

IBM® WebSphere Commerce



# 商店开发指南

版本 5.5



IBM® WebSphere Commerce



# 商店开发指南

版本 5.5

**注:**

在使用本资料及其支持的产品之前，请务必阅读『声明』部分中的信息。

**第一版，第一次修订（2003 年 9 月）**

本版本适用于 IBM WebSphere Commerce Business Edition V5.5、IBM WebSphere Commerce - Express V5.5 和 IBM WebSphere Commerce Professional Edition V5.5（产品号 5724-A18）及所有后续发行版和修订版，直到在新版本中另有说明为止。

本版本还适用于以上所列产品的所有后续发行版和修订版，直到在新版本中另有声明为止。确认您正在使用本产品级别的正确版本。

通过您当地的 IBM 代表或 IBM 分部可订购出版物。以下地址不备有出版物。

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

<http://www-3.ibm.com/software/genservers/commerce/rcf.html>

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

© Copyright International Business Machines Corporation 2000, 2003. All rights reserved.

# 目录

开始之前 . . . . .	ix
本书中使用的约定和术语 . . . . .	ix
本书中使用的变量 . . . . .	x
路径变量 . . . . .	xi
何处可找到新信息 . . . . .	xii

## 第 1 部分 概述 . . . . . 1

第 1 章 商店开发概述 . . . . .	3
了解 WebSphere Commerce 中的商店开发 . . . . .	3
商店的用途 . . . . .	3
具有代表性的商店业务模型 . . . . .	3
正在开发的商店的数目 . . . . .	4
商店的基础 . . . . .	4
必需定制的等级 . . . . .	7
方案: 开发和部署生产商店 . . . . .	8

## 第 2 部分 由 WebSphere Commerce 支持的业务模型 . . . . . 11

第 2 章 WebSphere Commerce 中受支持的业务模型 . . . . .	13
了解 WebSphere Commerce 中受支持的业务模型 . . . . .	13
直销 . . . . .	13
主管 . . . . .	14
价值链 . . . . .	15
WebSphere Commerce 中的样本商店 . . . . .	17

## 第 3 部分 WebSphere Commerce 体系结构 . . . . . 19

第 3 章 WebSphere Commerce 组织结构 . . . . .	21
了解 WebSphere Commerce 组织结构 . . . . .	21
组织结构如何支持业务模型? . . . . .	22
消费者直销 . . . . .	22
B2B 直销 . . . . .	23
需求链 . . . . .	24
供应链 . . . . .	26
主管 . . . . .	27
样本组织结构 . . . . .	29
创建组织结构 . . . . .	29

第 4 章 WebSphere Commerce 中的访问控制 . . . . .	31
了解 WebSphere Commerce 中的访问控制 . . . . .	31
访问控制策略 . . . . .	31
了解业务模型中的访问控制 . . . . .	34
基本访问控制结构 . . . . .	34

消费者直销 . . . . .	36
B2B 直销 . . . . .	39
需求链 . . . . .	40
供应链 . . . . .	46
主管 . . . . .	50
样本公司中的访问控制 . . . . .	55
向商店添加访问控制 . . . . .	55

## 第 5 章 WebSphere Commerce 业务策略框架 . . . . . 57

了解 WebSphere Commerce 业务策略框架 . . . . .	57
业务策略 . . . . .	57
商业帐户 . . . . .	57
合同和服务协议 . . . . .	57
条款和条件 . . . . .	58
样本公司中的业务策略 . . . . .	58
向站点添加业务策略 . . . . .	58

## 第 6 章 实例体系结构 . . . . . 59

WebSphere Commerce Server . . . . .	59
WebSphere Commerce Server 实例 . . . . .	59

第 7 章 商店体系结构 . . . . .	61
了解 WebSphere Commerce 商店体系结构 . . . . .	61
商店有用资源 . . . . .	61
单个实例中的多家商店 . . . . .	62
商店之间的关系 . . . . .	64
了解商店体系结构如何支持业务模型 . . . . .	64
面向客户的商店 . . . . .	64
代理商店 . . . . .	67
有用资源商店 . . . . .	67
受支持业务模型中的商店 . . . . .	68

## 第 4 部分 开发商店前台 . . . . . 73

第 8 章 开发商店前台 . . . . .	75
商店前台体系结构 . . . . .	75
缺省命令和视图 . . . . .	75
创建商店页面 . . . . .	75
列出商店页面的列表 . . . . .	76
列出命令和视图 URL 的列表 . . . . .	79
使 JSP 文件名与视图关联 . . . . .	80

第 9 章 高速缓存商店页面 . . . . .	83
规划高速缓存战略 . . . . .	83
应对哪些页面进行高速缓存 . . . . .	83
应将页面高速缓存为完整页面还是页面片段 . . . . .	83
制订更详细的高速缓存战略 . . . . .	84
如何请求页面或片段 . . . . .	84
页面或片段是否依赖于商店关系 . . . . .	84

如何使高速缓存数据无效 . . . . .	84
实现高速缓存战略 . . . . .	85
了解 cachespec.xml 文件 . . . . .	85
使 cachespec.xml 文件中的高速缓存数据失效 . . . . .	88
实现对使用商店关系的商店页面的高速缓存 . . . . .	89
以动态高速缓存来替换高速缓存命令功能 . . . . .	91

## 第 5 部分 商店数据概述 . . . . . 95

### 第 10 章 商店数据 . . . . . 97

什么是商店数据? . . . . .	97
商店数据信息模型 . . . . .	97
按子系统查看商店数据信息模型 . . . . .	99
按数据类型查看商店数据信息模型 . . . . .	99
商店数据类型和样本公司 . . . . .	103
创建数据的工具 . . . . .	103
WebSphere Commerce 装入程序软件包 . . . . .	103
管理控制台 . . . . .	103
WebSphere 贸易加速器 . . . . .	103
组织管理控制台 . . . . .	103
工具和商店数据摘要图表 . . . . .	104

## 第 6 部分 开发商店数据 . . . . . 107

### 第 11 章 站点有用资源 . . . . . 109

了解 WebSphere Commerce 中的站点有用资源 . . . . .	109
语言 . . . . .	110
成员属性 . . . . .	110
属性类型 . . . . .	110
成员组类型 . . . . .	110
用户 . . . . .	110
组织 . . . . .	110
角色 . . . . .	110
数量单位转换 . . . . .	110
数量单位 . . . . .	111
税款类型 . . . . .	111
计算用法 . . . . .	111
货币 . . . . .	111
数值用法 . . . . .	111
商品类型 . . . . .	111
设备格式 . . . . .	111
商店关系类型 . . . . .	111
站点级贸易协议数据 . . . . .	112
贸易协议类型 . . . . .	112
参与方角色 . . . . .	112
策略类型 . . . . .	112
条款和条件类型 . . . . .	113
个性化属性 . . . . .	113
属性类型 . . . . .	113
运算符 . . . . .	113
附件用法 . . . . .	113
在 WebSphere Commerce 中创建站点有用资源 . . . . .	113

### 第 12 章 成员有用资源 . . . . . 115

了解 WebSphere Commerce 中的成员有用资源 . . . . .	115
--	-----

成员 . . . . .	116
成员属性 . . . . .	116
角色 . . . . .	116
了解 WebSphere Commerce 中的客户有用资源 . . . . .	117
地址信息 . . . . .	117
兴趣列表 . . . . .	118

### 了解 WebSphere Commerce 中的卖方有用资源 . . . . . 118

商店 . . . . .	119
帐户 . . . . .	119
合同 . . . . .	119
产品集 . . . . .	120
价格列表 . . . . .	120
产品目录 . . . . .	120
供货中心 . . . . .	120
库存商品 . . . . .	121
了解 WebSphere Commerce 中的管理员有用资源 . . . . .	121
在 WebSphere Commerce 中创建成员有用资源 . . . . .	121

### 第 13 章 商店有用资源 . . . . . 123

了解 WebSphere Commerce 中的商店有用资源 . . . . .	123
商店实体 . . . . .	123
在 WebSphere Commerce 中创建商店有用资源 . . . . .	124
在 XML 文件中创建商店数据有用资源 . . . . .	124

### 第 14 章 商店间的关系 . . . . . 129

了解 WebSphere Commerce 中商店间的关系 . . . . .	129
商店关系 . . . . .	129
商店关系类型 . . . . .	129
商店关系类型描述 . . . . .	131
创建 WebSphere Commerce 中的商店关系 . . . . .	131

