

IBM WebSphere Commerce



# 新时尚样本商店：联机帮助文件

版本 5.4



IBM WebSphere Commerce



# 新时尚样本商店：联机帮助文件

版本 5.4

**注意！**

在使用本资料及其支持的产品之前，请务必阅读『声明』中的一般信息。

# 目录

<b>第 1 章 使用样本创建商店</b>	<b>1</b>
使用样本创建商店	1
创建样本商店归档文件	1
更改商店数据库有用资源	1
更改 Web 有用资源	13
从商店服务发布样本商店归档文件	13
使用命令行发布商店归档文件 (Windows)	15
设置商店的 Payment Manager	17
为样本商店创建调度作业	18
为样本商店配置电子邮件通知	19
<b>第 2 章 样本商店数据库有用资源</b>	<b>23</b>
商店数据库有用资源	23
新时尚产品目录数据库有用资源	24
“新时尚”合同和商务策略数据库有用资源	26
新时尚装运数据库有用资源	26
样本商店访问控制数据库有用资源	28
<b>第 3 章 新时尚样本商店</b>	<b>31</b>
“新时尚”样本商店	31
<b>第 4 章 新时尚商店页面</b>	<b>33</b>
新时尚样本商店搜索页面	33
样本商店通讯录页面	34
新时尚样本商店通讯录页面	36
网络时尚和新时尚样本商店捆绑销售商品显示页	38
样本商店产品目录组页面	39
样本商店页面: 公共实现技术	41
样本商店的与我们联系页面	43
样本商店页脚	43
样本商店页眉	45
样本商店帮助页面	46
样本商店主页	46
样本商店登录页面	47
样本商店隐私页面	50
样本商店产品页面	50
样本商店注册页面	52
样本商店新品页面	54
新时尚样本商店订单确认页面	55
网络时尚和新时尚电子邮件通知页	55
样本商店订单摘要页面	56

“新时尚”样本商店检查产品可用性页面	57
网络时尚和新时尚样本商店打包销售商品显示页	58
样本商店左侧导航框架	60
新时尚样本商店左侧导航框架	61
样本商店选择开票地址页面	63
“新时尚”样本商店选择开票地址页面	64
样本商店购物车	65
新时尚样本商店购物车	67
样本商店装运方式页面	68
新时尚样本商店选择装运方式页面	69
样本商店选择送货地址页面	70
新时尚样本商店选择送货地址页面	71
样本商店快速结帐摘要页面	72
“网络时尚”和“新时尚”样本商店查看订单页面	73
网络时尚和新时尚样本商店希望列表页面	75
样本商店使用案例	78

<b>第 5 章 样本商店使用案例</b>	<b>79</b>
添加新地址使用案例 (商务版)	79
网络时尚和新时尚向希望列表添加商品使用案例	80
查看产品类别使用案例	80
购物车结帐使用案例	81
网络时尚和新时尚显示捆绑销售商品页面使用案例	84
网络时尚和新时尚显示打包销售商品页面使用案例	84
编辑地址使用案例 (商务版)	85
主页使用案例	86
登录使用案例	87
管理个人帐户使用案例	88
网络时尚和新时尚查看订单使用案例	89
显示产品页面使用案例	89
更改个人信息使用案例	90
注册使用案例	91
显示购物车使用案例	92
“网络时尚”和“新时尚”查看希望列表使用案例	93

<b>第 6 章 新时尚商店购物流程</b>	<b>95</b>
新时尚样本商店购物车流程	95
新时尚样本商店购物流程	95

<b>声明</b>	<b>97</b>
-----------	-----------




---

## 第 1 章 使用样本创建商店

---

### 使用样本创建商店

要使用样本商店其中之一创建商店，请执行以下操作：

1. （可选）  创建担任卖方的组织。  
要确定您是否希望创建一个新的组织来担任卖方，请参阅共享的数据有用资源。
2. 为以下角色创建用户：
  - 站点管理员（如果不在使用缺省的站点管理员）
  - 卖方管理员
  - 商店管理员
  - 商店开发者**重要信息：**要发布商店归档文件，您必须具有站点管理员、商店管理员或所有商店的商店开发者访问权限。  
如果正在创建具有商店管理员权限的用户，请确保访问组适用于所有的商店。
3. 使用商店服务创建商店归档文件
4. 更改商店数据库有用资源。
5. 更改 Web 有用资源
6. 发布商店归档文件。
7. （可选）配置商店。
8. 设置商店的 Payment Manager。

#### 重要信息：

1. 在某些实例中，样本商店在所有功能部件能正常工作前需要一些设置。如果您的商店正基于样本商店，您可能需要完成设置。关于更多信息，请参阅设置样本商店。
2. 某些商店数据有用资源（例如产品目录和实现中心）是在商店之间共享的。因此，如果您根据相同的样本商店发布多个商店，则对每个商店而言，产品目录和实现中心将是相同的。如果您在一个产品目录中做出更改，则发布基于同一个样本的另一个商店将覆盖这些更改。有关如何避免覆盖任何更改的更多信息，请参阅共享的数据有用资源。

---

### 创建样本商店归档文件

要用“商店服务”创建可以用作样本的商店归档文件，请参阅《*IBM WebSphere 商店开发者指南*》。

---

### 更改商店数据库有用资源

如果使用商店服务中的工具创建商店归档文件，则新的商店归档文件一开始就包含与您基于的样本商店归档文件（例如 infashion.sar）相同的商店数据库有用资源。在商店归档文件中，商店数据库有用资源采取 XML 文件的形式。

大多数情况下，要更改商店数据库有用资源，您必须直接编辑 XML 文件。在某些情况下，您可以使用商店服务中的工具来编辑数据库有用资源。

您可以选择直接编辑数据库有用资源，即一旦将商店归档文件发布到了贸易服务器，就可以使用 WebSphere 贸易加速器、装入程序软件包或直接通过 SQL 插入，来编辑数据库。如果选择编辑数据库而不是编辑商店归档文件中的有用资源，则必须更新商店归档文件以与数据库中的更改相匹配，或停止使用该商店归档文件。

下表中列出了选项。

**重要信息：**

1. “商店服务”中的工具将搜索在下表中列出的有用资源名称。因此，如果您希望使用“商店服务”中的工具编辑商店归档文件，必须使用与商店归档文件中相同的有用资源。
2. 当向 WebSphere Commerce Server 发布商店归档文件时，数据库信息将按照以下有用资源列中指定的顺序装入。因此，在 sarinfo.xml 文件中指定的有用资源的顺序应当与以下指定的有用资源顺序相匹配。
3. 在每个 XML 文件中的数据库信息的顺序不一定必须与以下“数据库表”列中指定的顺序相匹配。然而，父表中的信息必须在子表中的信息之前。
4. 创建正在营运的商店不需要那些标记为可选的信息。
5. 对于“数据库编辑选项”列，请注意，除非另行指定，否则所有数据库有用资源都可使用 SQL 插入或装入程序软件包进行编辑。因此，该列反映那些有用资源可以由 WebSphere 贸易加速器编辑。

商店数据库有用资源	数据库表	商店归档文件编辑选项	数据库编辑选项
实现	FFMCENTER (0...1)	<ul style="list-style-type: none"> <li>• 编辑商店归档文件 XML 文件。</li> </ul>	<ul style="list-style-type: none"> <li>• WebSphere 贸易加速器中的“新建实现中心”和“更改实现中心”页面。</li> <li>• 请参阅创建实现中心和更改实现中心</li> </ul>



商店数据库有用资源	数据库表	商店归档文件编辑选项	数据库编辑选项
商店	STOREENT	<ul style="list-style-type: none"> <li>• 编辑商店归档文件 XML 文件。</li> <li>• 使用“商店简要表”笔记本更改商店信息。“商店简要表”笔记本编辑以下数据库列： <ul style="list-style-type: none"> <li>– IDENTIFIER</li> <li>– DIRECTORY</li> <li>– SETCURR</li> </ul> </li> </ul>	
	STADDRESS	<ul style="list-style-type: none"> <li>• 使用“商店简要表”笔记本更改商店信息。</li> </ul>	<ul style="list-style-type: none"> <li>• WebSphere 贸易加速器中的“新建实现中心”和“更改实现中心”页面。</li> <li>• 请参阅创建实现中心和更改实现中心</li> </ul>
	STOREENTDS	<ul style="list-style-type: none"> <li>• 使用“商店简要表”笔记本更改商店信息。“商店简要表”笔记本编辑以下数据库列： <ul style="list-style-type: none"> <li>– DESCRIPTION</li> <li>– DISPLAYNAME</li> </ul> </li> </ul>	
	STORELANG	<ul style="list-style-type: none"> <li>• 使用“商店简要表”笔记本更改商店信息。“商店简要表”笔记本编辑以下数据库列： <ul style="list-style-type: none"> <li>– LANGUAGE_ID</li> </ul> </li> </ul>	
	STORELANGDS	<ul style="list-style-type: none"> <li>• 编辑商店归档文件 XML 文件。</li> </ul>	
	STORE	<ul style="list-style-type: none"> <li>• 使用“商店简要表”笔记本更改商店信息。“商店简要表”笔记本编辑以下数据库列： <ul style="list-style-type: none"> <li>– PHONE1</li> <li>– CITY</li> <li>– STORE_ID</li> <li>– COUNTRY</li> <li>– STATE</li> <li>– EMAIL1</li> <li>– ADDRESS1</li> <li>– ADDRESS2</li> <li>– FAX</li> <li>– ZIPCODE</li> </ul> </li> </ul>	
	DISPENTREL (缺省 CATENTRY 模板, CATENTRY_ID = -1)	<ul style="list-style-type: none"> <li>• 使用“商店简要表”笔记本更改商店信息。“商店简要表”笔记本编辑 PAGENAME 数据库列。</li> </ul>	
	DISPCGPREL (缺省 CATEGORY 模板, CATGROUP_ID = -1)	<ul style="list-style-type: none"> <li>• 使用“商店简要表”笔记本更改商店信息。“商店简要表”笔记本编辑 PAGENAME 数据库列。</li> </ul>	
	VENDOR	<ul style="list-style-type: none"> <li>• 编辑商店归档文件 XML 文件。</li> </ul>	

商店数据库有用资源	数据库表	商店归档文件编辑选项	数据库编辑选项
产品目录	CATGROUP	<ul style="list-style-type: none"> <li>更改产品目录信息</li> <li>编辑商店归档文件 XML 文件。</li> </ul>	<ul style="list-style-type: none"> <li>WebSphere 贸易加速器中的产品管理工具</li> </ul>
	CATGRPATTR	<ul style="list-style-type: none"> <li>更改产品目录信息</li> <li>编辑商店归档文件 XML 文件。</li> </ul>	
	CATGRPDSC	<ul style="list-style-type: none"> <li>更改产品目录信息</li> <li>编辑商店归档文件 XML 文件。</li> </ul>	<ul style="list-style-type: none"> <li>WebSphere 贸易加速器中的产品管理工具</li> </ul>
	CATALOG	<ul style="list-style-type: none"> <li>更改产品目录信息</li> <li>编辑商店归档文件 XML 文件。</li> </ul>	
	CATALOGDSC	<ul style="list-style-type: none"> <li>更改产品目录信息</li> <li>编辑商店归档文件 XML 文件。</li> </ul>	
	CATTOGRP	<ul style="list-style-type: none"> <li>更改产品目录信息</li> <li>编辑商店归档文件 XML 文件。</li> </ul>	
	CATGRPREL	<ul style="list-style-type: none"> <li>更改产品目录信息</li> <li>编辑商店归档文件 XML 文件。</li> </ul>	<ul style="list-style-type: none"> <li>WebSphere 贸易加速器中的产品管理工具</li> </ul>
	CATENTRY	<ul style="list-style-type: none"> <li>更改产品目录信息</li> <li>编辑商店归档文件 XML 文件。</li> </ul>	<ul style="list-style-type: none"> <li>WebSphere 贸易加速器中的产品管理工具</li> </ul>
	CATENTDSC	<ul style="list-style-type: none"> <li>更改产品目录信息</li> <li>编辑商店归档文件 XML 文件。</li> </ul>	<ul style="list-style-type: none"> <li>WebSphere 贸易加速器中的产品管理工具</li> </ul>
	ATTRIBUTE	<ul style="list-style-type: none"> <li>更改产品目录信息</li> <li>编辑商店归档文件 XML 文件。</li> </ul>	<ul style="list-style-type: none"> <li>WebSphere 贸易加速器中的产品管理工具</li> </ul>
	ATTRVALUE	<ul style="list-style-type: none"> <li>更改产品目录信息</li> <li>编辑商店归档文件 XML 文件。</li> </ul>	<ul style="list-style-type: none"> <li>WebSphere 贸易加速器中的产品管理工具</li> </ul>
	CATGPENREL	<ul style="list-style-type: none"> <li>更改产品目录信息</li> <li>编辑商店归档文件 XML 文件。</li> </ul>	<ul style="list-style-type: none"> <li>WebSphere 贸易加速器中的产品管理工具</li> </ul>
	CATENTREL	<ul style="list-style-type: none"> <li>更改产品目录信息</li> <li>编辑商店归档文件 XML 文件。</li> </ul>	
	BASEITEM	<ul style="list-style-type: none"> <li>更改产品目录信息</li> <li>编辑商店归档文件 XML 文件。</li> </ul>	<ul style="list-style-type: none"> <li>WebSphere 贸易加速器中的产品管理工具</li> </ul>
	ITEMSPC	<ul style="list-style-type: none"> <li>更改产品目录信息</li> <li>编辑商店归档文件 XML 文件。</li> </ul>	<ul style="list-style-type: none"> <li>WebSphere 贸易加速器中的产品管理工具</li> </ul>
	VERSIONSPC	<ul style="list-style-type: none"> <li>更改产品目录信息</li> <li>编辑商店归档文件 XML 文件。</li> </ul>	
DISTARRANG	<ul style="list-style-type: none"> <li>更改产品目录信息</li> <li>编辑商店归档文件 XML 文件。</li> </ul>		

商店数据库有用资源	数据库表	商店归档文件编辑选项	数据库编辑选项
产品目录	RECEIPT	<ul style="list-style-type: none"> <li>• 更改产品目录信息</li> <li>• 编辑商店归档文件 XML 文件。</li> </ul>	
	RCPTAVAIL	<ul style="list-style-type: none"> <li>• 更改产品目录信息</li> <li>• 编辑商店归档文件 XML 文件。</li> </ul>	
	STOREITEM	<ul style="list-style-type: none"> <li>• 更改产品目录信息</li> <li>• 编辑商店归档文件 XML 文件。</li> </ul>	<ul style="list-style-type: none"> <li>• WebSphere 贸易加速器中的产品管理工具</li> </ul>
税	JURSTGROUP (在税和装运费之间公用)	<ul style="list-style-type: none"> <li>• 编辑商店归档文件 XML 文件。</li> <li>• 使用“税款”笔记本更改税设置。“税款”笔记本编辑以下数据库列: <ul style="list-style-type: none"> <li>- JURSTGROUP_ID</li> <li>- DESCRIPTION</li> <li>- SUBCLASS</li> <li>- STOREENT_ID</li> <li>- CODE</li> </ul> </li> </ul>	
	JURST (在税和装运费之间公用)	<ul style="list-style-type: none"> <li>• 使用“税款”笔记本更改税设置。“税款”笔记本编辑以下数据库列: <ul style="list-style-type: none"> <li>- JURST_ID</li> <li>- COUNTRY</li> <li>- STOREENT_ID</li> <li>- CODE</li> <li>- SUBCLASS</li> <li>- STATE</li> </ul> </li> </ul>	
	JURSTGPREL (在税和装运费之间公用)	<ul style="list-style-type: none"> <li>• 使用“税款”笔记本更改税设置。“税款”笔记本编辑以下数据库列: <ul style="list-style-type: none"> <li>- JURST_ID</li> <li>- JURSTGROUP_ID</li> <li>- SUBCLASS</li> </ul> </li> </ul>	
	CALMETHOD (在税和装运费之间公用)	<ul style="list-style-type: none"> <li>• 编辑商店归档文件 XML 文件。</li> </ul>	
	TXCDCLASS (可选, 仅用于对 CALCODE 分类, 由 WebSphere 贸易加速器使用)	<ul style="list-style-type: none"> <li>• 编辑商店归档文件 XML 文件。</li> </ul>	
	TAXCGRY	<ul style="list-style-type: none"> <li>• 使用“税款”笔记本更改税设置。“税款”笔记本编辑以下数据库列: <ul style="list-style-type: none"> <li>- TAXCGRY_ID</li> <li>- STOREENT_ID</li> <li>- NAME</li> </ul> </li> </ul>	

商店数据库有用资源	数据库表	商店归档文件编辑选项	数据库编辑选项
税	CALCODE (在税和装运费用之间公用)	<ul style="list-style-type: none"> <li>使用“税款”笔记本更改税设置。“税款”笔记本编辑以下数据库列: <ul style="list-style-type: none"> <li>- CALCODE_ID</li> <li>- CODE</li> <li>- CALUSAGE_ID</li> <li>- STOREENT_ID</li> <li>- GROUPBY</li> <li>- CALMETHOD_id</li> <li>- CALMETHOD_id_app</li> <li>- CALMETHOD_id_qfy</li> </ul> </li> </ul>	
	CALCODEDSC (在税和装运费用之间公用, 可选)	<ul style="list-style-type: none"> <li>编辑商店归档文件 XML 文件。</li> </ul>	
	CALRULE (在税和装运费用之间公用)	<ul style="list-style-type: none"> <li>使用“税款”笔记本更改税设置。“税款”笔记本编辑以下数据库列: <ul style="list-style-type: none"> <li>- CALRULE_ID</li> <li>- CALCODE_ID</li> <li>- TAXCGRY_ID</li> <li>- CALMETHOD_ID</li> </ul> </li> </ul>	
	CALSCALE (在税和装运费用之间公用)	<ul style="list-style-type: none"> <li>使用“税款”笔记本更改税设置。“税款”笔记本编辑以下数据库列: <ul style="list-style-type: none"> <li>- CALSCALE_ID</li> <li>- CODE</li> <li>- STOREENT_ID</li> <li>- CALUSAGE_ID</li> <li>- SETCURR</li> <li>- CALMETHOD_ID</li> </ul> </li> </ul>	
	CALSCALEDS (在税和装运费用之间公用)	<ul style="list-style-type: none"> <li>编辑商店归档文件 XML 文件。</li> </ul>	
	CALRANGE (在税和装运费用之间公用)	<ul style="list-style-type: none"> <li>使用“税款”笔记本更改税设置。“税款”笔记本编辑以下数据库列: <ul style="list-style-type: none"> <li>- CALRANGE_ID</li> <li>- CALSCALE_ID</li> <li>- CALMETHOD_ID</li> <li>- RANGESTART</li> </ul> </li> </ul>	

商店数据库有用资源	数据库表	商店归档文件编辑选项	数据库编辑选项
税	CALRLOOKUP (在税和装运费用之间公用)	<ul style="list-style-type: none"> <li>使用“税款”笔记本更改税设置。“税款”笔记本编辑以下数据库列:               <ul style="list-style-type: none"> <li>- CALSCALE_ID</li> <li>- CALRULE_ID</li> </ul> </li> </ul>	
	CRULESCALE (在税和装运费用之间公用)	<ul style="list-style-type: none"> <li>使用“税款”笔记本更改税设置。“税款”笔记本编辑以下数据库列:               <ul style="list-style-type: none"> <li>- CALSCALE_ID</li> <li>- CALRULE_ID</li> </ul> </li> </ul>	
	CALRULEMGP (可选)	<ul style="list-style-type: none"> <li>编辑商店归档文件 XML 文件。</li> </ul>	
	CALCOTXEX (免税, 用于折扣)	<ul style="list-style-type: none"> <li>编辑商店归档文件 XML 文件。</li> </ul>	
	STENCALUSG (公共, 商店的缺省 CALCODE, 有两个条目: 一个表示销售税, 另一个表示装运税)	<ul style="list-style-type: none"> <li>使用“税款”笔记本更改税设置。</li> </ul>	
税实现	TAXJCRULE	<ul style="list-style-type: none"> <li>编辑商店归档文件 XML 文件。</li> <li>使用“税款”笔记本更改税设置。“税款”笔记本编辑以下数据库列:               <ul style="list-style-type: none"> <li>- CALRULE_ID</li> <li>- FFMCENTER_ID</li> <li>- JURSTGROUP_ID</li> </ul> </li> </ul>	
商店产品目录 - 税	CATENCALCD (可选)	<ul style="list-style-type: none"> <li>编辑商店归档文件 XML 文件。</li> </ul>	<ul style="list-style-type: none"> <li>WebSphere 贸易加速器中的产品管理工具</li> </ul>
	CATGPCALCD (可选)	<ul style="list-style-type: none"> <li>编辑商店归档文件 XML 文件。</li> </ul>	

商店数据库有用资源	数据库表	商店归档文件编辑选项	数据库编辑选项
装运	JURSTGROUP (在税和装运费用之间公用)	<ul style="list-style-type: none"> <li>• 编辑商店归档文件 XML 文件。</li> <li>• 使用“装运”笔记本更改装运设置。 “装运”笔记本编辑以下数据库列:               <ul style="list-style-type: none"> <li>- JURSTGROUP_ID</li> <li>- DESCRIPTION</li> <li>- SUBCLASS</li> <li>- STOREENT_ID</li> <li>- CODE</li> </ul> </li> </ul>	
	JURST (在税和装运费用之间公用)	<ul style="list-style-type: none"> <li>• 使用“装运”笔记本更改装运设置。“装运”笔记本编辑以下数据库列:               <ul style="list-style-type: none"> <li>- JURST_ID</li> <li>- COUNTRY</li> <li>- STOREENT_ID</li> <li>- CODE</li> <li>- SUBCLASS</li> <li>- STATE</li> </ul> </li> </ul>	
	JURSTGPREL (在税和装运费用之间公用)	<ul style="list-style-type: none"> <li>• 使用“装运”笔记本更改装运设置。“装运”笔记本编辑以下数据库列:               <ul style="list-style-type: none"> <li>- JURST_ID</li> <li>- JURSTGROUP_ID</li> <li>- SUBCLASS</li> </ul> </li> </ul>	
	SHIPMODE	<ul style="list-style-type: none"> <li>• 使用“装运”笔记本更改装运设置。“装运”笔记本编辑以下数据库列:               <ul style="list-style-type: none"> <li>- CODE</li> <li>- CARRIER</li> <li>- SHIPMODE_ID</li> </ul> </li> </ul>	
	SHPMODEDSC	<ul style="list-style-type: none"> <li>• 使用“装运”笔记本更改装运设置。“装运”笔记本编辑以下数据库列:               <ul style="list-style-type: none"> <li>- SHIPMODE_ID</li> <li>- LANGUAGE_ID</li> </ul> </li> </ul>	
	CALMETHOD (在税和装运费用之间公用)	<ul style="list-style-type: none"> <li>• 编辑商店归档文件 XML 文件。</li> </ul>	

商店数据库有用资源	数据库表	商店归档文件编辑选项	数据库编辑选项
装运	CALCODE (在税和装运费用之间公用)	<ul style="list-style-type: none"> <li>使用“装运”笔记本更改装运设置。“装运”笔记本编辑以下数据库列: <ul style="list-style-type: none"> <li>- CALCODE_ID</li> <li>- CODE</li> <li>- CALUSAGE_ID</li> <li>- STOREENT_ID</li> <li>- GROUPBY</li> <li>- CALMETHOD_ID</li> </ul> </li> </ul>	
	CALCODEDSC (可选)	<ul style="list-style-type: none"> <li>使用“装运”笔记本更改装运设置。</li> </ul>	
	CALRULE (在税和装运费用之间公用)	<ul style="list-style-type: none"> <li>使用“装运”笔记本更改装运设置。“装运”笔记本编辑以下数据库列: <ul style="list-style-type: none"> <li>- CALRULE_ID</li> <li>- CALCODE_ID</li> <li>- TAXCGRY_ID</li> <li>- CALMETHOD_ID</li> </ul> </li> </ul>	
	CALSCALE (在税和装运费用之间公用)	<ul style="list-style-type: none"> <li>使用“装运”笔记本更改装运设置。“装运”笔记本编辑以下数据库列: <ul style="list-style-type: none"> <li>- CALSCALE_ID</li> <li>- CODE</li> <li>- STOREENT_ID</li> <li>- CALUSAGE_ID</li> <li>- SETCURR</li> <li>- CALMETHOD_ID</li> </ul> </li> </ul>	
	CALRULEMGP (可选)	<ul style="list-style-type: none"> <li>编辑商店归档文件 XML 文件。</li> </ul>	
	CALSCALEDS (在税和装运费用之间公用)	<ul style="list-style-type: none"> <li>编辑商店归档文件 XML 文件。</li> </ul>	

