server 动态高速缓存用于高速缓存 JSP 文件的方法是基于 JSP 是如何编写的。如果根据 URL 参数和请求属性，某个特定 WebSphere Commerce 命令的页面输出总是产生相同的结果，则可使用属性元素 `consume-subfragments` (CSF) 连同 WebSphere Commerce 控制器 servlet (`com.ibm.commerce.server.RequestServlet`) 一起作为 servlet 名称，来用高速缓存条目高速缓存此页面输出。当以此方法定义高速缓存条目时，则将以称为满页高速缓存的方法高速缓存页面输出。对控制器 servlet 使用 `consume-subfragments` 在性能方面有很大优势，然而如果此机制被用来高速缓存 Web 页面，则页面输出无法拥有个性化的信息。

如果该页面输出具有与用户相关的部分，则以称为分段高速缓存的方法高速缓存页面输出。也就是说，以分开的高速缓存条目形式高速缓存 JSP 页面，然后当它们被请求时，再重新组合在一起。对于分段 (JSP) 高速缓存，WebSphere Commerce 必须在动态高速缓存机制决定是否从高速缓存提供 JSP 前，执行此命令以决定执行哪个 JSP。此方法的优势是灵活性好，因为不同的高速缓存条目是在用户信息的基础上重新组合成页面的。

满页高速缓存

当使用属性元素 `consume-subfragments` (CSF) 时，父条目 (标记了 CSF 的条目) 将在它的高速缓存条目中包含所有分段的所有内容，形成一个大的高速缓存条目，该条目没有包含或转发，但内容来自整个条目树。

高速缓存 servlet 时，仅存储该 servlet 的内容。高速缓存包含任何其它分段的占位符，它包含这些分段或转发到这些分段。消费者段 (CSF) 告诉高速缓存当它包含子 servlet 时，不要停止保存内容。父条目 (标记了 CSF 的条目) 将在它的高速缓存条目中包含所有分段的所有内容，导致一个大的高速缓存条目，该条目没有包含或转发，但内容来自整个条目树。这样可节省可观的应用程序服务器处理量，但通常只有当外部 HTTP 请求包含决定被包含分段的整个树所需的所有信息时才有用。

例如，如果 `<cache-entry>` 定义如下：

```
<cache-entry>
  <class>servlet</class>
  <name>com.ibm.commerce.server.RequestServlet.class</name>
  <property name="consume-subfragments">true</property>
  <property name="save-attributes">false</property>
  <property name="store-cookies">false</property>

  <!-- StoreCatalogDisplay?storeId=<storeId> -->
  <cache-id>
    <component id="" type="pathinfo">
      <required>true</required>
      <value>/StoreCatalogDisplay</value>
    </component>
    <component id="storeId" type="parameter">
      <required>true</required>
    </component>
  </cache-id>
</cache-entry>
```

注意当 `save-attributes` 属性设为 `false` 时，将不对高速缓存条目保存请求属性。当 `store-cookies` 属性设置为 `false` 时，将不在高速缓存条目中保存请求 cookie。

在上面的示例中，高速缓存 servlet 条目将包含 StoreCatalogDisplay.jsp 的已消费包含，StoreCatalogDisplay.jsp 是由 StoreCatalogDisplay 命令转发的 JSP 文件。

分段高速缓存

每个动态包含的 JSP 文件必须在 cachespec.xml 文件中定义它自己的 <cache-entry>，以便当它接收到请求时可由动态高速缓存向其提供服务。否则，将为每个请求重新执行每个动态包含的 JSP 文件。例如，考虑如果 StoreCatalogDisplay.jsp 动态包含 header.jsp 和 footer.jsp 而您仅设置了 StoreCatalogDisplay.jsp 的 <cache-entry>。则当您请求 StoreCatalogDisplay 页面时，如果没有高速缓存 header.jsp 和 footer.jsp 文件，则执行它们。这里是如何定义 StoreCatalogDisplay.jsp 的 <cache-entry> 的示例：

```
<cache-entry>
  <class>servlet</class>
  <name>/ToolTech/ShoppingArea/CatalogSection/CategorySubsection/StoreCatalogDisplay.jsp</name>
  <property name="save-attributes">>false</property>

  <cache-id>
    <component id="storeId" type="parameter">
      <required>>true</required>
    </component>
    <component id="catalogId" type="parameter">
      <required>>false</required>
    </component>
  </cache-id>
</cache-entry>
```

高速缓存 servlet 过滤

WebSphere Commerce V5.5 之前，WebSphere Commerce 提供了它自己的高速缓存机制。使用以前的机制，基于两种类型的方法无法高速缓存 Web 页：

- 会话独立（SI）高速缓存 - 基于 URL 参数高速缓存页面
- 会话相关（SD）高速缓存 - 基于 URL 参数、用户语言、首选货币、父组织、合同标识以及成员组高速缓存页面

基于 URL 参数生成 SI 高速缓存的高速缓存标识，而对于 SD 高速缓存则使用 URL 参数加上会话信息创建高速缓存标识。

为了提供与以前的会话相关高速缓存相同的功能但使用 WebSphere Application Server 动态高速缓存机制，WebSphere Commerce 5.5 引入了称为高速缓存过滤器的 servlet 过滤器。此高速缓存过滤器设计为从动态高速缓存要使用的会话信息中设置请求属性来构造高速缓存标识。因为会话信息由 WebSphere Commerce 服务器运行时设置，高速缓存过滤器将无法设置所有请求属性，直到对该 Web 站点的第二次请求。

下表列出了由高速缓存过滤器设置的请求属性：

表 10. 由高速缓存过滤器设置的请求属性

请求属性	描述
DC_curr	用户的首选货币
DC_lang	用户的首选语言
DC_porg	用户的父组织
DC_cont	用户的当前合同
DC_mg	用户的显式成员组

表 10. 由高速缓存过滤器设置的请求属性 (续)

请求属性	描述
DC_storeId	商店标识
DC_userId	用户标识

因为用户有资格获得多个合同且可属于多个成员组，请求属性 DC_cont 和 DC_mg 可能包含多个值。对于这样的用户，则对这些值排序并使用分号（“;”）作为分隔符将它们并置。另外，将定义多个合同和成员组请求属性。（例如，DC_cont0、DC_cont1、... DC_contN，其中 N 是用户被授权的合同数）。例如，如果用户有资格获得合同 10004 和 10005，则将设置以下请求属性：DC_cont 是 10004;10005，DC_cont0 是 10004，DC_cont1 是 10005。

设置请求属性 DC_cont 的目的是允许构建一个具有有限数量组件的高速缓存标识，设置单独请求属性 DC_cont0、DC_cont1、...DC_contN 的目的是允许为更多的粒状高速缓存失效构建相关性标识。

因为成员组信息不是会话数据的一部分，高速缓存过滤器必须根据用户标识从数据库检索此信息。为防止由于重复数据库查询而导致性能降级，高速缓存过滤器使用 WebSphere 命令高速缓存来完成此任务。从 WebSphere 命令框架直接扩展的新命令（com.ibm.commerce.dynacache.command.MemberGroupsCacheCmdImpl）用来根据用户标识高速缓存用户所属的成员组。WebSphere Commerce 将以下高速缓存条目添加到 cachespec.xml 以通知动态高速缓存服务高速缓存此命令：

```
<cache-entry>
  <class>command</class>
  <name>com.ibm.commerce.dynacache.commands.MemberGroupsCacheCmdImpl</name>
  <cache-id>
    <component type="method" id="getUserId">
      <required>true</required>
    </component>
  </cache-id>
  <dependency-id>DC_userId</dependency-id>
  <dependency-id>DC_userId
    <component type="method" id="getUserId">
      <required>true</required>
    </component>
  </dependency-id>
</cache-entry>
```

WebSphere Application Server 动态高速缓存将基于类 MemberGroupsCacheCmdImpl 的 getUserId() 方法 getUserId() 作为其高速缓存标识，高速缓存该类的命令对象。因此，下次有相同用户标识的成员组的请求时，动态高速缓存将返回带有已计算的成员组的已高速缓存的命令对象。然后可使用命令方法 getMemberGroups() 现成地访问此成员组信息，高速缓存过滤器调用此命令方法在请求属性 DC_mg 中设置此用户的成员组。

注：WebSphere Commerce 对例如上面的高速缓存过滤器中的 MemberGroupsCacheCmdImpl 内部使用 WebSphere 命令高速缓存；但是，WebSphere Commerce 并不正式支持 WebSphere Commerce 命令的 WebSphere 命令高速缓存。

商店页面高速缓存策略

在设计您的商店页面时，有关高速缓存注意事项的信息，请参阅《*WebSphere Commerce 商店开发指南*》中的『高速缓存您的商店页面』部分。包括有关以下问题的指南：

- 什么页面应该被高速缓存？
- 应该高速缓存整个页面还是页面片段？
- 制定高速缓存策略

此外，请参阅《*WebSphere Commerce 商店开发指南*》中讨论有关高速缓存的商店关系提示的部分。

WebSphere Commerce 中的高速缓存失效机制

WebSphere Application Server 动态高速缓存服务生成的高速缓存条目包含的信息包括 servlet 输出、servlet 执行的结果（例如对其它 servlet 或 JSP 文件的调用）以及关于条目的元数据（例如超时和条目优先级信息）。不时地，当信息留在高速缓存中时，WebSphere Commerce 就需要验证此信息是当前的、一致的和准确的。因此 WebSphere Commerce 需要高效的机制来识别并除去不再有效的高速缓存信息。WebSphere Application Server 动态高速缓存服务使用称为高速缓存失效的过程对维护高速缓存信息提供支持。高速缓存失效允许 WebSphere Commerce 利用由服务提供的不同失效机制以在不同的情况下执行失效。这些机制各有不同，从基于规则、基于时间、基于组到程序化的方法。

本节仅重点描述如何在 WebSphere Commerce 中执行高速缓存失效，并不打算涉及一般高速缓存方面。它描述了 WebSphere Application Server 动态高速缓存提供的失效机制：

- 在 `cachespec.xml` 中定义了失效策略
- 通过 WebSphere Application Server 提供的高速缓存监视器除去高速缓存条目
- 在 WebSphere Commerce 应用程序中程序化地调用 WebSphere Application Server 动态高速缓存失效 API

在 `cachespec.xml` 中定义失效策略

WebSphere Application Server 动态高速缓存通过使用 `cachespec.xml` 文件支持高速缓存失效配置。可在用来配置可高速缓存对象的同一 `cachespec.xml` 中动态地定义失效策略，可在 Web 应用程序压缩文档（WAR）WEB-INF 或企业 bean WEB-INF 目录中找到该文件。关于 `cachespec.xml` 语法的详细信息，请参阅 WebSphere Application Server 信息中心

（<http://www.ibm.com/software/webservers/appserv/infocenter.html>）中的主题“`cachespec.xml` file”。

可在带有 `<invalidation>` 标记（该标记定义了必须使对应高速缓存条目失效的事件）或带有 `<timeout>` 标记（该标记配置保留高速缓存条目的持续时间）的高速缓存条目中指定失效规则。定义它们的方法与定义相关性标识的方法完全相同，由失效规则生成的标识用来使具有这些相同相关性标识的高速缓存条目失效。每个高速缓存条目可存在多个失效规则。分别执行所有失效规则。

以下各部分讨论了 `cachespec.xml` 中这些失效规则的语法并显示了如何在 WebSphere Commerce 中应用以下规则的示例：

- 基于 Servlet 的失效规则

- 基于命令的失效规则
- 超时和优先级失效规则

基于 Servlet 的失效规则

WebSphere Application Server 动态高速缓存提供的此种失效方法用于一旦 servlet 请求执行则使其它高速缓存条目失效。

cachespec.xml 中的语法:

```
<cache-entry>
  <class>servlet</class>
  <name>servlet_name</name>
  <invalidation>invalidation_id
    <component id="" type="pathinfo">
      ...
    </component>
  </invalidation>
</cache-entry>
```

注:

<class>

值 “servlet” 表示该失效由 servlet 触发。

<invalidation>

一个元素，用来标识触发失效的事件以及要使带有指定标识的哪些高速缓存条目失效。

pathinfo

在 WebSphere Commerce 中，所有 servlet 请求通过一个单独的控制器 servlet，该控制器为二者之一：

com.ibm.commerce.server.RequestServlet.class 或
com.ibm.commerce.tools.common.ToolsRequestServlet.class。

要标识唯一请求 URI，“pathinfo” 组件类型可用来从请求中过滤 pathinfo 名称。

cachespec.xml 中定义的策略的示例：当发出 InterestItemDelete URL 从列表中删除产品目录条目时，要使由其引用号标识的兴趣商品列表高速缓存页面失效：

```
<cache-entry>
  <class>servlet</class>
  <name>com.ibm.commerce.server.RequestServlet.class</name>

  <invalidation>listId
    <component id="" type="pathinfo" ignore-value="true">
      <required>true</required>
      <value>/InterestItemDelete</value>
    </component>
    <component id="listId" type="parameter">
      <required>true</required>
    </component>
  </invalidation>
</cache-entry>
```

基于命令的失效规则

此方法使作为运行命令的副作用的其它高速缓存条目失效。它提供了事件驱动的机制以在 WebSphere Commerce 中以命令级别触发失效。用户可检查命令中所写的业务逻辑并识别命令成功执行时可能受影响的特定高速缓存对象，然后相应地定义失效规则。

此类命令的一个条件是它们必须实现 WebSphere Commerce 编程模型，且如果将根据输入参数来生成失效标识，则处理这些输入参数的命令中的方法和字段必须得到具体化。

请参阅 WebSphere Application Server 文档以获取此支持的详细信息。另见第 81 页的『为基于命令的失效写入命令』。

cacheSpec.xml 中的语法:

```
<cache-entry>
  <class>command</class>
  <name>fully_qualified_class_name_of_the_command</name>
  <invalidation>invalidation_rules</invalidation>
</cache-entry>
```

注:

<class>

值 command 表示该失效由命令触发。

<name>

包含该命令的全限定路径。

cacheSpec.xml 中定义的策略的示例: 当调用 CatalogEntryUpdate 命令来更改产品目录条目时，要使按其引用号高速缓存的产品显示页失效:

```
<cache-entry>
  <class>command</class>
  <name>com.ibm.commerce.catalogmanagement.commands.CatalogEntryUpdateCmdImpl</name>
  <invalidation>productId
    <component id="getCatentryId" type="method">
      <required>true</required>
    </component>
  </invalidation>
</cache-entry>
```

基于超时和优先级的失效规则

策略中使用 <timeout> 标记的超时机制提供了一个简单的方法在预设的持续时间后除去所有高速缓存条目。当设置限时除去高速缓存条目的特定失效规则或触发器事件不可行时，该机制是很有用的。

将根据修改过的最近最少使用 (LRU) 算法在高速缓存已满时由 WebSphere Application Server 动态高速缓存服务自动除去高速缓存条目，在该算法中，使用 <priority> 标记的优先级权重用于确定要从高速缓存中除去哪些条目。特定高速缓存条目的优先级值相对于其它高速缓存条目越高，该高速缓存条目停留在此高速缓存中的时间相对于其它高速缓存条目就越长。

与高速缓存已满时从内存中除去条目不同，您可配置磁盘卸载并将条目复制到文件系统以供以后使用。（位置是可配置的。）当服务器关闭时也将把高速缓存条目卸载到磁盘，并可在服务器重新启动时重新使用。另外，以下动态高速缓存服务设置也将影响失效过程:

高速缓存大小

决定了高速缓存保留的最大条目数。

缺省优先级

决定缺省情况下一个条目可在已满的高速缓存中保留多久

磁盘卸载

指定是否启用磁盘卸载以及磁盘卸载位置。

cacheSpec.xml 中的语法:

```
<cache-entry>
  .....
  <cache-id>
    <component >.....</component>
    <priority>priority_value</priority>
    <timeout>time_in_cache</timeout>
  </cache-id>
</cache-entry>
```

cacheSpec.xml 中定义的策略的示例: 在创建或使用购物车 3600 秒之后, 要除去购物车高速缓存内容:

```
<cache-id>
  <component id="" type="pathinfo">
    <required>true</required>
    <value>/OrderItemDisplay</value>
  </component>
  <component id="orderId" type="parameter">
    <required>true</required>
  </component>
  <timeout>3600</timeout>
</cache-id>
```

通过高速缓存监视器除去高速缓存条目

WebSphere Application Server 高速缓存监视器提供了可手工使高速缓存条目失效的选项。存在一个清除高速缓存按钮可清空整个高速缓存内容。可单独地使每个高速缓存条目无效, 也可将一组高速缓存条目视为一个模板使之无效(例如, 所有与 ProductDisplay.jsp 相关联的条目)。在监视器上也可按块使由相关性标识所标识的高速缓存条目组失效。

调用动态高速缓存失效 API

WebSphere Application Server 动态高速缓存提供了以下 API 来支持程序化失效:

- com.ibm.websphere.cache.invalidateById
- com.ibm.websphere.cache.invalidateByTemplate

请参阅 WebSphere Application Server 文档以获取关于使用这些 API 的详细信息。

WebSphere Commerce 利用这些 API, 提供自制的失效包装器 DynaCacheInvalidation 命令以在 WebSphere Commerce 中支持表驱动的失效 API。它通过处理 CACHEIVL 表中的记录使高速缓存对象失效。关于更多的详细信息, 请参阅第 85 页的『高速缓存失效 API (DynaCacheInvalidation)』。

为基于命令的失效写入命令

为了允许动态高速缓存拦截命令调用, 必须将该命令与从 CacheableCommandImpl (在 com.ibm.websphere.command 数据包中) 扩展的该命令的实现类一起写入 WebSphere 命令框架中。为简化基于命令的失效的命令写入, WebSphere Commerce 已更新抽象类 ControllerCommandImpl 和 TaskCommandImpl 以从 CacheableCommandImpl 扩展, 这样任何从这些抽象类扩展的命令也都将从 CacheableCommandImpl 扩展并因此可用于基于命令的失效。

当编写此类命令时，同样重要的是要知道失效标识是什么，并理解拦截命令调用的失效规则。因为根据在命令中作为输入参数出现的方法和字段来生成失效标识，所以在命令接口中应提供构造失效标识所需的所有方法并实现它们。

在 WebSphere Commerce 中使用命令失效的示例

以下示例显示了 WebSphere Commerce 如何使用命令失效。当删除属于特定成员组的特定成员的命令 `DeleteMemberGroupMemberCmdImpl` 成功执行时，动态高速缓存将使定义在失效规则中的高速缓存条目失效。此示例中，它被定义为“`DC_userId:userId`”，其中 `userId` 是从 `getMemberId` 方法返回的值。（有关在 WebSphere Commerce 中高速缓存成员组信息的详细信息，请参阅第 76 页的『高速缓存 servlet 过滤』。）例如，`DC_userId:-1000`、`DC_userId:-1001` 等等。此命令具有 `get` 方法 `getMemberId()`，此方法检索正被删除的用户标识且此方法用于计算根据用户标识要删除带相关性标识的哪些高速缓存条目。同一逻辑应用于命令 `AddMemberGroupMemberCmdImpl`，该命令也具有 `get` 方法 `getMemberId()`：

```
<cache-entry>
  <class>command</class>
  <name>com.ibm.commerce.membergroup.commands.AddMemberGroupMemberCmdImpl</name>
  <name>com.ibm.commerce.membergroup.commands.DeleteMemberGroupMemberCmdImpl</name>

  <invalidation>DC_userId
    <component type="method" id="getMemberId">
      <required>true</required>
    </component>
  </invalidation>
</cache-entry>
```

注：以上所有失效规则都随 WebSphere Commerce 在样本高速缓存策略中提供。在 `/samples/dynacache/invalidation` 子目录下的 WebSphere Commerce 安装目录中可找到更多样本失效规则。请查看标题为“Dynacache 的样本失效高速缓存策略”的自述文件，以获取关于如何将失效规则合并到 `cachespec.xml` 文件中的更多信息。

高速缓存失效示例

以下示例说明了如何在 `cachespec.xml` 文件中设置高速缓存策略，以高速缓存 WebSphere Commerce 中消费者直销业务模型的 `ProductDisplay JSP` 页面，也说明了如何通过同一个 XML 文件中定义失效规则来使高速缓存条目失效。此示例定义了多个相关性标识以及 JSP 文件的高速缓存标识生成规则。当高速缓存条目在不同环境下更新时，每个相关性标识用来使高速缓存条目失效。此示例只说明了用来使 `ProductDisplay JSP` 失效的策略子集。关于完整的示例和详细信息，请参阅在 `WC_installdir/samples/dynacache/invalidation` 目录中的自述文件。

```
<cache>

<cache-entry>
  <class>servlet</class>
  <name>/FashionFlow/ShoppingArea/CatalogSection/CatalogEntrySubsection/ProductDisplay.jsp</name>
  <property name="save-attributes">false</property>

  <!-- Cache ProductDisplay.jsp -->
  <cache-id>
    <component id="storeId" type="parameter">
      <required>true</required>
    </component>
  </cache-id>
</cache-entry>
</cache>
```

```

        <component id="catalogId" type="parameter">
            <required>true</required>
        </component>
        <component id="productId" type="parameter">
            <required>true</required>
        </component>
        <component id="DC_lang" type="attribute">
            <required>true</required>
        </component>
        <component id="DC_curr" type="attribute">
            <required>true</required>
        </component>
        <component id="DC_porg" type="attribute">
            <required>true</required>
        </component>
        <component id="DC_cont" type="attribute">
            <required>true</required>
        </component>
        <component id="DC_mg" type="attribute">
            <required>true</required>
        </component>
    </cache-id>

    <!-- Used for invalidating the product display cache entry -->
    <!-- that belongs to a specific store -->
    <dependency-id>storeId
    <component id="storeId" type="parameter">
        <required>true</required>
    </component>
    </dependency-id>

    <!-- Used for invalidating the cache entry of a specific product -->
    <dependency-id>productId
        <component id="productId" type="parameter">
            <required>true</required>
        </component>
    </dependency-id>

    <!-- Used for invalidating the product display cache entry -->
    <!-- that belongs to a specific catalog in the store -->
    <dependency-id>storeId:catalogId
    <component id="storeId" type="parameter">
        <required>true</required>
    </component>
        <component id="catalogId" type="parameter">
            <required>true</required>
        </component>
    </dependency-id>

    <!-- Used for invalidating the product display cache entry -->
    <!-- that is under a specific contract -->
    <dependency-id>contractId
        <component id="DC_cont0" type="attribute">
            <required>true</required>
        </component>
    </dependency-id>

</cache-entry>
<cache-entry>
    <class>command</class>
    <sharing-policy>not-shared</sharing-policy>
    <name>com.ibm.commerce.catalogmanagement.commands.AddCatalogDescCmdImpl</name>
    <name>com.ibm.commerce.catalogmanagement.commands.UpdateCatalogDescCmdImpl</name>

    <!-- ***** -->