### 第 15 章 命令、视图和 URL 注册表数据 . . . . . 135

了解 WebSphere Commerce 中的命令、视图和 URL 注册表 . . . . .	135
URL 注册表 . . . . .	135
命令注册表 . . . . .	136
视图注册表 . . . . .	136
创建新命令、视图和 URL . . . . .	136
在 WebSphere Commerce 中注册命令、视图和 URL . . . . .	137
创建 XML 文件以注册命令、视图和 URL . . . . .	137

### 第 16 章 产品目录有用资源 . . . . . 141

了解 WebSphere Commerce 中的产品目录 . . . . .	141
产品目录 . . . . .	142
产品目录组 . . . . .	142
产品目录条目 . . . . .	143
产品集 . . . . .	144
属性 . . . . .	144
属性值 . . . . .	144
打包销售商品属性 . . . . .	144
打包销售商品属性值 . . . . .	144
在 WebSphere Commerce 中创建产品目录有用资源 . . . . .	145
创建主产品目录 . . . . .	145
显示商店产品目录有用资源 . . . . .	161
创建销售产品目录 . . . . .	163

向另一类别添加产品 . . . . .	163
在 WebSphere Commerce 中管理产品目录有用资源	165
产品目录组 . . . . .	166
产品目录条目 . . . . .	166
产品管理工具 . . . . .	167
装入程序软件包 . . . . .	168

## 第 17 章 对有用资源进行定价 . . . . . 169

了解 WebSphere Commerce 中的定价 . . . . .	169
报价 . . . . .	170
卖价 . . . . .	170
贸易状态容器 . . . . .	170
条款和条件 . . . . .	170
定价条款和条件的类型 . . . . .	170
贸易协议 . . . . .	171
参与方 . . . . .	171
参与方角色 . . . . .	171
合同 . . . . .	171
业务策略 . . . . .	172
价格策略 . . . . .	172
产品目录条目送货 . . . . .	172
其它定价有用资源 . . . . .	172
在 WebSphere Commerce 中创建定价有用资源 . . . . .	172
在 XML 文件中创建定价有用资源 . . . . .	173

## 第 18 章 合同有用资源 . . . . . 177

了解 WebSphere Commerce 中的合同 . . . . .	178
帐户 (商业帐户) . . . . .	178
合同 . . . . .	179
贸易协议 . . . . .	179
参与方 . . . . .	180
条款和条件 . . . . .	180
业务策略 . . . . .	182
附件 . . . . .	183
订购商品 . . . . .	183
在 WebSphere Commerce 中创建缺省合同有用资源 . . . . .	184
创建业务策略 XML 文件 . . . . .	185
创建缺省合同文件 . . . . .	186

## 第 19 章 供货有用资源 . . . . . 193

了解 WebSphere Commerce 中的供货有用资源 . . . . .	194
供货中心 . . . . .	194
接收 . . . . .	194
RaDetail . . . . .	195
库存 . . . . .	195
送货安排 . . . . .	195
其它供货有用资源 . . . . .	195
在 WebSphere Commerce 中创建供货有用资源 . . . . .	195
创建商店供货有用资源 (非 ATP) . . . . .	197

## 第 20 章 竞销有用资源 . . . . . 199

了解 WebSphere Commerce 中的竞销 . . . . .	199
在 WebSphere Commerce 中创建竞销有用资源 . . . . .	201

## 第 21 章 Payments 工具 . . . . . 203

使用 XML 文件创建支付有用资源 . . . . .	203
-----------------------------	-----

定制支付卡匣的环境 . . . . .	204
修改商店 .jsp 文件 . . . . .	205
检查卡匣的卡匣柜概要文件 . . . . .	206
检查卡匣的 .jsp 文件 . . . . .	208
配置 WebSphere Commerce Payments 中的商家设置 . . . . .	208

## 第 22 章 语言有用资源 . . . . . 209

了解 WebSphere Commerce 中的语言有用资源 . . . . .	209
缺省语言 . . . . .	209
支持的语言 . . . . .	210
备用语言 . . . . .	210
在 WebSphere Commerce 中创建语言有用资源 . . . . .	210

## 第 23 章 货币有用资源 . . . . . 211

了解 WebSphere Commerce 中的货币有用资源 . . . . .	211
货币格式 . . . . .	212
数值用法 . . . . .	212
货币格式描述 . . . . .	212
支持的货币 . . . . .	212
货币转换规则 . . . . .	212
对等货币 . . . . .	212
在 WebSphere Commerce 中创建货币有用资源 . . . . .	213
使用 XML 文件创建货币有用资源 . . . . .	213

## 第 24 章 计量单位有用资源 . . . . . 217

了解 WebSphere Commerce 中的计量单位 . . . . .	217
数量单位和数量单位格式 . . . . .	218
在 WebSphere Commerce 中创建计量单位 . . . . .	219

## 第 25 章 地区有用资源 . . . . . 221

了解 WebSphere Commerce 中的地区有用资源 . . . . .	221
在 WebSphere Commerce 中创建地区有用资源 . . . . .	222

## 第 26 章 送货有用资源 . . . . . 223

了解 WebSphere Commerce 中的送货有用资源 . . . . .	223
送货方式 . . . . .	224
计算代码 . . . . .	224
地区和地区组 . . . . .	225
在 WebSphere Commerce 中创建送货有用资源 . . . . .	225
使用 XML 文件创建送货有用资源 . . . . .	226
创建送货供货有用资源 . . . . .	233
创建商店 - 产品目录 - 送货有用资源 . . . . .	235
创建缺省送货方式 . . . . .	236

## 第 27 章 税款有用资源 . . . . . 239

了解 WebSphere Commerce 中的税款有用资源 . . . . .	239
税类别 . . . . .	240
计算代码 . . . . .	240
地区和地区组 . . . . .	241
在 WebSphere Commerce 中创建税款有用资源 . . . . .	242
使用 XML 文件创建税款有用资源 . . . . .	242
创建税款供货有用资源 . . . . .	251
创建商店 - 产品目录 - 税款有用资源 . . . . .	252

## 第 28 章 折扣有用资源 . . . . . 255

了解 WebSphere Commerce 中基于规则的折扣	255
存储缺省货币	256
计算代码	256
RLPromotion	256
Blaze 规则项目	257
Blaze 规则服务	257
折扣服务	257
Blaze Rule Server	257
了解 WebSphere Commerce 中基于模式的折扣	257
计算代码	257
在 WebSphere Commerce 中创建折扣有用资源	258

## 第 29 章 库存有用资源 259

了解 WebSphere Commerce 中的库存有用资源	259
ATP 库存	260
非 ATP 库存	261
在 WebSphere Commerce 中创建库存有用资源	262
管理库存调整代码	262
添加库存调整代码	263
更改库存调整代码	264
删除库存调整代码	265

## 第 30 章 订单有用资源 267

了解 WebSphere Commerce 中的订单有用资源	267
订单和订购商品	267
订购商品	268
订单报价关系	270
在 WebSphere Commerce 中创建订单有用资源	270

## 第 31 章 供应商有用资源 271

了解 WebSphere Commerce 中的供应商有用资源	271
创建供应商有用资源	272

## 第 32 章 客户概要文件 273

了解 WebSphere Commerce 中的客户概要文件	273
--------------------------------	-----

## 第 7 部分 向商店添加访问控制 275

### 第 33 章 商店中的访问控制 277

了解 WebSphere Commerce 中的访问控制	277
商店中的访问控制	277
向商店添加访问控制	280
创建或编辑商店中的访问控制	280

## 第 8 部分 使您的商店全球化 285

### 第 34 章 全球化 287

支持全球化	287
样本商店	288
显示格式	291
创建新的显示格式	291
创建全球化商店	294
创建商店	294
管理全球化站点的模板	295
所有商店和语言程序设计模块使用一个模板	296

向商店添加语言	298
创建全球化联机产品目录	298
管理全球化有用资源	299
翻译属性文件	299

## 第 9 部分 封装商店 301

### 第 35 章 封装商店 303

创建商店归档文件	304
创建样本商店归档文件	306

## 第 10 部分 发布商店 309

### 第 36 章 发布完整的商店 311

了解 WebSphere Commerce 中的发布	311
启动发布	312
从商店归档文件中解包有用资源	314
更新发布参数	315
发布数据	315
发布日志文件	320
使商店归档文件可用于管理控制台	321
在 SARRegistry.xml 文件中注册商店归档文件	321
将商店归档文件复制到适用的商店归档文件目录	322