商店数据库有用资源	数据库表	商店归档文件编辑选项	数据库编辑选项
装运	CALRANGE (在税和装运费用之间公用)	<ul style="list-style-type: none"> <li>使用“装运”笔记本更改装运设置。“装运”笔记本编辑以下数据库列: <ul style="list-style-type: none"> <li>CALRANGE_ID</li> <li>CALSCALE_ID</li> <li>CALMETHOD_ID</li> </ul> </li> </ul>	
	CALRLOOKUP (在税和装运费用之间公用)	<ul style="list-style-type: none"> <li>使用“装运”笔记本更改装运设置。“装运”笔记本编辑以下数据库列: <ul style="list-style-type: none"> <li>CALRLOOKUP_ID</li> <li>SETCURR</li> <li>CALRANGE_ID</li> <li>VALUE</li> </ul> </li> </ul>	
	CRULESCALE (在税和装运费用之间公用)	<ul style="list-style-type: none"> <li>使用“装运”笔记本更改装运设置。“装运”笔记本编辑以下数据库列: <ul style="list-style-type: none"> <li>CALSCALE_ID</li> <li>CALRULE_ID</li> </ul> </li> </ul>	
	STENCALUSG (公共, 商店的缺省 calcode。条目 1 表示装运 caluage 1。)	<ul style="list-style-type: none"> <li>使用“装运”笔记本更改装运设置。</li> </ul>	
装运实现	SHPJCRULE (商店至少有一个缺省规则)	<ul style="list-style-type: none"> <li>使用“装运”笔记本更改装运设置。“装运”笔记本编辑以下数据库列: <ul style="list-style-type: none"> <li>CALRULE_ID</li> <li>SHPARRANGE_ID</li> <li>JURSTGROUP_ID</li> </ul> </li> </ul>	
	SHPARRANGE	<ul style="list-style-type: none"> <li>使用“装运”笔记本更改装运设置。“装运”笔记本编辑以下数据库列: <ul style="list-style-type: none"> <li>SHARRAND_ID</li> <li>STORE_ID</li> <li>FFMCENTER_ID</li> <li>SHIPMODE_ID</li> </ul> </li> </ul>	
商店一产品目录	STORECAT	<ul style="list-style-type: none"> <li>编辑商店归档文件 XML 文件。</li> </ul>	
	STORECENT	<ul style="list-style-type: none"> <li>编辑商店归档文件 XML 文件。</li> </ul>	
	STORECGRP	<ul style="list-style-type: none"> <li>编辑商店归档文件 XML 文件。</li> </ul>	
	DISPENTREL	<ul style="list-style-type: none"> <li>编辑商店归档文件 XML 文件。</li> </ul>	
	DISPCGPREL	<ul style="list-style-type: none"> <li>编辑商店归档文件 XML 文件。</li> </ul>	
商店实现	INVENTORY	<ul style="list-style-type: none"> <li>编辑商店归档文件 XML 文件。</li> </ul>	



商店数据库有用资源	数据库表	商店归档文件编辑选项	数据库编辑选项
出售物	TRADEPOSCN (1)	• 编辑商店归档文件 XML 文件。	
	MGTRDPSCN (对购物者组可选)	• 编辑商店归档文件 XML 文件。	
	OFFER	• 编辑商店归档文件 XML 文件。	• WebSphere 贸易加速器中的产品管理工具
	OFFERDESC	• 编辑商店归档文件 XML 文件。	
	OFFERPRICE	• 编辑商店归档文件 XML 文件。	• WebSphere 贸易加速器中的产品管理工具
	LISTPRICE	• 编辑商店归档文件 XML 文件。	
命令	URLREG	• 编辑商店归档文件 XML 文件。	
	CMDREG	• 编辑商店归档文件 XML 文件。	
	VIEWREG	• 编辑商店归档文件 XML 文件。	
货币	CURCONVERT	• 编辑商店归档文件 XML 文件。	
	CURFORMAT	• 编辑商店归档文件 XML 文件。	
	CURFMTDESC	• 编辑商店归档文件 XML 文件。	
	CURCVLIST	• 编辑商店归档文件 XML 文件。	
	CURLIST	• 使用“商店简要表”笔记本更改商店信息。“商店简要表”笔记本编辑 CURRSTR。	
竞销	EMSPOT	• 编辑商店归档文件 XML 文件。	• WebSphere 贸易加速器中的电子广告位管理工具
	CAMPAIGN	• 编辑商店归档文件 XML 文件。	• WebSphere 贸易加速器中的竞销管理工具
商店产品目录装运	CATENCALCD	• 编辑商店归档文件 XML 文件。	• WebSphere 贸易加速器中的产品管理工具
	CATGPCALCD	• 编辑商店归档文件 XML 文件。	
商店一缺省	STOREDEF	• 编辑商店归档文件 XML 文件。	
一致性检查		• 编辑商店归档文件 XML 文件。	
支付	CMDREG	• 更改支付设置。 • 编辑商店归档文件 XML 文件。	
	VIEWREG	• 更改支付设置。 • 编辑商店归档文件 XML 文件。	

商店数据库有用资源	数据库表	商店归档文件编辑选项	数据库编辑选项
商务策略	POLICY	• 编辑商店归档文件 XML 文件。	
	POLICYCMD	• 编辑商店归档文件 XML 文件。	

商店数据库有用资源	数据库表	商店归档文件编辑选项	数据库编辑选项
 Business 组织	ORGENTITY	• 编辑商店归档文件 XML 文件。	
	MBRREL	• 编辑商店归档文件 XML 文件。	
	ADDRBOOK	• 编辑商店归档文件 XML 文件。	
	ADDRESS	• 编辑商店归档文件 XML 文件。	
 Business 商务帐户	TERMCOND	帐户有用资源不能通过商店归档文件进行更新或重新发布。如果需要更改帐户有用资源，则必须使用 WebSphere 贸易加速器中的商务关系管理工具在数据库中编辑有用资源。	WebSphere 贸易加速器的商务关系管理工具  <b>重要信息：</b> 不能使用装入程序软件包装入帐户。
	ACCOUNT		
	TRADING		
	TCDESC		
	PATTRVALUE		
	CREDITLINE		
	TRDDESC		
	POLICYTC		
	BUYERPO		
	PARTICIPNT		
	ATTACHMENT		
	TRDATTACH		

商店数据库 有用资源	数据库表	商店归档文件编辑选项	数据库编辑选项
 Business 合同	CONTRACT	<ul style="list-style-type: none"> <li>编辑商店归档文件中的合同 XML 文件</li> </ul>	WebSphere 贸易加速器的商务关系管理工具  <b>重要信息:</b> 不能使用装入程序软件包装入合同。
	TERMCOND		
	PRODUCTSET		
	TRADING		
	TCDESC		
	PATTRVALUE		
	TRDDESC		
	POLICYTC		
	PARTICIPNT		
	TRADEPOSCN		
	ATTACHMENT		
	OFFER		
	TRDATTACH		
	OFFERPRICE		
	STORECNTR		
	PURCHASELT		
	STOREDEF (由商店服务使用)		
APRVSTATUS			
FLINSTANCE			

## 更改 Web 有用资源

要更改商店归档文件中的 Web 有用资源，有以下选项：

- 将商店归档文件导入 WebSphere Studio，使用 Page Designer 或选择的工具为 Web 有用资源做必要的更改。如果需要，创建新商店页面。然后将 Web 有用资源导出并返回至商店归档文件或运营的商店。
- 使用商店服务中的“Web 有用资源”对话框从商店归档文件下载 Web 有用资源到选择的位置，然后使用选择的工具更改它们。或者使用商店服务中的“Web 有用资源”对话框将商店归档文件中的 Web 有用资源替换为现有的 Web 有用资源。如果需要，创建新的商店页面。
- 手工打开商店归档文件中的 Web 有用资源压缩归档文件，并更改现有的文件或添加新文件。

## 从商店服务发布样本商店归档文件

将样本商店归档文件发布到 Websphere Commerce Server 使您可以创建运营的商店。要发布商店归档文件，请完成以下步骤：

1. 请确保您具有站点管理员或商店管理员访问权限。如果具有商店管理员访问权限，请确保访问权限是针对所有商店的。
2. 请确保已完成所有在确保相应的服务和服务器正在运行中描述的任务。

3. 打开商店服务。
4. 从**商店归档文件**列表中，选择您想要发布的商店归档文件。
5. 单击**发布**。  
显示“发布商店归档文件”页面。
6. 选择期望的发布选项。关于发布选项的更多信息，请单击**帮助**。  
**技巧：**要创建全功能的商店，请在第一次发布商店归档文件时，选择所有的发布选项（包括产品数据选项）。
7. 选择**确定**。  
当商店发布时，您将返回到“商店归档文件”列表页面。“发布”状态列中反映了发布状态。单击**刷新**来更新状态。
8. 从列表中选择商店归档文件，并单击**发布摘要**来查看发布的结果。
9. 发布完成时，单击**启动商店**来查看并测试您的商店。当完成后，对该站点作书签并关闭浏览器。

#### 重要信息:

1. 如果更改 Web 应用程序 Web 路径或 Web 应用程序文档根路径，则必须确保它们与 Websphere Commerce Server 中定义的路径相匹配。
2. 一次只可发布一个商店归档文件。不支持并行发布，此类发布将导致两个商店的发布都会失败。
3. 在发布期间，一致性检查程序确认商店归档文件引用的文件是否存在。如果一致性检查找到错误，则此错误将写入日志。发布照常继续。
4. 在重新发布商店之前，请从以下目录中删除文件:

▶ NT	<code>drive:\WebSphere\CommerceServer\instances\instancename\cache</code>
▶ 2000	<code>drive:\Program Files\WebSphere\CommerceServer\instancename\cache</code>
▶ AIX	<code>/usr/WebSphere/CommerceServer/instances/instancename/cache</code>
▶ Solaris	<code>/opt/WebSphere/CommerceServer/instances/instancename/cache</code>
▶ 400	<code>/QIBM/UserData/WebCommerce/instances/instancename/cache</code>

在商店开发阶段，您可能想要禁用高速缓存。关于更多信息，请参阅配置高速缓存。

5. 从商店服务启动商店时，用与登录到商店服务时所用相同的用户名和密码登录到商店。如果更改商店中的密码，则也为商店服务更改了密码。相反，要测试商店中的特征（包含更改密码），请保存站点地址，关闭所有浏览器窗口，然后重新登录商店。
6. 如果作为缺省管理员登录，则不能浏览基于“商家到商家”样本商店的商店，而应创建属于缺省组织的新用户，然后浏览该商店。
7. 某些商店数据有用资源（例如产品目录和实现中心）是在商店之间共享的。因此，如果您根据相同的样本商店发布多个商店，则对每个商店而言，产品目录和实现中心将是相同的。如果您在一个产品目录中做出更改，则发布基于同一个样本的另一个商店将覆盖这些更改。有关如何避免覆盖任何更改的更多信息，请参阅共享的数据有用资源。

## 使用命令行发布商店归档文件 ( Windows )

虽然发布商店归档文件的主要方法是使用“商店归档文件服务”，但也可以用命令行进行发布。要使用命令行进行发布，请执行以下操作：

1. 请确保您具有站点管理员或商店管理员访问权限。如果具有商店管理员访问权限，请确保访问权限是针对所有商店的。

2. 使用商店归档文件的有效参数，输入以下命令：

```
publishstore sarName hostname logonId logonPwd {insert|update}  
destination1=webapp.zip,destination2=properties.zip
```

其中：

- sarName 是商店归档文件名称。sarName 是区分大小写的。确保使用正确的大小写。
- hostname 其中 *hostname* 是 WebSphere Commerce Server 的全限定 TCP/IP 名称和实例的工具端口号。您可以在配置管理器中找到工具端口号，位于**实例属性 > WebSphere** 下。缺省情况下，是 hostname:8000。
- logonId 是 WebSphere Commerce 用户标识。
- logonPwd 是 WebSphere Commerce 的用户登录密码
- insert|update 确定商店是创建（插入）还是更新（更新）。
- {ALL|NOCATLG} 确定应发布 SAR 中的哪些 XML 文件。要全部发布，请使用“所有”。要发布除产品目录以外的全部，请使用 NOCATLG。
- destination1=webapp.zip,destination2=properties.zip 是 SAR 中文件有用资源文件的列表（例如 webapp.zip），以及它们的发布路径，例如，destination1 是

```
> NT drive:\Websphere\AppServer\installedApps\  
WC_Enterprise_App_instancename.ear\wcstores.war
```

```
> 2000 drive:\Program Files\Websphere\AppServer\installedApps\  
WC_Enterprise_App_instancename.ear\wcstores.war
```

destination2 是

```
> NT  
drive:\WebSphere\AppServer\installedApps\WC_Enterprise_App_demo.ear\  
wcstores.war\WEB-INF\classes
```

```
> 2000 drive:\ProgramFiles\WebSphere\AppServer\installedApps\  
WC_Enterprise_App_demo.ear\wcstores.war\WEB-INF\classes
```

以下是此命令的示例：

```
publishstore mysar.sar myhost wcsadmin wcsadmin insert ALL  
"d:\websphere\AppServer\installedApps\WC_Enterprise_App_demo.ear\  
wcstores.war=webapp.zip,d:\websphere\AppServer\installedApps\  
WC_Enterprise_App_demo.ear\wcstores.war\WEB-INF\classes=properties.zip"
```

3. 打开 Internet Explorer。转至以下 Web 地址：

[http://hostname/webapp/wcs/stores/store\\_directory/index.jsp](http://hostname/webapp/wcs/stores/store_directory/index.jsp)，其中 *store\_directory* 是您刚发布的商店的目录。

显示您的商店。

**注意:** 如果正在发布用 WebSphere Commerce Suite 版本 5.1 创建的商店归档文件, 则需要在通过 URL 启动此商店之前完成以下步骤:

1. 如果您正在使用 DB2, 请执行以下操作。如果使用的是 Oracle, 请转至步骤 2。
  - a. 当命令运行结束时, 从**开始**菜单, 依次选择**程序**、**DB2 Windows NT 版和命令窗口**。
  - b. 在 **DB2 CLP** 窗口中, 输入 db2 connect to *dbname*, 其中 *dbname* 是向其发布商店的数据库。按 **Enter** 键。
  - c. 在命令行下, 输入 db2 select \* from store。按 **Enter** 键。显示商店列表。记录下您所创建的商店的号码。
  - d. 在命令行下, 输入 db2 select \* from catalog。按 **Enter** 键。显示产品目录的列表。记录下样本商店产品目录的号码。
  - e. 转至步骤 3。
2. 如果您正在使用 Oracle, 请执行以下操作:
  - a. 当命令运行完成时, 从**开始**菜单, 选择**程序**, **Oracle - HomeOra81**, **应用程序开发**, 然后是 **SQL Plus**。
  - b. 在窗口中, 输入您的 用户名 和 密码 以及 主机字符串。
  - c. 在 SQL Plus 窗口中, 输入 select \* from store;。按 **Enter** 键。显示商店列表。记录下您所创建的商店的号码。
  - d. 在 SQL Plus 窗口中, 输入 select \* from catalog;。按 **Enter** 键。显示产品目录的列表。记录下样本商店产品目录的号码。
  - e. 转至步骤 3。
3. 打开 Internet Explorer。转至下列 URL:  
`http://hostname/webapp/wcs/stores/servlet/StoreCatalogDisplay?storeId=storeId from step1c or 2c&langId=-1&catalogId=catalogId from step1d or 2d`  
显示您的商店。  
如果有关于显示商店的问题, 请参阅有关发布的疑难解答。

#### **重要信息:**

1. 在发布期间, 一致性检查程序确认商店归档文件所引用的文件是否存在。如果一致性检查找到错误, 则此错误将写入日志。发布照常继续。
2. 在重新发布商店之前, 请从以下目录中删除文件:  
`drive:\WebSphere\CommerceServer\instances\instancename\cache` `drive:\Program Files\WebSphere\CommerceServer\instances\instance name\cache`
3. 在商店开发阶段, 请禁用高速缓存触发器和高速缓存。保留高速缓存打开可能引起以下问题:
  - 对 JSP 文件的更改在浏览器中可能不显示。
  - 当更新数据库时, 将在发布期间调用高速缓存触发器。高速缓存触发器可能生成不必要的数据库活动, 由此产生数据库事务日志溢出。关于更多信息, 请参阅配置高速缓存。
4. 如果作为缺省管理员登录, 则不能浏览基于“商家到商家”样本商店的商店, 而应创建属于缺省组织的新用户, 然后浏览该商店。

---

## 设置商店的 Payment Manager

您可以使用管理控制台或 Payment Manager 用户界面来完成商店的 Payment Manager 设置。如果使用管理控制台，则将会在 **Payment Manager** 菜单上出现菜单项。如果使用 Payment Manager 用户界面，则菜单项将出现在导航框架中的**管理**下。

如果使用样本商店归档文件（建议使用）创建商店，则将会部分配置 Payment Manager。

要完成商店的 Payment Manager 的设置，请执行以下操作：

1. 打开管理控制台或 Payment Manager 用户界面。
2. 按照需要将 Payment Manager 用户角色指定给 WebSphere Commerce 用户。要指定 Payment Manager 用户角色，请选择**用户**。

缺省情况下，将 Payment Manager 管理员角色指定给缺省 WebSphere Commerce 站点管理员 wcsadmin。您可能希望给其它 WebSphere Commerce 用户指定各种不同的 Payment Manager 角色。

3. 要授权您的商店的卡匣，执行以下操作：
  - a. 选择**商家设置**。
  - b. 单击**商家名称**列中您的商店的名称。
  - c. 选择希望授权给您的商店的卡匣。
  - d. 单击**更新**。

**重要信息：**如果您手工创建了商店，则必须添加新的商家（您的商店）以便为商店授权卡匣。当创建新商家时，指定的商家号必须与 WebSphere Commerce 商店标识相匹配。您可以通过选择**商家设置**再单击**添加商家**来创建新的商家。

4. 要配置您的商店的卡匣，执行以下操作：
  - a. 选择**商家设置**。
  - b. 要选择卡匣，请单击出现在商店所在行和希望配置的卡匣所在列处的图标。
  - c. 单击您的商店的卡匣页面上的**帐户**并执行以下操作之一：
    - 要更改现有的帐户，请单击帐户名称。
    - 要创建新的帐户，请单击**添加帐户**。


有关配置 Cassette for BankServACH 的更多信息，请参阅 *IBM WebSphere Payment Manager for Multiplatforms, Cassette for BankServACH Supplement, Version 3.1*。

有关配置 OfflineCard cassette 和 CustomOffline cassette 的信息，请参阅《*IBM WebSphere Payment Manager 多平台版本，管理员指南，版本 3.1*》。

有关配置 cassette for SET 的信息，请参阅《*IBM WebSphere Payment Manager 多平台版本，Cassette for SET 补充文档，版本 3.1*》。

有关配置 Cassette for CyberCash 的信息，请参阅 *IBM WebSphere Payment Manager for Multiplatforms, Cassette for CyberCash Supplement, Version 3.1*。

有关配置 Cassette for VisaNet 的信息，请参阅 *IBM WebSphere Payment Manager for Multiplatforms, Cassette for VisaNet Supplement, Version 3.1*。

当从管理控制台或 Payment Manager 用户界面中使用 Payment Manager 时，要获得帮助，请单击您正在其上工作的 Payment Manager 页面右上角的 。

有关以上 Payment Manager 管理任务或其他任何 Payment Manager 管理任务的更多信息，请参阅《IBM WebSphere Payment Manager 多平台版本，管理员指南，版本 3.1》。

关于安装 WebSphere Payment Manager 的详细信息，请参阅以下文档：

- 《IBM WebSphere Commerce 商务版安装指南，版本 5.4》
- IBM WebSphere Payment Manager for Multiplatforms, Install Guide, Version 3.1。

## 为样本商店创建调度作业

当发布样本商店后，需要为商店创建调度作业。下表显示了必须为每个样本商店创建的调度作业：

样本商店	必需的调度作业
“流行时尚” 商城	<ul style="list-style-type: none"> <li>• BalancePayment</li> <li>• PayCleanup</li> <li>• ReturnCreditAndCloseScan</li> </ul>
“新时尚” 商城	<ul style="list-style-type: none"> <li>• BalancePayment</li> <li>• PayCleanup</li> <li>• ProcessBackorders</li> <li>• RAReallocate</li> <li>• ReleaseExpiredAllocations</li> <li>• ReleaseToFulfillment</li> <li>• ReturnCreditAndCloseScan</li> </ul>
“网络时尚” 商城	<ul style="list-style-type: none"> <li>• BalancePayment</li> <li>• PayCleanup</li> <li>• ReturnCreditAndCloseScan</li> </ul>
 “多乐五金商店”	<ul style="list-style-type: none"> <li>• BalancePayment</li> <li>• PayCleanup</li> <li>• ProcessBackorders</li> <li>• RAReallocate</li> <li>• ReleaseExpiredAllocations</li> <li>• ReleaseToFulfillment</li> <li>• ReturnCreditAndCloseScan</li> </ul>
“韦博拍卖行”	<ul style="list-style-type: none"> <li>• BalancePayment</li> <li>• PayCleanup</li> <li>• ReturnCreditAndCloseScan</li> </ul>

以下是这些作业的简短描述：

### BalancePayment

一旦订单已装运，则此作业将调用 DoDepositCmd 任务命令以便划款支付。此命令实现了 WebSphere Commerce 的自动支付划款功能。



### PayCleanup

此作业将取消那些支付请求被相应金融机构拒绝的时间长于商店规定时间的 WebSphere Commerce 订单。

### ProcessBackorders

此作业将库存分配给在库存不可提供时创建的延迟交货订单。

### RReallocate

(根据预期库存重新分布分配) 此作业根据现有延迟交货订单, 重新分配开放的预期库存记录 (EIR)。需要这样做是为了更准确地预计, 当添加或修改 EIR 信息和删除或分配先前延迟交货的商品时, 何时将可以提供延迟交货的订单商品。

### ReleaseExpiredAllocations

此作业将已分配库存从先前分配的超过其失效时间期限的订单限额商品返回接收表。

### ReleaseToFulfillment

此作业将订单上的已分配商品发货到实现中心。

### ReturnCreditAndCloseScan

此作业扫描以获取退货商品授权, 该授权可以是返款, 也可以标记为已结束。

要创建调度作业, 请按在调度商店级别作业中找到的对每个作业的指导进行操作。下表列出了每个作业的推荐参数:

调度作业名称	推荐的开始时间	推荐的时间间隔 (秒)	推荐的优先级
BalancePayment	00:00	86400	1
PayCleanup	00:00	86400	1
ProcessBackorders	00:00	43200	8
RReallocate	00:00	86400	1
ReleaseExpiredAllocations	00:00	3600	8
ReleaseToFulfillment	00:00	600	10
ReturnCreditAndCloseScan	00:00	86400	1

设置开始时间为 00:00 将立即启动调度作业。

**注意:** 对于这些作业, 无需填写“调度作业”窗口中的作业参数字段。

---

## 为样本商店配置电子邮件通知

以下过程在授权支付、授权订单或取消订单时启用顾客电子邮件通知。注意, 要将电子邮件发送给顾客, 您必须设置邮件服务器。

**注意:** 如果不设置邮件服务器, 将无法从商店发送电子邮件通知, 但样本商店的其它功能仍将正常工作。

不同的样本商店支持不同的电子邮件通知。下表显示了每个商店支持的电子邮件通知:

样本商店	支持的电子邮件通知	消息类型
流行时尚	密码重新设置	密码重新设置的通知消息
新时尚	已授权订单	关于已授权订单的通知消息
	密码重新设置	密码重新设置的通知消息
	提交订单	关于已接收订单的通知消息
	已取消订单	已取消订单的通知消息
	装运通知	用于将订单发布申明通知给顾客的消息
	希望列表	广播式消息
网络时尚	已授权订单	关于已授权订单的通知消息
	密码重新设置	密码重新设置的通知消息
	希望列表	广播式消息

**注意：**“韦博拍卖行”样本商店是基于“网络时尚”的。要设置电子邮件通知，您需要执行“网络时尚”的所有步骤再加上附加的与拍卖相关的步骤。有关“韦博拍卖行”电子邮件通知步骤的更多信息，请参阅下面的相关任务。

要启用电子邮件通知，请执行以下操作：

1. 确保已经启动 IBM WebSphere Application Server 管理服务器。
2. 使用站点管理员标识打开管理控制台。
3. 在“管理控制台站点 / 商店选择”页面上选择**商店**。显示“选择商店和语言”部分。
4. 从**名称**下拉列表中，选择商店。
5. 从**语言**下拉列表中，选择语言。单击**确定**。显示商店管理控制台主页。
6. 从**配置**菜单中，单击**传送**。显示“传送配置”页面。
  - a. 验证**电子邮件**传送的状态是“活动”。  
如果电子邮件不在活动状态，请选择它，然后单击**更改状态**。
  - b. 选择**电子邮件**，然后单击**配置**。显示“传送配置参数”页面。
  - c. 在**主机**字段中，输入全限定邮件服务器名称，例如 `myserver.ibm.com`。
  - d. 在**协议**字段中，以小写字母输入 `smtp`，或自己选择的协议。单击**确定**。
7. 从**配置**菜单中，单击**消息类型**。显示“消息类型配置”页面。
8. 如下创建在授权支付时发送的通知：
  - a. 单击**新建**。显示“消息传送指定”页面。
  - b. 从**消息类型**下拉列表中选择消息类型。关于商店中使用的消息类型，请参阅上表。
  - c. 在**消息严重性**字段中，输入 `0` 至 `0`。
  - d. 从**传送**下拉列表中，选择**电子邮件**。
  - e. 从**设备格式**下拉列表中，选择**标准设备格式**。
  - f. 单击**下一步**。显示“消息传送指定参数”页面。
  - g. 按照如下完整填写字段：