```

```

<!-- Invalidate all the product page cache entries that -->
<!-- might be affected when the catalog description is changed -->
<!-- ***** -->
<invalidation>storeId:catalogId
  <component id="getStoreId" type="method">
    <required>true</required>
  </component>
<component id="getCatalogId" type="method">
  <required>true</required>
</component>
</invalidation>

</cache-entry>

<cache-entry>
  <class>command</class>
  <sharing-policy>not-shared</sharing-policy>
  <name>com.ibm.commerce.catalogmanagement.commands.ListpriceAddCmdImpl</name>
  <name>com.ibm.commerce.catalogmanagement.commands.ListpriceDeleteCmdImpl</name>
  <name>com.ibm.commerce.catalogmanagement.commands.ListpriceUpdateCmdImpl</name>
  <name>com.ibm.commerce.catalogmanagement.commands.OfferAddCmdImpl</name>
  <name>com.ibm.commerce.catalogmanagement.commands.OfferDeleteCmdImpl</name>
  <name>com.ibm.commerce.catalogmanagement.commands.OfferUpdateCmdImpl</name>
  <name>com.ibm.commerce.catalogmanagement.commands.ProductAttributeUpdateCmdImpl</name>
  <name>com.ibm.commerce.catalogmanagement.commands.AttributeValueUpdateCmdImpl</name>
  <name>com.ibm.commerce.catalogmanagement.commands.AddListpriceCmdImpl</name>
  <name>com.ibm.commerce.catalogmanagement.commands.DeleteListpriceCmdImpl</name>
  <name>com.ibm.commerce.catalogmanagement.commands.UpdateListpriceCmdImpl</name>
  <name>com.ibm.commerce.catalogmanagement.commands.AddOfferCmdImpl</name>
  <name>com.ibm.commerce.catalogmanagement.commands.DeleteOfferCmdImpl</name>
  <name>com.ibm.commerce.catalogmanagement.commands.UpdateOfferCmdImpl</name>
  <name>com.ibm.commerce.catalogmanagement.commands.UpdateAttributeCmdImpl</name>
  <name>com.ibm.commerce.catalogmanagement.commands.UpdateAttributeValueCmdImpl</name>

  <!-- ***** -->
  <!-- Invalidate the specific product page cache entry when the -->
  <!-- product is updated -->
  <!-- ***** -->
  <invalidation>productId
    <component id="getCatentryId" type="method">
      <required>true</required>
    </component>
  </invalidation>

</cache-entry>

<cache-entry>
  <class>command</class>
  <sharing-policy>not-shared</sharing-policy>
  <name>com.ibm.commerce.contract.commands.ContractSuspendCmdImpl</name>
  <name>com.ibm.commerce.contract.commands.ContractTCDeployCmdImpl</name>

  <!-- ***** -->
  <!-- Invalidate all the product page cache entries under a -->
  <!-- specific contract -->
  <!-- ***** -->
  <invalidation>contractId
    <component id="getContractId" type="method">
      <required>true</required>
    </component>
  </invalidation>
</cache-entry>

<cache-entry>
  <class>command</class>
  <name>com.ibm.commerce.tools.devtools.store.commands.StoreProfileUpdateCmdImpl</name>
  <name>com.ibm.commerce.tools.devtools.flexflow.ui.commands.impl.FlexflowUpdateCmdImpl</name>

```



```

<name>com.ibm.commerce.store.commands.StoreOpenCmdImpl</name>
<name>com.ibm.commerce.store.commands.StoreCloseCmdImpl</name>

<!-- ***** -->
<!-- Invalidate all the product page cache entries that -->
<!-- belong to the store when the store is updated -->
<!-- ***** -->
<invalidation>storeId
  <component id="getStoreId" type="method">
    <required>true</required>
  </component>
</invalidation>

</cache-entry>

<cache-entry>
  <class>command</class>
  <sharing-policy>not-shared</sharing-policy>
  <name>com.ibm.commerce.catalogimport.commands.CatalogImportJobAddCmd</name>

  <!-- ***** -->
  <!-- Invalidate all the product page cache entries that -->
  <!-- belong to the store when the store catalog is updated -->
  <!-- ***** -->
  <invalidation>storeId
    <component id="getStoreId" type="method">
      <required>true</required>
    </component>
  </invalidation>
</cache-entry>

</cache>

```

高速缓存失效 API (DynaCacheInvalidation)

WebSphere Commerce 调度程序定期调用 DynaCacheInvalidation 命令来处理 CACHEIVL 表中的记录，然后调用动态高速缓存失效 API 来使这些高速缓存条目失效。从检查点文件检索 startTime。将把检查点文件写入 instance_name/cache 目录。当第一次执行 DynaCacheInvalidation 时创建此目录。在命令的结束阶段，将把最后一条记录的时间戳记写入检查点文件。

CACHEIVL 表

当处理 CACHEIVL 表时 DynaCacheInvalidation 命令的规则如下：

- 如果设置了模板列，则 DynaCacheInvalidation 命令调用动态高速缓存失效 API (invalidateByTemplate) 并使用该名称作为模板标识（例如 /webapp/wcs/stores/ToolTech/ShoppingArea/CatalogSection/CategorySubsection /StoreCatalogDisplay.jsp）。
- 如果设置了 DATA_ID 列且没有设置模板名称，则 DynaCacheInvalidation 命令调用动态高速缓存失效 API (invalidateById) 并使用 DATA_ID 作为标识（例如 StoreCatalogDisplay:storeId:10151）。
- 当调用动态高速缓存失效 API 时，它使高速缓存条目失效。

外部高速缓存中的失效

可将结果高速缓存推向外部高速缓存（例如 IBM HTTP Server 或 Edge Server）。当发生以下情况时将在外部高速缓存中发生失效：

- 高速缓存充满
- 发生超时
- 从 WebSphere Application Server 发送失效消息

Web 服务器插件高速缓存

对于 WebSphere Application Server 5.0，Web 服务器插件包含内置 Edge Side Include (ESI) 处理器。ESI 处理器具有高速缓存整个页面以及分段的能力，提供更高的高速缓存命中率。ESI 处理器实现的高速缓存是内存中高速缓存，而非磁盘高速缓存。因此，当您重新启动 Web 服务器时不会保存高速缓存条目。

从 Web 服务器插件接收到请求时，除非禁用了 ESI 处理器，否则将把请求发送到 ESI 处理器。（缺省情况下启用 ESI 处理器。）如果发生高速缓存不命中，将把 Surrogate-Capabilities 头添加到该请求并将请求转发到 WebSphere Application Server。如果在应用程序服务器中启用了动态 servlet 高速缓存且响应是可移动的，则应用程序服务器在对 WebSphere 插件的响应中返回 Surrogate-Control 头。

通过 WebSphere Web 服务器插件配置文件 `plugin-cfg.xml` 可配置 ESI 处理器。例如：

```
<Property Name="esiEnable" Value="true"/>
<Property Name="esiMaxCacheSize" Value="1024"/>
<Property Name="esiInvalidationMonitor" Value="true" />
```

其中：

- `esiEnable` 用来禁用 ESI 处理器（通过将值设为 `false`）。缺省情况下启用 ESI。
- `esiMaxCacheSize` 是以 1K 字节为单位的最大高速缓存大小。缺省最大高速缓存大小是 1 MB。如果高速缓存已满，要从高速缓存中除去的第一个条目是最接近失效的条目。（ESI 高速缓存条目的缺省失效时间是 24 小时。）
- `esiInvalidationMonitor` 指定 ESI 处理器是否应从应用程序服务器接收失效。

可用三种方式将高速缓存条目从 ESI 高速缓存中除去：

- 调用条目的失效超时。
- 清除条目以为新条目释放空间。
- 应用程序服务器发送了条目组的显式失效。

为启用此机制，必须将 `esiInvalidationMonitor` 设置为 `true`，且位于 `installableApps` 目录中的 `DynaCacheEsi.ear` 应用程序必须安装在应用程序服务器上。

注： `DynaCacheEsi.ear` 应将 `VH_instance_name` 作为虚拟主机。

关于 ESI 处理器的更多信息，请参阅 WebSphere Application Server 信息中心（<http://www.ibm.com/software/webservers/appserv/infocenter.html>）中的主题“Configuring Edge Side Include caching”。

简单文件 servlet

缺省情况下，WebSphere Commerce 不会高速缓存静态数据（例如图像和 HTML）。可通过将系统属性 `com.ibm.servlet.file.esi.timeOut` 设置为值 0（零），来设置此情况。如果希望启用图像和 HTML 高速缓存，您可使用以下方法之一：

- 在 `cachespec.xml` 文件中为 `SimpleFileServlet` 设置高速缓存条目。在 `cachespec.xml` 中设置高速缓存条目是一个更好的方法，因为控制正从 WebSphere Application Server 高速缓存的内容同时也使您能够在需要时使这些高速缓存条目失效。

请注意您正在应用程序服务器处将可能所有的图像存储为高速缓存条目，当到达高速缓存条目的最大数目时这可能危及性能。建议为 `SimpleFileServlet` 高速缓存条目设置相对于其它高速缓存条目较低的优先级值。

- 将系统属性 `com.ibm.servlet.file.esi.timeOut` 设置修改为非零超时值。

某些图像文件需要失效的原因是在某些业务模型中，可方便地修改徽标和标题。

例如：

```
<cache-entry>
  <class>servlet</class>
  <name>com.ibm.ws.webcontainer.servlet.SimpleFileServlet.class</name>
  <property name="EdgeCacheable">true</property>
  <cache-id>
    <component id="" type="pathinfo">
      <required>true</required>
    </component>
  </cache-id>
  <dependency-id>
    <component id="" type="pathinfo">
      <required>true</required>
    </component>
  </dependency-id>
  <timeout>1800</timeout>
</cache-entry>
```

这里是用于使图像失效的高速缓存条目的示例。

```
<cache-entry>
  <class>command</class>
  <name>com.ibm.commerce.tools.devtools.store.commands.StoreLogoUpdateCmdImpl</name>
  <invalidation>FashionFlow/images/logo.gif</invalidation>
</cache-entry>

<cache-entry>
  <class>command</class>
  <name>com.ibm.commerce.tools.devtools.store.commands.StoreBannerUpdateCmdImpl</name>
  <invalidation>FashionFlow/images/banner.gif</invalidation>
</cache-entry>
```

问题确定

如果您在使用动态高速缓存使页面得以正确高速缓存时遇到问题，请参阅以下信息：

- **问题：**无法访问高速缓存监视器。

解决方案：确保您已重新生成 Web 服务器插件并重新启动 Web 服务器和应用程序服务器，这样将装入新的虚拟主机和端口号映射。

- **问题：**无论什么页面都没有被高速缓存。

解决方案: 验证已将 cachespec.xml 文件装入 WebSphere Application Server。如果已装入 cachespec.xml, 则应可通过在侧栏的**高速缓存策略**链接上单击来从高速缓存监视器查看策略。另外 WebSphere Application Server SystemOut.log 文件将显示类似于以下内容的行:

```
[6/6/03 21:21:12:635 EDT] 65c60609 ConfigManager I DYNA0047I: Successfully loaded
cache configuration file D:\WebSphere\AppServer5\installedApps\buzz\WC_demo.ear\
Stores.war\WEB-INF/cachespec.xml.
```

请注意以上行被分隔开只是因为显示的需要。

- **问题:** 只在高速缓存您的某些页面。

解决方案: 很可能 cachespec.xml 文件有问题。您可打开跟踪, 然后访问正被调查的页面。跟踪应当表示问题。

有关启用跟踪的信息, 请参阅第 57 页的『启用跟踪组件』。

要打开动态高速缓存跟踪 (WebSphere Commerce 方), 跟踪规范应包含 “com.ibm.websphere.commerce.WC_CACHE=all=enabled”。

要打开动态高速缓存跟踪 (WebSphere Application Server 方), 跟踪规范应包含 “com.ibm.ws.cache.*=all=enabled”。

- **问题:** DynaCacheInvalidation 命令不能完成失效功能。

解决方案: 从 URL 验证命令起作用。如果您需要从 URL 运行命令, 则 cacheinvalidation.jsp 文件必须在 docpath 中。即使它不在该路径中, 命令仍将运行, 但您将接收不到报告其完成的页面。一旦 URL 调用正确, 您可验证是否已调度它通过 WebSphere Commerce 调度程序运行, 缺省值是以未调度方式运行。

- **问题:** 通过相关性标识的失效不能正常工作。

解决方案: 验证所要 servlet 或 JSP 文件的相关性标识在同一高速缓存条目块中。如果是这样, 您必须确保在不同高速缓存条目块中的相同相关性标识的多个定义具有完全相同的定义。即, 如果如下为某些高速缓存条目定义了相关性标识, 则任何其他高速缓存条目块中的相关性标识的定义必须严格相同:

```
<dependency-id>storeId
  <component id="storeId" type="parameter">
    <required>true</required>
  </component>
</dependency-id>
```

第 6 章 系统维护





本章讨论数据库清理实用程序以及您可以如何使用它来维护系统上的数据库。

数据库清理实用程序

数据库清理实用程序允许您同时从数据库删除多个对象。如果您已更改了数据库中的许多信息且有一些未使用过的表或行，就可能想进行这样的删除。有几个对象是您可以从数据库中删除的。关于这些对象的更多信息，请参阅第 98 页的『清除数据库』。

当数据库清理实用程序删除对象时，为了保持数据库的引用完整性，对象表中的记录将被删除。数据库清理实用程序命令用以下两种方式之一清理数据库：自上而下或自下而上。自上而下方式使用自动删除关联从子表中删除所有的行。如果在引用完整性中指定了删除限制，则“自动删除关联”将失败，您将只能使用自下而上的方式。

    要使用自下而上的方法，请将命令语法中的 FORCE 参数指定为 yes，这样先删除子表，然后删除父表。

    另一种触发自下而上方法的途径是将命令语法中的 LOGLEVEL 参数指定为 2。指定 0 将不记录任何内容，而指定 1 只记录顶级表的删除语句。LOGLEVEL 2 记录从每个删除的子表直到顶级表的删除语句。虽然选择 2 触发自下而上的方法，但如果在引用完整性中有删除限制，则它不能保证成功删除。要用删除限制删除记录，请将 FORCE 参数指定为 yes。

如果表中包含的子表很多，则使用自下而上的方法会需要较长的响应时间。例如，MEMBER 表就包含 500 个以上的子表。从性能因素考虑，我们建议您使用自上而下的方法。

+ 注：仅在 Loglevel 0 或 Loglevel 1 失败的情况下使用 Loglevel 2。仅在设置为 no
+ 的 FORCE 参数失败时使用设置为 yes 的 FORCE 参数。

数据库清理实用程序是可配置、可扩展且可自适应的实用程序。除了预设清理配置外，您还可以向数据库表添加新对象，以定义要清除的表和行。请参阅“添加新配置至数据库清理实用程序”部分。

如果您已经通过创建新表扩展了数据库模式，则可以使用数据库清理实用程序清理新表。如果您更改了数据库模式（例如向表中添加新列、更改外键主键关系或在引用完整性路径中添加新子表）则数据库清理实用程序将会自动适应这些更改。如果您更改了列名，则应在 CLEANCONF 表中更新配置数据。

数据库清理实用程序依据数据库模式中引用完整性定义的删除规则，删除子表中的记录。可以将删除规则设置为 on delete cascade、on delete set null 或 on delete restrict。如果添加新表，请确保引用完整性和删除规则的定义正确。否则，数据库清理实用程序无法处理新表。

注意：您只能在登台服务器上运行数据库清理实用程序以清除登台日志对象。登台数据库不同于生产数据库。登台数据库只有配置数据而没有运作数据。删除配置数据有

可能造成运作数据上的自动删除关联。当“登台传播”实用程序将删除传播到生产数据库时，这可能造成对运作数据（此数据是您想保留的）的级联删除。要清除配置数据，请在生产数据库上运行数据库清理实用程序。

数据库清理实用程序对象

数据库清理实用程序引用 CLEANCONF 表来确定在指定了特定对象和对象类型时要删除哪些表及哪些行。下表描述了 CLEANCONF 表中的预配置删除方案。您可以通过向 CLEANCONF 表中添加相似的行来配置自己的删除对象。

对象	类型	语句
account	obsolete	delete from account where markfordelete = 1 and trdtype_id = 0 and trading_id not in (select account_id from trading) and trading_id not in (select distinct account_id from ordpymthd)
address	obsolete	delete from address where status = 'T' and (days(CURRENT TIMESTAMP) - days(lastcreate)) >= ? and (address_id not in (select distinct address_id from orderitems where address_id is not null)) and (address_id not in (select distinct address_id from orders where address_id is not null)) and (address_id not in (select distinct allocaddress_id from orderitems where allocaddress_id is not null))
atp_inventory	obsolete	delete from receipt where qtyonhand = 0 and qtyinkits = 0 and receipt_id not in (select distinct receipt_id from ordpickhst where receipt_id is not null) and receipt_id not in (select distinct receipt_id from ordshipst where receipt_id is not null)
attachment	obsolete	delete from attachment where days(current timestamp) - days(timeupdated) >=? and attachment_id not in (select attachment_id from trdattach)
auction	retracted	delete from auction where austatus = 'R' and (days(CURRENT TIMESTAMP) - days(updatetime)) >= ?
auction	settlement_closed	delete from auction where austatus = 'SC' and (days(CURRENT TIMESTAMP) - days(updatetime)) >= ?
auctionlog	obsolete	delete from auctionlog where (days(CURRENT TIMESTAMP) - days(actiontime)) >= ?
autobidlog	obsolete	delete from autobidlog where (days(CURRENT TIMESTAMP) - days(actiontime)) >= ?

对象	类型	语句
baseitem	obsolete	delete from baseitem where markfordelete = 1 and baseitem_id not in (select baseitem_id from catentry) and baseitem_id not in (select distinct baseitem_id from itemspc where markfordelete = 0 and itemspc_id in (select distinct itemspc_id from orderitems) or itemspc_id in (select distinct itemspc_id from oicomplis) or itemspc_id in (select distinct itemspc_id from versionspc where versionspc_id in (select distinct versionspc_id from receipt)) or itemspc_id in (select distinct itemspc_id from radetail) or itemspc_id in (select distinct itemspc_id from bkordalloc) or itemspc_id in (select distinct itemspc_id from invreserve) or itemspc_id in (select distinct itemspc_id from rmaitem) or itemspc_id in (select distinct itemspc_id from rmaitemcmp) or itemspc_id in (select distinct itemspc_id from catentry))
bidlog	obsolete	delete from bidlog where (days(CURRENT_TIMESTAMP) - days(actiontime)) >= ?
calculation_code	obsolete	delete from calcode where published = 2 and calcode_id not in (select distinct calcode_id from ordadjust where calcode_id is not null) and calcode_id not in (select distinct calcode_id from stencalusg where calcode_id is not null) and calcode_id not in (select distinct calcode_id from ordcalcd where calcode_id is not null) and calcode_id not in (select distinct calcode_id from ordicalcd where calcode_id is not null)
catalog_group	obsolete	delete from catgroup where markfordelete = 1
catentry	without_orderitems	delete from catentry where markfordelete = 1 and (days(CURRENT_TIMESTAMP) - days(lastupdate)) >= ? and catentry_id not in (select distinct catentry_id from auction) and catentry_id not in (select distinct catentry_id from orderitems where catentry_id is not null) and catentry_id not in (select distinct catentry_id from oicomplis where catentry_id is not null) and catentry_id not in (select distinct catentry_id from rmaitem where catentry_id is not null) and catentry_id not in (select distinct catentry_id from offer where offer_id in (select distinct offer_id from orderitems))
catentry	without_orderitems-iitems	delete from catentry where markfordelete = 1 and (days(CURRENT_TIMESTAMP) - days(lastupdate)) >= ? and catentry_id not in (select distinct catentry_id from auction) and catentry_id not in (select distinct catentry_id from orderitems) and catentry_id not in (select distinct catentry_id from oicomplis where catentry_id is not null) and catentry_id not in (select distinct catentry_id from iitem) and catentry_id not in (select distinct catentry_id from rmaitem where catentry_id is not null) and catentry_id not in (select distinct catentry_id from offer where offer_id in (select distinct offer_id from orderitems))

对象	类型	语句
contract	obsolete	<ol style="list-style-type: none"> 1. delete from trading where markfordelete = 1 and trdtype_id = 1 and trading_id not in (select distinct trading_id from orderitems) and trading_id not in (select distinct trading_id from rma) and trading_id not in (select distinct trading_id from ordpaymthd) and trading_id not in (select distinct account_id from ordpaymthd) 2. delete from productset where markfordelete = 1 and productset_id not in (select distinct productset_id from tradeposcn where tradeposcn_id in (select distinct tradeposcn_id from offer where offer_id in (select distinct offer_id from orderitems))) 3. delete from tradeposcn where markfordelete = 1 and tradeposcn_id not in (select distinct tradeposcn_id from offer where offer_id in (select distinct offer_id from orderitems))
coupon_promotion	expired	delete from cppmn where days(current timestamp) - days(enddate) >=?
cpnlog	obsolete	delete from cpnlog
cpnstats	obsolete	delete from cpnstats
expected_inventory_ records	obsolete	delete from ra where markfordelete = 1 and ra_id not in (select distinct ra_id from receipt, radetail where receipt.radetail_id = radetail.radetail_id)
expected_inventory_ record_details	obsolete	delete from radetail where markfordelete = 1 and radetail_id not in (select distinct radetail_id from receipt)
forummsg	obsolete	delete from forummsg where msgstatus = 'D' or (days(CURRENT_TIMESTAMP) - days(posttime)) >= ?
fulfillment_center	obsolete	delete from ffmcenter where markfordelete = 1 and ffmcenter_id not in (select distinct ffmcenter_id from radetail) and ffmcenter_id not in (select distinct ffmcenter_id from inventory) and ffmcenter_id not in (select distinct ffmcenter_id from rma) and ffmcenter_id not in (select distinct ffmcenter_id from orderitems) and ffmcenter_id not in (select distinct allocffmc_id from orderitems) and ffmcenter_id not in (select distinct ffmcenter_id from store) and ffmcenter_id not in (select distinct rtnffmctr_id from store) and ffmcenter_id not in (select distinct ffmcenter_id from receipt) and ffmcenter_id not in (select distinct ffmcenter_id from auction) and ffmcenter_id not in (select distinct ffmcenter_id from auctionlog)
inventory_adjustments	obsolete	delete from invadjust where days(CURRENT_TIMESTAMP) - days(adjustmentdate) >= ?
inventory_adjustment_ codes	obsolete	delete from invadjcode where markfordelete = 1 and invadjcode_id not in (select distinct invadjcode_id from invadjust)