### 第 37 章 装入商店数据的概述 323

了解 WebSphere Commerce 中的数据装入	324
用于装入商店数据的装入程序软件包命令	327
用于转换和抽取数据的装入程序软件包命令	345
装入程序软件包命令的相关工具	354
装入商店数据	355
使用装入程序软件包命令和脚本	356
解析标识的示例	357
装入数据的示例	364

## 第 38 章 装入 WebSphere Commerce

### 数据库有用资源组 367

数据库有用资源组	367
数据库有用资源装入顺序	367
装入商店	368
装入数据库有用资源组	373

### 第 39 章 发布商业帐户和合同 377

使用管理控制台或命令行来发布商业帐户和合同	377
使用命令发布商业帐户和合同	377
发布商业帐户有用资源	378
发布合同有用资源	378

### 第 40 章 发布商店前台有用资源和商店配置文件 379

使用管理控制台或命令行来发布商店前台有用资源和商店配置文件	379
通过复制到 WebSphere Commerce Server 发布商店前台有用资源和商店配置文件	379



---

**第 11 部分 向商店添加 WebSphere Commerce 功能部件 . . . . . 383**

**第 41 章 向商店添加客户关心 . . . . . 385**

了解商店中的客户关心 . . . . . 385

- 使用框架集 . . . . . 386
- 在没有框架集的情况下使用客户关心 . . . . . 387
- 定义客户关心 . . . . . 389
- 使用客户关心监视客户 . . . . . 396
- 将请求直接发送给客户关心队列 . . . . . 400
- 定制客户关心 . . . . . 401

对商店添加客户关心 . . . . . 403

- 第 1 部分: 安装先决条件 . . . . . 403
- 第 2 部分: 从样本商店复制客户关心集成文件 . . . . . 404
- 第 3 部分: 添加可确定客户正在浏览哪些页面的代码 . . . . . 405
- 第 4 部分: 添加指向客户关心的链接 . . . . . 405
- 第 5 部分: 创建将重定向至客户关心框架集页面的入口页面 . . . . . 406

**第 42 章 向商店添加电子广告位 . . . . . 407**

电子广告位 . . . . . 407

- 电子广告位 bean . . . . . 411
- 向商店页面添加电子广告位 . . . . . 411

---

**第 12 部分 附录 . . . . . 413**

**附录 A. UML 图注 . . . . . 415**

**附录 B. 创建数据 . . . . . 417**

创建样本商店的数据 . . . . . 417

**附录 C. 数据库有用资源组 . . . . . 419**

数据库有用资源组相关性 . . . . . 419

**声明 . . . . . 425**

商标 . . . . . 427



---

## 开始之前

《IBM® WebSphere® Commerce 商店开发者指南》提供了关于 WebSphere Commerce 商店体系结构和商店开发过程的信息。同时还特别提供了关于以下主题的详细信息:

- 商店开发过程
- 受 WebSphere Commerce 支持的业务模型
- WebSphere Commerce 体系结构
- 开发商店前台
- 开发商店数据
- 商店数据体系结构
- 商店数据信息模型
- 使商店全球化
- 向商店添加访问控制
- 封装商店
- 发布商店
- 向商店添加 WebSphere Commerce 功能部件

---

## 本书中使用的约定和术语

本书使用以下突出显示约定:

<b>粗体字</b>	表示命令或者诸如字段名、图标或菜单选项之类的图形用户界面 (GUI) 控件。
等宽字	表示完全按显示原样输入的文本示例、文件名、目录路径和名称。
斜体字	用于强调字。斜体还表示必须用系统的相应值来替代的名称。



此图标对技巧作出标记, 标记是可帮助您完成任务的附加信息。

---

### 重要信息

这些部分突出显示了特别重要的信息。

### 注意

这些部分突出显示重要的信息。



表示特定于 WebSphere Commerce Business Edition 的信息。

	表示特定于 WebSphere Commerce Professional Edition 的信息。
	表示特定于 WebSphere Commerce - Express 的信息。
	  表示特定于 WebSphere Commerce 开发环境的信息。开发环境为 WebSphere Commerce Studio V5.5。
	 表示特定于 WebSphere Commerce 开发环境的信息。开发环境是 WebSphere Commerce - Express Developer Edition, V5.5。
	表示特定于运行在 AIX <sup>®</sup> 上的程序的信息。
	表示特定于运行在 OS/400 <sup>®</sup> 上的程序的信息。
	表示特定于 WebSphere Commerce for Linux for xSeries <sup>™</sup> 的信息、特定于 WebSphere Commerce for Linux for Eserver zSeries <sup>™</sup> and S/390 <sup>®</sup> 的信息、特定于 WebSphere Commerce for Linux for Eserver iSeries <sup>™</sup> 的信息以及特定于 WebSphere Commerce for Linux for Eserver pSeries <sup>™</sup> 的信息。
	表示特定于运行在 Solaris Operating Environment 上的程序的信息。
	表示特定于运行在 Windows <sup>®</sup> 2000 上的程序的信息。
	表示特定于 DB2 Universal Database <sup>™</sup> 的信息。
	表示特定于 Oracle9i Database 的信息。

---

## 本书中使用的变量

本书中一些关键的变量如下所示:

### *businessmodel*

您正在使用的样本业务模型的名称 (例如, 消费者直销或 B2B 直销)。

  *cell\_name*

单元是 WebSphere Application Server 分布式网络中一起受管的一个或多个节点的任意逻辑分组。在本定义中, 一个节点就是单个的 WebSphere Application Server。由 WebSphere Application Server 部署管理器一次性管理的一个或多个单元被称为 WebSphere Application Server 部署管理器单元。

 *cell\_name*

在 WebSphere Commerce - Express 中, *cell\_name* 等同于 *host\_name*。

### *host\_name*

该变量代表 WebSphere Commerce Server 的全限定主机名 (例如, `server.mydomain.ibm.com` 是全限定的)。

*instance\_name*

该变量代表您正在使用的 WebSphere Commerce 实例的名称（例如，mall1）。

*storedir*

该变量代表您的商店所在的商店目录的名称。

*WAS\_instance\_name*

该变量代表与您的 WebSphere Commerce 实例相关联的 WebSphere Application Server 的名称。

---

## 路径变量

本指南使用以下变量表示目录路径：

*WC\_installdir*

这是 WebSphere Commerce 的安装目录。以下是 WebSphere Commerce 在各种操作系统上的缺省安装目录：

 /usr/WebSphere/CommerceServer55

 /QIBM/ProdData/CommerceServer55

 /opt/WebSphere/CommerceServer55

 /opt/WebSphere/CommerceServer55

 C:\Program Files\WebSphere\CommerceServer55

*WCDE\_installdir*

WebSphere Commerce 开发环境的安装目录。对于 WebSphere Commerce Business Edition 和 WebSphere Commerce Professional Edition，您的开发环境为 WebSphere Commerce Studio，V5.5。以下为缺省安装目录：

C:\WebSphere\CommerceStudio55。

对 WebSphere Commerce - Express，开发环境为 WebSphere Commerce - Express Developer Edition，V5.5。以下为缺省安装目录：

C:\WebSphere\CommerceDev55

 *WC\_userdir*

这是 WebSphere Commerce 使用的所有数据的目录，用户可以修改它或它需要由用户配置。这种数据的示例是 WebSphere Commerce 实例信息。该目录对 OS/400 是唯一的。

*WC\_userdir* 变量代表以下目录：

/QIBM/UserData/CommerceServer55

*WAS\_installdir*

这是 WebSphere Application Server 的安装目录。以下是 WebSphere Application Server 在各种操作系统上的缺省安装目录：

▶ AIX	/usr/WebSphere/AppServer
▶ 400	/QIBM/ProdData/WebAS5/Base
▶ Linux	/opt/WebSphere/AppServer
▶ Solaris	/opt/WebSphere/AppServer
▶ Windows	C:\Program Files\WebSphere\AppServer

#### ▶ 400 *WAS\_userdir*

这是 WebSphere Application Server 使用的所有数据的目录，用户可以修改它或它需要由用户配置。这种数据的示例是 WebSphere Application Server 实例信息。该目录对 OS/400 是唯一的。