主机  
协议

邮件服务器的全限定名称，例如 `example.ibm.com`  
输入 `smtp`（必须使用小写字母），或您在使用的协议。

收件人  
发件人

输入有效的电子邮件地址。此地址将在运行时替换为顾客电子邮件地址。  
输入希望用作消息发件人的电子邮件地址，例如 `orders@example.ibm.com`。  
地址必须是邮件服务器上有效用户的电子邮件地址。  
主题  
输入希望显示为消息主题行的文本，例如您的订单已经接受。

h. 单击**完成**。显示“消息类型配置”页面。

9. 对于样本商店中的每个消息类型，重复步骤 8。

**注意:**

- 在发送出已取消订单的通知电子邮件之前，通常需要较长时间。可通过以下方法缩短该时间：
  1. 将 STORE 数据库表中的 REJECTEDORDEREXPIRY 列的值设置为较小的值。
  2. 更改 PayCleanup 调度作业以使用较短的调度时间间隔。
- 关于装运通知的更多信息，请参阅 ReleaseShipNotify 消息



---

## 第 2 章 样本商店数据库有用资源

---

### 商店数据库有用资源

商店数据是装入到 WebSphere Commerce Server 数据库的信息，它使商店能够运行。要正常运作，商店必须具有适当的数据以支持所有的顾客活动。例如，为了让顾客购买商品，商店必须包含待售商品的产品目录（产品目录数据）、与处理订单相关的数据（税款和装运费用数据）以及实现请求的库存（库存和实现数据）。

数据可以由一个商店专用，或者由多个商店共享。关于更多信息，请参阅共享数据库有用资源。

WebSphere Commerce 提供的样本商店归档中的商店数据库有用资源是对装入程序软件包有效的格式良好的 XML 文件，它带有以下异常：商店归档 XML 文件倾向于可移植的，且不应包含特定于数据库某一特定实例的已生成主键。相反，它们使用内部别名（在商店归档文件装入程序约定中有概述），此内部别名在发布时由标识解析器解析。商店归档文件使用一组 DTD 宏（在 XML 中称为实体）。宏作为您在商店创建期间在商店服务中选择的值的占位符。这两个约定的使用允许多次复制和发布这些样本商店。

样本商店归档文件包括所有对创建功能性商店必备的数据库有用资源。您可以修改这些文件以在您自己的商店归档文件中使用，或将它们用作创建您自己的 XML 文件的指南。WebSphere Commerce 要求某些数据必须装入到 WebSphere Commerce 数据库中以创建可运作的商店，并且此数据必须以模式确定的顺序装入。例如，FFMCENTER 表必须在 STOREENT 表之前填充。由于样本商店包括了以 WebSphere Commerce 所要求的顺序和结构存储的所有必须数据，所以使用数据库有用资源作为基础将大大节省您在初始创建期间的的时间。

关于样本商店归档文件中使用的数据库有用资源文件列表，请参阅样本商店归档文件数据库有用资源。关于商店数据的更多详细信息，请参阅《IBM WebSphere Commerce 商店开发者指南》。

**注意：**样本商店数据库有用资源的 XML 文件的 DTD 不在商店归档文件中。它们位于以下目录：

▶ NT

`drive:\WebSphere\CommerceServer\xml\sar`

▶ 2000

`drive:\Program Files\WebSphere\CommerceServer\xml\sar`

▶ AIX

`/usr/WebSphere/CommerceServer/xml/sar`

▶ Solaris

/opt/WebSphere/CommerceServer/xml/sar

▶ 400

/QIBM/ProdData/WebCommerce/xml/sar

---

## 新时尚产品目录数据库有用资源

catalog.xml 文件存储 WebSphere Commerce 内样本商店的产品目录信息。每个样本商店都有其自己的 catalog.xml 文件。关于更多信息，请参阅 catalog.xml 文件。

“新时尚”产品目录数据库有用资源分为以下部分：

- 产品目录组
- 产品目录实体
- ATP 特色

### 产品目录组

产品目录组是类别和产品的组。类别是一个产品目录构成的自身组。例如，男式时装是构成男式时装类别的组（例如，长裤和衬衫），而长裤类别则是产品的组。

在“新时尚”样本商店中，有若干产品目录组：

- 男式时装
- 女式时装
- 新品
- 家居促销

产品目录组在 CATGROUP 表中创建，并与 CATTOGRP 表中的特定产品目录相关联。产品目录组可具有顶级类别和子类别。例如，顶级类别可以是“男式时装”，而子类别则为“长裤”。如果有子类别，您必须将其置于 CATGRPREL 表内的顶级类别之下。

**注意：**家用促销类别或新品类别中的产品仅在特价商品类别中出现。促销结束后，必须将这些产品移回其日常类别中。

### 产品目录实体

“新时尚”样本商店产品目录由产品目录实体组成。这些实体由以下各项构成：

- 产品
- 商品
- 打包销售商品
- 捆绑销售商品

关于产品目录实体的信息存储在 CATENTRY 表中。CATENTREL 表存储产品目录实体间的关系，如产品项、捆绑销售商品和打包销售商品的关系。关于实体的更多信息，请参阅打包销售商品和捆绑销售商品。

产品

“新时尚”商店内的每一产品在 BASEITEM 表中都有一个条目。基本商品就是产品。每件产品只能有一个由其失效日期决定的版本。因此，您只能有一件 12 号、2010 年 1 月 1 日失效的女式红色衬衫。每个产品的版本信息存储在 ITEMVERSN 表中。

在新时尚商店中，库存不与其它商家共享。因此，产品目录中的每件产品在 DISTARRANG 表中仅有一个条目。DISTARRANG 表的每一行都表示一种分发安排，使商店能够销售它自己的库存。

## 商品

一件商品是产品的一个实例，由属性定义。关于特定商品的信息存储在 ITEMSPC 表中。每个商品的库存信息都保留在 RECEIPT 表中。预期收到延迟交货商品的日期可在 RADETAIL 表中找到。

**注意：**在“新时尚”商店中，产品目录使用 OFFERPRICE 表中的商品定价。关于更多信息，请参阅 offering.xml 文件。

## 打包销售商品

打包销售商品是一起销售的产品，如一套高尔夫球用品。一套打包销售商品在购物车上呈现为一行。不能单独购买打包销售商品中的产品。产品在 product.xml 文件中设置并在 CATENTRY 表中组成打包销售商品。打包销售商品中的每一件产品都有独立条目。

## 捆绑销售商品

捆绑销售商品是组合在一起但可以单独销售的产品。一套捆绑销售商品在购物车上呈现为一行。捆绑销售商品存储在 CATENTRY 表中，而与捆绑销售商品相关联的产品存储在 CATGPENREL 表中。产品与捆绑销售商品的关系在 CATENTREL 表中定义。

产品目录实体、打包销售商品和捆绑销售商品的属性在 PKGATTR 和 PKGATTRVAL 表中设置。PKGATTR 表反向引用属于打包销售商品中产品的一个现有属性表。PKGATTRVAL 表反向引用打包销售商品中的产品属性值。

## ATP 特色

新时尚商店具有一些特别的商品，这些商品显示了 ATP 功能，例子如下：

- 男式服装、运动装和队服的现有库存为零。
- 女式服装、装饰品和夜用包为强制延迟交货订单。
- 男式服装、运动装和队服没有 RADetails。

如果商品为零在手或强制延迟交货订单，则无论顾客订购了多少此类商品，这些商品将自动转为延迟交货订单。无 RADetail 意指延迟交货订单的预期日期将是当前日期加上商店 30 天的缺省延迟交货补偿时间（商店表中的 DEFAULTBOOFFSET 列）。

---

## “新时尚”合同和商务策略数据库有用资源

在 WebSphere Commerce 中，所有商店必须具有缺省合同。缺省合同由存储在 `contract.xml` 文件中的条款和条件所组成。如果条款和条件引用商务策略，则此商务策略存储在 `businesspolicy.xml` 文件。

“新时尚”合同和商务策略数据库有用资源信息分为以下部分：

- 合同
- 商务策略

### 合同

#### 条款和条件

样本商店的所有合同必须具有条款和条件。条款和条件是一系列在合同中同意的规则，并控制买方和销售商间的购买过程。每个合同都必须包含一组价格条款和条件。

您也可为其它类型的收费（例如装运费用）具备条款和条件。每个合同都必须只有一个装运费用条款和条件。

`contract.xml` 中必须指出买方和卖方。

### 商务策略

#### 条款和条件

策略概括了企业对于特定商务过程所遵循的规则。如果您合同的条款和条件引用商务策略，则在导入 `contract.xml` 文件之前，您必须填充 `businesspolicy.xml`。

**注意：**在 `contract.xml` 中引用的一些策略不包含在 `businesspolicy.xml` 中。这些策略是引导程序数据的一部分。关于引导程序数据的更多信息，请参阅引导程序文件。

---

## 新时尚装运数据库有用资源

新时尚装运数据库有用资源存储在以下 XML 文件：

- `shipping.xml`
- `store-catalog-shipping.xml`
- `store-defaults.xml`
- `shipfulfill.xml`

“新时尚”装运数据库有用资源分为以下部分：

- 地区
- 装运方式
- 计算代码
- 计算规则
- 计算标度
- 计算范围
- 计算查找



- 计算组合
- 装运实现

## 地区

shipping.xml文件标识装运地区。在 JURST 表中定义地区，JURSTGROUP 将地区指定到一个分组和子类，且 JURSTPREL 将地区和地区分组指定到同一子类。

## 装运方式

装运方式是装运公司及其装运服务的组合。例如，“XYZ 承运人，隔夜”是一种装运方式。关于装运方式的信息存储在 SHIPMODE 表中。

## 计算代码

计算代码用于计算折扣、装运费用、销售税和装运税。shipping.xml文件包含所有装运计算代码。CALCODE 表设置装运的计算代码。displaylevel 字段显示一个数字，此数字显示计算得到的金额。

- 0 = 订购商品
- 1 = 订单
- 2 = 产品
- 3 = 商品
- 4 = 合同

## 计算规则

每个计算代码都有一套定义如何作出计算的计算规则。例如，如果您正在将货物装运到某个区域，则您可以对计算应用某些规则，而如果您正在将货物装运到另一个区域，则您可以对计算应用不同的规则。CALRULE 表存储装运的计算规则。flag 字段指示是否要调用特定 CalculationCode 的 CalculationCodeQualifyMethod。

- 0 = 将不调用此方法。
- 1 = 将调用此方法。

## 计算标度

计算标度是应用于计算的范围集合。例如，对于装运费用，您可能有一系列重量范围，其中每个都对应特定的费用。即，重量 0 到 5 千克的产品装运费用可能为人民币 80.00 元，重量 5 到 10 千克的产品装运费用可能为人民币 120.00 元。CALSCALE 表存储装运的标度代码，一个用于每张订单，一个用于每个商品。

**注意：**如果货币应用于相应范围时，则 CALSCALE 表存储货币的标度代码。

## 计算范围

标度代码的范围存储在 CALRANGE 表中。calmethod\_id\_10 用于按订单装运，calmethod\_id\_11 用于按商品装运。

## 计算查找

计算查找价格是与计算标度关联的价格。对于重量 0 到 5 千克的产品装运费用可能为人民币 80 元，重量 5 到 10 千克的产品装运费用为人民币 120 元，它们的计算查找价格将分别为人民币 80 元和人民币 120 元。对于给定的 CALRANGE 标识，每种货物只有一个查找价格。CARLOOKUP 表定义了查找标识和价格。

### 计算组合

计算规则和标度范围组合在 CRULESCALE 表中，如以下代码所见：计算方法和规则组合在 STENCALUSG 表中。商店的计算缺省值也存储在此表中。usageflag 字段控制 OrderPrepare 命令如何使用该计算。

1 = 使用 — 使用此 CalculationUsage。

2 = 检查 — 如果计算不为订购商品产生值，则抛出 ECAApplicationException。

### 装运实现

装运实现有用资源将装运地区分组与计算规则关联，并为商店将实现中心关联到装运方式。装运实现信息存储在 SHPICRULE 和 SHPARRANGE 表中，能够如以下所见显示在 shipfulfill.xml 中。

---

## 样本商店访问控制数据库有用资源

在 WebSphere Commerce 中所有商店具有访问控制策略。访问控制策略授权用户或用户组执行特定的操作。每个商店有两个访问控制策略文件：

- *samplestorenameAccessPolicies.xml*
- *samplestorenameAccessPolicies\_locale.xml*

*AccessPolicies.xml* 和 *AccessPolicies\_locale.xml* 都是高级本地访问控制文件。*AccessPolicies.xml* 独立于本地语言，而 *AccessPolicies\_locale.xml* 则依赖于本地语言。每个文件都包含样本商店中使用的可能操作、操作组、资源以及策略定义。这些文件分别转换成 *AccessPoliciesOut.xml* 和 *AccessPoliciesOut\_locale.xml* 文件。每个转换后的文件都填充数据库。关于转换访问策略文件的更多信息，请参阅以下相关链接。

**注意：**仅转换后的文件可以大量装入或直接在 SAR 文件中使用，而转换前的文件不可以。

样本商店的数据库有用资源信息可分为以下部分：

- 操作
- 资源类别
- 资源组
- 操作组
- 策略定义

### 操作

在访问控制策略下可执行的操作在每个样本商店的 *AccessPolicies.xml* 文件中定义。

### 资源类别

资源类别定义可保护的资源。

### **资源组**

资源组包含由访问控制策略控制的资源。资源组可以包含商务对象，例如“合同”、“交换贸易状态”或一系列相关的命令。每个样本商店的 `AccessPolicies.xml` 文件定义策略中的资源组。

### **操作组**

操作组定义对访问控制策略中的资源组可执行的操作。这些组在每个商店的 `AccessPolicies.xml` 文件中定义。

### **策略定义**

每个样本商店的策略在单个商店的 `AccessPolicies.xml` 文件中定义。“多乐五金”样本商店有两个策略。



---

## 第 3 章 新时尚样本商店

---

### “新时尚” 样本商店

“新时尚”是 WebSphere Commerce 提供的商家到消费者的在线时尚商店之一。它实现了许多当今顶尖零售站点最常用的特征。“新时尚”样本商店包含的一些特征如下：

- 多文化支持
- 订购商品的有效日期
- 当前没有库存的商品的延迟交货
- 根据顾客喜好分割订单
- 跟踪订单状态
- 订单状态的电子邮件通知
- 搜索能力
- 通过实时顾客支持的协作

新时尚商店提供运营的网上商店所需的所有页面和功能。新时尚作为商店归档文件与 WebSphere Commerce 中打包，所以无需进一步的安装。要想查看样本商店，只需使用商店服务工具在“新时尚”的基础上创建新的商店归档文件，并将其发布到 WebSphere Commerce Server 上即可。关于更多信息，请参阅使用商店服务创建商店归档文件。

由于在 WebSphere Commerce 中的商店创建是建立在选择样本商店归档文件并修改此文件的基础上，因此“新时尚”设计为您的商店的基础。它基于一个简单而经过验证的购物流程，所有样本商店页面可以简单地定制。

关于“新时尚”购物流程的更多信息，请参阅样本商店购物流程图表和使用案例。使用案例详述商店中每个用户交互的流程，例如，注册或显示产品。关于每个页面如何工作的更多技术详细信息，请参阅每个页面的相应参考信息。



---

## 第 4 章 新时尚商店页面

---

### 新时尚样本商店搜索页面

在顾客在主页 sidebar.jsp 或高级搜索页面 advancedsearch.jsp 中输入搜索请求之后，显示搜索结果页面 resultlist.jsp。

**注意：**页面 subcategory.jsp 不将 sidebar.jsp 作为页面的辅助栏包含在内。因此，已在此页面内构建搜索功能，且这些搜索功能与 sidebar.jsp 的搜索功能相同。

#### Bean

resultlist.jsp 使用以下 Bean:

- CatalogDataBean
- CategoryDataBean
- CatEntrySearchListDataBean
- CatalogEntryDataBean

advancedsearch.jsp 使用以下 Bean:

- CatalogDataBean
- CategoryDataBean

#### 命令

resultlist.jsp 使用以下命令:

- ProductDisplay
- AdvancedSearchView
- CatalogSearchResultView

advancedsearch.jsp 使用以下命令:

- CatalogSearchResultView

sidebar.jsp 使用以下命令:

- CatalogSearchResultView
- AdvancedSearchView
- LogonForm (仅用于搜索)
- HelpView (仅用于搜索)
- StoreCatalogDisplay (仅用于搜索)
- SetCurrencyPreference (仅用于搜索)

subcategory.jsp 使用以下命令:

- CatalogSearchResultView
- AdvancedSearchView

**注意：** 页面 `subcategory.jsp` 不将 `sidebar.jsp` 作为页面的辅助栏包含在内。因此，已在此页面内构建搜索功能，且这些搜索功能与 `sidebar.jsp` 的搜索功能相同。

## 实现

**注意：** 关于所有样本商店页面共用的实现技术的信息（包含多文化信息），请参阅样本商店页面：公共实现技术。

当顾客单击左侧导航屏幕 `sidebar.jsp` 或 `subcategory.jsp`，中的**转至**，或单击高级搜索页面 `advancedsearch.jsp` 中的**提交**时，将显示 `resultlist.jsp`。

当激活 `CatEntrySearchListDataBean` 时，它转至数据库并检索所有的搜索结果，如下面的代码所示。

```
com.ibm.commerce.beans.DataBeanManager.activate(catEntSearchListBean, request);
```

`request` 参数是 URL 字符串，此参数传递给 `resultlist.jsp` 页面。在 `CatEntrySearchListDataBean` 中，方法 `getResultList()` 返回一个 `CatalogEntryDataBean` 的数组。每个 `CatalogEntryDataBean` 就是列在搜索结果页面上的一个结果。`getResultList()` 方法返回每个页面显示的结果的特定数目。此数目是当 `CatEntrySearchListBean` 激活后用 `pageSize` 变量在 URL 请求地址中指定的。当 `result.jsp` 显示，并且用户单击**上一页**或**下一页**按钮时，`CatalogSearchResultView` 将列出结果并且 `beginindex` URL 参数会传递到 `CatalogSearchResultView`。`CatalogSearchResultView` 告知 `CatEntrySearchListDataBean` 显示应从搜索结果列表的第几号结果开始。

在新时尚样本商店中，搜索结果只显示产品。当调用 `CatalogSearchResultView` 命令时，`resulttype` 变量会作为 URL 参数传递给命令。`resulttype` 告知 `CatEntrySearchListBean` 列出产品、商品或两者都列出。`resulttype` 的值是：

- 1 = 仅列出商品。
- 2 = 仅列出产品。
- 3 = 列出商品和产品。

以下代码仅列出产品。

```
<input type="hidden" name="resultType" value="2">
```

## 限制

商店产品目录遵循的层次结构以顶部类别位于顶部（例如“男士用品和女士用品”和“新货”）而子类别位于它们下方（例如“衬衫”和“裤子”）。

当顾客选择**高级搜索**时，他们只可以在选择的特定顶部类别或子类别下搜索。如果顾客在“男士用品”类别下搜索，那么将仅显示该类别正下方的产品。将不会显示“男式衬衫”或“男裤”类别下的产品。要搜索“裤子或衬衫”子类别，顾客必须选择该特定类别，以显示搜索结果。

---

## 样本商店通讯录页面

该样本商店通讯录页面允许注册顾客向通讯录添加地址，包含送货和开票地址。



注册顾客登录，然后单击“我的帐户”（`account.jsp`）页面中的**编辑通讯录**。将显示“通讯录”（`addressbookform.jsp`）页，从此页面中顾客可添加新地址或编辑现有地址。要获取更多信息请参阅添加新地址使用案例和编辑地址使用案例。

样本商店通讯录过程使用以下 JSP 文件：

- `account.jsp`（“我的帐户”页面）
- `addressbookform.jsp`（通讯录页）
- `addressbookform.jsp`（包含命令 `AddressForm` 的参数。不对顾客显示）
- `address.jsp`（添加地址和更新地址页面）

**注意：**`address.jsp` 同时用于“添加地址”和“更新地址”页面。如果提供了 `addressId`，则装入 `address.jsp` 作为“更新地址”页面。否则，将作为“添加地址页”装入。如果提供 `addressId` 作为 `AddressAdd` 命令的参数，则此命令将更新指定 `addressId` 的地址。否则，创建一个新地址。

### 命令

`account.jsp` 使用以下命令：

- `UserRegistrationForm`
- `AddressBookForm`

`addressbook.jsp` 使用以下命令：

- `AddressForm`
- `AddressDelete`

`address.jsp` 使用以下命令：

- `AddressAdd`
- `PrivacyView`

### Bean

`addressbook.jsp` 使用以下 Bean：

- `AddressAccessBean`

`address.jsp` 使用以下 Bean：

- `ErrorDataBean`
- `AddressDataBean`

### 实现详细信息

**注意：**关于所有样本商店页面共用的实现技术的信息（包含多文化信息），请参阅流行时尚页面：公共实现技术。

当顾客在“我的帐户”页面中单击**编辑通讯录**，命令 `AddressBookForm` 被调用。`AddressBookForm` 然后装入“通讯录”页面（`addressbook.jsp`）。如果顾客单击**添加新地址**，则调用 `AddressForm` 命令。`AddressForm` 用 `AddressForm.jsp` 在数据库中注册并检查 `page` 参数。如果 `page` 设置为 `newshipaddress`，则装入“结帐 1: 添加开票地址页面（`billingaddress.jsp`）”，否则装入“添加地址”页面（`address.jsp`）。

**注意:** 由于对添加开票地址的出错处理, 如果设置为 newshipaddress, 则 AddressForm 将装入 billingaddress.jsp 页。如果 page 等于 newshipaddress, 则意指顾客正从“开票地址页”创建新地址, 并试图转至“装运地址页”, 但出现了错误。因此, 将顾客送回“开票地址页”。

在顾客完成“添加地址页”(address.jsp)中的字段后, 它检查 addressId 是否存在。如果 addressId 已存在, 则更新通讯录, 否则创建新地址。

如果提供了 addressId, 则装入 address.jsp 作为“更新地址”页面。否则, 将作为“添加地址页”装入。在“更新地址页”, 用先前输入的值预先填充了条目字段, 如下所见:

在创建新地址之前, 使用以下 JavaScript 为此地址创建一个昵称(对地址是唯一标识符, 包含时间和日期):

当顾客完成地址并同时单击“添加地址”页面(address.jsp)和“更新地址”页面(address.jsp)中的**提交**时, 将调用 AddressAdd 命令。通讯录页面(addressbook.jsp)显示现有的地址。

#### 错误处理

如果顾客未完成“添加地址”页面(address.jsp)或“更新地址”页面(address.jsp)中的必需字段, 系统会要求重新输入这些字段。

---

## 新时尚样本商店通讯录页面

该样本商店通讯录页面允许注册顾客向通讯录添加送货和开票地址。

注册顾客登录, 然后单击“我的帐户”(account.jsp)页面中的**编辑通讯录**。显示通讯录(addressbookform.jsp)页。在此页面上, 顾客可以添加新地址或编辑现有的地址。要获取更多信息请参阅添加新地址使用案例和编辑地址使用案例。

样本商店通讯录使用以下 JSP 文件:

- account.jsp (“我的帐户”页面)
- addressbookform.jsp (通讯录页)
- addressbookform.jsp (包含命令 AddressForm 的参数。不对顾客显示)
- address.jsp (添加地址和更新地址页面)

**注意:** address.jsp 同时用于“添加地址”和“更新地址”页面。如果提供了 addressId, 则装入 address.jsp 作为“更新地址”页面。否则, 它装入作为“添加地址”页面。如果提供 addressId 作为命令 AddressAdd 的参数, 此命令将更新指定 addressId 中的地址。否则, 创建一个新地址。

#### 命令

account.jsp 使用以下命令:

- AddressBookForm
- InterestItemDisplay
- TrackOrderStatus

addressbookform.jsp 使用以下命令:

- AddressForm
- AddressDelete
- OrderItemDisplay

address.jsp 使用以下命令:

- AddressAdd
- PrivacyView

## Bean

addressbook.jsp 使用以下 Bean:

- AddressAccessBean
- AddressBookDataBean

address.jsp 使用以下 Bean:

- ErrorDataBean
- AddressDataBean

## 实现详细信息

**注意:** 关于所有样本商店页面共用的实现技术的信息 (包含多文化信息), 请参阅样本商店页面: 公共实现技术。

当顾客在“我的帐户”页面中单击**编辑通讯录**时, 将调用 AddressBookForm 命令。AddressBookForm 则装入通讯录页 (addressbookform.jsp)。如果顾客单击**添加新地址**, 将调用 AddressForm 命令。AddressForm 用 AddressForm.jsp 在数据库中注册并检查 page 参数。如果 page 设置为 newshipaddress, 则装入“结帐 1: 添加开票地址页面 (billingaddress.jsp)”, 否则装入“添加地址”页面 (address.jsp)。