对象	类型	语句
itemspecification	obsolete	delete from itemspc where markfordelete = 1 and itemspc_id not in (select distinct itemspc_id from orderitems) and itemspc_id not in (select distinct itemspc_id from oicomplst) and itemspc_id not in (select distinct itemspc_id from versionspc where versionspc_id in (select distinct versionspc_id from receipt)) and itemspc_id not in (select distinct itemspc_id from radetail) and itemspc_id not in (select distinct itemspc_id from bkordalloc) and itemspc_id not in (select distinct itemspc_id from invreserve) and itemspc_id not in (select distinct itemspc_id from rmaitem) and itemspc_id not in (select distinct itemspc_id from rmaitemcmp) and itemspc_id not in (select distinct itemspc_id from catentry)
message	obsolete	delete from message where message_id not in (select message_id from msgmemrel) or (days(CURRENT_TIMESTAMP) - days(posttime)) >= ?
msgmemrel	obsolete	delete from msgmemrel where message_id in (select m.message_id from message ms, msgmemrel m where ms.message_id = m.message_id and (status = 'D' or ((status = 'O' or sendstat = 'S') and (days(CURRENT_TIMESTAMP) - days(posttime)) >= ?)))
order	canceled	delete from orders where status = 'X' and (days(CURRENT_TIMESTAMP) - days(lastupdate)) >= ?
order	completed	delete from orders where status = 'C' and (days(CURRENT_TIMESTAMP) - days(lastupdate)) >= ?
order	deposited	delete from orders where status = 'D' and (days(CURRENT_TIMESTAMP) - days(lastupdate)) >= ?
order	shipped	delete from orders where status = 'S' and (days(CURRENT_TIMESTAMP) - days(lastupdate)) >= ?
order	stale_guest	delete from orders where (status = 'P' or status = 'I' or status = 'W' or status = 'N') and (days(CURRENT_TIMESTAMP) - days(lastupdate)) >= ? and orders.member_id in (select distinct users_id from users where registertype = 'G') and (orders_id not in (select distinct orders_id from orderitems where inventorystatus != 'NALC' and inventorystatus is not null))
order	stale_non_guest	delete from orders where (status = 'P' or status = 'I' or status = 'W' or status = 'N') and (days(CURRENT_TIMESTAMP) - days(lastupdate)) >= ? and orders.member_id in (select distinct users_id from users where registertype != 'G') and (orders_id not in (select distinct orders_id from orderitems where inventorystatus != 'NALC' and inventorystatus is not null))
organization	obsolete	delete from member where member_id in (select orgentity_id from orgentity where orgentity_id = ?)
pastats	obsolete	delete from pastats
pcstats	obsolete	delete from pcstats
pestats	obsolete	delete from pestats

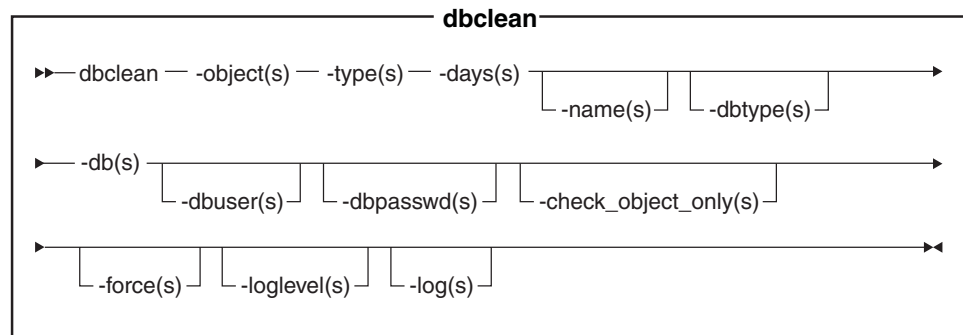
对象	类型	语句
policy	obsolete	delete from policy where days(current timestamp) - days(endtime) > ? and policy_id not in (select distinct policy_id from ordpaymthd) and policy_id not in (select distinct policy_id from rma)
rfq	obsolete	delete from trading where markfordelete = 1 and trdtype_id in (2, 3, 4) and trading_id not in (select distinct trading_id from orderitems) and trading_id not in (select distinct trading_id from rma) and trading_id not in (select distinct trading_id from ordpaymthd) and trading_id not in (select distinct account_id from ordpaymthd)
rma	abandoned	delete from rma where status in ('PRC', 'EDT') and (days(CURRENT TIMESTAMP) - days(lastupdate)) >= ? and rma_id not in (select rma_id from rmaitem where rmaitem.status in ('APP', 'MAN')) and rma_id not in (select rma_id from rtnreceipt)
rma	approved_or_partly_approved	delete from rma where status = 'PND' and (days(CURRENT TIMESTAMP) - days(lastupdate)) >= ? and rma_id in (select rma_id from rmaitem where rmaitem.status in ('APP', 'MAN')) and rma_id not in (select rma_id from rtnreceipt)
rma	canceled	delete from rma where status = 'CAN' and (days(CURRENT TIMESTAMP) - days(lastupdate)) >= ?
rma	completed	delete from rma where status = 'CLO' and (days(CURRENT TIMESTAMP) - days(lastupdate)) >= ? and rma_id not in (select rma_id from rtnreceipt)
rma	not_approved	delete from rma where status = 'PND' and (days(CURRENT TIMESTAMP) - days(lastupdate)) >= ? and rma_id not in (select rma_id from rmaitem where rmaitem.status in ('APP', 'MAN')) and rma_id not in (select rma_id from rtnreceipt)
rtnreasons	obsolete	delete from rtnreason where markfordelete = 1 and rtnreason_id not in (select distinct rtnreason_id from rtnrcptdsp) and rtnreason_id not in (select distinct rtnreason_id from rmaitem)
sastats	obsolete	delete from sastats
staglog	obsolete	delete from staglog where stgprocessed = 1 and (days(CURRENT TIMESTAMP) - days(stgstmp)) >= ?
store	obsolete	delete from storeent where storeent_id = ? and type='S'
users	guest	delete from member where member_id in (select users_id from users where registertype='G' and (days(CURRENT TIMESTAMP) - days(lastsession)) >= ? And (users_id not in (select member_id from orders where status != 'Q')) and (users_id > 0) and (users_id not in (select member_id from address where address_id in (select address_id from orderitems where address_id is not null and status != 'Q') or address_id in (select allocaddress_id from orderitems where allocaddress_id is not null and status != 'Q') or address_id in (select address_id from orders where address_id is not null and status != 'Q'))))

对象	类型	语句
users	registered	delete from member where member_id in (select users_id from users where registertype= 'R' and (days(CURRENT TIMESTAMP) - days(lastsession)) >= ? and (users_id not in (select member_id from orders where status != 'Q')) and (users_id > 0) and (users_id not in (select member_id from address where address_id in (select address_id from orderitems where address_id is not null and status != 'Q') or address_id in (select allocaddress_id from orderitems where allocaddress_id is not null and status != 'Q') or address_id in (select address_id from orders where address_id is not null and status != 'Q'))))
usrtraffic	obsolete	delete from usrtraffic where (days(CURRENT TIMESTAMP) - days(stmp)) >= ?
vendor	obsolete	delete from vendor where markfordelete = 1 and vendor_id not in (select distinct vendor_id from ra) and vendor_id not in (select distinct vendor_id from receipt where vendor_id is not null)
productset	obsolete	delete from productset where markfordelete = 1 and productset_id not in (select productset_id from tradeposcn where productset_id is not null)
productset	obsolete	delete from productset where productset_id in (select productset_id from tradeposcn where productset_id is not NULL and markfordelete = 1 and type = 'C')
tradeposcn	obsolete	delete from tradeposcn where markfordelete = 1 and type = 'S'

数据库清理命令

数据库清理实用程序命令 (AIX、iSeries、Linux、Solaris 和 Windows 2000)

数据库清理实用程序从您的数据库中除去对象。要运行数据库清理实用程序，请在命令行输入以下命令。在一行中输入整个命令。



400 iSeries 现在使用 dbclean.sh。要在 iSeries 上运行外壳程序脚本，请执行以下操作：

1. 使用 CCSID 不是 65535 的用户概要文件登录。
2. 在 OS/400 命令行上输入以下命令打开 QSHELL 命令窗口: STRQSH.
3. 如下运行命令: /QIBM/ProdData/CommerceServer55/bin/dbclean.sh (parameters . . .)

+ 注: 必须始终指定参数 -dbuser。

Oracle 注:

- 即使您当前正在用相同的用户标识运行此实用程序, 也必须在命令中包含可选参数、登录用户标识和密码。

+ **Linux 400** Oracle 引用不适用于 iSeries 或 Linux 平台。

参数值:

object 对象的名称, 将从此对象中删除废弃记录。输入以下对象名称之一:

- 输入 account 删除帐户对象。
- 输入 address 删除地址对象。
- 输入 atp_inventory 删除接收信息对象。
- 输入 attachment 删除附件对象。
- 输入 auction 删除拍卖对象。
- 输入 auctionlog 删除拍卖日志对象。
- 输入 autobidlog 删除自动投标代理对象。
- 输入 baseitem 删除产品信息对象。
- 输入 bidlog 删除投标日志对象。
- 输入 calculation_code 删除计算代码对象。
- 输入 catentry 删除产品目录条目对象。
- 输入 catalog_group 删除产品目录组对象。
- 输入 contract 删除合同对象。
- 输入 coupon_promotion 删除赠券对象
- 输入 cpnlog 删除竞销对象。
- 输入 cpnstats 删除竞销统计信息对象。
- 输入 expected_inventory_records 删除库存对象。
- 输入 expected_inventory_records_details 删除库存详细信息对象。
- 输入 forummsg 删除站点管理员和客户之间的消息对象。
- 输入 fulfillment_center 删除供货中心对象。
- 输入 inventory_adjustments 删除库存对象。
- 输入 inventory_adjustment_codes 删除库存代码对象。
- 输入 itemspecification 删除指定商品对象。
- 输入 jdbccustomizer 指定定制程序参数的位置以与 dbclean 一起使用。要配置 dbclean 以将工具箱 jdbc 驱动程序用于 iSeries, 您必须将 DB2/iSeries 指定为 dbtype, 并且您的定制程序文件应包含:
`jdbcDriver=com.ibm.as400.access.AS400JDBCdriver`
`jdbcUrlPrefix=jdbc:as400://`。iSeries 上 DB2 的缺省值为:

`jdbcDriver=com.ibm.db2.jdbc.app.DB2Driver jdbcUrlPrefix=jdbc:db2://`。
其它平台上 DB2 的缺省值为: `jdbcDriver=COM.ibm.db2.jdbc.app.DB2Driver`
`jdbcUrlPrefix=jdbc:db2`。

注: 其它组合也可能工作, 但尚未测试过。

- 输入 `message` 删除与拍卖有关的消息对象。
- 输入 `msgmemrel` 删除消息成员关系对象。
- 输入 `orders` 删除订单对象。
- 输入 `organization` 删除组织对象。
- 输入 `pastats` 删除“产品顾问”统计信息对象。
- 输入 `pcstats` 删除“产品比较”统计信息对象。
- 输入 `pestats` 删除“产品探测”统计信息对象。
- 输入 `policy` 删除策略对象。
- 输入 `product_sets` 删除产品集对象。
- 输入 `rfq` 删除报价请求对象。
- 输入 `rma` 删除退货商品对象。
- 输入 `rtnreasons` 删除退货原因对象。
- 输入 `sastats` 删除“销售辅助”统计信息对象。
- 输入 `staglog` 删除已登台对象。
- 输入 `store` 删除商店对象。
- 输入 `user` 删除用户对象。
- 输入 `usrtraffic` 删除用户通信量日志对象。
- 输入 `vendor` 删除供应商对象。

type 要删除的对象的类型。请参阅清理数据库下的各个命令。

days 要删除的记录的最短存在天数。

name (可选) 要删除的对象标识。如果 `member` 表示 `organization` 参数的值且 `organization` 表示 `type` 值, 则此参数是必需的。

dbtype

+
+

(可选) 数据库类型为 DB2、DB2/iSeries 或 Oracle。缺省值为 DB2。▶ 400
缺省值为 DB2/iSeries。

db 数据库的名称。▶ 400 可在关系数据库目录中找到的数据库名称。

注: 如果您已配置 `jdbccustomizer` 参数使用工具箱 `jdbc` 驱动程序, 则必须指定数据库所驻留的主机的名称, 而非数据库名, 例如 `-db hostname.ibm.com`。

▶ Oracle

使用 `host:port:sid`。例如, `myhost:1521:mydb`。

dbuser

(可选) 已创建模式的数据库管理员或数据库的站点管理员的登录标识。如果没有指定此参数, 则将使用调用此实用程序的用户的标识。▶ 400 (必需) 与 WebSphere Commerce 实例相关联的用户概要文件。这也是模式名称。

dbpasswd

(可选) 由 `dbuser` 参数为 DB2 数据库指定的登录标识的密码。如果没有指定, 系统将提示您输入密码。

check_object_only

(可选) 数据库清理实用程序列出所有可能受数据库清理影响的子表, 并在您指定 `yes` 的情况下删除限制。如果您保持此参数为 `no` (缺省值) 的话, 实用程序不会执行检查。

force (可选) 强制选项可以设置成 `yes` 或 `no`。如果您指定了 `yes`, 则实用程序删除子表, 接着删除父表。


loglevel


(可选) 在数据库清理期间要执行的记录级别。如果没有指定值, 则缺省记录级别是 0。

- 输入 0 指定不记录日志活动。这是缺省值。
- 输入 1 指定将只为指定表记录 `DELETE` 语句。
- 输入 2 指定将只为指定的表及任何子表记录 `DELETE` 语句。记录级别 2 将强制数据库清理实用程序使用自下而上的方法。

log (可选) 日志文件的路径和名称, 实用程序在此日志文件中记录自身活动。该命令的发出者必须有对指定路径 (该路径必须已经存在) 的写权限。如果没有指定此参数, 则在以下日志目录中创建称为 `dbclean_yyyy.mm.dd_hh.mm.ss.zzz.log` 的日志文件。

   `WC_installdir/logs`

 `WC_installdir\logs`

 如果没有指定此参数, 则在 `WC_userdir/instances` 目录中创建称为 `dbclean_dbuser_yyyy.mm.dd_hh.mm.ss.zzz.log` 的日志文件。

instancexml

当删除商店时使用。 `instancexml` 以 `WC_installdir/instances/instancexml/xml/instancexml.xml` 格式取得绝对文件名的值。

注: 此参数对一些拥有文件资源 (例如用户、组织、商店和附件) 的对象是必需的。

script (可选) 指定要保存的输出脚本的路径和名称。缺省情况下, 它保存为

 `WC_installdir\logs\assetclean_userName_timestamp_.cmd`

 `/WC_userdir/instances/assetclean_userName_timestamp_.sh`

, 而对于其它平台为 `WC_installdir/logs/assetclean_userName_timestamp_.sh`。

清除数据库

要使用数据库清理实用程序从 WebSphere Commerce 数据库中删除不使用或废弃的对象, 请选择以下操作之一:

- 帐户对象
- 地址对象
- 可承诺 (ATP) 库存对象
- 附件对象

- Professional 拍卖对象
- Professional 拍卖日志对象
- Professional 自动投标代理日志对象
- 基本商品对象
- Professional 投标代理日志对象
- 计算代码对象
- 竞销对象
- 竞销统计信息对象
- 目录条目对象
- 目录组对象
- 合同对象
- 赠券对象
- 预期库存记录对象
- 预期库存记录详细信息对象
- 论坛消息对象
- 实现中心对象
- 库存调整代码对象
- 库存调整对象
- 商品规格对象
- 成员消息对象
- 消息对象
- 订单对象
- 组织对象
- Professional 产品顾问统计信息对象
- Professional 产品比较统计信息对象
- Professional 产品探测统计信息对象
- 策略对象
- 产品集对象
- 报价请求 (RFQ) 对象
- 退货商品对象
- 退货原因对象
- Professional 销售辅助统计信息对象
- 登台对象
- 商店对象
- 用户对象
- 用户通信量日志对象
- 供应商对象

注：在可以使用数据库清理实用程序之前，必须配置数据库并设置 PATH 环境变量。

为 WebSphere Commerce 实用程序设置 PATH 环境变量

注：▶ 400 本节不适用于 iSeries 平台。

在使用“数据库清理”实用程序或登台服务器之前，请确保在 PATH 环境变量中设置了以下内容：

- ▶ 2000 ▶ DB2
`WC_installdir/bin` 和 `WC_installdir\sqllib\bin`。
- ▶ 2000 ▶ Oracle
`WC_installdir/bin` 和 `WC_installdir\ora9i\bin`
- ▶ 2000 ▶ DB2
`WC_installdir/bin` 和 `WC_installdir\sqllib\bin`
- ▶ 2000 ▶ Oracle
`WC_installdir/bin` 和 `WC_installdir\ora9i\bin`
- ▶ AIX ▶ DB2
`WC_installdir/bin` 和 `WC_installdir/lpp/db2_08_01`
- ▶ AIX ▶ Oracle
`WC_installdir/bin` 和 `/usr/ora9i/bin`
- ▶ Solaris ▶ DB2
`WC_installdir/bin` 和 `WC_installdir/IBMdb2/V8.1`
- ▶ Linux ▶ DB2
`WC_installdir/bin` 和 `opt/IBM/db2/V8.1`
- ▶ Solaris ▶ Oracle
`WC_installdir/bin` 和 `WC_installdir/ora9i/bin`

删除对象

本节包含使用数据库清理实用程序从 WebSphere Commerce 数据库删除特定的未使用的或废弃的对象的内容。









删除帐户对象

要删除帐户对象，请执行以下操作：

1. 输入以下命令：

- ▶ Windows ▶ DB2
`dbclean -object account -type obsolete -db dbname -days daysold -loglevel loglevel`
- ▶ Windows ▶ Oracle
`dbclean -object account -type obsolete -db dbname -days daysold -loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password`

+
+
+

-     
 `. dbclean.sh -object account -type obsolete -db dbname -dbuser user -days daysold -loglevel loglevel`
-   
 `. dbclean.sh -object account -type obsolete -db dbname -days daysold -loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password`

将 `host:port:sid` 用作 Oracle 数据库名称。例如, `myhost:1521:mydb`。













2. 检查 `dbclean_yyyy.mm.dd_hh.mm.ss.zzz.log` 文件来验证命令是否成功。

关于帐户对象的附加示例, 请参阅第 123 页的『删除对象的示例』。

删除地址对象

要删除地址对象, 请执行以下操作:

1. 输入以下命令:

-  
 `dbclean -object address -type obsolete -db dbname -days daysold -loglevel loglevel`
-  
 `dbclean -object address -type obsolete -db dbname -days daysold -loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password`
-     
 `. dbclean.sh -object address -type obsolete -db dbname -dbuser user -days daysold -loglevel loglevel`
-   
 `. dbclean.sh -object address -type obsolete -db dbname -days daysold -loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password`

+
+
+

将 `host:port:sid` 用作 Oracle 数据库名称。例如, `myhost:1521:mydb`。





2. 检查 `dbclean_yyyy.mm.dd_hh.mm.ss.zzz.log` 文件来验证命令是否成功。






关于删除地址对象的附加示例, 请参阅第 123 页的『删除对象的示例』。




删除可承诺库存对象

要删除可承诺 (ATP) 库存对象, 请执行以下操作:

1. 输入以下命令:

-  
 `dbclean -object atp_inventory -type typename -db dbname -loglevel loglevel`
-  
 `dbclean -object atp_inventory -type typename -db dbname -logLevel loglevel -dbtype oracle -dbuser user -dbpasswd password`

```
+
+
+
•     
  . dbclean.sh -object atp_inventory -type typename -db dbname -dbuser user
  -loglevel loglevel
```

```
•   
  . dbclean.sh -object atp_inventory -type typename -db dbname -loglevel
  loglevel -dbtype oracle -dbuser user -dbpasswd password
```

将 *host:port:sid* 用作 Oracle 数据库名称。例如, *myhost:1521:mydb*。



2. 检查 *dbclean_yyyy.mm.dd_hh.mm.ss.zzz.log* 文件来验证命令是否成功。



关于删除 ATP 库存对象的附加示例, 请参阅第 123 页的『删除对象的示例』。



删除附件对象




要删除附件对象, 请执行以下操作:

1. 输入以下命令:

```
•  
  dbclean -object attachment -type obsolete -db dbname -days daysold
  -loglevel loglevel
```

```
•  
  dbclean -object attachment -type obsolete -db dbname -days daysold
  -loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password
```

```
+
+
+
•     
  . dbclean.sh -object attachment -type obsolete -db dbname -dbuser user
  -days daysold -loglevel loglevel
```

```
•   
  . dbclean.sh -object attachment -type obsolete -db dbname -days daysold
  -loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password
```

将 *host:port:sid* 用作 Oracle 数据库名称。例如, *myhost:1521:mydb*。

注: 附件的 *-type* 也可为 *markfordelete*。这种情况下, *-instancexml* 是必需的。
例如:

```
dbclean.sh -object attachment -type markfordelete -instancexml
  /WC_userdir/instances/instance/xml/instance.xml
  -db dbname -dbuser instance -loglevel
```



2. 检查 *dbclean_yyyy.mm.dd_hh.mm.ss.zzz.log* 文件来验证命令是否成功。











关于删除附件对象的附加示例, 请参阅第 123 页的『删除对象的示例』。

删除拍卖对象

要删除拍卖对象, 请执行以下操作:

1. 输入以下命令:

```
•  
  dbclean -object auction -type typename -db dbname -days daysold -loglevel
  loglevel
```

-  
`dbclean -object auction -type typename -db dbname -days daysold -loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password`
- +
+
+
•     
`. dbclean.sh -object auction -type typename -db dbname -dbuser user -days daysold -loglevel loglevel`
-   
`. dbclean.sh -object auction -type typename -db dbname -days daysold -loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password`

将 `host:port:sid` 用作 Oracle 数据库名称。例如, `myhost:1521:mydb`。注: 对于 `-type` 参数, 可以指定 `settlement_closed` 以表示完成的拍卖记录, 或指定 `retracted` 以表示已撤销的拍卖。













2. 检查 `dbclean.log_yyyy.mm.dd_hh.mm.ss.zzz` 文件以验证该命令是否成功。

关于删除拍卖对象的附加示例, 请参阅第 123 页的『删除对象的示例』。

删除拍卖日志对象

要删除拍卖日志对象, 请执行以下操作:

1. 输入以下命令:

-  
`dbclean -object auctionlog -type obsolete -db dbname -days daysold -loglevel loglevel`
-  
`dbclean -object auctionlog -type obsolete -db dbname -days daysold -loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password`
- +
+
+
•     
`. dbclean.sh -object auctionlog -type obsolete -db dbname -dbuser user -days daysold -loglevel loglevel`
-   
`. dbclean.sh -object auctionlog -type obsolete -db dbname -dbuser user -days daysold -loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password`

将 `host:port:sid` 用作 Oracle 数据库名称。例如, `myhost:1521:mydb`。













2. 检查 `dbclean_yyyy.mm.dd_hh.mm.ss.zzz.log` 文件来验证命令是否成功。

关于删除拍卖日志对象的附加示例, 请参阅第 123 页的『删除对象的示例』。

删除自动投标代理日志对象

要删除自动投标代理日志对象, 请执行以下操作:

1. 输入以下命令:

-  
`dbclean -object autobidlog -type obsolete -db dbname -days daysold -loglevel loglevel`
-  
`dbclean -object autobidlog -type obsolete -db dbname -days daysold -loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password`
- +     
`. dbclean.sh -object autobidlog -type obsolete -db dbname -dbuser user -days daysold -loglevel loglevel`
- +   
`. dbclean.sh -object autobidlog -type obsolete -db dbname -days daysold -loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password`

将 `host:port:sid` 用作 Oracle 数据库名称。例如, `myhost:1521:mydb`。













2. 检查 `dbclean_yyyy.mm.dd_hh.mm.ss.zzz.log` 文件来验证命令是否成功。

关于删除自动投标代理日志对象的附加示例, 请参阅第 123 页的『删除对象的示例』。

删除基本商品对象

要删除基本商品对象 (它包含有关带有相同名称和描述的一般商品系列的信息), 请执行以下操作:

1. 输入以下命令:

-  
`dbclean -object baseitem -type obsolete -db dbname -loglevel loglevel`
-  
`dbclean -object baseitem -type obsolete -db dbname -loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password`
- +     
`. dbclean.sh -object baseitem -type obsolete -db dbname -dbuser user -loglevel loglevel`
- +   
`. dbclean.sh -object baseitem -type obsolete -db dbname -loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password`

将 `host:port:sid` 用作 Oracle 数据库名称。例如, `myhost:1521:mydb`。

2. 检查 `dbclean_yyyy.mm.dd_hh.mm.ss.zzz.log` 文件来验证命令是否成功。










关于删除基本商品对象的附加示例, 请参阅第 123 页的『删除对象的示例』。

删除投标日志对象

要删除投标日志对象, 请执行以下操作:

1. 输入以下命令:

+
+
+

-  
`dbclean -object bidlog -type obsolete -db dbname -days daysold -loglevel loglevel`
-  
`dbclean -object bidlog -type obsolete -db dbname -days daysold -loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password`
-     
`. dbclean.sh -object bidlog -type obsolete -db dbname -dbuser user -days daysold -loglevel loglevel`
-   
`. dbclean.sh -object bidlog -type obsolete -db dbname -days daysold -loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password`

将 `host:port:sid` 用作 Oracle 数据库名称。例如, `myhost:1521:mydb`。













2. 检查 `dbclean_yyyy.mm.dd_hh.mm.ss.zzz.log` 文件来验证命令是否成功。

关于删除竞标日志对象的附加示例, 请参阅第 123 页的『删除对象的示例』。

删除计算代码对象

要删除计算代码对象, 请执行以下操作:

1. 输入以下命令:

-  
`dbclean -object calculation_code -type obsolete -db dbname -loglevel loglevel`
-  
`dbclean -object calculation_code -type obsolete -db dbname -loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password`
-     
`. dbclean.sh -object calculation_code -type obsolete -db dbname -dbuser user -loglevel loglevel`
-   
`. dbclean.sh -object calculation_code -type obsolete -db dbname -loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password`

将 `host:port:sid` 用作 Oracle 数据库名称。例如, `myhost:1521:mydb`。

2. 检查 `dbclean_yyyy.mm.dd_hh.mm.ss.zzz.log` 文件来验证命令是否成功。













关于删除计算代码对象的附加示例, 请参阅第 123 页的『删除对象的示例』。

删除竞销对象

要删除竞销对象, 请执行以下操作:

1. 输入以下命令:

+
+
+

-  
`dbclean -object cpnlog -type all -db dbname -loglevel loglevel`
-  
`dbclean -object cpnlog -type all -db dbname -loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password`
-     
`. dbclean.sh -object cpnlog -type all -db dbname -dbuser user -loglevel loglevel`
-   
`. dbclean.sh -object cpnlog -type all -db dbname -loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password`

将 `host:port:sid` 用作 Oracle 数据库名称。例如, `myhost:1521:mydb`。













2. 检查 `dbclean_yyyy.mm.dd_hh.mm.ss.zzz.log` 文件来验证命令是否成功。

关于删除竞销对象的附加示例, 请参阅第 123 页的『删除对象的示例』。

删除竞销统计信息对象

要删除竞销统计信息对象, 请执行以下操作:

1. 输入以下命令:

-  
`dbclean -table cpnstats -type all -db dbname -loglevel loglevel`
-  
`dbclean -table cpnstats -type all -db dbname -loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password`
-     
`. dbclean.sh -object cpnstats -type all -db dbname -dbuser user -loglevel loglevel`
-   
`. dbclean.sh -object cpnstats -type all -db dbname -loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password`

将 `host:port:sid` 用作 Oracle 数据库名称。例如, `myhost:1521:mydb`。

2. 检查 `dbclean_yyyy.mm.dd_hh.mm.ss.zzz.log` 文件来验证命令是否成功。













有关从 CPGNSTATS 表中删除竞销统计信息记录的附加示例, 请参阅第 123 页的『删除对象的示例』。

删除产品目录条目对象

要删除产品目录条目对象, 请执行以下操作:

1. 输入以下命令:

+
+
+

-  
`dbclean -object catentry -type typename -db dbname -days daysold -loglevel loglevel`
-  
`dbclean -object catentry -type typename -db dbname -days daysold -loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password`
-     
`. dbclean.sh -object catentry -type typename -db dbname -dbuser user -days daysold -loglevel loglevel`
-   
`. dbclean.sh -object catentry -type typename -db dbname -days daysold -loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password`

将 `host:port:sid` 用作 Oracle 数据库名称。例如, `myhost:1521:mydb`。

注: 对于 `-type` 参数, 可以指定 `without_orderitems` 以表示不指向任何订购商品的产品目录条目, 或指定 `without_orderitems_iitems` 以表示不指向任何订购商品或兴趣列表商品的产品目录条目。

2. 检查 `dbclean_yyyy.mm.dd_hh.mm.ss.zzz.log` 文件来验证命令是否成功。













关于删除产品目录条目对象的附加示例, 请参阅第 123 页的『删除对象的示例』。

删除产品目录组对象

要删除产品目录组对象, 请执行以下操作:

1. 输入以下命令:

+
+
+

-  
`dbclean -object catalog_group -type obsolete -db dbname -loglevel loglevel`
-  
`dbclean -object catalog_group -type obsolete -db dbname -loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password`
-     
`. dbclean.sh -object catalog_group -type obsolete -db dbname -dbuser user -loglevel loglevel`
-   
`. dbclean.sh -object catalog_group -type obsolete -db dbname -loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password`

将 `host:port:sid` 用作 Oracle 数据库名称。例如, `myhost:1521:mydb`。













2. 检查 `dbclean_yyyy.mm.dd_hh.mm.ss.zzz.log` 文件来验证命令是否成功。

关于删除产品目录组对象的附加示例, 请参阅第 123 页的『删除对象的示例』。






删除合同对象

要删除合同对象，请执行以下操作：

1. 输入以下命令：

-  
`dbclean -object contract -type obsolete -db dbname -loglevel loglevel`
-  
`dbclean -object contract -type obsolete -db database -loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password`
-     
`. dbclean.sh -object contract -type obsolete -db dbname -dbuser user -loglevel loglevel`
-   
`. dbclean.sh -object contract -type obsolete -db dbname -loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password`

将 `host:port:sid` 用作 Oracle 数据库名称。例如，`myhost:1521:mydb`。

注：     对于 `-dbor DATABASE` 参数，必须指定以下三个数据库表中的一个：`productset`、`tradeposn` 或 `trading`。













2. 检查 `dbclean_yyyy.mm.dd_hh.mm.ss.zzz.log` 文件来验证命令是否成功。

关于删除合同对象的附加示例，请参阅第 123 页的『删除对象的示例』。

删除赠券对象

要删除赠券对象，请执行以下操作：

1. 输入以下命令：

-  
`dbclean -object coupon_promotion -type expired -db dbname -days daysold -loglevel loglevel`
-  
`dbclean -object coupon_promotion -type expired -db dbname -days daysold -loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password`
-     
`. dbclean.sh -object coupon_promotion -type expired -db dbname -dbuser user -days daysold -loglevel loglevel`
-   
`. dbclean.sh -object coupon_promotion -type expired -db dbname -days daysold -loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password`

将 `host:port:sid` 用作 Oracle 数据库名称。例如，`myhost:1521:mydb`。

2. 检查 `dbclean_yyyy.mm.dd_hh.mm.ss.zzz.log` 文件来验证命令是否成功。

关于删除赠券对象的附加示例，请参阅第 123 页的『删除对象的示例』。

删除预期库存记录对象

要删除预期库存记录对象，请执行以下操作：

1. 输入以下命令：






•  

```
dbclean -object expected_inventory_records -type obsolete -db dbname -days daysold -loglevel loglevel
```




•  

```
dbclean -object expected_inventory_records -type obsolete -db dbname -days daysold -loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password
```

+
+
+

•     

```
. dbclean.sh -object expected_inventory_records -type obsolete -db dbname -dbuser user -days daysold -loglevel loglevel
```

•   

```
. dbclean.sh -object expected_inventory_records -type obsolete -db dbname -days daysold -loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password
```

将 *host:port:sid* 用作 Oracle 数据库名称。例如，*myhost:1521:mydb*。

2. 检查 *dbclean_yyyy.mm.dd_hh.mm.ss.zzz.log* 文件来验证命令是否成功。

关于删除预期库存记录对象的附加示例，请参阅第 123 页的『删除对象的示例』。

删除预期库存记录详细信息对象

要删除预期库存记录详细信息对象，请执行以下操作：

1. 输入以下命令：

•  

```
dbclean -object expected_inventory_record_details -type obsolete -db dbname -loglevel loglevel
```

•  

```
dbclean -object expected_inventory_record_details -type obsolete -db dbname -loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password
```

+
+
+

•     

```
. dbclean.sh -object expected_inventory_record_details -type obsolete -db dbname -dbuser user -loglevel loglevel
```

•   

```
. dbclean.sh -object expected_inventory_record_details -type obsolete -db dbname -loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password
```

将 *host:port:sid* 用作 Oracle 数据库名称。例如，*myhost:1521:mydb*。













2. 检查 *dbclean_yyyy.mm.dd_hh.mm.ss.zzz.log* 文件来验证命令是否成功。

关于删除预期库存记录对象详细信息的附加示例，请参阅第 123 页的『删除对象的示例』。

删除论坛消息对象

要删除论坛消息对象，请执行以下操作：

1. 输入以下命令：

-  
`dbclean -object forummsg -type obsolete -db dbname -days daysold -loglevel loglevel`
-  
`dbclean -object forummsg -type obsolete -db dbname -days daysold -loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password`
- +     
+ `. dbclean.sh -object forummsg -type obsolete -db dbname -dbuser user -days daysold -loglevel loglevel`
+
-   
`. dbclean.sh -object forummsg -type obsolete -db dbname -days daysold -loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password`

将 `host:port:sid` 用作 Oracle 数据库名称。例如，`myhost:1521:mydb`。













2. 检查 `dbclean_yyyy.mm.dd_hh.mm.ss.zzz.log` 文件来验证命令是否成功。

关于删除论坛消息对象的附加示例，请参阅第 123 页的『删除对象的示例』。

删除供货中心对象

要删除供货中心对象，请执行以下操作：

1. 输入以下命令：

-  
`dbclean -object fulfillment_center -type obsolete -db dbname -loglevel loglevel`
-  
`dbclean -object fulfillment_center -type obsolete -db dbname -loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password`
- +     
+ `. dbclean.sh -object calculation_code -type obsolete -db dbname -dbuser user -loglevel loglevel`
+
-   
`. dbclean.sh -object fulfillment_center -type obsolete -db dbname -loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password`

将 `host:port:sid` 用作 Oracle 数据库名称。例如，`myhost:1521:mydb`。













2. 检查 `dbclean_yyyy.mm.dd_hh.mm.ss.zzz.log` 文件来验证命令是否成功。

关于删除供货中心对象的附加示例，请参阅第 123 页的『删除对象的示例』。

删除库存代码调整对象

要删除库存代码调整对象，请执行以下操作：

1. 输入以下命令：

-  
`dbclean -object inventory_adjustment_codes -type obsolete -db dbname -loglevel loglevel`
-  
`dbclean -object inventory_adjustment_codes -type obsolete -db dbname -loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password`
-     
`. dbclean.sh -object inventory_adjustment_codes -type obsolete -db dbname -dbuser user -loglevel loglevel`
-   
`. dbclean.sh -object inventory_adjustment_codes -type obsolete -db dbname -loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password`

将 `host:port:sid` 用作 Oracle 数据库名称。例如，`myhost:1521:mydb`。













2. 检查 `dbclean_yyyy.mm.dd_hh.mm.ss.zzz.log` 文件来验证命令是否成功。

关于删除库存代码调整对象的附加示例，请参阅第 123 页的『删除对象的示例』。

删除库存调整对象

要删除库存调整对象，请执行以下操作：

1. 输入以下命令：

-  
`dbclean -object inventory_adjustments -type obsolete -db dbname -days daysold -loglevel loglevel`
-  
`dbclean -object inventory_adjustments -type obsolete -db dbname -days daysold -loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password`
-     
`. dbclean.sh -object inventory_adjustments -type obsolete -db dbname -dbuser user -days daysold -loglevel loglevel`
-   
`. dbclean.sh -object inventory_adjustments -type obsolete -db dbname -days daysold -loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password`

将 `host:port:sid` 用作 Oracle 数据库名称。例如，`myhost:1521:mydb`。













2. 检查 `dbclean_yyyy.mm.dd_hh.mm.ss.zzz.log` 文件来验证命令是否成功。

关于删除库存调整对象的附加示例，请参阅第 123 页的『删除对象的示例』。

删除指定商品信息对象

要删除指定商品信息对象，请执行以下操作：

1. 输入以下命令：

-  
`dbclean -object itemspecification -type obsolete -db dbname -loglevel loglevel`
-  
`dbclean -object itemspecification -type obsolete -db dbname -loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password`
-     
`. dbclean.sh -object itemspecification -type obsolete -db dbname -dbuser user -loglevel loglevel`
-   
`. dbclean.sh -object itemspecification -type obsolete -db dbname -loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password`

将 `host:port:sid` 用作 Oracle 数据库名称。例如，`myhost:1521:mydb`。













2. 检查 `dbclean_yyyy.mm.dd_hh.mm.ss.zzz.log` 文件来验证命令是否成功。

关于指定商品信息对象的附加示例，请参阅第 123 页的『删除对象的示例』。

删除成员消息关系对象

要删除成员消息关系对象，请执行以下操作：

1. 输入以下命令：

-  
`dbclean -object msgmemrel -type obsolete -db dbname -days daysold -loglevel loglevel`
-  
`dbclean -object msgmemrel -type obsolete -db dbname -days daysold -loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password`
-     
`. dbclean.sh -object msgmemrel -type obsolete -db dbname -dbuser user -days daysold -loglevel loglevel`
-   
`. dbclean.sh -object msgmemrel -type obsolete -db dbname -days daysold -loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password`

将 `host:port:sid` 用作 Oracle 数据库名称。例如，`myhost:1521:mydb`。

2. 检查 `dbclean_yyyy.mm.dd_hh.mm.ss.zzz.log` 文件来验证命令是否成功。

关于删除成员消息关系对象的附加示例，请参阅第 123 页的『删除对象的示例』。

删除消息对象

要删除消息对象，请执行以下操作：

1. 输入以下命令：



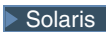

-  

```
dbclean -object message -type obsolete -db dbname -days daysold -loglevel  
loglevel
```


-  

```
dbclean -object message -type obsolete -db dbname -days daysold -loglevel  
loglevel -dbtype oracle -dbuser user -dbpasswd password
```

+
+
+

-     

```
. dbclean.sh -object message -type obsolete -db dbname -dbuser user -days  
daysold -loglevel loglevel
```

-   

```
. dbclean.sh -object message -type obsolete -db dbname -days daysold  
-loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password
```

将 *host:port:sid* 用作 Oracle 数据库名称。例如，*myhost:1521:mydb*。

2. 检查 *dbclean_yyyy.mm.dd_hh.mm.ss.zzz.log* 文件来验证命令是否成功。

关于删除消息对象的附加示例，请参阅第 123 页的『删除对象的示例』。

删除订单对象

要删除订单对象，请执行以下操作：

1. 输入以下命令：

-  

```
dbclean -object order -type typename -db dbname -days daysold -loglevel  
loglevel
```

-  

```
dbclean -object order -type typename -db dbname -days daysold -loglevel  
loglevel -dbtype oracle -dbuser user -dbpasswd password
```

+
+
+

-     

```
. dbclean.sh -object order -type typename -db dbname -dbuser user -days  
daysold -loglevel loglevel
```

-   

```
. dbclean.sh -object order -type typename -db dbname -days daysold  
-loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password
```

将 *host:port:sid* 用作 Oracle 数据库名称。例如，*myhost:1521:mydb*。

注：对于 *-type* 参数，可以指定六种不同类型之一：*completed* 表示已完成的订单、*canceled* 表示已取消的订单、*shipped* 表示已装运的订单、*deposited* 表示已圈出的订单、*stale_guest* 表示来自临时客户的过期订单或 *stale_non_guest* 表示非临时客户的过期订单。










2. 检查 `dbclean_yyyy.mm.dd_hh.mm.ss.zzz.log` 文件来验证命令是否成功。

关于删除订单对象的附加示例，请参阅第 123 页的『删除对象的示例』。

删除组织对象

要删除组织对象，请执行以下操作：

1. 输入以下命令：

-  
`dbclean -object organization -type specified -db dbname -loglevel loglevel
-name organizationid -instancexml
WC_installdir/instances/INSTANCE_NAME/xml/INSTANCE_NAME.xml`
-  
`dbclean -object organization -type specified -db dbname -loglevel loglevel
-dbtype oracle -dbuser user -dbpasswd password -name organizationid
-instancexml
WC_installdir/instances/INSTANCE_NAME/xml/INSTANCE_NAME.xml`
- +     
+ `. dbclean.sh -object organization -type specified -db dbname -dbuser user
+ -loglevel loglevel -name organizationid -instancexml
+ WC_installdir/instances/INSTANCE_NAME/xml/INSTANCE_NAME.xml`
-   
`. dbclean.sh -object organization -type specified -db dbname -loglevel
loglevel -dbtype oracle -dbuser user -dbpasswd password -name
organizationid -instancexml
WC_installdir/instances/INSTANCE_NAME/xml/INSTANCE_NAME.xml`

将 `host:port:sid` 用作 Oracle 数据库名称。例如，`myhost:1521:mydb`。注：对于 `-type` 参数，可以指定 `organization` 以表示任何组织记录。










2. 检查 `dbclean_yyyy.mm.dd_hh.mm.ss.zzz.log` 文件来验证命令是否成功。

关于删除组织对象的附加示例，请参阅第 123 页的『删除对象的示例』。

删除产品顾问统计信息对象

要删除产品顾问统计信息对象，请执行以下操作：

1. 输入以下命令：

-  
`dbclean -object pastats -type all -db dbname -loglevel loglevel`
-  
`dbclean -object pastats -type all -db dbname -loglevel logLevel -dbtype
oracle -dbuser user -dbpasswd password`
- +     
+ `. dbclean.sh -object pastats -type all -db dbname -dbuser user -loglevel
+ loglevel`

-   

```
. dbclean.sh -object pastats -type all -db dbname -loglevel loglevel
-dbtype oracle -dbuser user -dbpasswd password
```

将 *host:port:sid* 用作 Oracle 数据库名称。例如, *myhost:1521:mydb*。

2. 检查 *dbclean_yyyy.mm.dd_hh.mm.ss.zzz.log* 文件来验证命令是否成功。

关于删除产品顾问统计信息对象的附加示例, 请参阅第 123 页的『删除对象的示例』。

删除产品比较统计信息对象

要删除产品比较 (产品顾问) 统计信息对象, 请执行以下操作:

1. 输入以下命令:

-  

```
dbclean -object pcstats -type all -db dbname -loglevel loglevel
```

-  

```
dbclean -object pcstats -type all -db dbname -loglevel loglevel -dbtype
oracle -dbuser user -dbpasswd password
```

+
+
+

-     

```
. dbclean.sh -object pcstats -type all -db dbname -dbuser user -loglevel
loglevel
```

-   

```
. dbclean.sh -object pcstats -type all -db dbname -loglevel loglevel
-dbtype oracle -dbuser user -dbpasswd password
```