*WAS\_userdir* 变量代表以下目录：

*QIBM/UserData/WebAS5/Base/WAS\_instance\_name*

*WC\_userdir*

*WC\_userdir* 变量代表以下目录：*QIBM/UserData/WebAS5/Base/WAS\_instance\_name*

*workspace\_dir*

用于开发环境。该变量代表 *drive:\WebSphere\workspace\_db2*

---

## 何处可找到新信息

以后可能会对本书进行更新。请检查以下 WebSphere Commerce Web 站点以获取更新：

<http://www.ibm.com/software/commerce/library/>

更新可能包括新信息。

---

## 第 1 部分 概述





---

## 第 1 章 商店开发概述

本章提供了 WebSphere Commerce 中站点或商店开发过程的概述，并介绍了在本指南中讨论的许多概念。

注：本指南使用短语 *商店开发* 同时指称创建单个商店所涉及的过程以及创建多个商店或站点环境所涉及的过程。

---

### 了解 WebSphere Commerce 中的商店开发

在使用 WebSphere Commerce 开始开发站点或商店之前，您需要了解下列因素是如何影响商店开发过程的。本章介绍了其中每个因素，但它们在本指南中大多都有更加详细的说明，另外有些在 WebSphere Commerce 库中的其它文档中也有说明。

在 WebSphere Commerce 中如何选择开发商店取决于以下因素：

- 商店的用途
- 具有代表性的商店业务模型
- 正在开发的商店的数目及其类型
- 商店的基础
- 必需定制的等级

#### 商店的用途


开发商店的目的通常为以下之一：

- 生产：生产商店是生产环境中功能完整的商店，可供客户或合作伙伴使用。
- 演示：演示商店出于销售目的演示某些能力。演示商店可能只有部分功能。
- 样本：样本商店是功能完整的商店，旨在作为您创建您的网上商店的基础。

#### 具有代表性的商店业务模型

开发商店之前，您需要了解何种由 WebSphere Commerce 支持的业务模型可以最好地代表您的商店。WebSphere Commerce 支持作为以下业务模型之一的实例的站点或商店：

注：这些业务模型在第 13 页的第 2 章，『WebSphere Commerce 中受支持的业务模型』中有更详细的讨论，这儿仅提供每个业务模型的简短介绍。

- 直销业务模型：与前发行版一样，WebSphere Commerce 支持直销业务模型。通过使用 WebSphere Commerce 您可以创建这样的站点或商店，它们支持关于产品、服务或信息的商业交易，此交易可以直接在公司和消费者之间进行，也可以在两个公司或合作伙伴之间进行。WebSphere Commerce 支持以下类型的直销业务模型：
  - 消费者直销业务模型：消费者直销支持在企业 and 消费者之间涉及产品、服务或信息的商业交易。在消费者直销方案中消费者通常直接公司那里购买商品或服务。
  -  B2B 直销业务模型：B2B 直销支持两个企业或合作伙伴之间涉及产品、服务或信息的商业交易。典型的 B2B 直销交易发生在买方、供应商、生产商、转销商、经销商和贸易伙伴当中。

- **Business** 主管业务模型: WebSphere Commerce 还通过因特网服务提供商或其它主管供应商来支持对商家或其它企业的主管。
- **Business** 价值链业务模型: 价值链支持涉及多个企业或合作伙伴的交易。产品、商品、服务或信息通过价值链组从生产方交付到最终用户手中。价值链也具有关系和管理方面的功能, 也就是说, 您可以管理合作伙伴或企业在价值链中的关系, 也可以向各方提供一些管理服务。WebSphere Commerce 支持以下两种类型的价值链的交易和关系管理:
  - 需求链: 需求链支持间接销售渠道和直销渠道。
  - 供应链: 供应链支持商品的采购和供应。WebSphere Commerce 通过私有市场支持商品的供应。私有市场为供应商提供了论坛, 以便向与他们有合同关系的买方给出其待售商品和服务的报价。

注: 这些业务模型在第 13 页的第 2 章, 『WebSphere Commerce 中受支持的业务模型』中有更详细的讨论。

## 正在开发的商店的数目

由公司而定, 您或许需要开发不止一个商店或不止一类商店。例如, 如果您的公司向客户直接销售, 您可能只需要一个客户可以访问并从中购买商品的商店。但是如果您支持需求链, 则您的公司可能就需要一个主中心商店和若干允许您连接并管理渠道的商店。您也可以选择按您的渠道主管组织或公司的各个商店。关于需求链的更多信息, 请参阅第 13 页的第 2 章, 『WebSphere Commerce 中受支持的业务模型』。

如果您正在为商家或其它公司主管商店, 则您也需要为管理商家和处理注册请求开发一个中心商店, 并需要一种为您正在主管的商店开发站点的方法。关于主管业务模型的更多信息, 请参阅第 13 页的第 2 章, 『WebSphere Commerce 中受支持的业务模型』。

您可以用 WebSphere Commerce 开发许多类型的站点和站点中的多个商店。关于商店类型的更多详细信息, 请参阅第 61 页的第 7 章, 『商店体系结构』。

## 商店的基础

在用 WebSphere Commerce 创建站点或商店之前, 您必须决定从何处开始开发。WebSphere Commerce 提供了若干样本, 您可以用作开发的起点, 或者您可以选择从头开始。关于随 WebSphere Commerce 提供的样本的更多详细信息, 请参阅《WebSphere Commerce 样本商店指南》。

### 从样本开始

随 WebSphere Commerce 提供的样本打包为商店归档文件。

**商店归档文件:** 商店归档文件 (.sar) 是一个 ZIP 归档文件, 包含创建站点或商店所必需的所有有用资源。它主要用作封装和传递商店的媒介。商店归档文件只需发布到 WebSphere Commerce 服务器上即可创建可以查看、浏览和购物的能运作的商店。

通常, 商店归档文件由以下文件组成:

- **Web 有用资源:** 创建商店页面的文件, 如 HTML 文件、JSP 文件、图像、图形和包含文件。

- 属性资源束：包含商店页面的文本。如果您的商店支持多种语言，则商店归档文件将包含多个资源束，每种受支持的语言对应一个资源束，再加上一个缺省的资源束（它不包含语言环境）。例如，`AddressText_en_US.properties` 和 `AddressText.properties`。
- 商店数据有用资源：要装入到数据库中的数据。商店数据有用资源包括竞销、产品目录条目、货币、供货信息、定价、送货、存储和税务信息等数据。关于商店数据有用资源的更详细的列表，请参阅第 107 页的第 6 部分，『开发商店数据』。

随 WebSphere Commerce 提供的样本商店归档文件中的商店数据有用资源是对装入程序软件包有效的、格式良好的 XML 文件。商店归档 XML 文件计划成为可移植文件并且不应包含特定于数据库特定实例的已生成主键。相反，它们使用商店发布时由标识解析器解析的内部别名。这些约定的使用允许移植样本商店归档文件。关于更多信息，请参阅第 301 页的第 9 部分，『封装商店』。

关于装入程序软件包的更多信息，请参阅第 323 页的第 37 章，『装入商店数据的概述』。


**注：**商店数据有用资源还包含创建合同的信息。合同信息不是通过装入程序软件包装入的；它向创建合同的命令提供输入。

- 支付有用资源：WebSphere Commerce 支付的配置信息。支付信息不是通过装入程序软件包装入的；它向配置 WebSphere Commerce Payments 的命令提供输入。
- 描述符：描述关于它如何发布的商店归档文件和信息的 XML 文件。这些文件包括：`store-refs.xml`、`ibm-wc-load.xml`、`unpack.xml` 和 `ForeignKeys.dtd`。




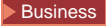
关于商店归档文件的更多信息，请参阅第 303 页的第 35 章，『封装商店』。

**发布商店归档文件：**您可以通过使用管理控制台中的发布实用程序或通过命令行来发布商店归档文件。关于如何发布商店归档文件的更多详细信息，请参阅 WebSphere Commerce 产品联机帮助中的主题“发布商店归档文件”。

**样本类型：**随 WebSphere Commerce 提供的样本分类如下：

- 组合商店归档文件
-  组件商店归档文件
- 基本商店归档文件


**组合商店归档文件：**组合商店归档文件包含所有创建工作站点所必需的有用资源。随 WebSphere Commerce 提供的样本组合商店归档文件通常包含用于为相应的业务模型创建适当环境的组织结构、预定义的用户角色和必要的访问控制策略。组合商店归档文件也包含创建需要的商店或站点所必需的有用资源。例如，需求链样本组合商店归档文件包含一个样本渠道中心站点、共享产品目录以及转销商和经销商商店。