**注意:** 由于对添加开票地址的出错处理, 如果设置为 newshipaddress, 则 AddressForm 将装入 billingaddress.jsp 页。如果 page 等于 newshipaddress, 则意指顾客正从“开票地址页”创建新地址, 并试图转至“装运地址页”, 但出现了错误。因此, 将顾客送回“开票地址页”。

在顾客完成“添加地址”页面 (address.jsp) 中的字段后, 系统检查 addressId 是否存在。如果 addressId 已存在, 则更新通讯录。否则, 创建一个新地址。如果提供了 addressId, 则装入 address.jsp 作为“更新地址”页面。否则, 将作为“添加地址页”装入。在“更新地址页”, 用先前输入的值预先填充了条目字段, 如以下所见:

```
if (addressId != null) bUpdateAddress = true; else bUpdateAddress = false;
```

当顾客完成地址并同时单击“添加地址”页面 (address.jsp) 和“更新地址”页面 (address.jsp) 中的**提交**, 将调用 AddressAdd 命令。通讯录页 addressbookform.jsp 显示现有的地址。

顾客可从“选择送货地址页”单击**编辑通讯录**转至通讯录。如果顾客从“选择送货地址页”转至通讯录页, 则通讯录页显示**返回结帐**链接。将不允许这些顾客从通讯录页除去地址。因此, 将添加新代码以检查顾客是否从“选择送货地址页”转来, 如下所见:

```
String mode = jhelper.getParameter("mode");
if (mode.equals("AddressBookReturnToCheckout"))
```

### 错误处理

如果顾客未完成“添加地址”页面（address.jsp）或“更新地址”页面（address.jsp）中的必需字段，系统会提示顾客重新输入这些字段。以下代码处理错误：

```
TypedProperty hshErrorProperties = bnError.getExceptionData();

if (hshErrorProperties != null)
{
//There is an error in the submitted address.
strErrorCode = hshErrorProperties.getString(ECConstants.EC_ERROR_CODE, "");
if (strErrorCode.equals(ECUserConstants.EC_ADDR_ERR_BAD_NICKNAME))
strErrorMessage = infashiontext.getString("ERROR_MESSAGE1");

...
}
```

---

## 网络时尚和新时尚样本商店捆绑销售商品显示页

捆绑销售商品显示页的特色是将网上商店中的产品分成促销组。它通常包含描述、构成捆绑销售商品的组件列表、每个组件的价格、图像和属性列表（大小和颜色，如果构成捆绑销售商品的产品具有不同变化的话），以及每个属性值的列表（对于颜色而言就是红和蓝，对于大小而言就是大号 and 特大号）。

只需轻轻点击一下，顾客就可以将捆绑销售商品中的所有产品添加到购物车或希望列表。捆绑销售商品中的每个产品在购物车中作为一个单独行项，而不像打包销售商品那样，整个打包销售商品作为一项显示在一行上。

关于样本商店捆绑销售商品显示页面和它们如何工作的更多信息，请参阅显示捆绑销售商品页面使用案例。

### 命令

bundledisplay.jsp 使用以下命令：

- OrderItemAdd
- InterestItemAdd

### Bean

bundledisplay.jsp 使用以下 Bean：

- BundleDataBean
- CompositeProductDataBean

### 实现详细信息

**注意：**关于所有样本商店页面共用的实现技术的信息（包含多文化信息），请参阅样本商店页面：公共实现技术。

`Bundledisplay.jsp` 在数据库（在 `DISPENTREL` 表中）中注册以显示商店中的所有产品。`Bundledisplay.jsp` 显示以下信息：

- 描述（包含构成捆绑销售商品的组件列表）和捆绑销售商品的图像
- 捆绑销售商品中每个产品的图像、价格、属性（大小和颜色）的列表。还有，每个属性值的列表（对于颜色而言就是红和蓝，对于大小而言就是大号 and 特大号）。
- 添加至购物车和添加至希望列表
- 数量字段（仅用于新时尚）

**描述（包含构成捆绑销售商品的组件列表）和捆绑销售商品的图像**

`BundleDataBean` 检索捆绑销售商品的描述和图像。

**产品图像、价格、属性和属性值**

`BundleDataBean` 调用 `ProductDataBean` 检索有关构成捆绑销售商品的每个产品的信息。有关 `ProductDataBean` 如何工作的更多信息，请参阅样本商店产品页面。

**添加至购物车和添加至希望列表**

有关更多信息，请参阅样本商店产品页面。

**数量字段（仅用于新时尚）**

对捆绑销售商品中的每种产品都有一个数量字段，它允许顾客指定要添加到购物车的每种产品的数量。

---

## 样本商店产品目录组页面

产品目录组页面帮助顾客浏览商店中的各个部门或各组产品或服务。第一个产品目录组页面将带领顾客来到他们想要购物的区域，通常会显示每个目录组的名称、简短描述和图像。后面的产品目录组页面将依照顾客希望浏览的产品类型逐步缩小选定范围。购物路径的最后一页产品目录组页面中包含到产品页面的链接。

通常有三种类型的产品目录组页面：

- 链接至子产品目录组页面的页面（产品目录组页面）
- 链接至产品页面的页面（产品列表页面）
- 链接至子产品目录组和产品页面的页面

关于样本商店产品目录组页面和它们如何使用的更多信息，请参阅查看产品类别使用案例。

**实现详细信息**

**注意：**关于对所有样本商店页面公共的实现技术的信息（包含多文化信息），请参阅样本商店页面：公共实现技术。

**产品目录组页面显示和产品列表页面显示**

样本商店特征产品目录组页面，它显示特色产品并在每个顶级类别（男式和女式）下显示子类别；产品列表页面，它在每个子类别（长裤和衬衫）下显示一个产品的列表。

所有到顶级类别的超级链接都包含一个称为 `top` 并设置为 `Y` 的额外参数。JavaServer Pages 文件 `categorydisplay.jsp` 当作显示商店里所有类别的页面在数据库（在 `DISPCGREL` 表中）中注册。在此页面中我们检查 `top` 参数是否存在。如果存在，则此页面包含 `topcategory.jsp`。否则包含 `subcategory.jsp`。`topcategory.jsp` 是产品目录组页面，`subcategory.jsp` 是产品列表页面。

### 产品目录组页面

`topcategory.jsp` 使用 `CategoryDataBean` 中的方法 `getSubCategories()` 检索子类别的列表。包含在“流行时尚”中的有特色的特价商品创建在产品目录中，而不是作为使用 WebSphere 贸易加速器的竞销的一部分。有特色的特价商品添加至产品目录组中，然后由 `topcategory.jsp` 使用 `CategoryDataBean` 中的 `getProducts()` 方法进行检索。

在“网络时尚”和“新时尚”中，男式的 `topcategory.jsp` 显示捆绑销售商品，而女式的 `topcategory.jsp` 显示打包销售商品。捆绑销售商品是使用 `CategoryDataBean` 中的 `getBundles()` 方法来检索的，而打包销售商品是使用 `CategoryDataBean` 中的 `getPackages()` 方法来检索的。

### 产品列表页面

`subcategory.jsp` 在产品目录组中显示所有产品的列表，在左导航栏中显示所有同级类别（在同一顶级类别中下的所有其它类别）的列表。对每个产品，`subcategory.jsp` 使用以下方法显示产品的简短描述、全图像和价格：`getDescription().getShortDescription()`、`getCalculatedContractPrice()` 和 `getDescription().getFullImage()`。要显示同级类别列表，父类别标识是必需的。要检索父类别标识，在超级链接中提供参数 `parent_category_rn`，然后此参数为父类别构造 `CategoryDataBean`。缺省情况下，`CategoryDataBean` 从参数 `CategoryId` 检索类别标识。然而，在此情况下标识存储在 `parent_category_rn` 中。因此，此类别标识必须明确设置如下：

```
String parentCategoryId = request.getParameter("parent_category_rn");
parentCategory = new CategoryDataBean ();
parentCategory.setCategoryId(parentCategoryId);
com.ibm.commerce.beans.DataBeanManager.activate(parentCategory,
request);
```

然后使用 `CategoryDataBean` 中的方法 `getProducts()` 检索属于此目录组的一个产品列表。

**注意：**在“新时尚”商店中，如果在“商店服务”中启用的话，`subcategory.jsp` 页面还包含到协作顾客关心功能的链接。`subcategory.jsp` 页面具有添加的链接，因为它没有属于自己的辅助栏，而且与其它页面一样不包含 `sidebar.jsp`。

如果顾客单击与**顾客关心现场聊天**，则将在屏幕上显示一个弹出的窗口，而顾客可以与客户服务代表进行实时的在线聊天。只有在“商店服务”中启用该功能时，才会显示该链接。与**客户助手现场聊天**链接包含在一对定制标记（`<flow:ifEnabled feature="customerCare">` 标记和 `</flow:ifEnabled>` 标记）之间的主体中，并且可以根据在“商店服务”选择的选项来启用或禁用它。只要定制标记仍保留在那里，“商店服务”就可以用于在具有协作支持的站点和不具有协作支持的站点之间自动切换，而不必修改 JavaServer Pages。要

在页面中永久启用或禁用协作支持，可以通过在“商店服务”图形用户界面中单击**永久应用**，以从 JavaServer Pages 中除去定制标记和**与顾客关心现场聊天**链接。不建议您手工除去或改变定制标记或它包含的代码。相反的，请使用“商店服务”中的**永久应用按钮**。

**注意：**围绕**与顾客关心现场聊天**链接的定制标记不能复制到其它商店的 JavaServer Pages 中。这些标记仅适合在原来包含它们的商店中起作用。有关协作的更多信息，请参阅下面有关链接。

---

## 样本商店页面：公共实现技术

大部分样本商店页面使用以下实现技术。此页面使用“流行时尚”商店作为示例，但此页面可应用到所有样本商店。关于某个别页面特定技术的更多信息，请参阅该页的参考文件。

### 多文化内容

样本商店页面还用于显示多文化内容。即相同的页面可用于不同的语言环境。启用多文化显示的大部分代码在 `getResource.jsp` 中。`getResource.jsp` 完成以下操作：

- 从命令上下文检索当前语言环境并存储为变量 `locale`。
- 检索商店目录并存储为变量 `storeDir`。
- 检索商店名并存储为变量 `storeName`。
- 使用 `ResourceBundle` API 装入特定于语言的属性文件并存储为变量 `infashiontext`。

JavaServer Pages 文件要访问以上变量，`getResource.jsp` 必须包含在此 JavaServer Pages 文件中，使用编译时 `INCLUDE` 操作：

```
<%@ include file="include/getResource.jsp"%>
```

由于 `getResource.jsp` 包含在几乎所有样本商店 JavaServer Pages 文件中，所以一个请求可能会引起它执行几次。要避免这种复制，从此页面检索的很多信息都存储在请求上下文中。例如：

```
String storeDir = (String) request.getAttribute("storeDir");
String includeDir = (String) request.getAttribute("includeDir");
String fileDir = (String) request.getAttribute("fileDir"); String
bundleDir = (String) request.getAttribute("bundleDir");

String storeName = "";

if (storeDir == null) {

storeDir = sdb.getJspPath(); fileDir = sdb.getFilePath(); includeDir =
storeDir + "include" + "/"; bundleDir = sdb.getDirectory();
storeName = sdb.getDescription(cmdcontext.getLanguageId()).getDisplayName();
request.setAttribute("storeName", storeName);
request.setAttribute("storeDir", storeDir);
request.setAttribute("fileDir", fileDir);
request.setAttribute("includeDir", includeDir);
request.setAttribute("bundleDir", bundleDir); }
```

## 特定于语言的消息

特定于语言的消息（如“感谢您的订购！”）存储在资源束属性文件中。这些文件都在以下目录中：

```
drive:\WebSphere\AppServer\installedApps\WC_Enterprise_App_instance_name.ear
\wcstores.war\WEB-INF\classes\storedir
drive:\Program
Files\WebSphere\AppServer\installedApps\WC_Enterprise_App_instance_name.ear
\wcstores.war\WEB-INF\classes\storedir
/usr/WebSphere/AppServer/installedApps/WC_Enterprise_App_instance_name.ear
/wcstores.war/WEB-INF/classes/storedir
/opt/WebSphere/Appserver/installedApps/WC_Enterprise_App_instance_name.ear
/wcstores.war/WEB-INF/classes/storedir
/opt/WebSphere/AppServer/installedApps/WC_Enterprise_App_instance_name.ear
/wcstores.war/WEB-INF/classes/storedir
/QIBM/UserData/WebASAdv4/WAS_instance_name/installedApps/WC_Enterprise_App_instance_name.ear
/wcstores.war/WEB-INF/classes/storedir
```

例如，如果您的商店目录是“storedir”，则英语的属性文件将是：

▶ NT

```
drive:\drive:\WebSphere\AppServer\installedApps\WC_Enterprise_App_instance_name.ear
\wcstores.war\WEB-INF\classes\storedir
\infashiontext_en_US.properties.
drive:\Program
Files\WebSphere\AppServer\installedApps\WC_Enterprise_App_instance_name.ear
\wcstores.war\WEB-INF\classes\storedir
/usr/WebSphere/AppServer/installedApps/WC_Enterprise_App_instance_name.ear
/wcstores.war/WEB-INF/classes/storedir/infashiontext_en_US.properties.
/QIBM/UserData/WebASAdv4/WAS_instance_name/installedApps/WC_Enterprise_App_instance_name.ear
/wcstores.war/WEB-INF/classes/storedir/infashiontext_en_US.properties
```

用 Java `java.util.ResourceBundle` API 可从 JSP 文件 `getResource.jsp` 装入此文件的内容。此捆绑销售商品存储为变量 `infashiontext`。特定于语言的消息显示如下：

```
<title><%=infashiontext.getString("REGISTER_TITLE")%></title>
```

## 设置内容编码

大部分新式浏览器都理解用 UTF-8 编码的 HTML 数据。然而，一些旧浏览器可能只理解本机的编码。例如，旧的日语浏览器只能理解用“Shift\_JIS”编码的 HTML 数据。要解决此问题，样本商店在资源束属性文件中将特定于语言的编码指定为属性名 `ENCODESTATEMENT`。例如，英语的属性文件 `infashiontext_en_US.properties` 包含了以下条目：

```
ENCODESTATEMENT = text/html; charset=ISO_8859-1
```

对个别 JSP 文件可使用 JSP 请求对象设置编码如下：



```
<% response.setContentType(infashiontext.getString("ENCODESTATEMENT")); %>
```

编码类型的设置在 JSP 文件需要尽可能的早，就象 HTTP 页眉要在任何 HTML 内容之前发送。因此，如果在 HTTP 内容之后才设置内容类型（当作 HTTP 页眉发送），可能无法达到期望效果。浏览器可能无法正确显示数据。

### 包含页眉、页脚和左导航框架

几乎所有样本商店页面都显示页眉（header.jsp）、页脚（footer.jsp）和左导航框架（sidebar.jsp）页面。使用以下执行时 INCLUDE 指令可将它们包含在其它 JSP 文件中：

```
<% String incfile;  
  
incfile = includeDir + "header.jsp"; %> <jsp:include  
page="<%=incfile%>" flush="true"/>
```

如果知道此 JSP 文件的确切位置，就可以使用以下操作简化 include 过程：  
<jsp:include page="/storedir/include/header.jsp"/>

其中，header.jsp 位于 Web 应用程序文档根路径下的 storedir 目录中。

---

## 样本商店的与我们联系页面

当顾客单击**与我们联系**链接时，显示“与我们联系”页面（contact.jsp）。它显示各种不同类型商店的联系信息，包括实际商店的电话号码和地理位置。

### 实现详细信息

**注意：**关于所有样本商店页面共用的实现技术的信息（包含多文化信息），请参阅样本商店页面：公共实现技术。

在 VIEWREG 表中创建称为 ContactView 的新视图，并建立此视图与 contact.jsp 的关联。此隐私页面的 URL 是：

```
http://machine_name/webapp/wcs/stores/servlet/ContactView?parameter_list
```

---

## 样本商店页脚

样本商店在每个页面的底部都有一个页脚（footer.jsp），其中包含到以下页面的链接：

- 主页
- 购物车
- 我的帐户
- 与我们联系
- 隐私策略
- 帮助

### 命令

footer.jsp 使用以下命令：

- StoreCatalogDisplay
- OrderItemDisplay
- LogonForm
- ContactView
- PrivacyView
- HelpView

## Bean

footer.jsp 使用以下 Bean:

- UserRegistrationDataBean
- ErrorDataBean

## 实现详细信息

**注意:** 关于所有样本商店页面共用的实现技术的信息（包含多文化信息），请参阅样本商店页面：公共实现技术。

**购物车** 页脚上的**购物车**链接链接至 OrderItemDisplay 控制器命令，此命令返回 OrderItemDisplayViewShiptoAssoc 视图命令。OrderItemDisplayViewShiptoAssoc 视图命令在数据库中用 OrderItemDisplay.jsp 注册。OrderItemDisplay.jsp 装入 shoppingcart.jsp 以显示“购物车”页面。

有关命令的更多信息，请参阅命令以及《WebSphere WebSphere Commerce 程序员指南》。

## 我的帐户

如果您是注册顾客，单击**我的帐户**将引导您转到“我的帐户”页面（account.jsp）。如果您不是注册顾客，单击**我的帐户**将引导您转到“注册或登录”页面（myaccount.jsp）。这是使用以下代码来完成的:

```
if (userType.equalsIgnoreCase("G")){ %>
<font class="buttonson"><a
href="LogonForm?langId=<%=languageId%>&storeId=<%=storeId%>&catalogId=
<%=catalogId%>" style="color:
#CCCC99"><%=infashiontext.getString("MY_ACCOUNT")%></a></font></td>
<%} else {%>
<font class="buttonson"><a
href="LogonForm?langId=<%=languageId%>&storeId=<%=storeId%>&catalogId=
<%=catalogId%>&page=account" style="color:
#CCCC99"><%=infashiontext.getString("MY_ACCOUNT")%></a></font></td>
```

## 与我们联系

单击“与我们联系”调用 ContactView 命令，它会装入“与我们联系”页面（contact.jsp）。

**帮助** 单击“帮助”调用 HelpView 命令，它会装入“帮助”页面（help.jsp）。

## 隐私策略

单击“隐私策略”调用 PrivacyView 命令，它会装入“隐私策略”页面（privacy.jsp）。

---

## 样本商店页眉

样本商店在每个页面的顶部都具有一个页眉（header.jsp），其中包含到以下页面的链接：

- 购物车
- 我的帐户
- 与我们联系
- 帮助
- 男士用品
- 女士用品
- 热销商品

### 命令

header.jsp 使用以下命令：

- OrderItemDisplay
- LogonForm
- ContactView
- HelpView
- StoreCatalogDisplay
- CategoryDisplay

### Bean

header.jsp 使用以下 Bean：

- UserRegistrationDataBean
- CatalogDataBean
- CategoryDataBean

### 实现详细信息

**注意：**关于所有样本商店页面共用的实现技术的信息（包含多文化信息），请参阅样本商店页面：公共实现技术。

**购物车** 页眉上的购物车链接链接至 OrderItemDisplay 命令，它将返回 OrderItemDisplayViewShiptoAssoc 视图命令。OrderItemDisplayViewShiptoAssoc 视图命令在数据库中用 OrderItemDisplay.jsp 注册。OrderItemDisplay.jsp 装入 shoppingcart.jsp 以显示“购物车”页面。

### 我的帐户

如果您是注册顾客，单击**我的帐户**引导您至“我的帐户”页面（account.jsp）。如果您不是注册顾客，**我的帐户**引导您至“注册”或“登录”页面（myaccount.jsp）。使用以下代码可完成此操作：

```
if (userType.equalsIgnoreCase("G")){ %>
<font class="buttonson"><a
href="LogonForm?langId=<%=languageId%>&storeId=<%=storeId%>&catalogId=
<%=catalogId%>" style="color:
#CCCC99"><%=infashiontext.getString("MY_ACCOUNT")%></a></font></td>
```

```
<%} else {%>
<font class="buttonson"><a
href="LogonForm?langId=<%=languageId%>&storeId=<%=storeId%>&catalogId=
<%=catalogId%>&page=account" style="color:
#CCCC99"><%=infashiontext.getString("MY_ACCOUNT")%></a></font></td>
```

### 与我们联系

单击**与我们联系**调用 `ContactView` 命令，它会装入“与我们联系”页面（`contact.jsp`）。

**帮助** 单击**帮助**调用 `HelpView` 命令，它会装入“帮助”页面（`help.jsp`）。

### 顶级类别（男式用品，女式用品和热销商品）

当顾客单击页眉中的一个顶级类别时，将调用 `CategoryDisplay` 命令。`CategoryDisplay` 在数据库中用 `CategoryDisplay.jsp` 注册。当 `top` 参数设置为 `Y`，在以下示例中，装入 `topcatagory.jsp`，如以下显示适当的类别页面：

```
< a
href="CategoryDisplay?catalogId=<%=catalogId%>&storeId=<%=storeId%>&categoryId=<%=
=category.getCategoryId()%>&langId=<%=languageId%>&top=Y">
```

---

## 样本商店帮助页面

当顾客单击**帮助**时，将显示帮助页面（`help.jsp`）。

### 实现详细信息

**注意：**关于所有样本商店页面共用的实现技术的信息（包含多文化信息），请参阅样本商店页面：公共实现技术。

在 `VIEWREG` 表中创建称为 `HelpView` 的新视图，并建立此视图与 `help.jsp` 的关联。隐私页面的 URL 是：

`http://machine_name/webapp/wcs/v5/stores/HelpView?parameter_list`。

---

## 样本商店主页

主页（`storecatalogdisplay.jsp`）作为商店的前台，将顾客引领到商店中。样本商店主页显示商店中所有的顶级类别（如男式用品和女式用品），并推出一些有特色的特价商品。有关样本商店主页的更多信息，请参阅主页使用案例。

### 命令

`storecatalogdisplay.jsp` 使用以下命令：

- `CategoryDisplay`
- `ProductDisplay`

### Bean

`storecatalogdisplay.jsp` 使用以下 Bean：

- `CatalogDataBean`
- `CategoryDataBean`
- `ProductDataBean`

- EMarketingSpotBean（仅用于网络时尚）

### 实现详细信息

**注意：**关于所有样本商店页面共用的实现技术的信息（包含多文化信息），请参阅样本商店页面：公共实现技术。

storecatalogdisplay.jsp 由 index.jsp 页面启动，此页面提供一个 URL 供调用样本商店的主页。index.jsp 调用 parameters.jsp 文件，此文件包含启动商店所需的参数。storecatalogdisplay.jsp 显示以下信息：

### 顶级类别

顶级类别注册在 CATTOGRP 表中。storecatalogdisplay.jsp 使用 CatalogDataBean 中的 getTopCategories() 方法检索顶级类别。

### 有特色的特价商品

包含在流行时尚和新时尚中的有特色的特价商品是在产品目录中创建的，而不是用 WebSphere 贸易加速器创建的。要创建促销，需要添加带有标识 HOMEPAGE\_PROMO 的顶级类别至产品目录中。然后属于此类别的产品被当作有特色的特价商品显示。

在网络时尚中，主页上的目标产品是竞销的一部分。此竞销是使用电子广告位和 WebSphere 贸易加速器创建的。主页上的电子广告位称为 StoreHomePage，它用于创建基于性别的促销。为了激活电子广告位，您必须使用 WebSphere 贸易加速器创建一个竞销。

临时顾客以及未提供性别信息的注册顾客将看见缺省的产品（HOMEPAGE\_PROMO）。

**注意：**家用促销类别或新品类别中的产品仅在特价商品类别中出现。促销结束后，必须将这些产品移回其日常类别中。

---

## 样本商店登录页面

样本商店登录页面使注册顾客能够登录。

当注册顾客在辅助栏上单击链接现在就注册并接收关于促销的高级通知！时，将显示“注册或登录”页面（account.jsp）。然后，顾客提供他们的电子邮件地址和密码，并单击登录，将显示“我的帐户”页面（myaccount.jsp）。要获取更多信息请参阅登录使用案例。

样本商店登录过程使用以下 JSP 文件：

- account.jsp（注册或登录页）
- myaccount.jsp（“我的帐户”页）
- LoginForm.jsp（包含 Logon 命令的参数）。不对顾客显示。）
- Logoff.jsp（包含 Logoff 命令的参数）。不对顾客显示。）
- forgetpassword.jsp（“已忘记您的密码”页面）
- ResetPasswordForm.jsp（包含复位密码命令的参数）。不对顾客显示。）
- password.jsp（向顾客显示消息，通知已对其设置了密码）
- ChangePasswordForm.jsp（“更改密码”页面）

- `ResetPasswordError.jsp`（在重新设置密码存在问题的情况下调用。不对顾客显示。）

## 命令

`account.jsp` 使用以下命令:

- `Logon`
- `Logoff`

`myaccount.jsp` 使用以下命令:

- `UserRegistrationForm`（流行时尚、网络时尚、新时尚和韦博拍卖行）
- `AddressBookForm`（流行时尚、网络时尚、新时尚和韦博拍卖行）
- `InterestItemDisplay`（网络时尚、新时尚和韦博拍卖行）
- `ProfileFormView`（网络时尚和韦博拍卖行）
- `TrackOrderStatus`（网络时尚、新时尚和韦博拍卖行）

`forgetpassword.jsp` 使用以下命令:

- `Logoff`
- `ResetPassword`

`ChangePasswordForm.jsp` 使用以下命令:

- `ResetPassword`

`forgetpassword_err.jsp` 使用以下命令:

- `ResetPassword`
- `Logoff`

`password.jsp` 使用以下命令:

- `LogonForm`

## Bean

`forgetpassword.jsp` 使用以下 Bean:

- `ErrorDataBean`

`forgetpassword_err.jsp` 使用以下 bean:

- `ErrorDataBean`

## 实现详细信息

**注意:** 关于所有样本商店页面共用的实现技术的信息（包含多文化信息），请参阅样本商店页面：公共实现技术。

顾客在“注册”页面中注册或在“登录”页面中输入其电子邮件地址和密码之后（`account.jsp`），使用以下代码将这些值转换为小写:

```
function prepareSubmit(form)
{
form.<%=ECUserConstants.EC_UREG_LOGONID%>.value =
```

```

form.<%= ECUserConstants.EC_UREG_LOGONID%>.value.toLowerCase()
form.submit()
}

```

account.jsp 还设置了登录命令期望的字段，例如：

```
<INPUT TYPE="hidden" NAME="URL" VALUE="LogonForm?page=account">
```

当顾客从“注册或登录”页面单击**登录**时，将调用 Logon 命令。Logon 命令在数据库中用 LoginForm.jsp 注册。LoginForm.jsp 使用 page 参数来确定装入哪个页面（myaccount.jsp 还是 account.jsp）。

```

String state = request.getParameter("page");
.
.
.
if (state == null)
{
incfile = "/" + storeDir + "/myaccount.jsp";

} else if (state.equals("account"))
{
incfile = "/" + storeDir + "/account.jsp";

}

```

如果输入了电子邮件地址和密码的正确组合，LoginForm.jsp 装入“我的帐户”页面（myaccount.jsp）。如果输入了电子邮件地址和密码的不正确组合，LoginForm.jsp 重新装入“注册或登录”页面（account.jsp）。

如果顾客忘记了密码，单击**您忘记了密码？**调用 Logoff 命令，并带有 state=forgetpassword 参数。Logoff 命令在数据库中用 Logoff.jsp 注册。Logoff.jsp 检查参数状态，如下所述：

```

if (state == null)
{
String [] arrstate = (String []) request.getAttribute("state");
if (arrstate != null)
state = arrstate[0];
}

if (state == null || state.length() == 0)
{
incfile = "/" + storeDir + "/UserRegistrationForm.jsp";

}
else if (state.equals("forgetpassword"))
{
incfile = "/" + storeDir + "/forgetpassword.jsp";

}

```

如果状态等于 forgetpassword, 则装入“已忘记您的密码”页面 (forgetpassword.jsp)。当顾客在页面中完成此字段并单击**将密码发送给我**, 将调用 ResetPassword 命令。旧密码设置为在数据库中失效, 将新密码以电子邮件发送给顾客。当顾客使用新密码登录时, 将强制他们更改其密码, 并将其带到“更改密码”页面 (ChangePasswordForm.jsp)。

**注意:** 如果顾客密码设置为失效, 则在下一次登录时将其自动带到“更改密码页面”。

#### 错误处理

如果顾客提供了不正确的电子邮件地址或密码, 或者未完成这些字段, 则显示错误消息, 且 Logon 命令重新装入注册或登录页, 而不设置 page 参数。如果输入了不正确的密码, 在登录前顾客必须等待数秒, 否则会显示以下错误:

请在尝试再次登录前等待数秒。

---

## 样本商店隐私页面

当顾客单击**隐私策略**时显示“隐私”页面 (privacy.jsp)。

#### 实现详细信息

**注意:** 关于所有样本商店页面共用的实现技术的信息 (包含多文化信息), 请参阅样本商店页面: 公共实现技术。

称为 PrivacyView 的新视图被创建在 VIEWREG 表中且与 privacy.jsp 关联。查看此隐私页面的 URL 是

`http://machine_name/webapp/wcs/v5/stores/PrivacyView?parameter_list`。

**注意:** 您必须创建您自己的隐私策略并将其包含在网上商店中。有关更多信息, 请参阅 IBM 隐私站点 <http://www.ibm.com/privacy/>。

---

## 样本商店产品页面

产品页面描述网上商店中特定产品的特征。它通常包含描述、价格和图像, 如果产品有不同的属性 (如不同的大小和颜色), 它还会允许顾客选择一个属性。

关于产品页面和它们如何工作的更多信息, 请参阅显示产品使用案例。

#### 命令

productdisplay.jsp 使用以下命令:

- OrderItemAdd
- InterestItemAdd (仅用于网络时尚和新时尚)

#### Bean

productdisplay.jsp 使用以下 Bean:

- CategoryDataBean
- ProductDataBean

#### 实现详细信息



**注意：**关于所有样本商店页面共用的实现技术的信息（包含多文化信息），请参阅样本商店页面：公共实现技术。

ProductDisplay.jsp 在数据库中（在 DISPENTREL 表中）注册以显示商店中的所有产品。ProductDisplay.jsp 显示以下内容：

- 描述、图像、属性和产品的属性值
- 父类别的简短描述
- 添加至购物车链接
- 数量文本框（仅用于新时尚）
- 添加至希望列表（仅用于网络时尚和新时尚）

#### 描述、图像、属性和产品的属性值

使用 ProductDataBean 属性显示产品描述和图像。

使用 ProductDataBean 的 getAttributes() 方法检索产品属性。使用 AttributeAccessBean 中的 getDistinctAttributeValues() 方法检索每个属性的值。如果 ProductDataBean 在此列中找到信息，则产品显示页面显示 HotMedia 图像，而不是全图像。

#### 父类别的简短描述

通过 parent\_category\_rn 参数提供父类别标识给产品页面。CategoryDataBean 检索父类别的简短描述。缺省情况下，CategoryDataBean 从参数 categoryId 获取类别标识。在以下示例中，参数名是 parent\_category\_rn，类别名有明确的设置：

```
String parentCategoryId = request.getParameter("parent_category_rn");
parentCategory = new CategoryDataBean ();
parentCategory.setCategoryId(parentCategoryId);
com.ibm.commerce.beans.DataBeanManager.activate(parentCategory,
request);
```

#### 添加至购物车

添加至购物车链接是通过创建调用 OrderItemAdd 命令的表单来实现的。缺省情况下，流行时尚和网络时尚如下使用隐藏字段将已订购的产品数量设置成 1：  
<input type="hidden" name="quantity" value="1">

您可以用一个文本框替换隐藏字段，这样顾客就可以输入不同的数量。

在新时尚商店中，缺省情况下使用如下的隐藏字段已订购的产品数量设置为 1：

```
<input type="text" name="quantity" value="1",size="2">
```

此文本字段允许顾客输入不同的数量。

#### 添加至购物车和添加至希望列表

当顾客选择添加至购物车或添加至希望列表时，将调用以下 JavaScript。

```
<SCRIPT language="javascript">
function Add2ShopCart(form){
form.action='OrderItemAdd'
```

```
        form.URL.value='OrderItemDisplay'  
        form.submit()  
    }  
  
    function Add2WishList(form){  
        form.action='InterestItemAdd'  
        form.URL.value='InterestItemDisplay'  
        form.submit()  
    }  
}</SCRIPT>
```

当顾客添加到购物车时，将调用 `OrderItemAdd` 命令。当顾客添加到希望列表时，将调用 `InterestItemAdd` 命令。

**注意：** 您可以使用 WebSphere 贸易加速器来创建产品。当您创建产品时，您必须创建产品的价格以便在样本商店产品页面中查看它。如果您没有创建价格，则您必须取走 `ProductDataBean` 的 `getCalculatedContractPrice` 方法。

---

## 样本商店注册页面

样本商店注册页面允许顾客在样本商店注册。当顾客注册时，必须提供他们的名字、姓氏和电子邮件地址，然后创建密码。

当顾客单击**现在注册并购买！**，显示“注册或登录”页面。然后顾客单击**注册**，显示“注册”页面 (`register.jsp`)。

当顾客希望更新注册信息时，单击“我的帐户”页面中的**更改个人信息**。显示“更改个人信息”页面 (`edit_registration.jsp`)。

### 命令

`register.jsp` 使用以下命令：

- `UserRegistrationAdd`
- `PrivacyView`

### Bean

`register.jsp` 使用以下 bean：

- `ErrorDataBean`

`edit_registration.jsp` 使用以下 bean：

- `DemographicsAccessBean`
- `UserRegistrationDataBean`
- `ErrorDataBean`

### 实现详细信息

**注意：** 关于所有样本商店页面共用的实现技术的信息（包含多文化信息），请参阅样本商店页面：公共实现技术。

**注册** 注册或登录页面 (`account.jsp`) 上的**注册**链接显示注册表单。WebSphere Commerce 使用 `UserRegistrationAdd` 命令创建新的注册。如果顾客已登录，

UserRegistrationUpdate 的行为如同 UserRegistrationAdd。也就是说，已登录的顾客无法再次注册并创建新帐户。既然样本商店允许用户多次注册并创建多个帐户，所以已登录的顾客在重新注册前必须先注销。要达到此目的，注册链接为已登录的顾客自动设置 Logoff 命令，为临时顾客设置 UserRegistrationForm 视图。Logoff 命令自动调用 LogoffView 任务。LogoffView 任务向数据库中的 Logoff.jsp 注册。Logoff.jsp 检查状态 URL 参数。如果它设置为 forgetpassword，则装入 forgetpassword.jsp，否则装入 UserRegistrationForm.jsp。UserRegistrationForm 视图与 VIEWREG 表中的 UserRegistrationForm.jsp 文件关联。UserRegistrationForm.jsp 检查“新” URL 参数。如果它设置为 Y，则通过包含 register.jsp 显示一个新的注册表单。否则，通过包含 edit\_registration.jsp 显示一个更新的注册表单。

register.jsp 包含一个新注册表单。新注册表单提交给 UserRegistrationAdd 命令。缺省情况下，UserRegistrationAdd 命令需要几个样本商店不需要的必需字段。因此，这些字段设置为隐藏的 HTML 字段，且值设置成“-”，如以下示例：

```
<INPUT TYPE="hidden" NAME="personTitle" Value="-">
```

**注意：**在新时尚商店，不提交“名称-值”对。如果不提交“名称-值”对，则 UserRegistrationAdd 命令不检查 URL 参数。当顾客注册并填写其简要表，其可能与 WebSphere 贸易加速器中的某一简要表匹配。如果 WebSphere 贸易加速器简要表具有与其关联的折扣，则顾客将收到此折扣。

#### 以电子邮件地址登录

样本商店需要顾客输入他们的电子邮件地址作为登录标识，但是 WebSphere Commerce 需要用户标识作为登录标识。作为一种解决方案，样本商店要求顾客在注册表单中输入他们的电子邮件地址。这样，在提交表单前，使用以下 Javascript 将电子邮件和用户标识字段设置电子邮件地址的值：

```
function prepareSubmit(form)
{
form.<%=ECUserConstants.EC_ADDR_EMAIL1%>.value =
form.<%= ECUserConstants.EC_UREG_LOGONID%>.value.toLowerCase()
form.<%=ECUserConstants.EC_UREG_LOGONID%>.value =
form.<%= ECUserConstants.EC_UREG_LOGONID%>.value.toLowerCase()
form.submit()
}
```

#### 更改个人信息

单击**更改个人信息**调用 UserRegistrationForm 命令，它在数据库中与 UserRegistrationForm.jsp 关联。如果 new 参数非空，它将装入 register.jsp。否则，将装入 edit\_registration.jsp。

UserRegistrationUpdate 命令用在 edit\_registration.jsp 中以更新用户的注册信息。如果顾客未完成密码或验证密码字段，系统将提供注册密码，使用以下代码：

```
function prepareSubmit(form)
{
form.<%= ECUserConstants.EC_ADDR_EMAIL1
%>.value=form.<%=ECUserConstants.EC_UREG_LOGONID%>.value
form.<%=ECUserConstants.EC_UREG_LOGONID%>.value =
```

```

form.<%= ECUserConstants.EC_UREG_LOGONID%>.value.toLowerCase()
if (form.<%=ECUserConstants.EC_UREG_LOGONPASSWORD%>.value.length == 0)
{
form.<%=ECUserConstants.EC_UREG_LOGONPASSWORD%>.value =
'<%=strPassword%>'
}
if (form.<%=ECUserConstants.EC_UREG_LOGONPASSWORDVERIFY%>.value.length
== 0)
{
form.<%=ECUserConstants.EC_UREG_LOGONPASSWORDVERIFY%>.value =
'<%=strPassword%>'
}
form.submit()
}如果顾客曾经输入过性别和年龄信息，则性别和年龄字段将预填充。
DemographicsAccessBean 从数据库中抽取性别和年龄信息。

```

### 出错处理

对 UserRegistrationAdd 命令调用 UserRegistrationErrorView 时出现的任何错误进行处理，它向数据库中的 UserRegistrationForm.jsp 注册。然而，UserRegistrationUpdate 也可以调用同样的错误视图。为了区分新注册表单和编辑表单，在新注册表单中包含了称为 new 的隐藏参数。如果该参数存在，则 UserRegistrationForm.jsp 包含 register.jsp。否则，将装入 edit\_registration.jsp。

register.jsp 在正常和错误情况下都可以使用。使用 ErrorDataBean 和误差检测法确定应在何种情况下执行 register.jsp。如果存在错误，则 register.jsp 显示错误消息。

**注意：**密码错误通过 UserRegistrationAdd 和 UserRegistrationUpdate 命令查出，但当发生密码错误时这些命令将调用 AuthenticationPolicyErrorView。

---

## 样本商店新品页面

当顾客单击**热销商品**时，显示“热销商品”页面 (newarrivals.jsp)。

### 命令

newarrivals.jsp 使用以下命令：

- ProductDisplay

### Bean

newarrivals.jsp 使用以下 Bean：

- CategoryDataBean
- ProductDataBean
- CatalogDataBean
- EMarketingSpotBean

### 实现详细信息

**注意：**关于所有样本商店页面共用的实现技术的信息（包含多文化信息），请参阅样本商店页面：公共实现技术。

“新品”页面（`newarrivals.jsp`）打算显示一个促销商品列表。流行时尚中包含的新品在产品目录中创建。结果，创建一个称为“新品”的顶级类别，所有促销商品都被添加至此类别中。由于流行时尚显示了此类别与其它顶级类别不同的内容，所以要将 `newarrivals.jsp` 文件在数据库（在 `DISPCGPREL` 表中）中注册以显示此类别。当用此类别标识调用 `CategoryDisplay` 命令时，装入 `newarrivals.jsp` 以显示指定为新品的产品。

`newarrivals.jsp` 页面使用 `CategoryDataBean` 显示类别信息。对于新时尚和流行时尚，此页面中的产品来自标识为“热销商品”的特殊产品目录。使用 `CategoryDataBean` 的 `getProducts()` 方法来检索促销产品的列表。

在网络时尚中，新品页面显示“热销”竞销。该竞销的目的是销售本季度最流行的商品。热销竞销使用推荐销售活动，此活动的目标是注册顾客，它根据顾客的年龄来推荐商品。如果注册顾客的年龄超过 29 岁，则将看见包含有三种产品的组合；而年龄小于 29 岁的顾客将看到另一个产品的组合。临时顾客以及未提供年龄信息的注册顾客将看到缺省产品。（热销商品）

热销竞销是使用 `WebSphere` 贸易加速器和电子广告位来创建的。电子广告位称为 `NewArrivalsPage`，它位于新品页面上。要激活电子广告位，您必须在 `WebSphere` 贸易加速器中设置一个竞销。

---

## 新时尚样本商店订单确认页面

当顾客在样本商店结账过程的第四步完成订单之后，顾客单击**立即订购**。将显示订单确认页（`confirmation.jsp`），以使顾客能够查看订单号、小计、总税款、装运费、总计以及整个订单的价格。如果处理两张订单，则将在确认页显示两张订单的信息。

要获取更多信息，请参阅购物车结账使用案例。

### Bean

`confirmation.jsp` 使用以下 Bean:

- `PayStatusPMDataBean`
- `OrderDataBean`

### 实现

`PayStatusPMDataBean` 从系统获取订单信息。`OrderDataBean` 获取信息，将其格式化，并将其显示到系统。

---

## 网络时尚和新时尚电子邮件通知页

在注册顾客通过单击**立即订购**（有关更多详细信息，请参阅购物车结账使用案例）完成订单并且订单由 `Payment Manager` 授权之后，系统将向顾客发送一封电子邮件（`OrderAuthorized.jsp`），声明已接受订单，授权支付。

在新时尚商店，将发送以下电子邮件通知:

- “订单已取消”电子邮件 (OrderCanceledNotification.jsp)。由于信用卡被拒绝, 无法完成订单。
- 装运通知电子邮件 (ReleaseShipNotify.jsp)。订单已经装运给购物者。
- 订单提交电子邮件 (OrderReceived.jsp)。在注册顾客通过单击**立即订购**完成订单后, 即发送订单提交电子邮件。

## Bean

OrderAuthorized.jsp 使用以下 bean:

- StoreDataBean
- OrderDataBean
- OrderItemDataBean
- OrderBean
- StoreEntityDescriptionAccessBean
- AddressDataBean

**注意:** 以上 bean 仅用于新时尚商店。

## 实现详细信息

**注意:** 关于所有样本商店页面共用的实现技术的信息 (包含多文化信息), 请参阅样本商店页面: 公共实现技术。

系统使用 OrderDataBean 检索有关顾客订单的信息。商店信息是使用 StoreDataBean 来检索的。StoreEntityDescriptionAccessBean 向系统提供商店的名称。AddressDataBean 向系统提供送货地址。

---

## 样本商店订单摘要页面

在样本商店结账过程的第四步“结账 4. 订单摘要页面” (orderdisplaypending.jsp) 中, 顾客可复查他们的详细订单信息, 包含所购买的商品的描述, 以及数量、单价、总价格、送货地址和装运费用。然后顾客必须通过提供支付信息并单击**立即订购**以完成订购。在新时尚商店中, 将显示预期装运日期。

要获取更多信息请参阅购物车结账使用案例。

## 命令

orderdisplaypending.jsp 使用以下命令:

- OrderProcess
- PrivacyView
- ContactView
- MultiOrderProcess (仅用于新时尚) (MultiOrderProcess 与 MultiOrderProcess.jsp 关联)

## Bean

orderdisplaypending.jsp 使用以下 Bean:

- OrderDataBean
- OrderItemDataBean
- AddressDataBean
- ErrorDataBean
- ShippingModeDescriptionDataBean
- UsablePaymentTCListDataBean

### 实现详细信息

#### 仅用于新时尚

当顾客单击**立即订购**时，调用 `MultiOrderProcess` 命令。`MultiOrderProcess` 是在 `VIEWREG` 表中注册的视图命令，并与 `MultiOrderProcess.jsp` 关联。`MultiOrderProcess.jsp` 根据“订单摘要”页面中的订单数目，多次执行 `OrderProcess`。在新时尚，结帐流程在“订单摘要”页面中仅允许最多两个订单。

**注意：**关于所有样本商店页面共用的实现技术的信息（包含多文化信息），请参阅样本商店页面：公共实现技术。

#### 对于所有样本商店

当顾客通过在“购物车”页面中单击**结帐**初始化结帐过程时，他们将通过一系列结帐页面，其中第四个是“结帐 4.订单摘要页面”（`orderdisplaypending.jsp`）。此页面包含一个允许顾客提交信用卡信息的表单。`UsablePaymentTCListDataBean` 用来从 `Payment Manager` 获取可用信用卡名称，且此表单的操作设置为 `OrderProcess`。提交此表单之后，如果订单处理成功，将调用 `OrderOKView`。`OrderOKView` 命令在数据库的 `VIEWREG` 表中注册，并与显示信息以确认订单的 `confirmation.jsp` 关联。在新时尚商店，如果处理两个订单，则这两个订单的信息都将显示在确认页上。

如果存在错误，则调用 `DoPaymentErrorView`。`DoPaymentErrorView` 在数据库中与 `OrderDisplayPending.jsp` 关联。结果，当存在错误时，将重新显示“结帐 4.订单摘要页面”（`orderdisplaypending.jsp`）并带有错误消息。

---

## “新时尚” 样本商店检查产品可用性页面

在第三步，即样本商店结帐过程的 3a 部分（如购物车结帐使用案例中所描述的），顾客可以检查产品的可用性，并查看订单中每个商品的预期装运日期（`ProductAvailability.jsp`）。

**注意：**仅在一些订购的商品不可用的情况下才显示此页面。

如果一些商品当前不在库存中，顾客可以选择装运首选项以使它们能分割订单或对商品延迟交货订单。顾客还可以选择从订单中除去商品。

### 命令

`ProductAvailability.jsp` 使用以下命令：

- `ProductDisplay`
- `OrderItemDelete`
- `OrderPrepare`

- OrderItemMove
- OrderItemDisplay

## Bean

ProductAvailability.jsp 使用以下 bean:

- OrderDataBean
- OrderItemDataBean
- CatalogEntryDataBean

## 实现

**注意:** 关于所有样本商店页面共用的实现技术的信息（包含多文化信息），请参阅样本商店页面：公共实现技术。

仅在一些订购商品不可用时才显示“结帐 3a, 检查产品可用性”页面（ProductAvailability.jsp）。AllocationCheck.jsp 检查所有商品是否都可用。如果所有商品都不可用或所有商品都可用，则它转至 OrderDisplayPending.jsp；否则它包含 ProductAvailability.jsp。ProductAvailability.jsp 使顾客能够查看订单中每个商品的可用日期。如果订单中的某些商品不在库存中，则顾客选择订单的**选择装运首选项**字段下的装运选项。

顾客可以选择三个装运选项:

1. 等待，直至整个订单都准备就绪可以装运。
2. 先装运库存中已有的商品，当剩余部分可用时再装运它们。
3. 先装运库存中已有的商品，将剩余部分保留在购物车中。

如果顾客选择选项 2 或 3，则 OrderItemMove 命令将订单分割成两个独立的订单。

- 新时尚样本商店

---

## 网络时尚和新时尚样本商店打包销售商品显示页

打包销售商品显示页的特色是将网上商店中的产品分成促销组。它通常包含描述、打包销售商品的图像、构成打包销售商品的组件列表和每个组件的图像、打包销售商品的价格、属性列表（大小和颜色，如果构成打包销售商品的产品具有不同变化的话），以及每个属性值的列表（对于颜色而言就是红和蓝，对于大小而言就是大号 and 特大号）。

只需轻轻点击一下，顾客就可以将打包销售商品添加到购物车或希望列表。打包销售商品在购物车中作为一个单独行项，而不像捆绑销售商品那样，捆绑销售商品中的每个商品以单独的一行显示。无法单独购买打包销售商品中的产品。

关于样本商店打包销售商品显示页和它们如何工作的更多信息，请参阅显示打包销售商品页面使用案例。

## 命令

PackageDisplay.jsp 使用以下命令:

- OrderItemAdd



- InterestItemAdd

## Bean

PackageDisplay.jsp 使用以下 bean:

- PackageDataBean
- CompositeProductDataBean

## 实现详细信息

**注意:** 关于所有样本商店页面共用的实现技术的信息 (包含多文化信息), 请参阅样本商店页面: 公共实现技术。

packagedisplay.jsp 在数据库 (在 DISPENTREL 表中) 中注册以显示商店中的所有产品。packagedisplay.jsp 显示以下信息:

- 打包销售商品的描述 (包含构成打包销售商品的组件的列表)、图像和价格
- 构成打包销售商品的产品的属性和属性值
- 添加至购物车和添加至希望列表链接
- 数量文本框 (仅用于新时尚)

### 打包销售商品的描述 (包含构成打包销售商品的组件的列表)、图像和价格

打包销售商品的描述、图像和价格是使用 PackageDataBean 来检索的。

### 构成打包销售商品的产品的属性和属性值

PackageDataBean 检索 ProductDataBean 以显示有关打包销售商品中的每个产品的信息。有关更多信息, 请参阅样本商店产品页面。

为了正确显示打包销售商品, 您必须对打包销售商品中每个产品的属性值使用相同的参数名称。例如:

```
<SELECT NAME="attrValue" >
<!-- Display product attribute values !-->
<%
Object values[] = attribute.getDistinctAttributeValues();
for (int j = 0; j < values.length; j++)
{
%>
<option><%=values[j]%></option>
<% } %>
</select>
```