将 *host:port:sid* 用作 Oracle 数据库名称。例如, *myhost:1521:mydb*。

2. 检查 *dbclean_yyyy.mm.dd_hh.mm.ss.zzz.log* 文件来验证命令是否成功。

关于删除产品比较 (产品顾问) 统计信息对象的附加示例, 请参阅第 123 页的『删除对象的示例』。

删除产品探测统计信息对象

要删除产品探测 (产品顾问) 统计信息对象, 请执行以下操作:

1. 输入以下命令:

-  

```
dbclean -object pestats -type all -db dbname -loglevel loglevel
```




-  

```
dbclean -object pestats -type all -db dbname -loglevel loglevel -dbtype
oracle -dbuser user -dbpasswd password
```

+
+
+

-     

```
. dbclean.sh -object pestats -type all -db dbname -dbuser user -loglevel
loglevel
```

-   

```
. dbclean.sh -object pestats -type all -db dbname -loglevel loglevel
-dbtype oracle -dbuser user -dbpasswd password
```

将 *host:port:sid* 用作 Oracle 数据库名称。例如, *myhost:1521:mydb*。



2. 检查 *dbclean_yyyy.mm.dd_hh.mm.ss.zzz.log* 文件来验证命令是否成功。

关于删除产品探测（产品顾问）统计信息对象的附加示例，请参阅第 123 页的『删除对象的示例』。



删除策略对象

要删除策略对象，请执行以下操作：






1. 输入以下命令：

-  




```
dbclean -object policy -type obsolete -db dbname -days daysold -loglevel
loglevel
```

-  

```
dbclean -object policy -type obsolete -db dbname -days daysold -loglevel
loglevel -dbtype oracle -dbuser user -dbpasswd password
```

- +     

```
. dbclean.sh -object policy -type obsolete -db dbname -dbuser user -days
daysold -loglevel loglevel
```

-   

```
. dbclean.sh -object policy -type obsolete -db dbname -days daysold
-loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password
```

将 *host:port:sid* 用作 Oracle 数据库名称。例如, *myhost:1521:mydb*。



2. 检查 *dbclean_yyyy.mm.dd_hh.mm.ss.zzz.log* 文件来验证命令是否成功。

关于删除策略对象的附加示例，请参阅第 123 页的『删除对象的示例』。



删除产品集对象

要删除产品集对象，请执行以下操作：






1. 输入以下命令：

-  

```
dbclean -object product_sets -type obsolete -db dbname -loglevel loglevel
```

-  

```
dbclean -object product_sets -type obsolete -db dbname -loglevel loglevel
-dbtype oracle -dbuser user -dbpasswd password
```

- +     

```
. dbclean.sh -object product_sets -type obsolete -db dbname -dbuser user
-loglevel loglevel
```


-   

```
. dbclean.sh -object product_sets -type obsolete -db dbname -loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password
```

将 `host:port:sid` 用作 Oracle 数据库名称。例如, `myhost:1521:mydb`。

2. 检查 `dbclean_yyyy.mm.dd_hh.mm.ss.zzz.log` 文件来验证命令是否成功。

关于产品集对象的附加示例, 请参阅第 123 页的『删除对象的示例』。

删除报价请求对象

要删除报价请求 (RFQ) 对象, 请执行以下操作:

1. 输入以下命令:

-  

```
dbclean -table rfq -type obsolete -db dbname -loglevel loglevel
```


-  

```
dbclean -object rfq -type obsolete -db dbname -loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password
```

+
+
+

-     

```
. dbclean.sh -object rfq -type obsolete -db dbname -dbuser user -loglevel loglevel
```

-   

```
. dbclean.sh -object rfq -type obsolete -db dbname -loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password
```

将 `host:port:sid` 用作 Oracle 数据库名称。例如, `myhost:1521:mydb`。

2. 检查 `dbclean_yyyy.mm.dd_hh.mm.ss.zzz.log` 文件来验证命令是否成功。

关于删除 RFQ 对象的附加示例, 请参阅第 123 页的『删除对象的示例』。

删除退货商品对象

要删除退货商品对象, 请执行以下操作:

1. 输入以下命令:


-  

```
dbclean -object rma -type typename -db dbname -days daysold -loglevel loglevel
```




-  

```
dbclean -object rma -type typename -db dbname -days daysold -loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password
```

+
+
+

-     

```
. dbclean.sh -object rma -type typename -db dbname -dbuser user -days daysold -loglevel loglevel
```

-   
`. dbclean.sh -object rma -type typename -db dbname -days daysold -loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password`

将 `host:port:sid` 用作 Oracle 数据库名称。例如, `myhost:1521:mydb`。

注: 对于 `-type` 参数, 可以指定 `abandoned` 以表示已放弃的记录、`canceled` 以表示已取消的记录、`not_approved` 以表示已拒绝的记录、`approved_or_partly_approved` 以表示已核准或部分核准的记录或 `completed` 以表示已完成的记录。

2. 检查 `dbclean_yyyy.mm.dd_hh.mm.ss.zzz.log` 文件来验证命令是否成功。



关于删除退货商品对象的附加示例, 请参阅第 123 页的『删除对象的示例』。






删除退货原因对象




要删除客户对商品不满的原因 (或简称为退货原因对象), 请执行以下操作:

1. 输入以下命令:

-  
`dbclean -object rtnreason -type obsolete -db dbname -loglevel loglevel`

-  
`dbclean -object rtnreason -type obsolete -db dbname -loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password`

- +     
`. dbclean.sh -object rtnreason -type obsolete -db dbname -dbuser user -loglevel loglevel`

-   
`. dbclean.sh -object rtnreason -type obsolete -db dbname -loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password`

将 `host:port:sid` 用作 Oracle 数据库名称。例如, `myhost:1521:mydb`。



2. 检查 `dbclean_yyyy.mm.dd_hh.mm.ss.zzz.log` 文件来验证命令是否成功。



关于删除退货原因对象的附加示例, 请参阅第 123 页的『删除对象的示例』。

删除销售辅助统计信息对象









要删除销售辅助 (产品顾问) 统计信息对象, 请执行以下操作:

1. 输入以下命令:

-  
`dbclean -object sastats -type all -db dbname -loglevel loglevel`

-  
`dbclean -object sastats -type all -db dbname -loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password`

+
+
+

-     
 `. dbclean.sh -object sastats -type all -db dbname -dbuser user -loglevel loglevel`
-   
 `. dbclean.sh -object sastats -type all -db dbname -loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password`

将 `host:port:sid` 用作 Oracle 数据库名称。例如, `myhost:1521:mydb`。













2. 检查 `dbclean_yyyy.mm.dd_hh.mm.ss.zzz.log` 文件来验证命令是否成功。

关于删除销售辅助（产品顾问）统计信息对象的附加示例，请参阅第 123 页的『删除对象的示例』。

删除已登台对象

要删除已登台对象，请执行以下操作：

1. 输入以下命令：

-  
 `dbclean -object staglog -type obsolete -db dbname -days daysold -loglevel loglevel`
-  
 `dbclean -object staglog -type obsolete -db dbname -dbuser user -days daysold -loglevel loglevel -dbtype oracle -dbuser <user> -dbpasswd <password>`
-     
 `. dbclean.sh -object staglog -type obsolete -db dbname -dbuser user -days daysold -loglevel loglevel`
-   
 `. dbclean.sh -object staglog -type obsolete -db dbname -days daysold -loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password`

+
+
+

将 `host:port:sid` 用作 Oracle 数据库名称。例如, `myhost:1521:mydb`。



2. 检查 `dbclean_yyyy.mm.dd_hh.mm.ss.zzz.log` 文件来验证命令是否成功。



关于删除已登台对象的附加示例，请参阅第 123 页的『删除对象的示例』。






删除商店对象




要删除商店对象，请执行以下操作：

1. 输入以下命令：

-  
 `dbclean -object store -type specified -db dbname -loglevel loglevel -name storeid -instancexml WC_installdir/instances/INSTANCE_NAME/xml/INSTANCE_NAME.xml`

-  

```
dbclean -object store -type specified -db dbname -loglevel loglevel -dbtype
oracle -dbuser user -dbpasswd password -name storeid -instancexml
WC_installdir/instances/INSTANCE_NAME/xml/INSTANCE_NAME.xml
```
- +     

```
. dbclean.sh -object store -type specified -db dbname -dbuser user
-loglevel loglevel -name storeid -instancexml
WC_installdir/instances/INSTANCE_NAME/xml/INSTANCE_NAME.xml
```
- +   

```
. dbclean.sh -object store -type specified -db dbname -loglevel loglevel
-dbtype oracle -dbuser user -dbpasswd password -name storeid -instancexml
WC_installdir/instances/INSTANCE_NAME/xml/INSTANCE_NAME.xml
```

将 *host:port:sid* 用作 Oracle 数据库名称。例如, *myhost:1521:mydb*。



2. 检查 *dbclean_yyyy.mm.dd_hh.mm.ss.zzz.log* 文件来验证命令是否成功。



关于删除商店对象的附加示例, 请参阅第 123 页的『删除对象的示例』。






删除用户对象




要删除用户对象, 请执行以下操作:

1. 输入以下命令:

-  

```
dbclean -object user -type typename -db dbname -days daysold -loglevel
loglevel -instancexml
WC_installdir/instances/INSTANCE_NAME/xml/INSTANCE_NAME.xml
```
-  

```
dbclean -object user -type typename -db dbname -days daysold -loglevel
loglevel -dbtype oracle -dbuser user -dbpasswd password -instancexml
WC_installdir/instances/INSTANCE_NAME/xml/INSTANCE_NAME.xml
```
- +     

```
. dbclean.sh -object user -type typename -db dbname -dbuser user -days
daysold -loglevel loglevel -instancexml
WC_installdir/instances/INSTANCE_NAME/xml/INSTANCE_NAME.xml
```
- +   

```
. dbclean.sh -object user -type typename -db dbname -days daysold -loglevel
loglevel -dbtype oracle -dbuser user -dbpasswd password -instancexml
WC_installdir/instances/INSTANCE_NAME/xml/INSTANCE_NAME.xml
```

将 *host:port:sid* 用作 Oracle 数据库名称。例如, *myhost:1521:mydb*。

注: 对于 *-type* 参数, 可以指定 *guest* 以表示临时客户或 *registered* 以表示注册客户。

2. 检查 *dbclean_yyyy.mm.dd_hh.mm.ss.zzz.log* 文件来验证命令是否成功。

关于删除用户对象的附加示例, 请参阅第 123 页的『删除对象的示例』。

删除用户通信量日志对象

要删除用户通信量日志对象，请执行以下操作：

1. 输入以下命令：


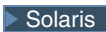
-  

```
dbclean -object usrtraffic -type obsolete -db dbname -days daysold
-loglevel loglevel
```


-  

```
dbclean -object usrtraffic -type obsolete -db dbname -days daysold
-loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password
```

+
+
+

-     

```
. dbclean.sh -object usrtraffic -type obsolete -db dbname -dbuser user
-days daysold -loglevel loglevel
```

-   

```
. dbclean.sh -object usrtraffic -type obsolete -db dbname -days daysold
-loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password
```

将 *host:port:sid* 用作 Oracle 数据库名称。例如，*myhost:1521:mydb*。

2. 检查 *dbclean_yyyy.mm.dd_hh.mm.ss.zzz.log* 文件来验证命令是否成功。

关于删除用户通信量日志对象的附加示例，请参阅第 123 页的『删除对象的示例』。

删除供应商对象

要删除供应商对象，请执行以下操作：

1. 输入以下命令：

-  

```
dbclean -object vendor -type obsolete -db dbname -loglevel loglevel
```

-  

```
dbclean -object vendor -type obsolete -db dbname -loglevel loglevel -dbtype
oracle -dbuser user -dbpasswd password
```

+
+
+

-     

```
. dbclean.sh -object vendor -type obsolete -db dbname -dbuser user
-loglevel loglevel
```

-   

```
. dbclean.sh -object vendor -type obsolete -db dbname -loglevel loglevel
-dbtype oracle -dbuser user -dbpasswd password
```

将 *host:port:sid* 用作 Oracle 数据库名称。例如，*myhost:1521:mydb*。





2. 检查 *dbclean_yyyy.mm.dd_hh.mm.ss.zzz.log* 文件来验证命令是否成功。


关于删除供应商对象的附加示例，请参阅第 123 页的『删除对象的示例』。

添加新配置至数据库清理实用程序

要添加新配置至数据库清理实用程序，请使用以下语法作为参考。例如，对象 o1 由表 R1 组成，该表包含以下列：col1、col2、lastupdate 和 col3。要配置数据库清理实用程序删除从 col1 > 10 且 lastupdate 为 n 天以前的所有对象，请执行以下操作：

DB2

- + 1.     打开 DB2 命令提示符。
- 2. 输入以下命令：

```
db2 insert into cleanconf (objectname, type, statement, namearg, sequence, daysarg) values ('o1', 'obsolete', 'delete from r1 where col1 > 10 and (days(CURRENT TIMESTAMP) - days(lastupdate)) > ?', 'no', 1, 'yes')
```
- 1.  运行以下 SQL 语句：

```
insert into cleanconf (objectname, type, statement, namearg, sequence, daysarg) values ('o1', 'obsolete', 'delete from r1 where col1 > 10 and (days(CURRENT TIMESTAMP) - days(lastupdate)) > ?', 'no', 1, 'yes')
```

Oracle



- 1. 打开 SQLPlus 命令窗口。
- 2. 输入以下命令：



```
insert into cleanconf (objectname, type, statement, namearg, sequence, daysarg) values ('o1', 'obsolete', 'delete from r1 where col1 > 10 and (sysdate - lastupdate) > ?', 'no', 1, 'yes')
```






其中 ? 由下面命令行中的 `-days` 参数替换。'no' 表示在此语句中不使用此名称参数。'yes' 表示在此语句中使用 `-days` 参数。'obsolete' 描述对象 o1 的清理类型。可以使用其它词，但是当调用数据库清理实用程序命令时，在 `-type` 参数中必须使用相同的词。




示例

要调用数据库清理实用程序命令从新表中清除已存在两天的记录，请输入以下命令：

-  

```
dbclean -object o1 -db dbname -type obsolete -days 2 -loglevel 1
```
-  

```
dbclean -object o1 -db dbname -type obsolete -days 2 -loglevel 1 -dbtype oracle -dbuser user -dbpasswd password
```
- +     

```
. dbclean.sh -object o1 -db dbname -dbuser user -type obsolete -days 2 -loglevel 1
```
- +   













```
. dbclean.sh -object o1 -db dbname -type obsolete -days 2 -loglevel 1 -dbtype oracle -dbuser user -dbpasswd password
```

注：对于 Oracle `dbname` 参数，使用 `host:port:sid`。例如，`myhost:1521:mydb`。













删除对象的示例

以下是使用数据库清理实用程序命令的可选参数从数据库表中删除对象的示例。有关详细的参数信息，参阅数据库清理实用程序命令。





示例 1: 要验证哪些表指定删除限制参数，输入以下内容：









-  
`dbclean -object objectname -type typename -db database -check_object_only yes`
-  
`dbclean -object objectname -type typename -db host:port:sid -check_object_only yes -dbtype oracle -dbuser user -dbpasswd <password>`
- +     
+ `. dbclean.sh -object objectname -type typename -db database -dbuser user`
+ `-check_object_only yes`
-   
`. dbclean.sh -object objectname -type typename -db host:port:sid -check_object_only yes -dbtype oracle -dbuser user -dbpasswd password`

示例 2: 要对指定了删除限制参数的表使用 `force` 选项，输入以下内容：

-  
`dbclean -object objectname -type typename -db database -days daysold -loglevel loglevel -force yes`
-  
`dbclean -object objectname -type typename -db host:port:sid -days daysold -loglevel loglevel -force yes -dbtype oracle -dbuser user -dbpasswd <password>`
- +     
+ `. dbclean.sh -object objectname -type typename -db database -dbuser user -days`
+ `daysold -loglevel loglevel -force yes`
-   
`. dbclean.sh -object objectname -type typename -db host:port:sid -days daysold -loglevel loglevel -force yes -dbtype oracle -dbuser user -dbpasswd <password>`

示例 3: 缺省日志文件名始终是 `dbclean_yyyy.mm.dd_hh.mm.ss.zzz.log` 的变体。要指定日志文件名，输入以下内容：

-  
`dbclean -object objectname -type typename -db database -days daysold -loglevel loglevel -log logfile`
-  
`dbclean -object objectname -type typename -db host:port:sid -days daysold -loglevel loglevel -log logfile -dbtype oracle -dbuser user -dbpasswd <password>`

- +
 - +
 - +
 -     
 . dbclean.sh -object *objectname* -type *typename* -db *database* -dbuser *user* -days *daysold* -loglevel *loglevel* -log *logfile*
 -   
 . dbclean.sh -object *objectname* -type *typename* -db *host:port:sid* -days *daysold* -loglevel *loglevel* -log *logfile* -dbtype *oracle* -dbuser *user* -dbpasswd *<password>*

第 7 章 性能

WebSphere Commerce 是许多产品间的复杂交互作用。每个产品都具有自身的性能特征并处于各种组件的交互作用中，在许多场合中性能可能会受不正确的配置或不充足的资源的影响。性能目标包括以及时方式处理以下类型的请求：

- 处理多个客户请求
- 访问 WebSphere Commerce 数据库中的数据
- 将数据格式编排为 Web 页
- 将响应返回到购物者的浏览器

要优化 WebSphere Commerce，请考虑以下组件：

- 硬件
确保您的机器符合《*WebSphere Commerce 安装指南*》中记录的最低机器要求。
在有很多并发用户的生产环境中，多 CPU 将有助于提高性能。使用更快的 CPU 通常将加速大多数操作。
- 数据库
对于 DB2 数据库调整，请参阅红皮书：*DB2/UDB WebSphere Performance Tuning Guide*（可从 <http://publib-b.boulder.ibm.com/Redbooks.nsf/RedbookAbstracts/sg246417.html?open> 获取）。
确保最大数据库连接池大小足以处理所有并发任务（例如，HTTP 连接、调度程序线程等）。
- WebSphere Commerce
确保服务器是 I/O 专用的 - 如果发生大量文件或网络访问，WebSphere Commerce 系统性能可能会受到影响。例如，如果打开所有日志记录和跟踪，系统可能会将大多数时间花在向磁盘写入数据而非处理工作负载上。
如第 71 页的第 5 章，『动态高速缓存』中所述，使用动态高速缓存
当您使用基于服务器的会话管理时，请参阅 *WebSphere Application Server 5.0 Tuning Guide* 中的表示信息。您可在 WebSphere Application Server 信息中心（<http://www.ibm.com/software/webservers/appserv/infocenter.html>）中找到该指南。展开**监视和疑难解答** -> **性能** -> **调整性能**。
- WebSphere Application Server
使用 *WebSphere Application Server 5.0 Tuning Guide* 作为如何调整系统的表示信息。
您可在 WebSphere Application Server 信息中心（<http://www.ibm.com/software/webservers/appserv/infocenter.html>）中找到该指南。展开**监视和疑难解答** -> **性能** -> **调整性能**。
- 其它注意事项：
 - WebSphere Datasource（最小的和最大的连接池大小，语句高速缓存大小）
 - Web 站点设计
 - 安全性（配置、超时、认证和访问控制）
 - Web 服务器问题（进程处理、资源使用、快速响应高速缓存加速键）

- WebSphere 引擎问题（Java 虚拟机或 JVM、传送队列、JSP 文件的高速缓存、EJB 容器）
- WebSphere Commerce 会话管理（将会话高速缓存并存储在内存中或将会话存储在数据库中）
- WebSphere Application Server 会话管理，（设置内存中的会话计数，允许溢出、超时时间间隔和分布式环境设置）。
- NFS（网络文件系统）性能调整（文件服务器调整）

使用 WebSphere Commerce PMI 模块进行性能监视

您可使用 WebSphere Application Server 性能监视基础结构（PMI）监视 WebSphere Commerce 系统的性能。先前的 WebSphere Commerce 性能监视器的功能已转移到 WebSphere Commerce PMI 模块。然后就可监视性能数据并使用 WebSphere Application Server 提供的各种工具进行分析。站点管理员可以使用从各工具收集的信息来检测性能问题并分析性能趋势。可使用各工具从本地或远程机器评测 WebSphere Commerce 应用程序服务器的性能。

WebSphere Commerce 应用程序服务器收集 URL、任务和 JSP 的统计信息。每个数据键都具有一组相关联的计数器，可以提供以下信息：

- 平均响应时间
- 最后响应时间
- 最小响应时间
- 最大响应时间
- 命中率
- 总响应时间
- 标准偏差

要设置 PMI，请参阅 WebSphere Application Server 信息中心

（<http://www.ibm.com/software/webservers/appserv/infocenter.html>）中的“监视性能”部分。

设置 PMI 模块

性能监视基础结构（PMI）使用客户机/服务器体系结构。服务器收集各个 WebSphere Application Server 组件的性能数据。客户机从一个或多个服务器检索性能数据并处理数据。

WebSphere Application Server 服务器收集内存中的 PMI 数据。此数据由计数器（例如 servlet 响应时间和数据连接池利用率）组成。然后使用 Web 客户机、Java 客户机或 JMX 客户机检索数据点。WebSphere Application Server 提供了 Tivoli® 性能查看器，它是一个显示并监视性能数据的 Java 客户机。

为了使用性能监视工具，需要如下设置 WebSphere Commerce PMI 模块：

1. 为了通过 PMI 接口监视性能数据，首先必须通过 WebSphere Application Server 管理控制台启用性能监视服务并重新启动服务器。请参阅 WebSphere Application Server 信息中心