WebSphere Commerce 包括若干组合商店归档文件，这些归档文件包含完全正常运作的网上样本站点，您可以将其用作创建您自己商店的基础。这些样本包括直销商店（消费者直销和  B2B 直销）、 需求链公司、 供应商公司和  主管站点，它们实现了当前顶级电子交易站点的很多最常用的功能，并提供所有必需的商店有用资源。关于随 WebSphere Commerce 提供的样本的更多信息，请参阅《WebSphere Commerce 样本商店指南》。

**为何要以样本组合商店归档文件开始？：**以样本组合商店归档文件开始就把所有必需的数据装入了 WebSphere Commerce 服务器，以便创建完全正常运作的站点。

WebSphere Commerce 需要将特定数据装入 WebSphere Commerce 服务器数据库以创建正常运作的站点，而且要求此数据必须以模式确定的顺序装入。由于样本组件商店归档文件在顺序和结构中包含了 WebSphere Commerce 服务器数据库所需的所有必填数据，因此使用样本商店作为您自己站点的基础在初始创建期间会节省大量时间。

在发布样本组合商店归档文件之后，可以根据您商店的需要，或多或少地对其进行编辑。例如，您可能仅需要使用 WebSphere Commerce 提供的工具编辑数据，并使用开发环境更改商店页面的外观。或者，可能需要直接编辑 XML 文件或数据库以对数据进行更全面的更改，并改编商店页面以更改商店流程和功能。又或者使用样本商店和开发新的商店有用资源两者的组合可能是最适合您的商店开发方法。例如，如果某个样本商店中的某些数据库有用资源与您的商店需要非常匹配，但是该商店页面的流程与您的不符，则开发全新的 Web 有用资源时可以从该商店复制数据库有用资源并进行定制。关于编辑商店数据的更多信息，请参阅第 107 页的第 6 部分，『开发商店数据』。

**组件商店归档文件:**  构成组合商店归档文件的每一部分还可用作独立的商店归档文件。这些商店归档文件称为组件商店归档文件。组件商店归档文件可以是组织结构商店归档文件（包含组织结构和预定义的用户角色），可以是能够运作的商店，也可以是由其它类型的商店用作资源的文件或数据有用资源的集合。关于随 WebSphere Commerce 提供的样本的更多信息，请参阅《WebSphere Commerce 样本商店指南》。

**为何要以样本组件商店归档文件开始? :** 以样本组件商店归档文件或样本组件商店归档文件的组合开始，与以样本组合商店归档文件开始相比，为您提供了更大的灵活性。比如通过发布组合商店归档文件来创建功能完整的站点。此站点的部分可能正是您所需要的，但其它的则可能不是。例如，如果您商店页面的流程明显不同于提供的所有样本，或者如果您计划对 WebSphere Commerce 服务器数据库模式进行大量定制，则您可能选择只发布所提供样本的某些部分，而不是整个样本。例如，您可以选择只发布样本组织结构，然后在站点中开发所有的有用资源来创建商店。或者您可以选择发布样本组织结构和一個或更多的样本组件归档文件，以在组织结构中创建商店或提供由其它商店使用的资源。

**注:** 如果您正在创建价值链业务模型实例，则建议您从发布样本组织结构开始，因为站点所需要的包含多个实体的组织结构是非常复杂的。关于 WebSphere Commerce 中组织结构如何工作的更多信息，请参阅第 21 页的『了解 WebSphere Commerce 组织结构』。

**基本商店归档文件:** WebSphere Commerce 也提供基本的样本商店，该样本商店提供在 WebSphere Commerce Server 中创建商店最低所需的一组有用资源。

**为何以样本基本商店开始? :** 以样本基本商店开始允许您在贸易服务器内建立商店实体，其中可以通过使用商店标识调用 JSP 文件。如果您正在创建与任何随 WebSphere Commerce 提供的样本商店都不同的商店，您可能要以样本基本商店开始，因为以基本商店开始允许开发人员在必要时添加有用资源，并且不需要除去或更改不适用于您商店的有用资源。关于随 WebSphere Commerce 提供的基本商店的更多信息，请参阅《WebSphere Commerce 样本商店指南》。

**注:** 您可以使用与样本组织结构之一一起提供的样本基本商店。

## 从头开始

也可能从头开始，就是说不使用随 WebSphere Commerce 提供的任何样本。

## 必需定制的等级

一旦决定了商店的基础，不管是样本商店、样本组织结构或样本基本商店，或是从头开始，您都需要确定将对其做何种更改。一般来说，WebSphere Commerce 中的大多商店开发都可归入以下类别之一：

- 添加或更改商店功能，其中包括添加新功能或更改商店流程
- 创建或更改商店外观
- 创建或更改商店数据

在许多案例中，您的商店开发工作将包含它们三个的组合。

### 添加或更改商店功能

添加或更改商店功能（包括更改商店流程或向商店添加新功能）通常必须更改业务逻辑。开发业务逻辑的工具（包括创建和扩展命令、创建定制的代码以及实现业务逻辑）将在《WebSphere Commerce 编程指南与教程》中讨论。

**注：**正在创建或更改业务逻辑的开发者必须具有 Java™、Enterprise JavaBeans™、WebSphere Studio Application Developer、J2EE 编程的编程技能，并熟悉 WebSphere Commerce 编程模型和对象模型。

WebSphere 贸易加速器提供了更改一些功能和商店流程（随消费者直销和 B2B 直销样本商店提供）的能力。关于您可以更改何种流程和功能以及如何更改的更多信息，请参阅 WebSphere Commerce 产品联机帮助。

### 创建或更改商店外观

更改商店外观通常涉及更改商店前台。商店前台有用资源包含 Web 有用资源（如 HTML 页面、JSP 文件、样式表、图像、图形和其它多媒体文件类型）。开发商店前台有用资源可能包括定制样本商店页面、用您自己现有的页面替换它们、创建新页面或结合执行上述三个操作。

WebSphere Commerce 提供了以下工具来创建或编辑商店前台有用资源：

- WebSphere Studio Application Developer

WebSphere Studio Application Developer（与 WebSphere Commerce Studio 封装在一起）包括创建和编辑商店前台有用资源（包括 HTML、图形、多媒体和 JavaServer Pages (JSP) 文件）所需的工具。WebSphere Studio Application Developer 中包含的 Page Designer 允许您创建 HTML 或 JSP 文件以及动画图像。您还可以配置 WebSphere Studio Application Developer，以便使用您选择的其它 Web 开发工具。关于注册您自己的工具的更多信息，请参阅 WebSphere Studio 联机帮助。

关于使用 WebSphere Commerce Studio 中的工具创建和编辑商店前台有用资源的更多信息，请参阅 WebSphere Commerce Studio 联机帮助。关于在 WebSphere Commerce 中创建商店前台的更多信息，请参阅第 75 页的第 8 章，『开发商店前台』。

**注：**正在创建或更改商店前台的开发者必须具有 Java、JavaScript™、HTML、JSP 技术的编程技能，并熟悉 WebSphere Commerce 商店体系结构。

- WebSphere 贸易加速器

WebSphere 贸易加速器包含以下工具，您可用来更改商店的外观：

- “更改页面” 笔记本
- “上载徽标” 笔记本

- “更改样式” 向导
- “管理文件” 笔记本
- “商店概要文件” 笔记本

关于这些工具的更多信息，请参阅 **WebSphere Commerce** 产品联机帮助。

**注：**上面列出的工具仅可处理基于消费者直销样本商店的商店，以及使用“商店创建”向导创建的 **Business** 受托管商店（使用托管模型和需求链模型）。

## 创建或更改商店数据

您有多种选项可用于在商店中开发和编辑数据库有用资源。

- **WebSphere Commerce 装入程序软件包**

**WebSphere Commerce 装入程序软件包**主要由准备数据并将数据装入到 **WebSphere Commerce** 数据库的实用程序组成。使用装入程序软件包装入大量数据并更新 **WebSphere Commerce** 数据库中的数据。此软件包中的装入实用程序使用有效、精心编成的 XML 作为输入将数据装入数据库。XML 文档的元素映射到数据库中的表名，元素属性映射到列。

关于使用装入程序软件包开发和装入数据有用资源的信息，请参阅第 323 页的第 37 章，『装入商店数据的概述』。

**何时使用 WebSphere Commerce 装入程序软件包：**使用 **WebSphere Commerce** 装入程序软件包将数据库有用资源装入 **WebSphere Commerce** 数据库并对其进行更新。装入程序软件包还可以用来自动化后端系统的定期数据供给。