### 添加至购物车和添加至希望列表

有关更多信息, 请参阅样本商店产品页面。

### 数量文本框

它允许顾客指定要添加到购物车或希望列表的打包销售商品的数目。

---

## 样本商店左侧导航框架

样本商店中的左侧导航框架（`sidebar.jsp`）使顾客能够选择显示商店所用的语言。导航框架还包含到注册和帮助页面的链接。

要获取更多信息请参阅主页使用案例。

### 命令

`sidebar.jsp` 使用以下命令：

- `StoreCatalogDisplay`
- `LogonForm`
- `HelpView`

### Bean

`sidebar.jsp` 使用以下 Bean：

- `StoreLanguageAccessBean`
- `LanguageDescriptionAccessBean`

### 实现详细信息

**注意：**关于所有样本商店页面共用的实现技术的信息（包含多文化信息），请参阅样本商店页面：公共实现技术。

当顾客从“选择国家或地区”下拉列表选择国家或地区并单击 **GO!** 时，将按照 `languageId`，使用以下代码重新装入商店产品目录页面：

```
String storelangId = storeLang.getLanguageId();  
.  
.  
.  
<option value="<%= storelangId %>"  
SELECTED><%=langDesc.getDescription()%></option>
```

将动态创建 JavaScript 代码，以便将 `StoreCatalogDisplay` 和 `SetCurrencyPreference` 命令链接在一起。这样做是为了将一种语言链接到一种缺省货币。当顾客选择了一种语言时，他们间接地为此语言选择了缺省货币，并且结果是不必在 `shoppingcart.jsp` 页中调用 `OrderPrepare` 命令。这在以下代码中显示：

```
<select NAME="currency"> ... <option value="<%=  
(String)currencyId.elementAt(iElementNum) %>" SELECTED>  
</option>
```

以下是基于选定的货币链接到一种语言、动态生成的 javascript 示例：

```
<SCRIPT language="javascript"> function  
ChangeLanguage(form) { if (form.currency[0].selected == true) {  
form.URL.value = "StoreCatalogDisplay?storeId=10151&catalogId=10151&langId=-1";  
} if (form.currency[1].selected == true) {  
form.URL.value = "StoreCatalogDisplay?storeId=10151&catalogId=10151&langId=-5";  
} form.submit(); } </SCRIPT>
```

---

## 新时尚样本商店左侧导航框架

新时尚商店中的左侧导航框架（`sidebar.jsp`）允许顾客选择显示商店所用的语言和货币，并搜索商品的产品目录。导航框架还包含到“注册”、“帮助”、“高级搜索”页面和“与顾客关心现场聊天”的链接。

**注意：**只有通过“商店服务”在商店中启用了与顾客关心现场聊天，“与顾客关心现场聊天”链接才可见。

要获取更多信息请参阅主页使用案例。

### 命令

`sidebar.jsp` 使用以下命令：

- `StoreCatalogDisplay`
- `SetCurrencyPreference`
- `LogonForm`
- `HelpView`
- `CatalogSearchResultView`
- `AdvancedSearchView`

### Bean

`sidebar.jsp` 使用以下 Bean：

- `SupportedLanguageAccessBean`
- `LanguageDescriptionAccessBean`
- `CurrencyDescriptionAccessBean`

### 实现详细信息

**注意：**关于所有样本商店页面共用的实现技术的信息（包含多文化信息），请参阅样本商店页面：公共实现技术。

当顾客从“选择语言”下拉列表中选择国家或地区和语言并单击 **GO!** 时，将按照 `languageId` 重新装入“商店产品目录”页面。下面的代码获取商店支持的语言并在下拉框中显示它们。

```
<jsp:useBean id="supportedLanguageDataBean"
class="com.ibm.commerce.common.objects.SupportedLanguageAccessBean"
scope="page" />
<%
Enumeration enStoreLangList =
supportedLanguageDataBean.findByStore(new Integer(storeId));
while (enStoreLangList.hasMoreElements()) {
SupportedLanguageDataBean storeLang =
(SupportedLanguageDataBean) enStoreLangList.nextElement();
String storelangId = storeLang.getLanguageId();
//Get the display name of the language in the language
//currently selected by the shopper.
```

```

LanguageDescriptionDataBean langDesc =
new LanguageDescriptionDataBean();
langDesc.setInitKey_languageId(languageId);
langDesc.setInitKey_descriptionLanguageId(storelangId);

//If this language is currently selected, select it
//in the drop down list.
if (languageId.equals(storelangId))
{
%>
<option value="<%= storelangId %>"
SELECTED><%=langDesc.getDescription()%></option>

```

当顾客从“选择货币”下拉列表中选择一个货币并单击 **GO!** 时，将按照 `supportedCurrencies` 重新装入“商店产品目录”页面。下面的代码获取支持的货币并在下拉框中显示它们:

```

CurrencyManager cm = CurrencyManager.getInstance();
String [] supportedCurrencies = (String [])
cm.getSupportedCurrencies(cmdcontext.getStore());
for (int i = 0; i < supportedCurrencies.length; ++i)
{
CurrencyDescriptionDataBean currDesc = new CurrencyDescriptionDataBean();
currDesc.setInitKey_languageId(languageId);
currDesc.setInitKey_currencyCode(supportedCurrencies[i]);
String currency = (String) cmdcontext.getCurrency();
// pre-select the appropriate value in the in the drop down list.
if (currency.equals(supportedCurrencies[i]))
{
%>
<OPTION Value="<%=supportedCurrencies[i]%>"
SELECTED><%=currDesc.getDescription()%></OPTION>
<%

```

当顾客在“搜索”字段中输入关键字并单击 **GO** 时，`CatalogSearchResultView` 命令将提交搜索条件，并显示 `resultlist.jsp` 页面，其中包含搜索结果。

如果顾客单击**与顾客关心现场聊天**，则将在屏幕上显示一个弹出的窗口，而顾客可以与客户服务代表进行实时的在线聊天。只有在“商店服务”中启用该功能时，才会显示该链接。**与客户助手现场聊天**链接包含在一对定制的标记（`<flow:ifEnabled feature="customerCare">` 标记和 `</flow:ifEnabled>` 标记）之间的主体中，并且可以根据在“商店服务”选择的选项来启用或禁用它。只要定制标记仍保留在那里，“商店服务”就可以用于在具有协作支持的站点和不具有协作支持的站点之间自动切换，而不必修改 `JavaServer Pages`。要在页面中永久启用或禁用协作支持，可以通过在“商店服务”图形用户界面中单击**永久应用**，以从 `JavaServer Pages` 中除去定制标记和**与顾客关心现场聊天**链接。不建议您手工除去或改变定制标记或它包含的代码。相反的，请使用“商店服务”中的**永久应用**按钮。

**注意：**围绕与顾客关心现场聊天链接的定制标记不能复制到其它商店的 JavaServer Pages 中。这些标记仅适合在原来包含它们的商店中起作用。有关协作的更多信息，请参阅下面有关链接。

---

## 样本商店选择开票地址页面

在样本商店结帐过程的第一步“结帐 1. 选择开票地址”页面 (`billingaddress.jsp`) 中，顾客可以选择现有的地址作为开票地址，或创建新的地址用作开票地址。

要获取更多信息请参阅购物车结帐使用案例。

### 命令

`billingaddress.jsp` 使用以下命令：

- `AddressForm`
- `OrderItemDisplay`
- `OrderCopy`
- `AddBillAddressView`

### Bean

`billingaddress.jsp` 使用以下 Bean：

- `OrderDataBean`
- `AddressAccessBean`
- `OrderItemAccessBean`
- `ErrorDataBean`

### 实现详细信息

**注意：**关于所有样本商店页面共用的实现技术的信息（包含多文化信息），请参阅样本商店页面：公共实现技术。

当顾客通过单击“购物车”页面中的**结帐**初始化结帐过程时，他们将通过一系列的结帐页面，其中第一个页面就是“结帐 1. 添加开票地址”页面 (`billingaddress.jsp`)。`billingaddress.jsp` 检查顾客在通讯录中是否有现有的地址。如果在通讯录中当前存在地址，它们将被显示，允许顾客选择一个作为开票地址。

顾客还可通过单击**创建新地址**创建新地址。单击**创建新地址**调用 `AddressForm` 命令，它在数据库中与 `AddressForm.jsp` 关联。`AddressForm.jsp` 调用 `address.jsp`，装入“添加地址”页面。`address.jsp` 检查 `page` 参数以确定要装入的下一页。如果将 `page` 值设置为 `billingaddress`，`AddressAdd` 表单中的 URL 值将设置为 `OrderItemDisplay`。`OrderItemDisplay` 调用 `billingaddress.jsp`，当顾客单击**提交**时，这将引导顾客返回到结帐 1. “选择开票地址”页面。检查 `page` 参数以确定装入的地址表单。

如果通讯录中不存在任何现有地址，将显示“添加开票地址”表单，提示顾客输入新地址。在此情况下的“添加开票地址”表单也由 `billingaddress.jsp` 生成。对此 HTML 表单的操作被设置为 `AddBillAddressView`，它在 `VIEWREG` 表中注册。`AddBillAddressView` 与 `AddBillAddress.jsp` 关联。当带 `AddBillAddressView` 的表单提交时，调用 `AddBillAddress.jsp`。

AddBillAddress.jsp 执行以下命令:

- AddressAdd
- OrderCopy

在 AddressAdd 执行后, AddBillAddress.jsp 使用 AddressAdd 返回的地址标识作为 OrderCopy 命令输入的一部分。然后, OrderCopy 给当前订单的开票地址分配地址标识并调用 OrderItemDisplay.jsp。page 参数的值设置为 newshipaddress, 因此 OrderItemDisplay.jsp 调用 shipaddress.jsp。

**注意:** 在注册期间, WebSphere Commerce 需要创建一个地址。由于样本商店在顾客注册期间不询问地址, 所以有些必需字段, 如“地址 1”, 可设置为未使用。当检查地址时, **billingaddress.jsp** 检查地址 1 的值是否为未使用。如果是的话, 将不显示地址。

---

## “新时尚” 样本商店选择开票地址页面

在样本商店结帐过程的第一步“结帐 1. 选择开票地址”页面 (billingaddress.jsp) 中, 顾客可以选择现有的地址作为开票地址, 或创建新的地址用作开票地址。

要获取更多信息请参阅购物车结帐使用案例。

### 命令

billingaddress.jsp 使用以下命令:

- AddressForm
- OrderItemDisplay
- OrderCopy
- AddressAdd

### Bean

billingaddress.jsp 使用以下 Bean:

- OrderDataBean
- AddressDataBean
- OrderItemDataBean
- ErrorDataBean

### 实现详细信息

**注意:** 关于所有样本商店页面共用的实现技术的信息 (包含多文化信息), 请参阅样本商店页面: 公共实现技术。

当顾客通过在“购物车”页面中单击**结帐**初始化结帐过程时, 他们将通过一系列结帐页面, 其中第一个是“结帐 1. 添加开票地址” (billingaddress.jsp)。billingaddress.jsp 将检查顾客在通讯录中是否有现有的地址。如果在通讯录中当前存在地址, 将显示它们, 这使顾客能够选择一个作为开票地址。

OrderItemDisplay 命令用来确定下一个要装入的页面。OrderItemDisplay 返回数据库中的 OrderItemDisplay.jsp。OrderItemDisplay.jsp 根据 page 参数包含不同 JSP 文件。如果 page 的值是 shipmethod，则装入第三页面“结帐 3.选择装运方式”页面 (shipping.jsp)。

顾客还可通过单击**创建新地址**创建新地址。单击**创建新地址**调用 AddressForm 命令，它在数据库中与 AddressForm.jsp 关联。AddressForm.jsp 调用 address.jsp，它装入“添加地址”页面。当顾客单击**提交**时，“添加地址”页面将顾客引导回“结帐 1. 添加开票地址”页面。

如果通讯录中不存在任何现有地址，将显示“添加开票地址”表单，提示顾客输入新地址。在此情况下，“添加开票地址”表单也由 billingaddress.jsp 生成。

在 AddressAdd 执行并调用 OrderCopy 命令之后，OrderCopy 将地址标识指定给当前订单的开票地址，并调用 OrderItemDisplay 命令，此命令调用 OrderItemDisplay.jsp。page 参数的值设置为 newshipaddress，因此 OrderItemDisplay.jsp 调用 shipaddress.jsp。

**注意：**在“流行时尚”和“网络时尚”商店中，AddBillAddress.jsp 用于调用 AddAddress 和 OrderCopy 命令。在“新时尚”商店中，由于使用 URL 参数来链接 AddAddress 和 OrderCopy 命令，这已经消除了 AddBillAddress.jsp。因为不需要额外的 JSP，所以此方法实现比较简单，但由于额外的重定向因而它有较长的系统执行路径。

---

## 样本商店购物车

如显示购物车使用案例所述，顾客可以查看并编辑在购物车 (shoppingcart.jsp) 中选择的商品。

### 命令

shoppingcart.jsp 使用以下命令：

- OrderItemUpdate
- OrderItemDelete
- StoreCatalogDisplay
- QuickCheckoutView

### Bean

shoppingcart.jsp 使用以下 Bean：

- OrderDataBean

### 实现详细信息

**注意：**关于所有样本商店页面共用的实现技术的信息（包含多文化信息），请参阅样本商店页面：公共实现技术。

当顾客在页眉或页脚中单击**购物车**，OrderItemDisplay 命令被调用，这将返回 OrderItemDisplayViewShiptoAssoc 视图命令。OrderItemDisplayViewShiptoAssoc 视图命

令在数据库中向 OrderItemDisplay.jsp 注册。OrderItemDisplay.jsp 根据 page 参数包含不同 JavaServer Page 文件。如果未提供 page 的值，将装入“购物车”页面 (shoppingcart.jsp)。

**注意：**OrderItemDisplay.jsp 运行 OrderPrepare，它重新计算订单总额并将缺省货币转换为顾客当前使用的货币。单一货币商店不需要运行 OrderPrepare。

“购物车”页面包含**更新总额**按钮和**结帐**按钮。单击**更新总额**更新订购的商品数量，然后重新显示“购物车”页面。单击**结帐**更新订购的商品数量并显示结帐 1：“选择开票地址”页面。

**结帐**和**更新总额**按钮都使用同一个 HTML 表单。然而，当顾客在提交此表单之前单击**结帐**时，JavaScript 被用来添加一个值设置为 billingaddress 的附加 page 参数。

用于完成“快速结帐”选项的命令是 QuickCheckoutView，它是在 VIEWREG 中注册并与 QuickCheckout.jsp 关联的视图命令。QuickCheckout.jsp 运行以下服务器命令：

- OrderItemUpdate (更新送货地址)
- OrderItemUpdate (更新装运方式)
- OrderCopy (更新开票地址和支付信息)
- OrderPrepare

QuickCheckout.jsp 使用 OrderAccessBean 从顾客的快速结帐简要表中检索开票和送货地址、装运方式和支付信息。然后，它将此信息指定给 orderId 中指定的订单，并执行 OrderPrepare 命令。

在快速结帐过程完成之后，此命令向前转至 URL 中指定的一个视图。网络时尚指定 QuickCheckoutSummaryView 作为 QuickCheckout 命令的 URL。这样，当完成快速结帐时，将显示快速结帐摘要页面。

### 错误处理

如果商店没有附属的实现中心，或产品不在库存中，OrderItemAdd/OrderItemUpdate 命令调用 ResolveFulfillmentCenterErrorView，后者在数据库中用 shoppingcart.jsp 注册。如果顾客在数量字段中输入了无效字符，将调用 InvalidInputErrorView。InvalidInputErrorView 在数据库中亦是用 shoppingcart.jsp 注册的。

由于 OrderItemUpdate 命令和 OrderItemAdd 命令都可能引起错误，shoppingcart.jsp 将检查是哪条命令引起问题并相应地显示错误。如果最后的命令是 OrderItemUpdate，购物车页面将随错误消息重新显示。在其它情况下，一个独立的错误页面将随错误消息显示。这在以下的代码中显示：

```
String lastCmdName = cmdcontext.getCommandName().trim();
```

shoppingcart.jsp 在正常和错误条件下都可以使用。ErrorDataBean 和错误检查确定在何种条件下显示 shoppingcart.jsp。如果存在错误，shoppingcart.jsp 显示一个相应的错误消息。



如果 QuickCheckout.jsp (QuickCheckoutView) 找不到快速结帐简要表, 它将调用 QuickCheckoutError.jsp 以显示错误消息。结果, 显示 quickcheckouterrorview.jsp。

quickcheckouterrorview.jsp 检查顾客是否已注册。如果顾客已注册, 则系统显示一条消息提示顾客创建快速结帐简要表。如果顾客没有注册, 则系统提示顾客先注册, 然后再创建快速结帐简要表。

---

## 新时尚样本商店购物车

如显示购物车使用案例所述, 顾客可以查看并编辑在购物车 (shoppingcart.jsp) 中选择的商品。在购物车中最多可以显示两个订单。

### 命令

shoppingcart.jsp 使用以下命令:

- OrderItemUpdate
- ProductDisplay
- StoreCatalogDisplay
- OrderItemMove

### Bean

shoppingcart.jsp 使用以下 Bean:

- OrderItemDataBean
- ErrorDataBean
- CatalogEntryDataBean
- OrderDataBean
- FormattedMonetaryAmountDataBean
- OrderAccessBean

### 实现详细信息

**注意:** 关于所有样本商店页面共用的实现技术的信息 (包含多文化信息), 请参阅样本商店页面: 公共实现技术。

购物车总是表现为作为单一订单显示, 然而在购物车中可以存在多个被分割的订单。这发生在顾客先前已经到了结帐流程, 然后又分割订单的情况下。当顾客单击**结帐**时, 将调用 OrderItemMove 将所有订单移到一个订单中。OrderItemMove 使用 deleteIfEmpty 参数删除在移动后留下的所有空订单。

当顾客在页眉或页脚中单击**购物车**时, 将调用 OrderItemDisplay 命令, 此命令在数据库中用 OrderItemDisplay.jsp 注册。OrderItemDisplay.jsp 根据 page 参数包含不同 JSP 文件。如果未提供 page 的值, 将装入“购物车”页面 (shoppingcart.jsp)。

OrderItemUpdate 命令干那更新购物车中每个商品的数量。

FormattedMonetaryAmountDataBean 格式化总价格和总金额。OrderDataBean 获取每个订单的购物车商品。每个购物车商品是一个 OrderItemBean。

“购物车”页面包含**更新总额**按钮和**结帐**按钮。单击**更新总额**更新订购的商品数量，然后重新显示“购物车”页面。单击**结帐**更新订购的商品数量，然后显示“结帐 1. 选择开票地址”页面。

**注意：**结帐和更新总额按钮都使用同一个 HTML 表单。

### 错误处理

如果商店没有实现中心附加其上，`ErrorDataBean` 将提供错误信息。`ErrorDataBean` 还提供有关顾客是否输入了无效数量（例如非数字字符）的信息。

`shoppingcart.jsp` 在正常和错误条件下都可以使用。`ErrorDataBean` 和错误检查确定在何种条件下显示 `shoppingcart.jsp`。如果存在错误，`shoppingcart.jsp` 显示一个相应的错误消息。有关错误处理的更多信息，请参阅以下相关概念。

---

## 样本商店装运方式页面

在样本商店结帐过程的第三步中（如购物车结帐使用案例中所述），顾客必须选择一种装运方式（`shipping.jsp`）。

### 命令

`shipping.jsp` 使用以下命令：

- `AddShipModeView`
- `OrderItemDisplay`

### Bean

`shipping.jsp` 使用以下 Bean：

- `OrderBean`
- `OrderItemAccessBean`
- `ShipModeAccesBean`

### 实现详细信息

**注意：**关于所有样本商店页面共用的实现技术的信息（包含多文化信息），请参阅样本商店页面：公共实现技术。

当顾客通过在“购物车”页面中单击**结帐**初始化结帐处理时，他们将通过一系列结帐页面。`OrderItemDisplay` 命令用来确定下一个要装入的页面。`OrderItemDisplay` 返回数据库中的 `OrderItemDisplay.jsp`。`OrderItemDisplay.jsp` 根据 `page` 参数包含不同 JSP 文件。如果 `page` 的值是 `shipmethod`，则装入第三页面“结帐 3. 选择装运方式”页面（`shipping.jsp`）。

“结帐 3. 选择装运方式”页面（`shipping.jsp`）包含一个允许顾客选择装运方式的表单。对表单的操作设置为 `AddShipModeVew`，它在 `VIEWREG` 数据库表中注册以与 `AddShipMode.jsp` 关联。当带 `AddShipModeVew` 的表单被提交时，`AddShipMode.jsp` 被调用。

`AddShipMode.jsp` 执行以下命令：

- `OrderItemUpdate`

- OrderPrepare.

OrderItemUpdate 命令更新带选定装运方式的订购商品。然后，OrderPrepare 命令被调用以预处理此订单。然后，OrderDisplay 被调用以显示结帐处理的下一页。如果 status 参数设置为 P，则接着显示“结帐 4: 订单摘要” (OrderDisplayPending.jsp)。

“结帐 3. 选择装运方式”页面 (shipping.jsp) 以显示每种装运方式的价格结构和大约的交付时间。此信息存储在 SHPMODEDESC 表的以下字段:

- DESCRIPTION 存储装运方式的描述。
- SHPMODEDESC.FIELD1 存储价格结构描述。
- SHPMODEDESC.FIELD2 存储大约的交付时间。

如果要更改数据库中的装运费用，请确保您也更改了 SHPMODEDESC 表中的描述以便在此页面上显示已更新过的值。

**注意:** SHPMODEDESC.FIELD1 和 SHPMODEDESC.FIELD2 已使用 SHPMODEDESC 表中的定制字段创建。

---

## 新时尚样本商店选择装运方式页面

在样本商店结帐过程的第三步中（如购物车结帐使用案例中所述），顾客必须选择一种装运方式 (shipping.jsp)。顾客可以为订单中的每个商品分别选择一种装运方式。

### 命令

shipping.jsp 使用以下命令:

- OrderItemUpdate
- Product Display
- OrderItemDisplay
- AllocationCheck

### Bean

shipping.jsp 使用以下 Bean:

- OrderDataBean
- ShippingDataBean
- OrderItemDataBean
- ShippingModeDescriptionDataBean
- CatalogEntryDataBean

### 实现详细信息

**注意:** 关于所有样本商店页面共用的实现技术的信息（包含多文化信息），请参阅样本商店页面: 公共实现技术。

当顾客通过在“购物车”页面中单击**结帐**初始化结帐过程时，他们将通过一系列结帐页面。OrderItemDisplay 命令用来确定下一个要装入的页面。OrderItemDisplay 返回数据

库中的 OrderItemDisplay.jsp。OrderItemDisplay.jsp 根据 page 参数包含不同 JSP 文件。如果 page 的值是 shipmethod, 则装入第三页面“结帐 3.选择装运方式”页面 (shipping.jsp)。

“结帐 3. 选择装运方式”页面 (shipping.jsp) 允许顾客为订单中的每个商品选择一个装运方式。顾客在每个商品的**装运方式**字段下选择一个装运方式。

OrderItemUpdate 命令用选定的装运方式更新订购商品。当顾客单击下一页时, 将调用 OrderItemDisplay 命令显示结帐过程中的下一页。如果一些商品不可用, 则下一步显示“结帐 3a. 产品可用性”页面 (ProductAvailability.jsp)。“产品可用性”页面将转到 AllocationCheck.jsp, AllocationCheck.jsp 根据是否存在可用库存, 决定转到哪一页。

“结帐 3. 选择装运方式”页面 (shipping.jsp) 以显示每种装运方式的价格结构。此信息存储在 SHPMODEDSC 表的以下字段:

- DESCRIPTION 存储装运方式的描述。
- SHPMODEDSC.FIELD1 存储价格结构描述。

如果要更改数据库中的装运费用, 请确保您也更改了 SHPMODEDSC 表中的描述以便在此页面上显示已更新过的值。

**注意:** SHPMODEDSC.FIELD1 是 SHPMODEDESC 表的定制字段。

---

## 样本商店选择送货地址页面

在样本商店结帐过程的第二步“结帐 2. 选择送货地址”页面 (shipaddress.jsp) 中, 顾客可以选择现有的地址作为送货地址, 编辑该地址, 或创建新地址以用来作为送货地址。

要获取更多信息, 请参阅购物车结帐使用案例。

### 命令

shipaddress.jsp 使用以下命令:

- OrderItemUpdate
- AddressForm
- OrderItemDisplay

### Bean

shipaddress.jsp 使用以下 Bean:

- OrderBean
- AddressAccessBean
- OrderItemDataBean

### 实现详细信息

**注意:** 关于所有样本商店页面共用的实现技术的信息 (包含多文化信息), 请参阅样本商店页面: 公共实现技术。

当顾客通过单击“购物车”页面中的**结帐**初始化结帐过程时，他们将通过一系列的结帐页面，其中第二个页面就是“结帐 2 选择送货地址”页面（`shipaddress.jsp`）。`shipaddress.jsp` 显示所有现有的地址，允许顾客选择其中一个作为送货地址。