- (<http://www.ibm.com/software/webservers/appserv/infocenter.html>) 中的“Enabling PMI services in application server through the administrative console”部分，以获取详细步骤。
2. 也可通过管理控制台启用 NodeAgent 中的性能监视服务，来通过 PMI 接口监视性能数据。请参阅 WebSphere Application Server 信息中心 (<http://www.ibm.com/software/webservers/appserv/infocenter.html>) 中的“Enabling PMI services in NodeAgent through the administrative console”部分，以获取详细步骤。
 3. 一旦设置了以上内容，您就可收集数据。
监视级别决定了要启用哪些数据计数器，您可动态设置监视级别而不用重新启动服务器。这可使用三种方法之一完成：
 - 通过 WebSphere Application Server 管理控制台启用数据收集。
 - 通过 Tivoli 性能查看器（以前是资源分析器）启用性能监视服务。
 - 使用命令行启用性能监视服务请参阅 WebSphere Application Server 信息中心 (<http://www.ibm.com/software/webservers/appserv/infocenter.html>) 中与以上任务对应的部分，以获取详细步骤。
 4. 监视并分析性能数据。
您可使用几种工具监视和分析数据：
 - a. 使用 Tivoli 性能查看器监视性能数据。此工具包含在 WebSphere Application Server 中。
 - b. 使用其它 Tivoli 监视工具监视性能数据。
 - c. 使用用户开发的监视工具监视性能数据。编写您自己的应用程序来监视性能数据。
 - d. 使用第三方监视工具监视性能。请参阅 WebSphere Application Server 信息中心 (<http://www.ibm.com/software/webservers/appserv/infocenter.html>) 中与以上任务对应的部分，以获取详细信息。

启动 Tivoli 性能查看器

Tivoli 性能查看器是随 WebSphere Application Server 5.0 提供的 WebSphere Application Server 性能监视器。查看器通过周期性地轮询管理服务器检索性能数据。按需要从查看器中不断收集和检索数据。启动性能查看器并指定要从 WebSphere Application Server 管理控制台中收集的数据的级别。使用图形界面来检索和查看表格或图表中的数据，或者在日志文件中存储数据。不仅可使用该查看器监视 Commerce PMI，也可使用它监视与 WebSphere Commerce 相关的其它一般 WebSphere Application Server 性能数据。

关于更多信息，请参阅 WebSphere Application Server 信息中心中的主题“Monitoring performance with Tivoli Performance Viewer (formerly Resource Analyzer)”。

启动 WebSphere Commerce 的 Tivoli 性能查看器时，您需要指定 `WC_instance_name` 应用程序服务器的 SOAP 或 RMI 端口号以测量与 WebSphere Commerce 相关的计数器。否则，缺省情况下，Tivoli 性能查看器将试图连接 `server1` 应用程序服务器。确定

WC_instance_name 应用程序服务器的 SOAP 或 RMI 端口号的一种方法是查看 WC_instance_name 的 System.Out 日志文件。当服务器启动时将显示 SOAP 或 RMI 端口号。

以下是启动 Tivoli 性能查看器的步骤

1. 通过管理控制台启用 PMI 服务。
2. 启动 Tivoli 性能查看器。可用两种方法完成:
 - a. 从命令行启动性能监视。转至 WAS_installdir/bin 目录并运行 tperfviewer 脚本。

在 Windows 2000 环境上, 您可将主机和端口指定为:

```
tperfviewer.bat host_name port_number connector_type
```

在 AIX 和其它 UNIX[®] 平台上, 请使用

```
tperfviewer.sh host_name port_number connector_type
```

例如: tperfviewer.bat localhost 8879 SOAP

connector_type 可以是 SOAP 或 RMI。8879 是 SOAP 连接器的缺省 ND 端口。9809 是 RMI 连接器的缺省 ND 端口。

- b. 单击开始 → 程序 → IBM WebSphere → Application Server V5.0 > Tivoli 性能查看器。

Tivoli 性能查看器将检测您正在使用哪个 WebSphere Application Server 数据包并使用缺省远程方法调用 (RMI) 连接器端口进行连接。如果连接失败, 将显示对话框来提供新的连接参数。通过使用命令行启动性能查看器, 您可连接到远程主机或不同的端口号。

3. 调整数据收集设置。

请参阅 WebSphere Application Server 信息中心

(<http://www.ibm.com/software/webservers/appserv/infocenter.html>) 中的主题 “Setting performance monitoring levels” 中的表示信息。

WebSphere Commerce PMI 模块报告值

启动 Tivoli 性能查看器后, 从 “资源选择” 面板展开 **Commerce 计数器组**。将按照任务、URL 和视图将数据分组到商店标识下。注意仅在 `StoreId=0` 下可找到所有任务命令。一旦从 “资源选择面板” 选择了特定 URL、视图或任务命令, “计数器选择” 面板将显示 WebSphere Commerce 计数器。

它们是对每个任务、URL 或视图包含的计数器, 注意所有时间值都是已经过时间:

计数器名称

定义

平均响应时间

任务的平均响应时间

最后响应时间

任务的最后响应时间。

最小响应时间

任务的最小响应时间。

最大响应时间

任务的最大响应时间。

命中率 调用任务的总次数。

总响应时间

任务的总响应时间。

标准偏差

响应时间的标准偏差。用于计算标准偏差的公式假定数据符合标准分布。

其它性能工具

站点管理员可能还需要使用以下工具:

- IBM Tivoli Web Site Analyzer

通过捕捉、分析、存储和报告 Web 站点的使用率、运行状况、完整性以及站点内容, IBM Tivoli Web Site Analyzer 可清楚地显示访问者站点交互及站点的整体性能。您可利用此信息来优化站点, 以提高客户忠诚度和电子商务有效性。Web Site Analyzer 可针对目标提供或对特定访问者或客户子市场的竞销来跟踪页面内容的受欢迎程度和产品购买。它可表示由于较少访问的 Web 或产品页面, 在 Web 导航中应减少投资或可能要修改的地方。

关于 Web Site Analyzer 的更多信息, 请参阅以下 Web 地址:

<http://www.ibm.com/software/sysmgmt/products/web-site-analyzer.html>

- Commerce Studio Page Detailer

用来分析 Web 页面并显示身份、大小、源和在页面上传送每件商品花费的时间。您可以使用这些详细信息来识别可以改进性能的区域, 从而改善最终用户的体验。您可以运行 Page Detailer 而无需运行其它 Commerce Studio 组件。

维护调度程序

为确保适当的性能, 必须对调度程序进行充分维护。确保通过提供常规维护使调度程序正确运行是站点管理员的职责。

SCHSTATUS 表对 SCHCONFIG 表中作业的每次执行都包含一个条目。由于调度程序运行大量作业, 所以保存在 SCHSTATUS 表中的作业状态记录变得相当庞大。这会对 WebSphere Commerce Server 的性能产生不利影响。所以建议您定期使用“调度程序状态显示”页面清理 SCHSTATUS 表。该页面位于管理控制台的配置菜单下。通过基于时间戳记和作业引用号清理作业, 删除作业状态记录将减小 SCHSTATUS 表的大小。

另外, 您可以调度 CleanJob 作业运行并自动修整 SCHSTATUS 表的大小。使用 endTime 参数指定是否希望除去前一个星期或前一个月的作业状态记录。关于 CleanJob 命令的语法和调度的信息, 请参阅您的平台的 WebSphere Commerce 产品和开发联机帮助。

注: 运行 CleanJob 时, 将从商店发布页面除去商店发布状态。

如果您不需要保留作业历史, 可一直将调度的 autoClean 属性设为“ON”。这样在作业完成执行后, 将自动从 SCHSTATUS 表除去作业。

关于调度程序的更多信息, 请参阅第 21 页的『调度服务』或参阅 WebSphere Commerce 联机帮助。

LDAP 的复制

WebSphere Commerce 支持对认证和存储认证以及概要文件数据使用 LDAP。某些数据可以在 WebSphere Commerce 数据库和 LDAP 服务器之间复制。大部分复制都可以使用 `ldapentry.xml` 文件来配置。与 WebSphere Commerce 的先前版本不同，在先前版本中，复制逻辑只能通过特定的成员子系统命令触发，WebSphere Commerce 的当前版本以对象级别执行复制，这意味着复制可以在需要时随时发生。因此，如果 LDAP 服务器当机，将会生成错误。

测试登台服务器上的站点

要测试登台服务器的站点，请执行以下操作：

1. 配置登台服务器。
2. 为所有定制表创建触发器。
3. 配置远程数据库（如果适用）。
4. 复制数据至登台数据库。
5. 测试站点。
6. 运行登台检查命令以确保没有唯一索引键冲突。
7. 传播数据至生产数据库。
8. 复制文件至生产服务器。
9. 从 STAGLOG 表中删除已登台的对象。

配置登台服务器

任何 WebSphere Commerce 机器都可设置为登台服务器。在安装期间或安装之后可以配置登台服务器。在 WebSphere Commerce 安装期间设置登台服务器的过程在《*WebSphere Commerce 安装指南*》中描述。要在安装之后设置登台服务器，请执行以下操作：

1. 使用配置管理器创建和配置要用作登台服务器的独立 WebSphere Commerce 实例。
2. 确保在数据库面板上选择使用登台服务器复选框，以将实例配置成登台服务器。
3. 确保在配置管理器的高速缓存面板中没有启用高速缓存。

为定制表创建触发器

触发器在 STAGLOG 表中创建一个条目，它标识数据库记录的更改。如果表中包含相同的数据作用域和键特征，则可以对新表修改现有触发器的设置。

如果您还没有创建任何新的表，您不需要执行这一步。如果已经创建了新表，请参阅为定制表配置登台服务器中的表示信息。

配置远程数据库

如果您在生产服务器以外的机器上设置了登台服务器，则必须配置远程数据库。如果您计划从登台服务器中运行登台实用程序，则需要配置生产数据库作为登台服务器中的远程数据库。如果您计划从生产服务器中运行登台实用程序，则需要配置登台数据库作为生产服务器中的远程数据库。

 对于 DB2 数据库，请参阅 *DB2 Administration Guide*。

► Oracle 对于 Oracle 数据库，请参阅产品文档。

复制数据至登台数据库

要将数据从生产数据库复制到登台数据库，请执行以下操作：

1. 设置 PATH 环境变量。
2. 配置数据库。
3. 切换至希望写入日志文件的目录。
4. 输入以下命令：

• Windows DB2

```
stagingcopy -scope _all_ -sourcedb production_database_name  
-destdb staging_database_name
```

• Windows Oracle

```
stagingcopy -scope _all_ -sourcedb production_database_name  
-destdb staging_database_name -dbtype oracle -sourcedb_user  
user -sourcedb_passwd password -destdb_user  
user -destdb_passwd password
```

+
+
+

• AIX 400 Solaris Linux DB2

```
. stagingcopy.sh -scope _all_ -sourcedb production_database_name  
-destdb staging_database_name -sourcedb_user user -destdb_user user
```

• AIX Solaris Oracle

```
. stagingcopy.sh -scope _all_ -sourcedb production_database_name  
-destdb staging_database_name -dbtype oracle -sourcedb_user  
user -sourcedb_passwd password  
-destdb_user user -destdb_passwd password
```

注：对 Oracle 数据库名，请使用 *host:port:sid*。例如，*myhost:1521:mydb*。

5. 检查 `stagingcopy_yyyy.mm.dd_hh.mm.ss.zzz.log` 文件以验证命令是否成功。
6. 停止并重新启动登台服务器实例。

运行登台检查命令

要检查登台数据库与生产数据库之间的潜在的唯一索引键冲突，请执行以下操作：

要运行登台检查命令以确保没有唯一索引键冲突，请执行以下操作：









1. 设置 PATH 环境变量。
2. 配置数据库。
3. 切换至希望写入日志文件的目录。
4. 输入以下命令：

• Windows DB2

```
stagingcheck -scope _unique_index_ -sourcedb staging_database_name  
-destdb production_database_name
```

• Windows Oracle

```
stagingcheck -scope _unique_index_ -sourcedb staging_database_name  
-destdb production_database_name -dbtype oracle -sourcedb_user  
user -sourcedb_passwd password -destdb_user  
user -destdb_passwd password
```

- +
 -     
 - stagingcheck.sh -scope _unique_index_ -sourcedb staging_database_name -destdb production_database_name -sourcedb_user user -destdb_user user
 -   
 - stagingcheck.sh -scope _unique_index_ -sourcedb staging_database_name -destdb production_database_name dbtype oracle -sourcedb_user user -sourcedb_passwd password -destdb_user user -destdb_passwd password



注：对 Oracle 数据库名，请使用 *host:port:sid*。例如，*myhost:1521:mydb*。



5. 检查 stagingcheck_yyyy.mm.dd_hh.mm.ss.zzz.log 文件以验证命令是否成功。

传播数据至生产数据库










要将数据从登台数据库传播到生产数据库，请执行以下操作：

1. 设置 PATH 环境变量。
2. 配置数据库。
3. 切换至希望写入日志文件的目录。
4. 输入以下命令：

-  

```
stagingprop -scope _all_ -sourcedb staging_database_name -destdb
production_database_name
```
-  

```
stagingprop -scope _all_ -sourcedb staging_database_name
-destdb production_database_name -dbtype oracle -sourcedb_user
user -sourcedb_passwd password -destdb_user
user -destdb_passwd password
```

- +
 -      
 - stagingprop.sh -scope _all_ -sourcedb staging_database_name -destdb production_database_name -sourcedb_user user -destdb_user user
 -   
 - stagingprop.sh -scope _all_ -sourcedb staging_database_name -destdb production_database_name dbtype oracle -sourcedb_user user -sourcedb_passwd password -destdb_user user -destdb_passwd password

注：对 Oracle 数据库名，请使用 *host:port:sid*。例如，*myhost:1521:mydb*。

5. 检查 stagingprop_yyyy.mm.dd_hh.mm.ss.zzz.log 文件以验证命令是否成功。

关于传播数据至生产数据库的更多信息，请参阅示例。

复制文件至生产服务器

如果在登台服务器上添加、更改或删除图像或 HTML 文件（与数据库中的数据进行比较），则必须将这些文件手工复制到生产服务器。

要复制文件至生产服务器，请执行以下操作：

1. 将文件复制到生产服务器的正确目录中：

- a. 创建一个压缩文件，其包含所有静态 HTML 文件、关联的图像文件及其它嵌入文件。此文件应该包含新的、更新的和未更改的文件。
 - b. 传送此压缩文件至生产服务器。
 - c. 解压缩每个压缩文件至生产服务器目录结构中相应的目录。如果已经移动了 HTML 文件，请编辑 WebSphere Commerce 配置以指向新目录中的文件。
2. 删除生产服务器上不用的目录。

删除已登台对象

要删除已登台对象，请执行以下操作：

1. 设置 PATH 环境变量。
2. 切换至希望写入日志文件的目录。
3. 输入以下命令：

-  

```
dbclean -object staglog -type obsolete -db dbname
        -days daysold -loglevel loglevel
```



-  

```
dbclean -object staglog -type obsolete -db dbname
        -days daysold -loglevel loglevel -dbtype oracle
        -dbuser user -dbpasswd password
```

+
+
+

-     

```
. dbclean.sh -object staglog -type obsolete -db dbname
        -dbuser user -days daysold -loglevel loglevel
```

-   

```
. dbclean.sh -object staglog -type obsolete -db dbname
        -days daysold -loglevel loglevel -dbtype oracle
        -dbuser user -dbpasswd password
```

注：对 Oracle 数据库名，请使用 *host:port:sid*。例如，*myhost:1521:mydb*。

4. 检查 `dbclean_yyyy.mm.dd_hh.mm.ss.zzz.log` 文件来验证命令是否成功。

关于删除已登台对象的附加示例，请参阅删除对象示例。

WebSphere Commerce Payments 性能调整参数

WebSphere Commerce Payments 包含调整参数，允许您控制 WebSphere Commerce Payments 内部资源分配。

注意： 这些参数只应由对 WebSphere Commerce Payments 非常熟悉的管理员进行修改。若将这些参数值中的任何一个设置得过高，将可能导致性能降级或启动时 WebSphere Commerce Payments 彻底失败。

强烈建议您：

- 仅以小的递增幅度进行更改。
- 每次只更改一个参数；然后在进行另一个更改之前测量一下效果。
- 在生产系统上使用它们之前，在非关键系统上彻底测试所做的更改。

WebSphere Commerce Payments 管理下列线程池集合:

协议线程池

已将处理卡匣中服务器端特定于协议的消息的任务指定给此池中的线程。每个卡匣都有一个这样的池，该池定义了自己的 ComPoints。

400 属性 `wpm.poolsize` 更改协议线程池大小。

通过在 WebSphere Application Server 管理控制台中完成以下步骤可更改池大小

1. 展开服务器。
2. 单击应用程序服务器 > **wpm Commerce Payments Server** > 进程定义 > **Java 虚拟机** > 定制属性 > **wpm.poolsize**。
3. 输入期望值。缺省值为 8。
4. 单击应用。

有了处理通常涉及服务器端协议消息的这一性质（提高了网络通信的级别），这些线程通常为每个给定请求保持很长一段时间。因此，对于依赖协议线程来处理协议消息的卡匣，应为其配置一个足够大的协议线程池，以容纳典型数量的涉及重要协议消息的并发交易。

服务线程池

此池中的线程由卡匣和框架用于在后台执行当前或未来任务。该池中的缺省线程数是 6。

400 属性 `wpm.spoolsize` 更改服务线程池大小。

通过在 WebSphere Application Server 管理控制台中执行以下操作可更改池大小。

1. 展开服务器。
2. 单击应用程序服务器 > **wpm Commerce Payments Server** > 进程定义 > **Java 虚拟机** > 定制属性 > **wpm.spoolsize**。
3. 在值字段中，指定期望的池大小。缺省值是 6。
4. 单击应用。

最初，服务线程最常用于处理调度在未来某一时刻执行的任务。此类任务的示例是使用卡匣后端处理器的重试交易，以及定期维护任务。

wpm.disableDuplicateOrderCheck

当处理新订单时，此参数表示 WebSphere Commerce Payments 忽略重复订单检查并除去对数据库的访问权。这将导致较高的交易吞吐量。仅当您的商家不会生成重复订单号时，才建议使用此参数。这样做会导致这些订单失败。要启用此参数，通过 WebSphere Application Server 管理控制台完成下列步骤:

1. 展开服务器。
2. 单击应用程序服务器 > **wpm Commerce Payments Server** > 进程定义 > **Java 虚拟机** > 定制属性。
3. 单击新建。
4. 在名称字段中，输入参数名称 `wpm.spoolsize`。
5. 输入 **1** 作为值。
6. 单击应用。

技巧: 您的数据库产品可能限制了在给定时间可以建立的到数据库的并发连接数。这可能是技术上的限制, 或者是您所购买数据库产品的许可证协议施加的限制。在调整上述参数时必须考虑到这些因素。如果由于将这些参数中的任何一个设置得过高而超过此类限制, 则 Payment Servlet 初始化将失败。

管理 iSeries 上的 Payments 实例数据库

随着 WebSphere Commerce Payments 处理新的订单和支付, Payments 实例数据库集合将增大。如果 iSeries 的存储空间开始被填满, 系统操作员可能会收到诸如 “CPI099C - 已达到临界存储下限” 这样的警告消息。如果发生了这种情况, 应该考虑释放一些 WebSphere Commerce Payments 实例数据库中的空间。

WebSphere Commerce Payments 实例数据库集合创建时将日志记录设置为 *YES。这意味着实例库将包含数据库日志接收方 (表明何时发生了诸如更改数据库文件之类事件的条目)。在 iSeries 中, 日志 QSQRN 在 WebSphere Commerce Payments 实例库中构建, 名为 QSQRNXXXX 的日志接收方在实例库中创建。如果 WebSphere Commerce Payments 实例库变得过大, 可删除实例库中旧的日志接收方 (名为 QSQRNXXXX, 类型为 *QRNRCV) 以释放空间。最新的日志接收方仍将附加到数据库, 而不应被删除。另外, 使用 IBM 服务提供的 Commerce Payments Pruneorders 实用程序可删改数据库表。

要获取有关清理存储器的更多信息, 请参阅 *AS/400 System Operation* 手册 (SC41-4203)。

第 8 章 疑难解答

站点管理员负责对系统问题进行疑难解答并查找解决方案。本节涵盖了 Crossworlds® 适配器和 WebSphere Commerce Payments 的疑难解答。关于对以下问题进行疑难解答的信息，请参阅《WebSphere Commerce 安装指南》：

- 可下载工具
- 日志文件
- WebSphere Application Server 问题
- Web 服务器问题
- WebSphere Commerce 问题
- 数据库问题

CrossWorlds® 适配器

本节列出了配置适配器时您可能遇到的潜在问题以及解决这些问题的操作。

- **问题：**在一台机器上已有 InterChange Server (ICS) 和资源库，而在另一台机器上有 WebSphere Commerce Server。Visigenic ORB 代理程序需要怎样的配置信息才能跨子网工作？故障现象是当试图进行连接时，弹出一个对话框，表示 WebSphere Commerce Server “无法定位 InterChange Server ‘<server name>’。服务器可能不在运行。”，即使可单独地验证服务器正在运行也是如此。

解决方案 1:

1. 创建用户级别的环境变量，名为 VBROKER_ADM，它指向您的 Visibroker 安装中的 adm 目录（例如 d:\inprise\vbroker\adm）。
2. 在 adm 目录中，创建名为 agentaddr 的文本文件，其中包含需要连接到此 ICS 的客户机的 IP 地址或主机名。
3. 重新启动 osagent 进程，确保 VBROKER_ADM 在启动此进程的用户或外壳程序的环境中。现在应能够将您的 WebSphere Commerce Server 连接到 ICS。注意每个 osagent 必须具有它们自己的 agentaddr 文本文件，该文件包含 WebSphere Commerce Server 的远程 IP。

注：agentaddr 没有扩展名，仅有的应当正在运行的 osagent 是在 ICS 端。

解决方案 2：有人已使用备用环境变量 OSAGENT_ADDR_FILE 更好地解决了此问题，该环境变量应当指定 agentaddr 文件的路径和文件名。

因为 ORB 需要几分钟的发现时间，放弃任何特定方法之前请花十分钟尝试重新连接。

- **问题：**WebSphere Commerce Server 无法连接到多个 InterChange Server。在 WebSphere Commerce Server 机器上设置了 OSAGENT_ADDR 环境变量，但只可以连接到正在指定机器上运行的 InterChange Server。您正在两个不同的子网上运行两个 InterChange Servers 并且 WebSphere Commerce Server 在第三台机器上。每台 InterChange Server 机器上正在运行 osagent 并且每台机器已适当地配置了 VBROKER_ADM 环境变量，使其指向包含 agentaddr 文件的目录，该文件包含每台 InterChange Server/osagent 机器的 IP 地址。