- **WebSphere 贸易加速器**

**WebSphere 贸易加速器**是主要用于通过各种商店操作维护网上商店的联机工具的工作台。然而，由于 **WebSphere 贸易加速器**允许您创建或编辑数据，您也可以将它用作商店开发工具，特别是当您更改少量数据的时候。关于您可以使用 **WebSphere 贸易加速器**编辑的数据库有用资源列表，请参阅第 107 页的第 6 部分，『开发商店数据』。

**何时使用 WebSphere 贸易加速器：**当您要创建或更新数据时，使用 **WebSphere 贸易加速器**。

- **直接编辑数据库**

您始终可以选择直接使用 SQL 插入、更新或删除来编辑数据库。

**注：**SQL 是特定于数据库的。Oracle 可能需要不同的 SQL 语法。请注意 SQL 语句将需要特定于数据库的值并且另一 **WebSphere Commerce** 服务器实例中可能不可再用这些 SQL 语句。

正在创建或更改商店数据的开发者必须熟悉 **WebSphere Commerce** 商店体系结构、商店数据和商店归档文件。要为实现定制的商店功能或与现有数据库信息集成而修改并扩展 **WebSphere Commerce** 数据库模式，开发者应该具备 DB2® 或 Oracle 数据库管理员技能。

---

## 方案：开发和部署生产商店

本节概述了使用 **WebSphere Commerce** 开发生产商店的推荐方案。

表 1. 方案: 开发和部署生产商店

任务	子任务	引用
确定哪个受支持的业务模型反映了您的公司		第 13 页的第 2 章, 『WebSphere Commerce 中受支持的业务模型』
确定商店流程		第 73 页的第 4 部分, 『开发商店前台』
创建使用案例		第 73 页的第 4 部分, 『开发商店前台』
分析随 WebSphere Commerce 提供的样本商店		《WebSphere Commerce 样本商店指南》
确定将哪个样本商店或其它样本用作起点		《WebSphere Commerce 样本商店指南》
创建一个基线组的商店有用资源	在开发环境（特定于 WebSphere Commerce 开发环境）中创建项目	WebSphere Studio 产品文档
	在开发环境中发布样本商店归档文件之一	WebSphere Commerce 产品联机帮助, 帮助主题“发布商店归档文件”
	如果可能, 请使用 WebSphere 贸易加速器中的更改流程工具来配置商店	WebSphere Commerce 产品联机帮助, 帮助主题“使用 WebSphere 贸易加速器更改商店流程”
	进行任何必要的数据库模式更改	《WebSphere Commerce 编程指南与教程》
	将商店有用资源登记到正在创建主副本的源控制系统中	WebSphere Commerce V5.5 Customization and Deployment Handbook SG24-6969 红皮书。
确定要从基线有用资源（商店前台、数据和服务器开发）创建商店所必须执行的开发	确定商店外观的更改	《WebSphere Commerce 样本商店指南》 第 73 页的第 4 部分, 『开发商店前台』
	确定商店页面的高速缓存策略	第 73 页的第 4 部分, 『开发商店前台』
	确定商店数据的更改	《WebSphere Commerce 样本商店指南》 第 107 页的第 6 部分, 『开发商店数据』
	了解样本商店中使用的实现	《WebSphere Commerce 样本商店指南》
	分析现有服务器功能, 确定何处需要增强或定制	WebSphere Commerce 产品和开发联机帮助
	确定与后终端系统集成的所需等级	WebSphere Commerce 产品和开发联机帮助

表 1. 方案: 开发和部署生产商店 (续)

设置组环境	每个开发者都在 IDE 中设置了一个开发项目, 用来自源控制系统中主副本的有用资源和与产品一起提供的服务器有用资源来填充项目	<i>WebSphere Commerce V5.5 Customization and Deployment Handbook</i> SG24-6969 红皮书。
	开发者使基线有用资源组开始运行	
	组在商店中分别从客户和管理员的角度来熟悉现有功能	《 <i>WebSphere Commerce 样本商店指南</i> 》
开发商店有用资源	修改并增强商店前台有用资源或创建新商店前台有用资源	第 73 页的第 4 部分, 『开发商店前台』
	开发附加服务器功能 (写新命令、EJB、与后端系统整合)	《 <i>WebSphere Commerce 编程指南与教程</i> 》
	修改数据并创建附加数据	第 107 页的第 6 部分, 『开发商店数据』
创建生产就绪数据		第 107 页的第 6 部分, 『开发商店数据』
将已开发的有用资源部署到生产		《 <i>WebSphere Commerce 编程指南与教程</i> 》 还可以在 <i>WebSphere Commerce V5.5 Customization and Deployment Handbook</i> , SG24-6969 红皮书中找到其它信息。



---

## 第 2 部分 由 WebSphere Commerce 支持的业务模型



## 第 2 章 WebSphere Commerce 中受支持的业务模型

在用 WebSphere Commerce 开始开发商店或站点之前，您需要了解 WebSphere Commerce 支持哪些业务模型。使用 WebSphere Commerce 创建的大多数商店将是这些业务模型中某一个模型的实例。

注：您也可以使用 WebSphere Commerce 来创建不符合本章中所述业务模型的商店。

### 了解 WebSphere Commerce 中受支持的业务模型

WebSphere Commerce 提供了体系化的基础结构，以将适合的公司放入以下在线业务模型中之一：

- 直销
-  主管
-  价值链

#### 直销

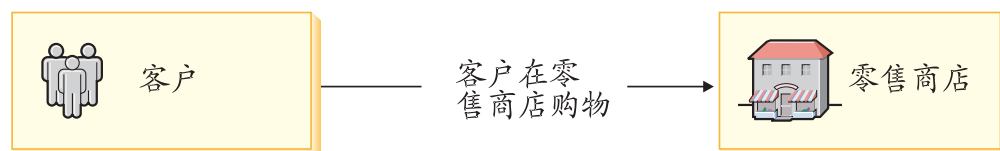
直销支持涉及产品、服务或信息的商业交易，此交易可以直接在公司和消费者之间进行，也可以在两个公司或合作伙伴之间进行。WebSphere Commerce 支持以下类型的直销业务模型：

- 消费者直销
-  B2B 直销

#### 消费者直销

消费者直销支持在公司和消费者之间的涉及产品、服务或信息的商业交易。在消费者直销方案中消费者通常直接从公司购买商品或服务。

下图说明了一个典型的消费者直销公司。



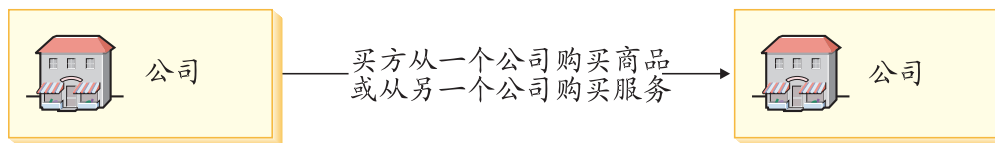
在典型的消费者直销公司中，客户直接从公司购买，正如本图表中显示的那样，此公司通常是零售商。此公司可以是零售商、通过自己的零售销路直接向消费者销售商品的生产商，也可以是其它任何直接向消费者销售商品或提供服务的公司。例如，一个通过产品目录向消费者销售的公司可以被认为是消费者直销公司。