顾客还可通过单击**创建新地址**创建新地址。单击**创建新地址**调用 `AddressForm` 命令，它在数据库中与 `AddressForm.jsp` 关联。`AddressForm.jsp` 调用 `address.jsp`，这将装入“添加地址”页面。`address.jsp` 检查 `page` 参数以确定要装入的下一页。如果设置 `page` 值为 `shipaddress`，`AddressAdd` 表单中的 `URL` 值将设置为 `OrderItemDisplay`。`OrderItemDisplay` 调用 `shipaddress.jsp`，当顾客单击**提交**时，这将引导顾客返回到结帐 2. “选择送货地址”页面

**注意：**在注册期间，WebSphere Commerce 需要创建一个地址。由于样本商店在顾客注册期间不询问地址，所以有些必需字段（例如“`address1`”）设置为“-”。当检查地址时，`billingaddress.jsp` 检查 `address1` 的值是否为“-”。如果是的话，将不显示地址。

---

## 新时尚样本商店选择送货地址页面

在样本商店结帐过程的第二步“结帐 2. 选择送货地址”页面（`shipaddress.jsp`）中，顾客可以选择现有的地址作为送货地址，编辑该地址，或创建新地址以用来作为送货地址。可以通过在**装运到**字段下选择别名来选择现有的地址。订单中的每个商品可以具有独立的送货地址。

要获取更多信息请参阅购物车结帐使用案例。

### 命令

`shipaddress.jsp` 使用以下命令：

- `OrderItemUpdate`
- `AddressForm`
- `AddressBookForm`
- `Product Display`
- `OrderItemDisplay`

### Bean

`shipaddress.jsp` 使用以下 Bean：

- `OrderBean`
- `AddressDataBean`
- `OrderItemDataBean`

### 实现详细信息

**注意：**关于所有样本商店页面共用的实现技术的信息（包含多文化信息），请参阅样本商店页面：公共实现技术。

当顾客通过单击“购物车”页面中的**结帐**启动结帐进程时，他们将通过一系列结帐页面，其中第二个页面是“结帐 2 选择送货地址”页面（`shipaddress.jsp`）。`shipaddress.jsp` 显示订单上所有商品以及链接到现有地址的昵称，这允许顾客为每个商品选择一个独立的昵称作为送货地址。

OrderItemDisplay 命令用来确定下一个要装入的页面。OrderItemDisplay 返回数据库中的 OrderItemDisplay.jsp。OrderItemDisplay.jsp 根据 page 参数包含不同 JSP 文件。如果 page 的值是 shipmethod, 则装入第三页面“结帐 3.选择装运方式”页面 (shipping.jsp)。

顾客可以通过单击**创建新地址**创建新地址。单击**创建新地址**调用 AddressForm 命令, 它在数据库中与 AddressForm.jsp 关联。AddressForm.jsp 调用 address.jsp, 这将装入“添加地址”页面。当顾客单击**提交**时, “添加地址”页面将顾客引导回“选择送货地址”页面。

顾客还可以通过单击**编辑通讯录**来编辑地址。单击**编辑通讯录**将调用 AddressBookForm 命令, 此命令与数据库中的 AddressBookForm.jsp 关联。AddressBookForm.jsp 调用 AddressBook.jsp, 它装入通讯录页面。显示所有与该顾客相关联的地址。然后, 顾客可以在选择的地址下单击**编辑**。单击**编辑**将显示更新地址页面, 在此页面中可以做出更改。当顾客单击**提交**时, 他们被带回到通讯录页面。当顾客单击**返回结帐**时, 通讯录页面将顾客带回到“结帐 2. 选择送货地址”页面。

**注意:** 在注册期间, WebSphere Commerce 需要创建一个地址。由于样本商店在顾客注册期间不询问地址, 在注册期间创建的地址字段为空。在检查地址时, shipaddress.jsp检查 LogonID 是否等于 NickName。如果是, 则表明此地址是自地址, 即注册期间创建的不被显示的地址。如果是的话, 将不显示地址。

---

## 样本商店快速结帐摘要页面

当顾客在购物车中单击**快速结帐**时, 将显示“快速结帐: 订单摘要”页面 (quickcheckoutsummary.jsp), 其中带有订单详细信息以及预填充的装运、开票和支付信息。然后顾客必须通过单击**立即订购**完成订单。

有关更多信息, 请参阅快速结帐使用案例。

### 命令

quickcheckoutsummary.jsp 使用以下命令:

- OrderProcess
- OrderItemDisplay
- PrivacyView
- ContactView

### Bean

quickcheckoutsummary.jsp 使用以下 Bean:

- OrderBean
- AddressDataBean
- ErrorDataBean
- ProfileCassetteAccountDataBean
- OrderPaymentInfoAccessBean

实现详细信息

**注意：**关于所有样本商店页面共用的实现技术的信息（包含多文化信息），请参阅样本商店页面：公共实现技术。

快速结帐摘要页面的行为方式如同样本商店订单摘要页面，但是有以下不同：

- 支付信息已用来自快速结帐摘要表的信息预填充。
- 使用 `OrderPaymentInfoAccessBean` 检索支付信息。

---

## “网络时尚”和“新时尚”样本商店查看订单页面

在注册顾客下订单之后，他们可以在任何时候查看此订单的状态。要查看一个订单，顾客单击**我的帐户**，然后从“我的帐户”页面中，单击**查看订单**。显示“订单状态”页面（`trackorderstatus.jsp`），其中带有顾客已经下的订单的列表。关于特定订单的更多信息，顾客单击此订单。显示“订单详细信息”页面（`orderdetail.jsp`）。

**注意：**不能查看：临时、未决、CSR 编辑、快速订单摘要表、私人需求列表、可共享需求列表、无库存订单以及已取消订单。

关于更多信息，请参阅查看订单使用案例。

### Bean

`trackorderstatus.jsp` 使用以下 Bean:

- `OrderDataBean`
- `PayStatusListPMDDataBean`
- `OrderAccessBean`
- `PriceDataBean`

`orderdetail.jsp` 使用以下 Bean:

- `OrderDataBean`
- `OrderItemDataBean`

### 实现详细信息

**注意：**关于所有样本商店页面共用的实现技术的信息（包含多文化信息），请参阅样本商店页面：公共实现技术。

当顾客单击**查看订单**时，显示“订单状态”页面（`trackorderstatus.jsp`），其中带有顾客已经下的订单列表。`OrderDataBean` 检索所有由该顾客下的订单，使用以下代码在列表中的所有订单中循环：

```
<jsp:useBean id="orderABFinder"
class="com.ibm.commerce.order.objects.OrderDataBean"
scope="page" />
Enumeration ordersABList = orderABFinder.findByMemberForUpdate(userId);
```

`OrderDataBean` 显示以下订单信息：

- 订单号
- 订单日期
- 总金额

- 支付状态

OrderDataBean 通过调用 `.getStatus()` 方法检索支付状态。`.getStatus()` 方法返回一个单字符状态字符串，它代表如下表所示的支付状态。

状态	简短描述	含义
P	未决	顾客可以修改此订单。
I	已提交	顾客无法修改此订单。
W	正在审批核准	CheckOrderApproval 任务命令指示一些订购的商品没有被核准。
N	已拒绝核准	CheckOrderApproval 任务命令指示对一些订购商品已拒绝了核准。
M	未决支付授权	等待支付授权
A	支付授权需要复查	支付授权遇到不常见的情况，例如地址验证警告。应当复查并接受支付授权，否则使用“订单管理”用户界面取消订单。如果接受授权，则用户界面将订单状态相应的更改为“B”或“C”。
B	延迟交货订单	支付授权可能需要重新执行，因为订单金额可能由于延迟交货订单库存分配而更改。
C	支付授权完成	支付授权已经完成。从现有的库存中分配所有订购商品。订单金额将不会更改。
E	CSR 编辑	客户服务代表正在处理订单。
R	已发货	所有订购商品已经发货以供实现。
S	已装运	所有订购商品都已装运。
D	已圈出	支付交易成功。
L	无库存	对于一个或多个订购商品，库存不可用。
T	临时	由“订单管理”用户界面使用来临时备份一个订单。
Q	快速订单简要表	订单保存有缺省订单信息，可以复制此信息来快速创建新的订单。
F	准备就绪等待远程实现	订单已经准备好被发送到远程系统以供实现。这个状态由MQAdapter 功能使用。
G	正在审批远程实现	订单已经被发送到远程系统以供实现。这个状态由MQAdapter 功能使用。
Y	专用需求列表	此订单是专用需求列表。



Z	可共享的需求列表	此订单是可共享的需求列表。
X	已取消	已取消此订单。

**注意:** Payment Manager 检查是否已经拒绝支付。如果支付状态是未拒绝, 则根据 .getStatus() 方法返回的内容显示消息。

```
if (payStatusBean.getPaymentState(sOrderId).equalsIgnoreCase("PAYMENT_VOID")
||
payStatusBean.getPaymentState(sOrderId).equalsIgnoreCase("PAYMENT_DECLINED"))
{
```

一旦顾客检索了他们的订单列表, 他们就可以单击指定的订单以获得更多信息 (orderdetail.jsp)。OrderDataBean 检索顾客的所有订购商品, OrderItemDataBean 检索关于每个商品的详细信息。OrderItemDataBean 检索以下详细信息:

- 订单的价格
- 订单的内容
- 订单的预计装运日期或实际装运日期
- 每个商品的跟踪号

每宗商品可能由多件具有不同装箱的货品组成, 且可能具有多于一个跟踪号。多宗商品可以共享同一跟踪号。

OrderDataBean 检索订单的预计装运日期或实际装运日期, 如以下代码所示:

```
orderDate = orderABFinder.findByOrderForUpdate(new
Long(orderId)).getActualShipDate();
```

---

## 网络时尚和新时尚样本商店希望列表页面

样本商店希望列表允许注册顾客将商品添加到他们的希望列表 (兴趣列表), 然后查看和编辑该列表 (interestitemdisplay.jsp), 根据自己的喜好将产品添加到购物车中。顾客还可以通过电子邮件 (sendwishlistmsg.jsp) 将他们的希望列表发送给家人和朋友。在新时尚商店中, 顾客可以发送带有希望列表的个人消息。

**注意:** 新时尚同时允许临时购物者和注册顾客向希望列表添加商品。网络时尚仅允许注册顾客向希望列表添加商品。

当顾客通过电子邮件接收希望列表时, 他们看见的是希望列表页面 (sharedwishlist.jsp)。除了没有**发送希望列表**和**除去商品**按钮之外, 此页面等同于 interestitemdisplay.jsp。顾客可以选择购买一个商品并将其添加到他们的购物车中。

有关更多信息, 请参阅查看希望列表使用案例和向希望列表添加商品使用案例。

样本商店希望列表过程使用以下 JSP 文件:

- interestitemdisplay.jsp (希望列表页面)
- sharedwishlist.jsp
- sendwishlistmsg.jsp

## 命令

interestitemdisplay.jsp 使用以下命令:

- OrderItemAdd
- SendWishListMsg

sharedwishlist.jsp 使用以下命令:

- OrderItemAdd

sendwishlistmsg.jsp 使用以下命令:

- SendMsgCmd

## Bean

interestitemdisplay.jsp 使用以下 Bean:

- InterestItemListDataBean
- InterestItemDataBean
- CatalogEntryAccessBean
- UserRegistrationDataBean

sharedwishlist.jsp 使用以下 Bean:

- InterestItemListDataBean
- InterestItemDataBean
- CatalogEntryAccessBean

sendwishlistmsg.jsp 使用以下 Bean:

- UserRegistrationDataBean (仅用于网络时尚)

## 实现详细信息

**注意:** 关于所有样本商店页面共用的实现技术的信息 (包含多文化信息), 请参阅样本商店页面: 公共实现技术。

### 添加至希望列表

当顾客单击**添加至希望列表**时, 将调用 `InterestItemAdd` 命令。有关更多信息, 请参阅样本商店产品页面。

在向希望列表添加商品之前, 系统使用 `UserRegistrationDataBean` 检查顾客是否已在商店中注册。如果顾客没有注册, 则 JSP 包含 `registerfirst.jsp`, 它接着显示一条消息提示顾客注册并再次尝试。

### 查看希望列表

当顾客单击**查看希望列表**时, 显示希望列表页面 (`interestitemdisplay.jsp`) 并带有希望列表内容的列表。`InterestItemListDataBean` 和 `InterestItemDataBean` 检索有关希望列表内容的信息。

### 添加至购物车和除去商品

在希望列表页面 (`interestitemdisplay.jsp`) 中, 顾客可以选择向他们的购物车添加商品或从希望列表中删除商品。当顾客单击**向购物车添加选定的商品**时, 将调用 `OrderItemAdd` 命令。当顾客单击**除去商品**时, 将调用 `InterestItemDelete` 命令。

## 发送希望列表

当顾客完成姓名和电子邮件地址字段并单击发送希望列表时，将调用 SendWishListMsg 命令。SendWishListMsg 在 VIEWREG 表中注册以显示 sendwishlistmsg.jsp。

在 SendWishListMsg 命令可以运行之前，你必须创建一个实例。网络时尚和新时尚使用以下命令创建一个实例：`SendMsgCmd sendMsgCmd = (SendMsgCmd) CommandFactory.createCommand(cmdEntry);`

例如：

```
CommandRegistryEntry cmdEntry =  
CommandFactory.locateCommandEntry("com.ibm.commerce.messaging.commands.SendMsgCmd",  
new Integer(storeId));  
SendMsgCmd sendMsgCmd = (SendMsgCmd)  
CommandFactory.createCommand(cmdEntry);
```

您可以设置“sendMsgCmd”任务命令的参数以立即发送消息：`sendMsgCmd.sendImmediate();`

**注意：**网络时尚仅有收件人的电子邮件地址字段。新时尚有收件人电子邮件字段、发件人姓名字段以及个人消息字段。收件人电子邮件字段和发件人姓名是必需的。

## 错误处理

如果顾客在希望列表中没有选择任何商品而单击向购物车添加选定的商品，将显示一个错误消息。以下代码执行此操作：



```
function checkForm(form)  
{  
var hasItem  
var i, e  
hasItem = false  
for (i = 0; i < form.elements.length; i++)  
{  
e = form.elements[i]  
if (e.type == "checkbox")  
{  
if (e.checked)  
{  
hasItem = true  
break  
}  
}  
}  
if (hasItem)  
form.submit()  
else  
alert("<%=infashiontext.getString("SELECTITEMS")%>")  
}  
</script>
```

---








## 样本商店使用案例

使用案例详述商店中每个用户交互的流程，例如：注册或结帐。在联机帮助中提供了一系列使用案例，详细描述了样本商店流行时尚、网络时尚和新时尚的交互。这些使用案例可以帮助您更彻底地理解样本商店的流程，而且它们可以用作创建自己商店的使用案例的向导。

### 注意:

1. 网络时尚是“流行时尚”商店的专业版。仅用于“网络时尚”的页面和功能由  标明。
2. 下面列出的未由  标明的页面可包含一些专业版功能。

提供了以下使用案例:

- 主页使用案例
- 注册使用案例
- 登录使用案例
- “管理个人帐户”使用案例
- 更改个人信息使用案例
- 查看产品类别使用案例
- 显示产品页面使用案例
-  显示打包销售商品页面使用案例
-  显示捆绑销售商品页面使用案例
- 显示购物车使用案例
- 购物车结帐使用案例
-  快速结帐使用案例
- 编辑地址使用案例
- 添加新地址使用案例
-  查看订单使用案例
-  向希望列表添加商品使用案例
-  查看希望列表使用案例
-  创建快速结帐简要表使用案例

有关这些页面如何一起工作的更多信息，请参阅样本商店购物流程图表。

---

## 第 5 章 样本商店使用案例

---

### 添加新地址使用案例（商务版）

---

**Business** 顾客可将新地址添加到他们的通讯录。

#### 操作者

顾客

#### 主要流程

顾客单击**添加新地址**。然后系统显示带有以下字段的页面：

- 昵称（仅用于新时尚）
- 名字
- 姓氏
- 街道地址（两个文本框组成）
- 市 / 县 / 区
- 省 / 直辖市
- 邮政编码
- 国家或地区
- 电话号码

顾客在字段中输入信息并单击**提交**。系统将一个新地址添加至通讯录（E1）中。

#### 备用流程

无

#### 异常流程

##### **E1: 遗漏的必需字段**

如果任一以下字段遗漏，系统将发出错误消息。

- 名字
- 姓氏
- 街道地址
- 市 / 县 / 区
- 省 / 直辖市
- 邮政编码
- 国家或地区

如果输入的昵称已在顾客的通讯录中存在，则系统将报告错误消息。

此使用案例恢复到开始状态。

---

## 网络时尚和新时尚向希望列表添加商品使用案例

通过购物车，顾客可以查看并编辑已选定要购买的产品。

在网络时尚商店，希望列表（或兴趣列表）使注册顾客能够向列表添加其日后要订购的产品。在新时尚商店，顾客无需注册即可向希望列表添加商品。希望列表可以用电子邮件的方式发送给家人或朋友，他们接着可以购买此商品作为给这位顾客的礼物。希望列表与购物车的不同在于顾客打算在当前的购物会话期间购买购物车中的产品。

### 操作者

注册顾客

### 主要流程

顾客查看产品、打包销售商品或捆绑销售商品（有关更多信息，请参阅显示产品页面使用案例、显示捆绑销售商品页面使用案例和显示打包销售商品页面使用案例），然后单击**添加至希望列表**。系统将产品、捆绑销售商品或打包销售商品添加到希望列表（E1）中，然后显示希望列表页面（如查看希望列表使用案例中所述）。

### 异常流程

#### E1: 临时购物者试图将商品添加到希望列表

如果临时购物者试图将商品添加到希望列表，则系统显示以下消息：要将商品保存到您的希望列表，请在网络时尚注册，然后重试。

在新时尚商店中，临时购物者可以在不注册的情况下将商品添加到希望列表。

---

## 查看产品类别使用案例

产品目录组页面显示子类别和产品的列表。产品目录组页面帮助顾客浏览产品。开头的产品目录组页面通向很广的范围，而接下来的产品目录组页面会将搜索范围缩减。通常有三种类型的产品目录组页面：

- 显示父类别中的子类别的页面
- 显示子类别中产品的页面
- 显示子类别和产品的页面

### 操作者

顾客

### 主要流程

顶级产品类别列在主页中，例如，男士时装、女士时装以及新品。当顾客单击“男士时装”或“女士时装”时，系统将从数据库在选定的类别中检索子类别，并显示相应的类别信息。当顾客单击“新到货品”链接时，将显示新产品。为每个类别和产品还将显示以下信息：

- 略图（仅用于产品）
- 每个类别或产品的名称
- 标识产品的简短描述。
- 产品价格。

新时尚和网络时尚的类别页面还包含了以打包销售商品或捆绑销售商品形式出现的富有特色的特价商品。特别的特价商品包含打包销售商品或捆绑销售商品的图像和简短描述。流行时尚富有特色的特价商品仅是单一产品。

顾客单击略图或名称。然后，系统显示相应的页面（A1, A2, A3, A4）。

### 备用流程

#### A1: 顾客选择类别

当顾客选择类别时，系统通过恢复此使用案例至初始状态显示不同的类别页面。

#### A2: 顾客选择产品

当用户选择产品时，如产品页面使用案例中所描述显示产品页面。

#### A3: 顾客选择捆绑销售商品（仅用于网络时尚和新时尚）

当顾客选择捆绑销售商品时，将显示捆绑销售商品页面（如显示捆绑销售商品页面使用案例中所述）。

#### A4 顾客选择打包销售商品（仅用于网络时尚和新时尚）

当顾客选择打包销售商品时，将显示打包销售商品页面（如显示打包销售商品页面使用案例中所述）。

### 异常流程

无

---

## 购物车结账使用案例

顾客在对他们的购物车中列出的产品进行支付时结账并提交订单。

### 操作者

顾客

### 主要流程

顾客单击**结账**初始化此使用案例。

系统显示“开票地址”页面。如果顾客在通讯录中有至少一个地址，系统提示顾客从通讯录中选择一个地址作为开票地址。顾客可以选择一个地址或单击**创建新地址**。如果顾客单击**创建新地址**，则使用添加新地址使用案例添加一个新地址。然后顾客选择一个地址。之后，系统将所选的地址设置为订单的开票地址。如果顾客在通讯录中不具有任何地址，将执行 A2 “输入开票地址”。

系统显示“送货地址”页面。显示通讯录中的一个地址列表。如果顾客在列表中找到适当的送货地址，则选择此地址。否则，顾客单击**创建新地址**并使用添加新地址使用案例添加新地址。系统将选定的地址设置为订单的送货地址。

系统显示一个相应于送货地址的装运方式的列表。对每种装运方式还将显示以下信息：

- 简短描述
- 装运费用:

- 每份订单的固定装运费用
- 订购的每项商品的装运费用
- 适当的交付时间（仅用于“流行时尚”和“网络时尚”）

顾客选择装运方式。系统将选定的方式设置为订单的装运方式。

注意：在新时尚商店中，为每个订购商品指定送货地址和装运方式。

### 仅用于新时尚

在新时尚样本商店，系统显示订单中商品的可用性，以及基于其可用性的数个装运商品选项。对于订单中的每个产品，系统显示：

- 数量
- 简短产品描述
- 属性值（例如，大小：x 表示大号；颜色：蓝色）
- 根据将来库存得出的预期有效日期
- “除去”按钮（从订单中除去商品）

如果对指定商品的请求数量不在库存中，则系统将订单分割成两个订单，一个包含可用订单商品，另一个包含库存不可用部分。顾客选择三个装运首选项之一：

- 选项 1，在装运前等到整个订单都准备就绪。系统显示整个订单的预期有效日期。
- 选项 2，现在先装运可提供的商品，稍后再装运剩下的。
- 选项 3，现在先装运可提供的商品，将剩下的留在购物车中稍后再购买。

然后顾客选择订单的装运首选项。

**注意：**如果所有商品都是可用的，则顾客将不会看到每个商品的有效日期。

有关新时尚库存子系统的更多信息，请参阅库存子系统。

### 用于流行时尚、网络时尚和新时尚

系统不仅显示产品信息，还显示订单的详细信息。对于订单中的每个商品，系统显示：

- 简短产品描述
- 属性值（例如，大小：x 表示大号；颜色：蓝色）
- 数量
- 单价
- 总价格
- 系统还显示：
  - 开票地址
  - 送货地址
  - 小计（所有订购产品的总费用）
  - 折扣（如果有的话）
  - 总税款（如果有的话）（多项税款应当分开显示）
  - 关税（如果有的话）



- 装运费用包含装运方式
- 总计（顾客将支付的费用总计）
- 预计装运日期（显示在顶部）（仅用于新时尚）

注意：在新时尚商店中，为订单中的每个商品显示送货地址和装运方式。

系统提示顾客输入信用卡信息。将请求输入以下信息：

- 信用卡类型（例如 Visa(R) 或 MasterCard(R)）
- 卡号
- 失效月份
- 失效年份

顾客输入信用卡信息并单击**立即订购**。系统存储支付信息（E1）并显示确认页面，带有以下有关订单的信息：

- 订单号
- 小计（所有订购产品的总费用）
- 总税款
- 装运费用
- 折扣（如果有的话）
- 总计

顾客打印此页面以供以后引用。

### 备用流程

#### A1: 添加开票地址

系统提示顾客添加新地址。顾客使用 添加新地址使用案例输入地址。系统将此新地址设置为订单的开票地址。使用案例继续进行。

### 异常流程

#### E1: 无效信用卡号

系统将检查信用卡号的有效性。如果检查失败，则系统显示错误消息声称：  
无效信用卡号

并提示顾客重新输入此信息。使用案例继续进行。

#### E2: 信用卡失效

如果信用卡的失效日期早于当前日期，系统显示一个错误消息。此使用案例恢复到开始状态。

**注意：**如果某个商品不可用，则将向顾客显示“产品可用性”页。

---

## 网络时尚和新时尚显示捆绑销售商品页面使用案例

捆绑销售商品页面显示有关构成捆绑销售商品的产品或商品的详细信息。

### 操作者

顾客

### 主要流程

顾客在类别页面中选择捆绑销售商品。然后系统从数据库中检索捆绑销售商品信息，然后显示一个页面，其中带有有关选定的捆绑销售商品的详细信息。捆绑销售商品页面显示以下信息：

- 简短描述
- 详细描述
- 捆绑销售商品的全尺寸图像
- 构成捆绑销售商品的产品的图像
- 捆绑销售商品中每个产品的价格
- 属性列表（大小和颜色，如果构成捆绑销售商品的产品具有不同变化的话），以及每个属性值的列表（对于颜色而言就是红和蓝，对于大小而言就是大号 and 特大号）

然后顾客为每个属性选择相应的值并单击**添加至购物车**。在新时尚商店，顾客可以指定添加到购物车的数量。系统添加此捆绑销售商品至购物车（E-1），并显示如显示购物车使用案例中所述的“购物车页面”。