解决方案: 在每台 osagent 机器的 VBROKER_ADM 环境变量中指定的目录中创建 localaddr 文本文件, 并指定其它 osagent 机器的 IP 地址或 DNS 别名。

- **问题:** 试图将 NT 机器的 WebSphere Commerce Server 连接到正在 Solaris 上运行的 ICS。

解决方案: 确保用于 WebSphere Commerce Server 和 ICS 位置的 agentaddr 文件不仅具有正在连接的机器的 IP 地址, 还有这些机器自己的 IP 地址。因此, NT 机器上的 agentaddr 文件必须具有它自己的 IP 地址以及 Solaris 机器的 IP 地址。

注:

1. 关于本节中没有涉及的其他问题, 请联系您的 IBM 支持代表。
2. CrossWorlds[®] 适配器在 iSeries、Linux、IBM eServer[™] zSeries[®] 或 S/390[®] Linux 上不受支持。

注: 关于本节中没有涉及的其他问题, 请联系您的 IBM 支持代表。

WebSphere Commerce Payments

每个业务模型拥有唯一的需求, 并因此以不同的方式利用 WebSphere Commerce Payments 功能。通常, 可以对 WebSphere Commerce Payments 进行调整, 以使其对给定业务环境最大程度地发挥其性能和功能。

调整以获得高性能环境

传输控制协议 / 网际协议 (TCP/IP) 套接字的等待时间

每个对 WebSphere Commerce Payments 的请求都将导致 TCP 套接字进入 TIME_WAIT 状态, 并在此状态保持几分钟。对于为大量请求提供服务的机器, 可能有大量套接字处于 TIME_WAIT 或 TIME_CLOSED 状态, 这就导致请求被拒绝 (即, 产生“无法连接到 WebSphere Commerce Payments”的返回码)。此行为对所有 TCP 连接都是预料中的和必要的。

TCP 套接字进入 TIME_WAIT 状态一段时间, 以确保此套接字上的任何后继通信不会被误认为是新绑定套接字上新建的通信。从理论上讲, 这段时间为 2 MSL (即, 两倍于最大段生命期)。事实上, TIME_WAIT 缺省值在 Windows 操作系统和 Solaris 上为 4 分钟, 而在 AIX 上为 2 分钟。在 iSeries 系统上缺省值大约为 2 分钟。通过更改您操作系统上的 TIME_WAIT 值, 大容量用户可以减少此问题。以下是如何在 Windows、AIX 和 Solaris 操作系统上重新配置时间间隔的示例。

例外: 如果使用备用堆栈, 则可能必须采取其它措施。

Windows:

1. 在注册表中找到:

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\tcpip\Parameters  
\TcpTimeWaitDelay
```

如果您的 Windows 注册表中不存在此条目, 则必须创建它并按照下面的步骤 2 继续。

2. 编辑此条目为新的 DWORD 项。
3. 将它设置为 30 到 300 之间的任意值 (该值表示秒数)。建议将该值设为 30。

AIX:

```
no -o tcp_timewait=1
```

该值（1）以 15 秒递增。例如，1 等于 15 秒，2 等于 30 秒，以此类推。建议您将该值设置为 1 或 2。

Solaris:

```
ndd -set /dev/tcp tcp_close_wait_interval 30000
```

该值（30000）以毫秒为单位。

附录 A. LDAP 方案

LDAP 方案: 作为成员资源库的 LDAP 服务器

一个将 LDAP 服务器用作成员资源库的方案是创建一个 WebSphere Commerce 新实例并指定它将 LDAP 服务器用作成员资源库。在此方案中, 向 WebSphere Commerce 数据库填入根组织、缺省组织和 “wcsadmin” 用户条目的引导程序数据。

对于此方案, 需要完成以下操作:

1. 创建目录服务器中必需的后缀。将由 WebSphere Commerce 使用的用户和组织实体条目存在于这些后缀下。
2. 在目录服务器上创建 “根组织”、“缺省组织” 和 “wcsadmin” 条目, 并使 “缺省组织” 和 “wcsadmin” 位于 “根组织” 之下。关于样本 LDIF 文件, 请参阅下面 “在 LDAP 服务器中创建引导程序条目” 相关链接。
3. 在 LDAP 服务器中对 “wcsadmin” 用户条目创建密码。
4. 在目录服务器中创建条目之后, 您应该以 “根组织” (使用 MEMBER_ID -2001) 和 “缺省组织” (使用 MEMBER_ID -2000) 的正确专有名称更新 ORGENTITY 表的 DN 列。
5. 设置 ldapentry.xml 文件以将 WebSphere Commerce 属性映射为 LDAP 属性。确保在 ldapentry.xml 文件中对用户指定搜索库, 以便可找到 “wcsadmin” 用户。
6. 提供正确密码以 “wcsadmin” 身份登录。

由于通过 WebSphere Commerce 或另一应用程序在目录服务器上创建了更多用户, 因此用户可使用其 RDN 值或 DN 值登录到 WebSphere Commerce。如果用户使用 RDN 登录, 请确保在 ldapentry.xml 文件中正确指定了搜索库。

注意: 如果您创建了 WebSphere Commerce 的多个实例, 并且它们共享同一 LDAP 服务器, 则您应该确保在每个 WebSphere Commerce 实例中的 LDAP 服务器的配置相同。

附录 B. LDAP 文件

本附录提供了以下 LDAP 文件的详细描述:

- ldapmap.dtd
- ldapentry.xml

ldapmap.dtd

ldapentry.dtd 替换了 WebSphere Commerce 先前版本中使用的 ldapmap.dtd 文件。以下是用于 LDAP 的 DTD。不应当修改此 DTD。

```
<!-- new mapping file -->
<!ELEMENT ldapentry (entry+)>
<!ELEMENT entry (ldapsetting,ldapmap)>
<!ELEMENT ldapsetting (ldaprdn,ldapocs,ldapbase)>
<!ELEMENT ldaprdn EMPTY>
<!ELEMENT ldapocs EMPTY>
<!ELEMENT ldapbase EMPTY>
<!ELEMENT ldapmap (map+)>
<!ELEMENT map (objectAttribute+,objectSeparator?,ldapAttribute)>
<!ELEMENT objectAttribute EMPTY>
<!ELEMENT objectSeparator EMPTY>
<!ELEMENT ldapAttribute EMPTY>

<!ATTLIST entry
entryName      (User|Organization|OrganizationalUnit) #REQUIRED>

<!ATTLIST ldapbase
defaultBase    CDATA      #REQUIRED
searchBase     CDATA      #REQUIRED>

<!ATTLIST ldaprdn
rdnName        CDATA      #REQUIRED  keyAttrName    CDATA      #REQUIRED
keyObjName     CDATA      #REQUIRED>

<!ATTLIST ldapocs
objClass       CDATA      #REQUIRED>

<!ATTLIST objectAttribute
attrName       CDATA      #REQUIRED>

<!ATTLIST objectSeparator
attrSeparator  CDATA      #IMPLIED>
<!ATTLIST ldapAttribute
name           CDATA      #REQUIRED
operation      (replace|add)
#REQUIREDflow (ldapToWcs|wcsToLdap|bothDirections) #REQUIRED>
<!-- End of DTD -->
```

ldapentry.xml

ldapentry.xml 文件替换了 WebSphere Commerce 的先前版本中使用的 ldapmap.xml 文件。请确保先前使用的 ldapmap.xml 文件的内容已传送到 ldapentry.xml 文件中。ldapentry.xml 文件为最常用的字段提供了缺省映射。缺省模式映射中的 LDAP 属性是基于缺省用户对象类 (person、organizationalPerson、inetOrgPerson 和ePerson) 和

缺省组织实体对象类 (organization 和 organizationalUnit) 的。这些对象类是为 WebSphere Commerce 支持的所有 LDAP 服务器定义的。要扩展缺省模式映射以超越这些对象类, 请确保首先在 LDAP 服务器上执行模式扩展, 然后将新属性的映射放置在 ldapentry.xml 文件中。建议不要将以下属性保存到 LDAP 中: WebSphere Commerce 中内部生成的键、经常更改的数据, 例如, USERS 表中的 Lastsession。

关于 LDAP 属性的信息, 请参阅以下站点: www.as400.ibm.com/ldap/schema。

```
<?xml version="1.0"?>
<!DOCTYPE ldapentry SYSTEM "ldapentry.dtd">
<ldapentry>
  <entry entryName="User">
    <ldapsetting>

<ldaprdn rdnName="uid"
  keyAttrName="logonId" keyObjName="UserRegistry"/>
    <ldapocs
objClass="top;person;organizationalPerson;inetOrgPerson"/>
    <ldapbase
defaultBase="o=Default Organization,o=Root Organization"
searchBase="o=Root Organization"/>
    </ldapsetting>
    <ldapmap>
      <map>

        <
objectAttribute attrName="logonPassword"/>

<ldapAttribute name="userPassword"
  operation="replace" flow="wcsToLdap"/>
      </map>
    <map>

<objectAttribute attrName="lastName"/>

<objectAttribute attrName="firstName"/>

<objectSeparator attrSeparator="/">

<ldapAttribute name="cn" operation="replace"
  flow="wcsToLdap"/>

    </map>
  <map>

    <objectAttribute attrName="lastName"/>

    <ldapAttribute name="sn" operation="replace"
" flow="bothDirections"/>

    </map>
    <map>

    <objectAttribute attrName="firstName"/>

    <ldapAttribute name="givenName" operation="replace
```

```

" flow="bothDirections"/>
    </map>
  <map>

  <objectAttribute attrName="phone1"/>

  <ldapAttribute name="homePhone" operation="replace"
" flow="bothDirections"/>
    </map>
  <map>

  <objectAttribute attrName="zipCode"/>

  <ldapAttribute name="postalCode" operation="replace"
" flow="bothDirections"/>
    </map>
  <map>

  <objectAttribute attrName="address1"/>

  <objectAttribute attrName="address2"/>

  <objectAttribute attrName="address3"/>

  <objectSeparator attrSeparator="/"/>

<ldapAttribute name="postalAddress" operation="replace"
" flow="bothDirections"/>
  </map>
</ldapmap>
</entry>

entry entryName="Organization">
  <ldapsetting>

  <ldaprdn rdnName="o" keyAttrName="orgEntityName"
keyObjName="Organization"/>
  <
  ldapocs objClass="top;organization"/>
  <
  ldapbase defaultBase="o=Root Organization" searchBase="
o=Root Organization"/>
  </ldapsetting>
  <ldapmap>
  <map>

  <objectAttribute attrName="businessCategory"/>

  <ldapAttribute name="businessCategory" operation="
replace" flow="bothDirections"/>
    </map>
  <map>

  <objectAttribute attrName="description"/>

  <ldapAttribute name="description" operation="
replace" flow="bothDirections"/>
    </map>
  <map>

  <objectAttribute attrName="address1"/>

  <objectAttribute attrName="address2"/>

  <objectAttribute attrName="address3"/>

```

```

<objectSeparator attrSeparator="/" />

<ldapAttribute name="postalAddress" operation="replace
" flow="bothDirections"/>
    </map>
    <map>

<objectAttribute attrName="phone1"/>

<ldapAttribute name="telephoneNumber" operation="replace
" flow="bothDirections"/>
    </map>
</ldapmap>
</entry>
    <entry entryName=
"OrganizationalUnit">
    <ldapsetting>

        <ldaprdn rdnName="ou" keyAttrName="orgEntityName"
keyObjName="Organization"/>

<ldapocs objClass="top;organizationalUnit"/>

<ldapbase defaultBase="o=Root Organization"
searchBase="o=Root Organization"/>
    </ldapsetting>

    <ldapmap>
        <map>

<objectAttribute attrName="businessCategory"/>

<ldapAttribute name="businessCategory" operation="replace
" flow="bothDirections"/>
    </map>
    <map>

<objectAttribute attrName="description"/>

<ldapAttribute name="description" operation="replac
e" flow="bothDirections"/>
    </map>
    <map>

<objectAttribute attrName="address1"/>

<objectAttribute attrName="address2"/>

<objectAttribute attrName="address3"/>

<objectSeparator attrSeparator="/" />

<ldapAttribute name="postalAddress" operation="replace
" flow="bothDirections"/>
    </map>
    <map>

<objectAttribute attrName="phone1"/>

<ldapAttribute name="telephoneNumber" operation="replace
" flow="bothDirections"/>
    </map>
</ldapmap>
</entry>
</ldapentry>

```

ldapentry.xml 文件的格式如下:

entry entry 元素标识了映射用于哪种成员类型。有效值为 User、Organization 或 OrganizationalUnit。

ldapsetting - ldaprDN - rdnName

指定哪个 LDAP 属性是 RDN 属性

ldapsetting - ldaprDN - keyAttrName

指定哪个 WebSphere Commerce 属性映射为 RDN 属性。对于用户，此属性名称对应 UserRegistrationAdd 命令语法中记载的属性名。此属性名区分大小写。同样情况适用于组织和组织单位，在此情况下应当使用 OrgEntityAdd 命令。

ldapsetting - ldaprDN - keyObjName

指定在哪个 WebSphere Commerce 访问 bean 中可找到在 keyAttrName 中指定的 WebSphere Commerce 属性

ldapsetting - ldapocs - objClass

指定用于创建 LDAP 条目的 LDAP 对象类

ldapsetting - ldapbase - defaultBase

指定在 RDN 用于登录情况下，登录期间 WebSphere Commerce 将在其中搜索用户的首个搜索库 DN。对于组织实体，defaultBase 指定每当需要时 WebSphere Commerce 将在其中搜索组织实体的首个搜索库 DN。

ldapsetting - ldapbase - searchBase

指定 WebSphere Commerce 将在其中搜索用户和组织实体的其余库 DN。

map -objectAttribute - attrName

WebSphere Commerce 属性名称。对于用户，此属性名称对应 UserRegistrationAdd 命令语法中记载的属性名。此属性名区分大小写。同样情况适用于组织和组织单位，在此情况下应当使用 OrgEntityAdd 命令。

map -ldapAttribute - name

要映射为 attrName 中指定的 WebSphere Commerce 属性的 LDAP 属性的名称。

map -ldapAttribute - flow

指定属性值是在 LDAP 中读出、写入，还是同时读出与写入。有效值为 ldapToWcs、wcsToLdap 或 bothDirections。

map -ldapAttribute - operation

指定是否应对 LDAP 修改属性值以及应如何修改它们。有效值为 replace 或 add。值 replace 以所提供的新信息更新当前用户信息（例如，以新的电话号码替换现有电话号码）。值 add 则包含用户的另一条目（例如，向当前用户信息添加新电话号码、为用户创建电话号码列表）。

map - objectSeparator - attrSeparator

向单个 LDAP 属性存储或从单个 LDAP 属性检索多个 WebSphere Commerce 属性时使用的分隔符。

附录 C. WebSphere Commerce Payments 教程

本教程指导您完成使用 OfflineCard 卡匣首次设置 WebSphere Commerce Payments 的整个过程。作为此初始设置的一部分，并且为了演示最常见的管理和支付功能，WebSphere Commerce Payments 使用 OfflineCard 卡匣和结帐样本提供教程支持。关于管理、配置和支付功能的详细信息，请参阅 WebSphere Commerce Payments 用户界面的联机帮助。

以下是 6 个必须完成的配置任务来使用 OfflineCard 卡匣设置可操作的 WebSphere Commerce Payments:

1. 访问 WebSphere Commerce Payments 用户界面
2. 创建 WebSphere Commerce Payments 商家并授权支付卡匣
3. 定义 WebSphere Commerce Payments 用户
4. 指定用户角色
5. 创建帐户
6. 创建品牌

一旦配置任务完成，您就准备好下订单并开始通常商家每日都要执行的以下支付处理任务。

7. 创建订单
8. 核准订单
9. 圈出支付金额
10. 结算批量
11. 发出返款
12. 查看每日批量总量

步骤 1: 访问 WebSphere Commerce Payments 用户界面

第一步是以 Payments 管理员登录 Payments。

要登录 WebSphere Commerce Payments 用户界面，请执行以下操作:

1. 在 Web 浏览器中，指向 `http://host_name:port/webapp/PaymentManager`，其中 `host_name` 是正在运行 Payments Web 服务器的主机名，而 `port` 是指 Payments 正在其上运行的端口号。
2. 输入您的 WebSphere Commerce 站点管理员标识。
3. 输入您的 WebSphere Commerce 站点管理员密码。
4. 单击**登录**。

重要事项: 如果 WebSphere Commerce Payments 实例使用的 HTTP 服务器配置为缺省值之外的端口号，请在本教程通篇 WebSphere Commerce Payments Web 地址链接中的主机名后包含端口号。

用户界面右上方的图标有以下用处:

- 单击多方向箭头刷新该页面。
- 单击左箭头返回访问的上一页。
- 单击问号访问该页面的上下文敏感联机帮助。

步骤 2: 创建 WebSphere Commerce Payments 商家并授权卡匣

如果您还没有登录，请以 Payments 管理员登录 WebSphere Commerce Payments。现在您具有全局视图和全局权限。配置 WebSphere Commerce Payments 的第一步是创建商家以及授权该商家使用支付卡匣。请执行以下操作创建商家并授权卡匣：

1. 从导航框架单击**商家设置**。
2. 从“商家设置”页面单击**添加商家**。
3. 在“商家设置”页面上，输入以下信息（注意：在教程的这些字段中必须输入等宽字体的文本）：

表 11. 创建商家字段

字段名	描述
商家名称	输入“测试商店”。这是您指定给商家的名称。它唯一的功能就是在用户界面中显示。
商家号	输入 123456789。这是您指定在所有交易数据中唯一标识商家的号码。
已授权的卡匣	选中 OfflineCard 框。选中此框授权商家使用此支付卡匣。

4. 单击**创建商家**保存商家配置。

如果已创建了要授权使用此卡匣的商家，请执行以下步骤：

1. 单击**商家设置**。
2. 单击“商家名称”。
3. 选择 **OfflineCard** 框。
4. 单击**创建商家**。

已授权商家使用此卡匣。

步骤 3: 定义 WebSphere Commerce Payments 用户

对于本教程，您将处理以下用户：

- 安装期间创建的缺省用户（关于更多信息，请参阅《WebSphere Commerce 安装指南》）。
- *Pat*，您将定义的用户

使用 WebSphere Commerce 组织管理控制台来完成诸如定义和管理用户的任务。在 WebSphere Commerce Payments 中定义用户是一个两部分过程。例如，要定义用户 *Pat*，您必须使用 WebSphere Commerce 组织管理控制台并将 *Pat* 指定为站点管理员角色。然后，您可直接在 Payments UI 中或通过组织管理控制台指定 *Pat* 的用户角色为商家管理员。注意可将访问权指定给用户之前，您必须创建商家。

要配置 Payments 用户，请执行以下操作：

1. 在 Web 浏览器中，指向 `https://host_name:port/adminconsole`

2. 单击访问管理 > 用户。
3. 单击新建。
4. 使用“新建用户”向导创建新用户 Pat。
5. 从“访问”页面，指定 Pat 站点管理员的角色。
6. 单击 **Payments > 用户**。
7. 输入用户名 Pat。
8. 单击搜索。
9. 单击 Pat 的名称。将显示该用户的配置页面。
10. 选择期望的商家名称。
11. 选择期望角色的单选按钮，在此例中为商家管理员。
12. 单击更新保存您的规范。

如果您正在对 WebSphere Commerce 产品而言为远程的系统上安装 WebSphere Commerce Payments 并希望您的 WebSphere Commerce Payments 用户界面中使用 WebSphere Commerce 样式表，则您必须将 `Payments_installdir\samples\wcs\PMCustomUI.properties` 目录中的 `PMCustomUI.properties` 文件复制到主 WebSphere Commerce Payments 安装目录。

步骤 4: 指定用户角色

必须将以下 WebSphere Commerce Payments 角色之一指定给用户，这些角色对于对应的 WebSphere Commerce 角色有相关映射：

表 12. 角色映射

Payments 角色	WebSphere Commerce 角色
Payments 管理员	站点管理员
商家管理员	站点管理员
主管	运作或销售经理
职员	客户服务主管

创建以下用户后：

- 用户，*Pat*
- 商家，*测试商店*

现在您已准备好在 WebSphere Commerce Payments 配置中指定 Pat 的角色。

例外：您还可指定无 *WebSphere Commerce Payments* 访问权角色来拒绝用户访问 WebSphere Commerce Payments。如果临时需要拒绝用户访问（例如，缺席的某个雇员），这可能是很有用的。要获取关于 WebSphere Commerce Payments 角色许可权的更多信息，请参阅 *WebSphere Commerce Payments Programming Guide and Reference* 中的角色许可权表。

要将“测试商店”的商家管理员角色指定给 Pat，请在 Payments UI 中执行以下操作：

1. 从导航框架单击用户。
2. 在“用户搜索”页面，输入用户名 Pat 并单击搜索。
3. 从“用户”页面，单击用户名 **Pat**。

4. 从商家滚动框，选择测试商店。
5. 选择商家管理员单选按钮。
6. 单击更新保存用户配置。

此时，从 WebSphere Commerce Payments 用户界面注销并再次登录，这次作为商家管理员 Pat 登录。

重要事项：在贸易服务提供者（CSP）建立远程服务于多个商家的 WebSphere Commerce Payments 的主管环境中，该 CSP 是 Payments 管理员并按照每个商家与提供者的合同操作。在此方案中，商家使用 CSP（Payments 管理员）授予他们的商家管理员权限配置其自己的商家设置。

以商家管理员登录

要注销并再次登录，请执行以下操作：

1. 从导航框架中，单击**注销 admin** 返回主 WebSphere Commerce Payments 登录页面。
2. 输入用户标识 Pat。
3. 输入为 Pat 定义的密码。
4. 单击**确定**。

对于本教程的剩余部分，您的角色将为用户 Pat，具有“测试商店”的商家管理员权限。您的 WebSphere Commerce Payments 用户界面视图将限于商家管理功能；而作为 Payments 管理员，您拥有商家和 Payments 管理员功能的全局视图。

步骤 5: 创建帐户

到现在为止，您已定义了一个商家（“测试商店”）并启用了—个支付卡匣（OfflineCard 卡匣）。您作为商家管理员的第一个任务是为 OfflineCard 卡匣建立帐户。

帐户是商家和金融机构之间的关系，它为商家处理交易。每个支付卡匣可以有多个帐户。对于本教程，您将为 OfflineCard 卡匣创建一个帐户。

要创建帐户，请执行以下操作：

1. 从导航框架单击**商家设置**。
2. 从“商家设置”页面，单击“测试商店”的 OfflineCard 卡匣图标。
3. 从“OfflineCard 卡匣”页面，单击**帐户**。
4. 在“帐户”页面上，单击**添加帐户**。
5. 完成以下字段（注意必须为教程输入等宽字体的文本）：

表 13. 添加帐户字段

字段名	描述
帐户名称	输入“测试帐户”。这是您为帐户指定的名称。它唯一的功能就是在用户界面中显示。
帐号	类型111111111。这是主管服务供应商或商家管理员指定的、在所有交易数据中唯一标识帐户的号码。
金融机构名称	输入“测试银行”。这是您持有其帐户的金融机构的名称。它唯一的功能就是在用户界面中提供显示信息。

表 13. 添加帐户字段 (续)

字段名	描述
货币	选择此帐户收的货币。
批量关闭时间	超过午夜的分钟数，在此时间之后卡匣将自动关闭此帐户的批量。0（零）值表示午夜。1439 是允许的最大值。空值禁用自动批量关闭。

6. 单击**创建帐户**创建 OfflineCard 卡匣的帐户。

指定帐户设置

要使用本教程，并不一定要为您刚刚为卡匣创建的帐户和品牌指定任何特定的基本或高级设置。然而，您应该了解可以为该帐户指定某些选项。例如，作为商家管理员，您可以定义帐户是否应该具有与支付核准关联的失效时段。

要从 WebSphere Commerce Payments 用户界面查看 OfflineCard 卡匣的帐户设置，执行下列操作：

1. 在导航框架上选择**商家设置**。
2. 从“商家设置”窗口中，单击“测试商店”窗口中的 OfflineCard 卡匣图标。
3. 从 OfflineCard 卡匣窗口中，单击**帐户**。
4. 在“帐户”窗口上选择**测试帐户**。
5. 从“测试帐户”窗口中，选择**帐户设置**。
6. 通过单击相应的选择，您可以查看或更改基本或高级设置。例如，选择**高级设置**以查看“自动支付处理”选项的“核准失效”设置。
7. 输入相应的选项并选择**更新**，以更新帐户设置。（要取消，在“帐户设置”窗口上选择“上一步”箭头。）

您可以参阅联机帮助，以获取关于处理选项设置的更多信息。然而，现在我们将探讨它们中的一个 - “核准失效”选项。

您可能要设置核准失效以控制支付核准有效的时间长度。它可能有助于设置核准失效时段，以避免您的金融机构对在核准失效之后圈出资金征收费用。例如，对于信用卡订购，支付核准仅持续信用卡发行商建立的某几天。如果您在核准失效后尝试为订单支付圈出资金，则您的金融机构可以拒绝资金圈出或收取更多的费用以作为押金。通过在 WebSphere Commerce Payments 中设置核准失效时段您可以避免这种情况，因为 WebSphere Commerce Payments 会阻止对核准授权已失效的资金进行划款。

如果支付核准已失效并且您试图圈出资金，您将接收到消息，表明支付不在核准状态中。在 WebSphere Commerce Payments 用户界面中，您可以使 0 金额支付无效。一旦支付为无效，您可以执行核准操作以将支付放回到核准状态。

要使用核准失效选项，您使用的卡匣必须支持核准失效。随 WebSphere Commerce Payments 提供的 OfflineCard、VisaNet、Paymentech 和 CustomOffline 卡匣支持核准失效。

重要事项： 您输入的作为核准失效时段的值应用于与帐户关联的所有品牌。如果您的品牌每个都具有不同支付核准失效规则，则将核准失效值设置为最小公分母，或为每个品牌创建独立的帐户。如果为每个品牌创建独立的帐户，则您必须为每个帐户单独结算批量支付。

有关核准订单、圈出支付金额和结算批量的更多信息在本章的稍后部分提供。

步骤 6: 创建品牌

要使用 OfflineCard 卡匣，您首先必须配置您的商店使用的信用卡公司的品牌名称。这样使购物者可以在结帐期间从列表中选择适当的信用卡。要为 OfflineCard 卡匣创建品牌，请执行以下操作（注意对于本教程，将 ROBO 用作品牌）：

1. 从导航框架单击**商家设置**。
2. 从“商家设置”页面，单击“测试商店”的 OfflineCard 卡匣图标。
3. 从“OfflineCard 卡匣”页面，单击**帐户**。
4. 从“帐户”页面单击帐户名称“Offline 帐户”。
5. 从“Offline 帐户”页面，单击**品牌**。
6. 从“品牌”页面，单击**添加品牌**。
7. 在**品牌名称**字段中，输入 ROBO。
8. 单击**创建品牌**。

步骤 7: 使用样本结帐创建订单

作为商家管理员，您拥有全局商家权限，这表示您可执行以下操作：

- 特定于商家的管理功能
- 所有支付处理功能

在实际业务方案中，您可将支付处理任务授权给其它商家定义的用户，这些用户拥有受限的支付处理权限（例如主管和职员）。在本教程中，作为商家管理员的您将执行这些任务。完成了为开始支付处理所需的所有 WebSphere Commerce Payments 和商家管理任务之后，现在您已准备好开始：

- 核准订单
- 圈出支付金额
- 结算批量
- 发出返款
- 查看每日批量总量

对于本教程，您将使用样本结帐工具为支付处理创建三个订单。样本结帐工具提供了一个用户界面，您可用它创建样本订单来测试您的卡匣实现。要访问样本结帐，您首先必须更改缺省用户标识和密码，如下所述。

要访问 WebSphere Commerce Payments 样本结帐并创建订单，请执行以下操作：

1. 从目录：