传统上不认为是公司的组织（如政府）也可以被认为是消费者直销公司。政府可以直接向客户提供商品和服务。

## B2B 直销

**Business** B2B 直销支持两个企业或合作伙伴之间涉及产品、服务或信息的商业交易。典型的 B2B 直销交易发生在买方、供应商、生产商、转销商、经销商和贸易伙伴之间。

下图说明了一个典型的 B2B 直销公司。



在一个典型的 B2B 直销公司中，公司直接从其它公司那里购买商品或服务。销售方公司可以是批发商、经销商、生产商或从其它公司向买方销售的零售商。

传统上不认为是公司的组织，例如政府和媒体，也可以被认为是 B2B 直销公司。政府可能直接向公司提供商品和服务。

## 主管

**Business** 主管模型通过因特网服务供应商（ISP）或其它主管供应商来支持对商家或其它公司的主管。

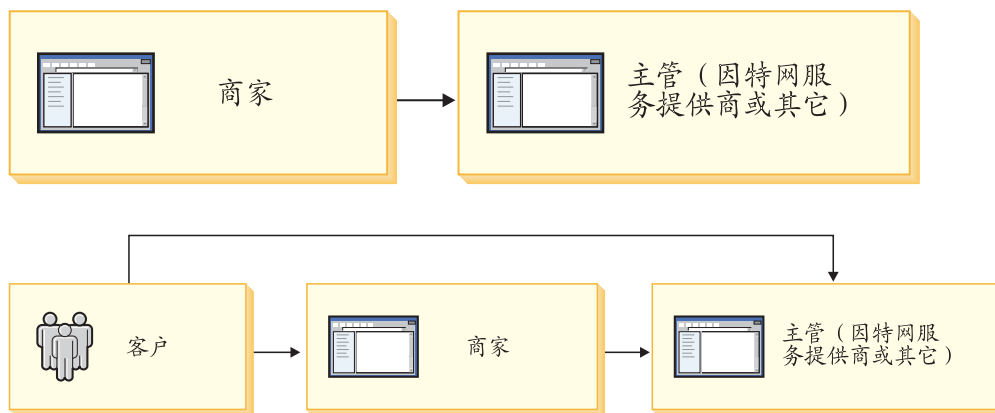
主管商务有两个可能具有的方面：

- 主管商店
- （可选）一个使客户可找到供应商主管的商店的站点

为了管理与主管商店的关系，主管模型通常包含一个中心（在 WebSphere Commerce 中称为中心商店）。该中心提供允许商家创建和管理商店的自供应工具以及允许主管供应商管理所有主管商店的工具。

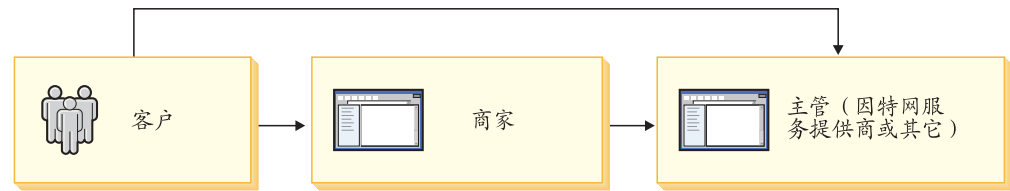
主管供应商通常还包含这样一个商店：客户可在其中找到并访问供应商主管的商店。

下图说明了主管的一个示例。



在此示例中，商家进入主管的站点并创建将由该站点主管的商店。主管供应商通常为商家提供简单的自供应工具，这些工具允许商家管理主管商店。当主管商店开始营业

时，客户可以通过主管的站点或通过直接进入主管商店来访问商店。



在此示例中，客户可以选择直接进入主管商店或商务，也可以选择浏览主管的站点然后再转移到主管商店或商务。

主管商店与消费者直销商店非常相似。有关两者之间的具体不同之处（如 WebSphere Commerce 样本商店中所实现的那样），请参阅《WebSphere Commerce 样本商店指南》。

## 价值链

**Business** WebSphere Commerce V5.5 的新增功能是支持涉及多个企业的在线商务交易。价值链支持涉及多个企业或多方的交易。产品、商品、服务或信息通过价值链组从生产方交付到最终用户手中。价值链也具有关系和管理方面的功能，也就是说，您可以管理合作伙伴或企业在价值链中的关系，也可以向各方提供一些管理服务。

因此，价值链必须管理公司的两个方面：公司的客户和直销以及公司的渠道伙伴和供应商。每个方面均需要其自身的管理渠道和做法。

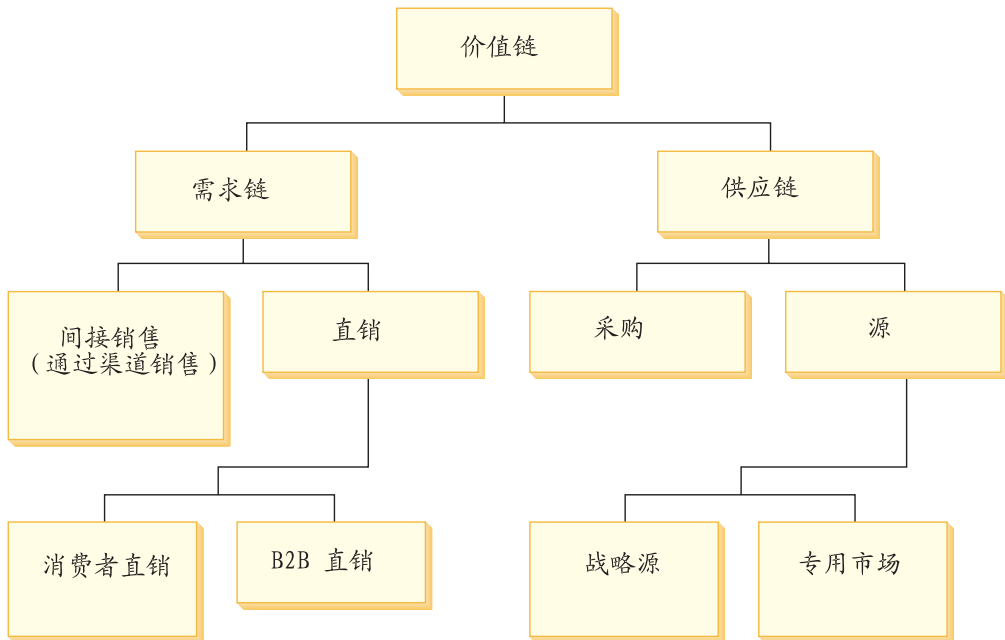
为了管理与伙伴或供应商的关系，价值链业务模型通常包含一个中心（在 WebSphere Commerce 中称为中心商店）。价值链管理员可以在中心商店中管理价值链的操作方面，包括使伙伴或供应商可以参与该价值链，即注册这些伙伴和供应商、对他们进行设置、建立协作。伙伴和供应商还可以访问中心商店以完成诸如注册用户之类的管理任务。

为了直接向客户销售（直销），价值链通常包含一个商店前台，客户可以在此直接购买商品或服务。

WebSphere Commerce 支持通过以下两种类型价值链的交易和此两种类型价值链的关系管理：

- 需求链
- 供应链

下图提供了在价值链中受支持的合作伙伴和关系的概述。



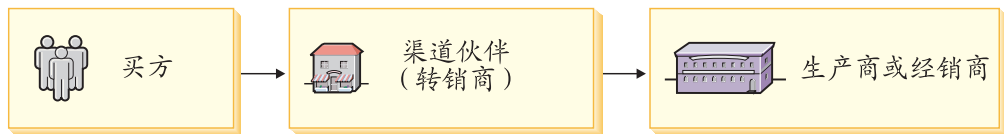
## 需求链

**Business** 需求链是由销售公司商品或服务的企业组成的。例如，需求链可以由发起销售交易的买方、销售生产商商品的转销商和制造商品的生产商组成。也可以由销售生产商商品的转销商、制造商品的生产商和把生产商商品供应给转销商的经销商组成。需求链也支持直销渠道，在此渠道中需求链所有者自己直接向客户或合作伙伴销售。关于直销的更多信息，请参阅第 13 页的『直销』。

**需求链主管：** 需求链所有者可以为其渠道合作伙伴（如转销商和经销商）主管商店。

下图说明了由 WebSphere Commerce 支持的一些需求链的示例。

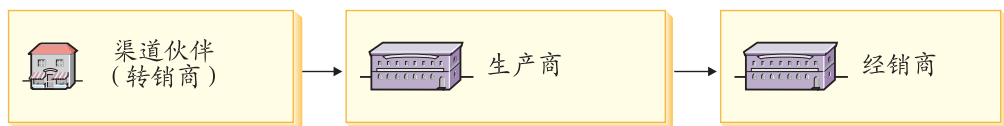
**买方、渠道伙伴（转销商）和生产商：**



在此示例中，买方从生产商转销商（渠道伙伴）那里购买商品。而转销商又通过生产商的中心从生产商那里获取商品。

**注：** 转销商可以由生产商主管管理，也可以是远程的。

**转销商、生产商和经销商：**



在本示例中，生产商提供了为其渠道合作伙伴（包括转销商）提供了一个中心。在此中心内，转销商和其它通道伙伴能够执行若干功能，包括对生产商商品的经销商的定位。

为了定位供应商，转销商可以在专用中心内浏览产品目录。如果不只一个经销商可以提供期望的产品，转销商就可以检查产品可供应量、经销商的位置和不同经销商的价格。然后（如果转销商作出这样的选择）他们可以将订单在若干经销商之间进行分割。随后订单将发送至经销商，由他们来完成交易并将商品或服务交付给转销商。然后转销商就把商品或服务直接销售给消费者。