**注意：** 缺省数量是 1。顾客可在添加捆绑销售商品至购物车后修改数量。

顾客还可以选择单击**添加至希望列表**。如果顾客这样做了，则系统将捆绑销售商品添加到希望列表，并如向希望列表添加商品使用案例中所述显示希望列表。

### 备用流程

无

### 异常流程

无

---

## 网络时尚和新时尚显示打包销售商品页面使用案例

打包销售商品页面显示有关构成打包销售商品的产品或商品的更多详细信息。

### 操作者

顾客

### 主要流程

顾客在类别页面中选择打包销售商品。系统从数据库中检索打包销售商品信息，然后显示一个页面，其中带有有关选定的打包销售商品的详细信息。打包销售商品页面显示以下信息：

- 简短描述
- 详细描述
- 构成打包销售商品的产品的图像
- 打包销售商品的价格
- 属性列表（大小和颜色，如果构成打包销售商品的产品具有不同变化的话），以及每个属性值的列表（对于颜色而言就是红和蓝，对于大小而言就是大号和特大号）

然后顾客为每个属性选择相应的值并单击**添加至购物车**。系统向购物车（E-1）添加打包销售商品，并如显示购物车使用案例中所述显示“购物车”页面。在新时尚商店中，顾客可以指定要添加到购物车的数量。缺省数量是 1。

**注意：**由于一个打包销售商品中的所有商品不能单独出售，因此一个打包销售商品在购物车中的计数为 1 件商品。

顾客还可以选择单击**添加至希望列表**。如果顾客这样做了，则系统将打包销售商品添加到希望列表，并如向希望列表添加商品使用案例中所述显示希望列表。

#### 备用流程

无

#### 异常流程

##### E1: 属性未列出

对于具有多种属性的产品，顾客必须为每个属性选择一个值。如果顾客没有选择值，则系统缺省值，并将打包销售商品添加到购物车或希望列表。

---

## 编辑地址使用案例（商务版）

**Business** 顾客可编辑其通讯录中的地址。

#### 操作者

顾客

#### 主要流程

顾客从通讯录中选择一个地址以退出。系统将检索并显示选定地址的详细信息：

- 昵称（仅用于新时尚）
- 名字
- 姓氏
- 街道地址
- 市/县/区
- 省/直辖市
- 邮政编码
- 国家或地区
- 电话号码

在“新时尚”中，顾客输入的昵称显示在页面的顶部。

顾客作出想要的更改并单击**提交**。系统更新地址（E1）。

#### 备用流程

无

#### 异常流程

**E1:** 系统通知顾客遗漏了必需信息，并请求这些可能包括以下内容的遗漏信息：

- 名字
- 姓氏
- 街道地址
- 市 / 县 / 区
- 省 / 直辖市
- 邮政编码
- 国家或地区

如果昵称在通讯录中已存在，则将显示错误消息。

顾客输入遗漏了必需信息。

---

## 主页使用案例

主页作为商店前台，带领顾客进入您的商店。

#### 操作者

顾客

#### 主要流程

顾客在 Web 浏览器中输入商店 URL。然后系统显示此主页。

主页包含以下内容：

- 带有到以下页面链接的导航栏：
  - 主页
  - 联系信息页面
  - 安全性和保密性页面
  - 购物车页面
  - 注册页面
  - 帐户页面
  - 帮助页面
- 到主要或顶级类别的链接。对于每个类别，显示以下信息：
  - 图像

- 到目标产品的链接。

如果顾客在注册时指定了他们的性别，则显示针对顾客性别的产品。如果顾客没有指定性别，则显示产品的一般集合。

对于每个产品，显示以下信息：

- 图像
- 简短描述

顾客单击图像。然后，系统显示相应的页面（A1，A2）。

### 备用流程

#### A1: 顾客选择类别

当顾客选择类别时，系统如查看产品类别使用案例中所述显示一个类别页面。

#### A2: 顾客选择产品

当用户选择产品时，如产品页面使用案例中所描述显示产品页面。

---

## 登录使用案例

登录过程允许注册购物者通过输入他们的用户名和密码访问账户。

### 操作者

顾客

### 主要流程

顾客选择**我的帐户**。然后系统显示一个页面，其中带有以下字段：

- 电子邮件地址
- 密码

顾客在以上字段中输入相应的信息，并选择**登录**。系统确保顾客的电子邮件地址和密码都是正确的，然后系统允许顾客进入他们的帐户。如果顾客忘记了密码，则他们选择**忘记密码**，并执行备用流程 A1。

### 备用流程

#### A1: 忘记密码

如果顾客忘记了密码，则他们选择**忘记密码**。然后系统显示一个页面，提示顾客输入电子邮件地址。顾客输入他们的电子邮件地址并单击**将密码发送给我**。然后系统发送密码至该电子邮件地址（E1）。

### 异常流程

#### E1: 系统中无匹配的电子邮件地址

如果系统用匹配的电子邮件地址无法定位一个顾客，将显示以下错误消息：无法用此电子邮件地址定位顾客。此使用案例异常终止。

如果顾客试图用同一用户名登录并失败了 6 次，则此顾客将被封锁在外。将显示消息说明帐户被锁住，且顾客必须联系商店代表以重新激活它。

---

## 管理个人帐户使用案例

顾客通过帐户页面管理他们的帐户。

### 操作者

顾客

### 主要流程

顾客选择**我的帐户**。之后系统显示带有以下选项的“我的帐户”页面：

- 更改个人信息
- 编辑通讯录
- 创建或更新快速结帐简要表（仅用于网络时尚）
- 查看希望列表
- 查看订单

如果顾客选择**更改个人信息**，则执行备用流程 A1: 更改电子邮件和密码。

如果顾客选择**编辑通讯录**，则执行备用流程 A2: 编辑通讯录。

如果顾客选择**创建或更新简要表**（快速结帐简要表），则如创建快速结帐简要表使用案例中所述显示快速结帐简要表页面。

如果顾客选择**查看希望列表**，则如查看希望列表使用案例中所述显示希望列表页面。

如果顾客选择**查看订单**，则如查看订单使用案例中所述显示订单状态页面。

### 备用流程

#### A1: 编辑电子邮件和密码

系统提示顾客使用更改个人信息使用案例中所描述的过程来修改电子邮件地址和密码。此使用案例恢复到开始状态。

#### A2: 编辑通讯录

系统显示一个页面，其列出所有已添加至通讯录的地址。在每个地址旁有两个按钮：**编辑**和**删除**。在地址列表的下面是一个**添加新地址**按钮。

如果顾客单击**删除**，系统从数据库删除相应的地址，然后显示一条消息指示此地址已正确删除。备用流程从头开始恢复。

如果顾客单击**编辑**，则系统提示顾客使用“编辑地址”使用案例中所述的过程来编辑地址。系统显示一个确认地址正确更新的消息，然后改变流至初始状态。

顾客单击**添加新地址**，则系统提示顾客使用添加新地址使用案例中所述的过程输入新的地址。该备用流恢复到开始状态。

### 异常流程

无。

---

## 网络时尚和新时尚查看订单使用案例

顾客可以跟踪和查看其订单的状态。

### 操作者

顾客

### 主要流程

顾客单击**我的帐户**，然后单击**查看订单**。系统显示订单状态页面，其中列出了该顾客所下的所有订单。如果顾客没有下任何订单，则执行 E1: 未发现订单。

对每个订单显示以下信息:

- 订单号（链接到订单详细信息页面）
- 订单日期
- 订单状态
- 总金额

要找到有关订单的更多信息，顾客可单击订单号。系统显示“订单详细信息”页面。订单详细信息页面显示有关订单中每个产品的以下信息:

- 数量
- 简短产品描述
- 属性（大小和颜色），以及每个属性的值（对于颜色而言就是红和蓝，对于大小而言就是大号 and 特大号）
- 单价
- 总价格
- 整个订单的装运日期（仅用于新时尚）
- 跟踪标识号（仅用于新时尚）

### 异常流程

#### E1: 未发现订单

如果系统找不到任何与此顾客关联的订单，则系统显示一条错误消息，声称：  
未发现订单

然后顾客可以单击**返回主页**或**我的帐户**。

---

## 显示产品页面使用案例

产品页面显示有关产品的更多详细信息。

### 操作者

顾客

### 主要流程

顾客在类别页面中选择一个产品。之后系统从数据库中检索产品信息，并显示带有关于所选产品的详细信息的页面。此“产品”页面显示以下信息:

- 产品名称
- 详细描述
- 价格
- 产品的全尺寸图像
- 属性列表（大小和颜色，如果产品具有不同变化的话），以及每个属性值的列表（对于颜色而言就是红和蓝，对于大小而言就是大号 and 特大号）

然后顾客为每个属性选择相应的值并单击**添加至购物车**。系统在购物车（E1）中添加选定的商品，并如显示购物车使用案例中所述显示“购物车”页面。在新时尚商店中，顾客可以指定要添加到购物车的数量。缺省数量是 1。

在网络时尚和新时尚，顾客可选择单击**添加至希望列表**。如果顾客这样做了，则系统将打包销售商品添加到希望列表，并如向希望列表添加商品使用案例中所述显示希望列表。

#### 备用流程

无

#### 异常流程

无

---

## 更改个人信息使用案例

顾客可以更改个人信息，例如电子邮件地址和密码。

#### 操作者

顾客

#### 主要流程

顾客单击**更改个人信息**（E1）。系统显示“更改个人信息”页面，此页面包括以下字段：

- **名字**，用当前名字预填充。
- **姓氏**，用当前姓氏预填充。
- **电子邮件地址**，用当前电子邮件地址预填充。
- **密码**
- **验证密码**
- **年龄**，用当前年龄预填充。
- **性别**，用性别预填充（仅用于网络时尚）
- **首选语言**（仅用于新时尚）
- **首选货币**（仅用于新时尚）

顾客按照需要改变字段。密码字段可以留下为空。如果顾客选择留下密码字段为空，则系统不能修改当前密码。然后顾客选择**提交**，系统更新电子邮件地址和密码（E2）。如果密码字段没有填写，保留未改动的先前密码。



## 备用流程

无

## 异常流程

输入的密码必须至少为 6 个字符，且包含至少 1 个数字，包含至少 1 个字母，不可包含同一字符总共 5 次或连续 4 次，否则将显示错误消息。如果密码与电子邮件地址相同，或顾客决定更改密码，但更改后的密码与原先的密码相同，也将显示错误消息。有关更多信息，请参阅以下相关任务。

### E1: 顾客没有登录

顾客必须登录才能更改电子邮件地址和密码。如果顾客没有登录，则系统提示顾客使用登录使用案例中所述的过程登录。此使用案例恢复到开始状态。

### E2: 对于不同的顾客已经存在电子邮件

输入的电子邮件地址无法与任何其它现有的顾客的电子邮件地址匹配。如果系统发现一个匹配，则它提示顾客选择一个不同的电子邮件地址。此使用案例恢复到开始状态。

### E3: 无法验证密码

输入的密码必须与在验证密码中指定的密码相匹配。否则，系统显示以下错误消息：密码与验证密码不匹配，请重新输入。  
此使用案例恢复到开始状态。

---

## 注册使用案例

注册进程允许顾客在数据库中输入个人信息。

### 操作者

顾客

### 主要流程

顾客选择**注册**。然后系统显示带有以下字段的页面：

- 电子邮件
- 密码
- 验证密码
- 名字
- 姓氏
- 年龄（可选）（仅用于网络时尚）
- 性别（可选）（仅用于网络时尚）
- 首选语言（仅用于新时尚）
- 首选货币（仅用于新时尚）

顾客在上面的字段中输入相应的信息，然后选择**提交**。系统中创建一个新的顾客，并保存顾客的信息（E1、E2、E3）。

系统提示顾客遵循在管理个人帐户使用案例中的进程来管理它们的帐户。

## 备用流程

无。

## 异常流程

### E1: 电子邮件地址已存在

如果系统已存在该电子邮件地址，则系统显示错误消息，要求用户输入不同的电子邮件地址。此使用案例恢复到开始状态。

### E2: 遗漏必需字段

如果以下字段（电子邮件、验证密码、名字和姓氏）中的任何一个没有完成，系统都会发出一条错误消息。此使用案例恢复到开始状态。

### E3: 非法密码

如果密码与验证密码不匹配，系统发出一个警告。

**注意：**输入的密码必须至少为六个字符，且包含至少一个数字和一个字母；对于同一字符，不可包含其总共五次或连续四次，否则将显示错误消息。如果密码与电子邮件地址相同，也将显示错误消息。

---

## 显示购物车使用案例

通过购物车，顾客可以查看并编辑已选定要购买的产品。

### 操作者

顾客

### 主要流程

顾客单击**购物车**。之后系统显示一个页面，上面包含购物车的内容。显示购物车中产品的列表（E1）。对于每个产品，显示以下信息。

- 简短产品描述
- 商品的“属性名称和属性值”对，例如尺寸和大号。
- 数量
- 单价
- 总价格

系统还显示在税款和装运前订单的总金额。

如果顾客单击**更新总额**，则执行 A1: 更新总额。

如果顾客单击**除去商品**，则执行 A2: 从购物车中除去商品。

如果顾客单击**结帐**，则执行 A3: 更新购物车并结帐。

如果顾客单击**返回购物**，则系统显示顾客来到“购物车”页面之前所在的那个页面。使用案例终止。

如果顾客单击“快速结帐”，则系统如快速结帐使用案例中所述显示“快速结帐：订单摘要”页面

### 子流程

#### A1: 更新总额

要更新任何商品的数量，顾客在文本框中输入新的数量，然后单击**更新总额**。系统更新购物车里的商品数量。如果商品新数量是 0，则从车里除去该商品。

#### A2: 从购物车中除去商品

要从购物车中删除一个商品，顾客对此商品单击**除去商品**。系统从车中除去商品。

#### A3: 更新购物车并结帐

系统更新购物车里的商品数量并启动结帐过程。请参阅“购物车结帐使用案例”，购物车结帐使用案例。

### 异常流程

#### E1: 清空购物车

如果在购物车中没有商品，则系统显示以下消息：您的购物车是空的。

如果在数量字段输入了无效数字（例如一个字符），将重新显示带错误消息的购物车页面。

---

## “网络时尚”和“新时尚”查看希望列表使用案例

顾客可以查看和编辑已经添加到希望列表（或兴趣列表）的产品。

### 操作者

顾客

### 主要流程

顾客单击**我的帐户**，然后单击**查看希望列表**。然后系统显示一个页面，显示希望列表的内容。显示希望列表中产品的列表（E1）。对于希望列表中的每个产品、捆绑销售商品或打包销售商品，将显示以下信息。

- 简短产品描述
- 商品的属性和值，例如，大小和颜色的值
- 单价
- 删除链接

系统在页面中也显示以下项：

- **返回购物按钮**
- **向购物车添加选定的商品按钮。**
- **发送希望列表按钮**
- **电子邮件地址字段（至：/收件人）**
- **电子邮件消息字段（仅用于“新时尚”）**
- **从：/发件人姓名（仅用于“新时尚”）**
- **电子邮件地址（仅用于“新时尚”）**

要从希望列表中删除一个商品，请单击**除去**。系统完成 A1 除去商品。

要将希望列表通过电子邮件发送给朋友和家人，顾客在**电子邮件地址**字段中输入收件人的电子邮件地址，并单击**发送希望列表**。系统完成 A2 发送希望列表。在“新时尚”商店中，顾客可以使用**电子邮件消息**字段发送电子邮件消息。顾客需要指定其姓名，并可选择指定其电子邮件地址。注册顾客的**姓名**字段已预先填好。

要将希望列表中的商品添加到购物车，顾客选择她想要添加到购物车的每个商品，并单击**向购物车添加选定的商品**。系统完成“A3 添加至购物车”。

## 备用流程

### A1: 除去商品

要从购物车中删除一个商品，顾客对此商品单击**除去**。系统从希望列表中除去商品。此使用案例恢复到开始状态。

### A2: 发送希望列表

要将希望列表通过电子邮件发送给朋友和家人，顾客在**电子邮件地址**字段中输入收件人的电子邮件地址。要将希望列表发送给多个朋友，必须用逗号分隔每个电子邮件地址，例如：info@infashion.com、wcs@infashion.com。然后，顾客单击**发送希望列表**。

系统编写电子邮件消息，包含以下信息：

- 关于如何购买希望列表商品的指导
- 到希望列表的链接，它包含以下信息：
  - 希望列表中每个产品、打包销售商品或捆绑销售商品的简短描述
  - 到产品、捆绑销售商品或打包销售商品显示页面的链接
  - 价格
  - **向购物车添加选定的商品**按钮

在“新时尚”商店中，顾客可以通过在**电子邮件消息**字段中输入消息，将个性化的电子邮件消息与系统消息一起发送。

### A3: 添加至购物车

要将希望列表中的商品添加到购物车，顾客选择想要添加的每个商品，然后单击**向购物车添加选定的商品**。系统将所有选定的商品添加到购物车，然后如显示购物车使用案例中所述显示购物车。

## 异常流程

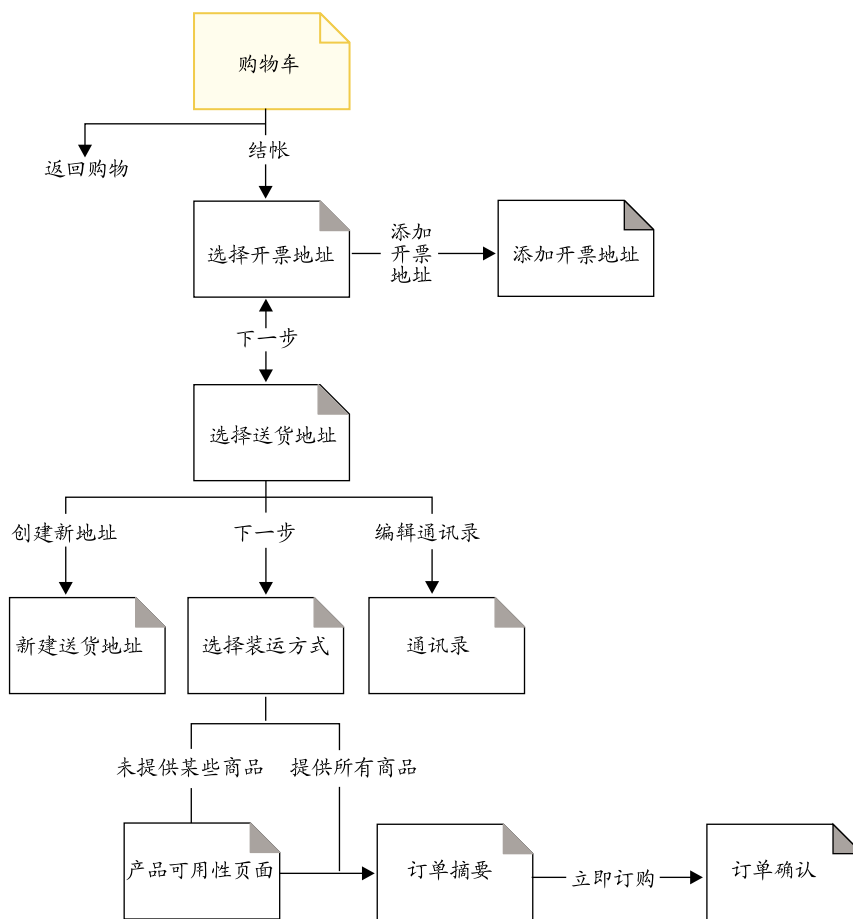
### E1: 希望列表是空的

如果在希望列表中没有商品，则系统显示以下消息：希望列表是空的。使用案例终止。


## 第 6 章 新时尚商店购物流程

### 新时尚样本商店购物车流程

以下图表详细描述了新时尚购物车流程。有关完整的购物流程的更多信息，请参阅以下相关参考。

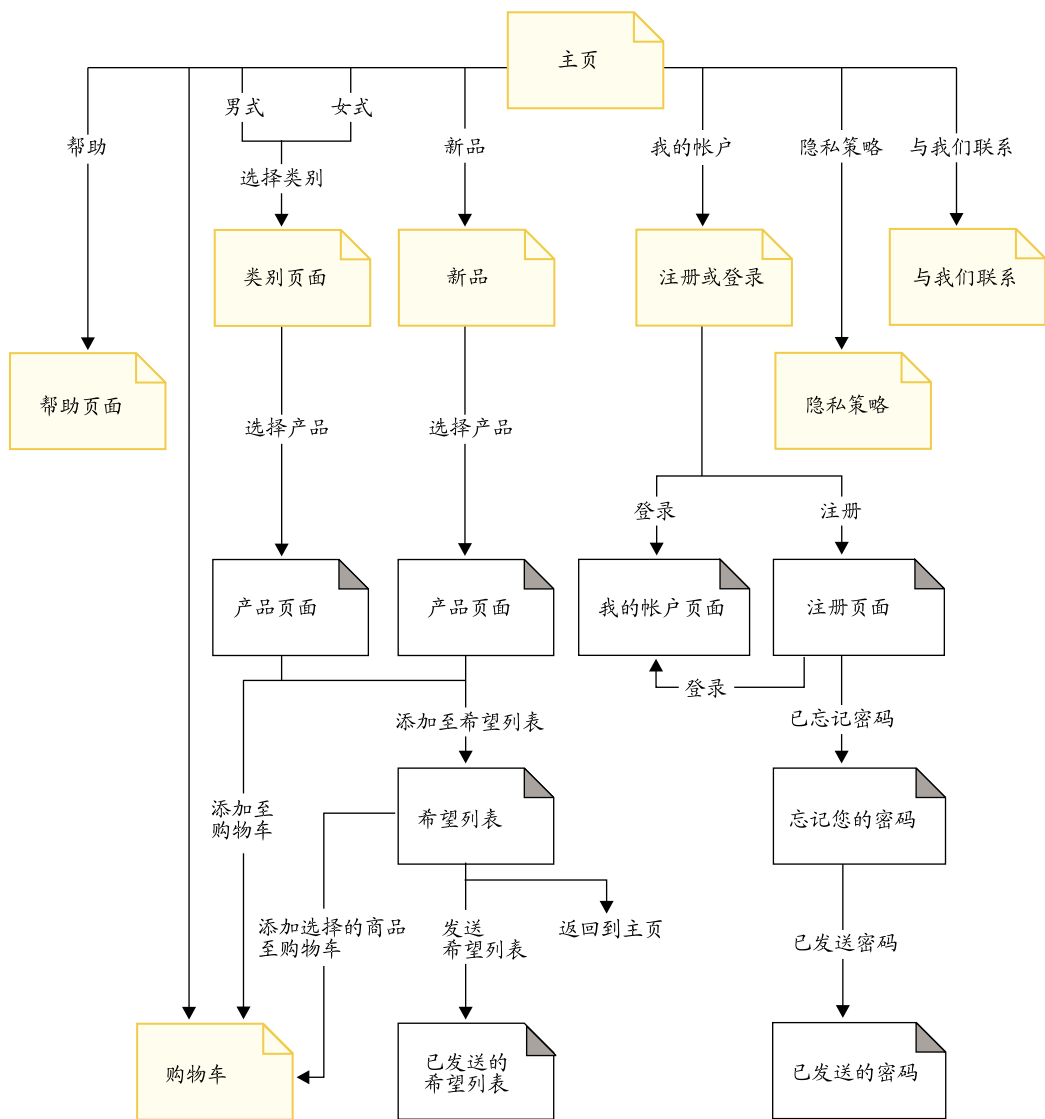


图注


 可以从本站点的任何页面访问这些页面。

### 新时尚样本商店购物流程

以下图表对新时尚购物流程中的顶级页面作详细描述：有关购物车和“我的帐户”流程的更多信息，请参阅以下相关参考部分。



图注

 可以从本站点的任何页面访问这些页面。

---

## 声明

任何对本文档中的 IBM 许可程序引用并不明示或暗示只能使用 IBM 的许可程序。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。在与其它产品结合使用时，除了那些由 IBM 明确指定的产品之外，其验证由用户自行负责。

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：

Director of Licensing  
Intellectual Property & Licensing  
North Castle Drive, MD-NC119  
Armonk, NY 10504-1785  
U.S.A.

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。

本文档可能包含日常商务运作中使用的数据和报表的示例。为了尽可能详尽地说明数据和报表，这些示例可能包含个人、公司、品牌和产品的名称。所有这些名称都是虚构的，任何与实际商务企业使用的名称和地址类似的名称和地址纯属巧合。

本文档可能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、完整性或使用不负任何责任。

本产品是基于 SET 协议的。

**Note to U.S. Government Users** — Documentation related to restricted rights — Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

### 商标

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

CICS

DB2

DB2 Extenders  
IBM  
SecureWay  
400

Encina  
iSeries  
VisualAge

HotMedia  
MQSeries  
WebSphere

Blaze Advisor 是 HNC Software, Inc. 在美国和 / 或其它国家或地区的商标。

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

Lotus 和 Domino 是 Lotus Development Corporation 在美国和 / 或其它国家或地区的商标。

Microsoft, Windows, Windows NT 和 Windows 徽标是 Microsoft Corporation 在美国和 / 或其它国家或地区的商标。

Oracle 是 Oracle Corporation 的注册商标。

SET 和 SET 徽标是 SET Secure Electronic Transaction LLC 的商标。有关更多信息, 请访问: <http://www.setco.org/aboutmark.html>。

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