```
WAS_installdir/installedApps/host_name/  
payments_instance_Commerce_Payments_App.ear/SampleCheckout.war
```

打开配置文件 SampleCheckout.xml。

2. 在 SampleCheckout 元素处，更改以下属性值：

```
pmHostName="fully_qualified_host_name "  
pmPort="port "  
default_userid="wc_userid "  
password="wc_password "
```

- 保存文件。使您的浏览器指向 `http://host_name:port/webapp/SampleCheckout`，其中 `host_name` 是正在运行 Payments Web 服务器的机器的主机名，而 `port` 是指 Payments 正在其上运行的端口号。
- 在“样本结帐”页面输入以下信息（注意：在教程的这些字段中必须输入等宽字体的文本）：

表 14. 样本结帐字段

名称	描述
商家号	输入 OfflineCard 样本结帐商家号 987654321。
订单号	输入表示订单号的任何数字。
金额	输入表示订单总金额数的数量。
货币	输入美元。下此订单使用的货币。
支付方式	选择 OfflineCard 作为支付类型。
品牌	选择希望使用的 OfflineCard 品牌。
信用卡号	输入 4111111111111111。
失效日期	为您的信用卡选择任何未来失效月份和年份。

▶ 400 对于 iSeries 系统，必须指定您正在使用的信用卡品牌。

- 单击**购买**。

再重复这些步骤两次，这样您就有了三个要处理其支付的订单。

步骤 8: 核准订单

一旦使用样本结帐创建了三个订单，您可核准这些订单。遵循这些步骤核准订单：

- 再次使您的浏览器指向 `http://host_name:port/webapp/PaymentManager`，其中 `host_name` 是正在运行 Payments Web 服务器的机器的主机名，而 `port` 是指 Payments 正在其上运行的端口号，并以 Pat 登录。
- 从导航框架单击**核准**。
- 从“核准”页面选择您所下订单之一的框。
- 单击**核准选定批量**。“核准结果”页面显示了您的核准请求状态。
- 完成核准后，单击**返回到核准屏幕**。

两个订单仍在等待核准。通过从“核准”页面单击**全部核准**，您可同时核准它们的全部金额。但是，为了更好地演示核准功能，此教程描述了如何单独地处理每个订单。

从订单页面核准订单

在此部分中，您将从“订单”页面核准订单的总订单金额的一部分。您会发现订单处理期间某些与此订单相关联的商品不可用于交付时（例如，延迟交货的商品），仅核准订单的一部分是很有用的。

- 从“核准”页面，单击所剩的等待核准的订单之一的**订单号**。
- 从“订单号”页面中，您可查看订单详细信息。单击**核准**，以核准此订单。

3. “订单号核准”页面显示了以下字段:

表 15. 订单号核准字段

字段名	描述
货币	下此订单使用的货币类型。这是一个只读字段。
订单金额	以用于下订单的货币表示的订单的总金额。这是一个只读字段。
已核准的金额	由于还没有核准订单金额，此只读字段显示为 0。
已圈出的金额	由于还没有核准或圈出金额，所以此只读字段显示为 0。
核准金额	这是订单金额总数。
授权代码	从手工脱机授权请求过程返回的授权代码。支付状态将更改为“已核准”。
拒绝原因	从手工脱机授权过程返回的拒绝原因。支付状态将更改为“已拒绝”。
AVS 结果代码	地址验证系统结果代码。

将核准金额更改为 3.00。可选地，指定授权原因来核准金额或指定拒绝原因表示已拒绝核准。

4. 单击**核准**对此订单核准 3 美元。完成核准处理之后，“订单”页面刷新并显示核准状态。

使用核收功能核准订单

因为您核准了处理的最后一个订单的一部分，所以在“核准”页面中仍有两个订单条目。在此步骤中，您将使用核收功能核准剩余的订单。

核收功能允许您核准订单并使之直接进入已圈出状态，而绕过核准状态。核收功能自动核准并圈出您的订单支付。因此，您可将核收视为带有自动圈出的核准。使用核收功能可加快对买方的货品交付并保证资金划款（例如，当销售可下载的软件或电子信息时）。但是，使用核收功能，您不能设置核准的授权原因或拒绝原因。因为核收将核准和圈出合并为一个交易，因此当按每个交易收费时，核收功能也是非常有用的。

请执行以下操作，使用核收功能核准订单：

1. 从导航框架单击**核准**。
2. 从“核准”页面单击**全部核收**。处理完成后，将显示提交以供核收的每个订单的核准状态。
3. 完成核收后，单击**返回到核准屏幕**。

步骤 9: 圈出支付金额

圈出功能允许您圈出订单支付。单一订单号可具有多个与之关联的支付。您可以看到在同一列表中，同一订单号多次出现，而每次出现都具有不同的支付信息。

要圈出支付金额，请执行以下操作：

1. 从导航框架单击**圈出**。
2. 选取所列出的某项支付的框并单击**圈出选定订单**。处理完成后，将显示提交以供圈出的支付的圈出状态。
3. 完成圈出后，单击**返回到圈出屏幕**。

您还可圈出支付金额的一部分。要圈出支付金额的一部分，请执行以下操作：

1. 从“圈出”页面中，单击等待圈出的某个支付的**支付号**。
2. 从“支付”页面单击**圈出**。
3. 在“订单支付”页面，将圈出金额更改为 2.00 并单击**圈出**。

步骤 10: 结算批量

批量是支付和返款的集合，金融机构将其作为一个单位进行处理。一个批量是与一个商家和一个帐户相关联的。前面的练习中圈出的支付将在一个批量中出现。您必须结算此批量以启动金融机构的处理。一旦完成结算，金融机构负责资金转移。

要结算批量，请执行以下操作：

1. 从导航框架单击**批量搜索**。您还可选择单击**结算**。
2. 在“批量搜索”页面中，您可输入以下信息来缩小搜索范围（注意对于本教程，您不需要完成这些字段）：

表 16. 批量搜索字段

字段名	描述
商家	您要搜索其批量的商家的名称。如果 WebSphere Commerce Payments 数据库中的商家数少于 500，则从下拉列表选择商家名称。如果 WebSphere Commerce Payments 数据库中的商家数多于 500，则输入商家名称。
批号	唯一标识商家中的批量的号码。支付已圈出时指定该批号。
状态	批量的状态： <ul style="list-style-type: none">• 打开• 已结束
结算状态	此批量的结算状态： <ul style="list-style-type: none">• 已结算 已成功结算的批量，即所有总额一致。• 结算失败 对此批量作出的结算尝试不成功，即所有总额不一致。
支付类型	标识用于下订单的支付类型或协议，例如 OfflineCard。
批量打开日期	使用“此后”和“此前”字段搜索在指定时间范围内打开的批量： <ul style="list-style-type: none">• 此后 指定一个日期，搜索在此日期和此日期之后打开的所有批量。• 此前 指定一个日期，搜索在此日期和此日期之前打开的所有批量。
批量结束日期	使用“此前”和“此后”字段搜索在指定时间范围内关闭的批量： <ul style="list-style-type: none">• 此后 指定一个日期，搜索在此日期和此日期之后关闭的所有批量。• 此前 指定一个日期，搜索在此日期和此日期之前关闭的所有批量。
帐户	正在处理的此订单的帐户。如果已定义了多于 500 个帐户，则在输入字段中输入帐号。

3. 单击**搜索**。

技巧：您还可使用“此前”和“此后”字段从搜索中排除某些批量来缩小搜索结果范围。例如，您可搜索在 08/01/2003 之前打开并在 08/15/2003 之后关闭的所有批量，这样就排除了在 08/02/2003 和 08/14/2003 之间打开的批量

4. 单击批号查看关于批量的信息。
5. 单击**批量详细信息**以查看此批量中所有支付和返款的详细信息列表。

6. 单击**结算**进行批量结算。处理完成后，在“结算结果”页面中显示结算状态。

要修剪过时信息，您可单击“结算结果”页面上的**删除**来删除已结算批量。删除批量后，关于此批量的所有附属信息（即支付、返款和特定于卡匣的数据）也将被删除。如果您需要保留所有支付数据（例如，出于审计目的），则您不应删除批量。

步骤 11: 发出返款

可对订单发出返款并可给定任意金额（最大到订单总额）。

要发出返款，请执行以下操作：

1. 为了找到要发出返款的订单，从导航框架单击**订单搜索**。
2. 在“订单搜索”页面中，您可输入以下内容（注意对于本教程，您不需要完成这些字段）：

表 17. 订单搜索字段

字段名	描述
商家	您要搜索其订单的商家的名称。如果 WebSphere Commerce Payments 数据库中的商家数少于 500，则从下拉列表选择商家名称。如果 WebSphere Commerce Payments 数据库中的商家数多于 500，则输入商家名称。
订单号	由商家指定的唯一标识订单的号码。
状态	订单的状态： <ul style="list-style-type: none">• 已请求• 已订购• 可返款• 已取消• 已结束
支付类型	标识用于下订单的支付类型或协议，例如 OfflineCard。
订单日期	使用下面的“此后”和“此前”字段搜索在指定时间范围内打开的订单： <ul style="list-style-type: none">• 此后 指定一个日期，搜索在此日期和此日期之后打开的所有订单。• 此前 指定一个日期，搜索在此日期和此日期之前打开的所有订单。
订单金额	<ul style="list-style-type: none">• 货币 用来下此订单的货币。从下拉列表中选择货币类型。• 大于 指定一个值，检索订单金额大于或等于指定值的所有订单。• 小于 指定一个值，检索订单金额小于或等于指定值的所有订单。
帐户	正在处理的此订单的帐户。如果已定义了多于 500 个帐户，则在输入字段中输入帐号。

3. 单击**搜索**。
4. 从“订单搜索结果”页面，单击处于“可返款”状态的订单的订单号来查看该订单的详细信息。
5. 从“订单”页面，单击**返款**以创建对此订单的返款。

6. “创建返款” 页面显示以下信息:

表 18. 创建返款字段

字段名	描述
货币	下此订单使用的货币类型。这是一个只读字段。
订单金额	以用于下订单的货币表示的订单的总金额。这是一个只读字段。
已核准的金额	已核准的订单总额，以下订单所使用的货币表示。这是一个只读字段。
已圈出的金额	已圈出的订单总额，以下订单所使用的货币表示。这是一个只读字段。
返款金额	商家管理员必须使用返回给购物者的总金额填写此字段。
授权原因	从手工脱机授权请求过程返回的授权代码。返款状态更改为“已退款”。
拒绝原因	从手工脱机授权过程返回的拒绝原因。返款状态更改为“已拒绝”。

输入返款金额并单击**返款**。

返款处理完成后，“订单” 页面刷新并显示返款状态。新创建的返款显示在**返款**下。

步骤 12: 查看批量总量

本教程的最后一个步骤是查看每日批量总量。WebSphere Commerce Payments 报表功能允许您查看关闭状态下的批量**每日总量**。

要生成每日批量总量报表，请执行以下操作:

1. 从导航框架单击**报表**。
2. 从“报表” 页面单击**每日批量总量**。
3. 在“批量总量报表” 页面中，输入希望用于批量总量报表的**日期**。保留此字段为空，则生成当日的报表。
4. 输入或选择**商家名称**。如果不输入商家名称，则将显示指定日期的所有批量的列表。如果存在多于 500 个批量，将仅显示前 500 个批量。
5. 单击**搜索**。

“每日批量总量” 报表计算**搜索**页面中所指定的日期关闭的所有批量的合计。因为您没有指定搜索日期，所以生成的报表仅包括当日的批量总量。对这些总量的计算是基于每个货币，因此每个货币有一行。注意这些总量包括了为所有支付类型创建的所有支付和返款，不仅仅是通过 OfflineCard 卡匣创建的。

您已经完成了作为 Payments 管理员和商家管理员一天的工作。虽然各个业务模型都可能有所不同，但是本教程概括了建立工作 WebSphere Commerce Payments 的基本途径，并演示了通过 WebSphere Commerce Payments 实现的基本支付处理。关于 WebSphere Commerce Payments 用户界面中特定字段的更多信息，请参阅联机帮助。

附录 D. WebSphere Commerce Payments 的本地语言支持 (NLS) 信息

本章提供了关于 WebSphere Commerce Payments 的本地语言支持 (NLS) 的补充信息。若您要安装 WebSphere Commerce Payments 的本地语言版本 (NLV)，则复查此信息非常重要。

NLS 提示和技巧

Adobe Acrobat Reader 的局限

在 AIX 平台上，对于某些语言，可能无法正确显示 PDF 文件。如果您无法查看或打印 WebSphere Commerce Payments PDF 文件（即，文档文件），则您可在 Windows[®] 或 Windows 2000 操作系统上从浏览器查看并打印文件。这是 AIX Adobe Acrobat Reader 的局限性。

代码页 如果没有正确地显示西班牙语的字符，请确保您使用的代码页为 ISO-8859-1。

WebSphere Commerce Payments 用户名

WebSphere Commerce Payments 仅支持用下表中的字符组成的用户名。无法访问包含该集合以外字符的用户名。

该表列出了用于指定 WebSphere Commerce Payments 用户名的支持字符集。

表 19. WebSphere Commerce Payments 用户名及其它的可移植字符集 (PCS)

ASCII 十六进制值	20	21	22	23	24	25	26	27	28	29	2A	2B	2C	2D	2E	2F
字符	空格	!	"	#	\$	%	&	'	()	*	+	,	_	.	/
ASCII 十六进制值	30	31	32	33	34	35	36	37	38	39	3A	3B	3C	3D	3E	3F
字符	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
ASCII 十六进制值	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F
字符	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
ASCII 十六进制值	50	51	52	53	54	55	56	57	58	59	5A	5B	5C	5D	5E	5F
字符	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
ASCII 十六进制值	60	61	62	63	64	65	66	67	68	69	6A	6B	6C	6D	6E	6F
字符	'	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
ASCII 十六进制值	70	71	72	73	74	75	76	77	78	79	7A	7B	7C	7D	7E	
字符	p	q	r	s	t	u	v	w	x	y	z	{		}	~	

受支持的语言和语言环境

下表列出了 WebSphere Commerce Payments 支持的每种语言及对应的语言环境。

表 20. *WebSphere Commerce Payments* 上受支持的语言和语言环境

语言	语言环境
巴西葡萄牙语	pt_BR
英语	en_US
法语	fr_FR
德语	de_DE
意大利语	it_IT
日语	Ja_JP
韩国语	ko_KR
简体中文	zh_CN
西班牙语	es_ES
繁体中文	Zh_TW

声明

Note to U.S. Government Users — — Documentation relating to restricted rights — — Use, duplication, or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

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.

有关双字节（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-chrome, Minato-ku
Tokyo 1061, 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；这些更改将编入本出版物的新版本中。IBM 可以随时对本出版物中描述的产品和 / 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：

Lab Director
IBM Canada Ltd. Laboratory
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。

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 没有对这些产品进行测试，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。

本信息包含日常业务运作中使用的数据和报表的示例。为尽可能表述完整，示例包括个人、公司、品牌和产品的名称。所有这些名称都是虚构的，任何与实际业务企业使用的名称和地址类似名称和地址纯属巧合。

本文档包含有关其它公司产品的信息，包括对这些公司因特网站点的引用。IBM 对这些信息的精确性、完整性和使用不承担任何责任。

本产品基于 SET 协议。

版权许可：

本信息包括源语言形式的样本应用程序，这些样本说明不同操作平台上的编程方法。如果是为按照在编写样本程序的操作平台上的应用程序编程接口（API）进行应用程序的开发、使用、经销或分发为目的，您可以任何形式对这些样本程序进行复制、修改、分发，而无须向 IBM 付费。这些示例并未在所有条件下作全面测试。因此，IBM 不能担保或暗示这些程序的可靠性、可维护性或功能。用户如果是为了按照 IBM 应用程序编程接口开发、使用、经销或分发应用程序，则可以任何形式复制、修改和分发这些样本程序，而无须向 IBM 付费。

这些样本程序的每份副本或任何部分或任何衍生产品必须包含如下版权声明：

(C) (贵公司的名称) (年份)。此部分代码是根据 IBM 公司的样本程序衍生出来的。(C) Copyright IBM Corp. 1996, 2003. All rights reserved.

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：

- AIX

- CrossWorlds
- DB2 Extenders™
- DB2 Universal Database™
- IBM iSeries
- Intelligent Miner™
- Lotus®
- OS/400
- pSeries™
- S/390
- VisualAge®
- WebSphere
- xSeries®
- zSeries
- Redbooks
- SecureWay
- DFS
- VisualAge
- DB2
- AS/400

Solaris、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。

Microsoft®、Windows 和 Windows 徽标是 Microsoft Corporation 在美国和 / 或其它国家或地区的注册商标。

SET 和 SET 徽标是 SET Secure Electronic Transaction™ LLC 拥有的商标。

UNIX 是 The Open Group 在美国和其它国家或地区的注册商标。

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