需求链样本站点 **Commerce Plaza** 就是此转销商、生产商和经销商方案的一个示例。

**注：**转销商可以由生产商主管管理，也可以是远程的。

**其它方案：**本段中描述的示例只是几个需求链实例。方案详细信息可能会根据所实施公司的类型而更改。例如，如果企业是生产商，那么中心的目的可能就是帮助生产商的转销商从若干经销商那里找到生产商的商品。如果企业是经销商，则中心的目的可能就是帮助经销商的转销商从若干不同供应商那里查找商品或服务。

## 供应链

Business

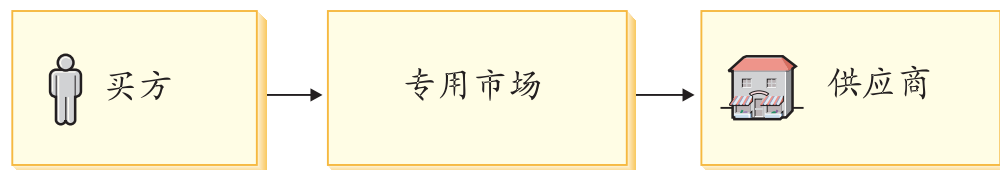
供应链由向公司提供服务的企业组成。**WebSphere Commerce** 提供体系性的基础结构，以支持形式为私有市场的供应链。

私有市场为供应商的待售商品报价提供了论坛。买方进入此论坛并在浏览了可用选项之后，可选择相应的商品或服务。

**注：**专用市场不支持竞争投标和反投标或其它竞争方法。

**供应链主管：**供应链所有者可以为其供应商主管商店。

下图说明了供应商公司的一个示例。



在此示例供应链中，买方进入供应商的中心进行交互并浏览综合产品目录，在该产品目录上列有来自多家供应商的产品和报价。然后买方可以选择期望报价或者请求来自多家供应商的报价。买方还可择直接从在线供应商处进行商务活动或进行采购。

## WebSphere Commerce 中的样本商店

**WebSphere Commerce** 提供了若干样本商店，您可以通过使用这些商店来熟悉 **WebSphere Commerce** 是如何支持本章中列出的不同业务模型的。以下是可用的样本（和相应的商店归档文件）：

表 2.

消费者直销	Business B2B 直销	Business 主管	Business 需求链	Business 供应链
FashionFlow (ConsumerDirect.sar) Express Express Store (Express Store.sar)	多乐五金 (B2BDirect.sar)	注: 下列所有商店在组合商店归档文件 Hosting.sar 中也有提供。建议您发布 Hosting.sar 以查看整个主管样本。	注: 下列所有商店在组合商店归档文件 DemandChain.sar 中也有提供。建议您发布 DemandChain.sar 以查看整个需求链样本。	注: 下列所有商店在组合商店归档文件 SupplyChain.sar 中也有提供。建议您发布 SupplyChain.sar 以查看整个供应链样本。
		商业主管中心 (Hosting Hub.sar)	Commerce Plaza (ChannelHub.sar)	商业供应商中心 (SupplierHub.sar)
		商店目录 (Store Directory.sar)	产品目录有用资源商店 (CatalogAsset Store.sar)	产品目录有用资源商店 (Catalog AssetStore.sar)
		产品目录有用资源商店 (CatalogAsset Store.sar)	转销商商店前台有用资源商店 (Resellerstore frontAsset Store.sar)	供应商有用资源商店 (Supplier AssetStore.sar)
		主管商店前台有用资源商店 (HostedStore FrontAsset Store.sar)	经销商有用资源商店 (Distributor-Asset Store.sar)	供应商
		主管商店	主管转销商商店	
			经销商商店	

有关这些示例中商店类型的更多信息, 请参阅第 64 页的『了解商店体系结构如何支持业务模型』。有关样本商店的更详细信息, 请参阅《WebSphere Commerce 样本商店指南》。

注: 每个示例还包含组件商店归档文件, 该归档文件包含业务模型的组织结构。

请注意, 这些样本代表每个业务模型中的特定商店实例, 而不是为了演示业务模型中所有可能的可用变体。然而, 即使您特定的公司实例和所提供的样本有很大区别, 您也可以使用样本作为自己站点的一个起点, 或在创建自己站点的时候使用样本的部分。关于随 WebSphere Commerce 提供的样本的更多详细信息, 请参阅《WebSphere Commerce 样本商店指南》。



---

## 第 3 部分 WebSphere Commerce 体系结构

本部分提供 WebSphere Commerce 体系结构如何支持开展联机业务的概述。尤其是，本部分讨论了 WebSphere Commerce 体系结构的组件如何允许业务中的各方（例如，客户、业务合作伙伴或经销商、转销商和供应商）进行在线交互。

为了使与您的业务相关的各方（例如，客户、业务合作伙伴、承包商、供应商、生产商、经销商和管理员）能够与您的公司在线交互或者相互之间在线交互，WebSphere Commerce 包含了以下体系结构组件：

- 组织结构
- 访问控制模型
- 业务策略框架
- 实例体系结构
- 商店体系结构

结合这些组件创建的体系结构支持您公司的不同业务合作伙伴在相互间进行交互。



## 第 3 章 WebSphere Commerce 组织结构

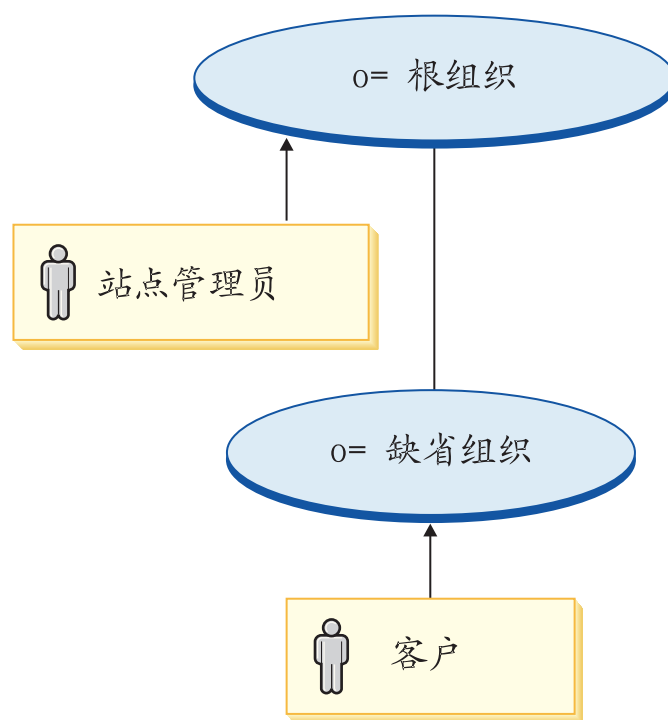
为了允许客户或买方访问您的站点、浏览产品目录并下订单；或允许员工管理站点（包括更新产品目录、创建新促销或管理订单）；或允许转销商或其他业务合作伙伴在站点上完成交易，必须在 WebSphere Commerce 组织结构中为您所有的公司方案操作者都指定职位。

### 了解 WebSphere Commerce 组织结构

WebSphere Commerce 组织结构为您公司方案中的操作者或实体提供了框架。在层次结构中组织此框架，此层次结构通过实体为组织和组织单位及用户模仿典型的组织层次结构。框架中的组织和组织单位对您公司的部件充当所有者的角色。公司各方（包括客户、管理员、商店、产品目录和经销商）必须属于组织或组织单位。

第 31 页的第 4 章，『WebSphere Commerce 中的访问控制』中讨论的组织结构和访问控制模型是紧密相关的，因为访问控制模型将访问控制策略应用于组织，而不是将其应用于单个实体（商店、客户、管理员等等）。适用于实体（或资源）的策略将应用于拥有实体或资源的组织。

下图概述了基本的 WebSphere Commerce 组织结构。不管业务模型如何，基本的组织结构将在实例创建期间安装。



- **根组织**：根组织是顶级组织并且是自己的父组织。WebSphere Commerce 组织结构中的所有组织都是根组织的子代。站点管理员属于根组织。

- **缺省组织**: 缺省组织属于根组织。消费者直销方案中的所有临时客户和所有客户都属于缺省组织。B2B 直销和价值链方案中的客户可以属于缺省组织，也可以属于其它组织。

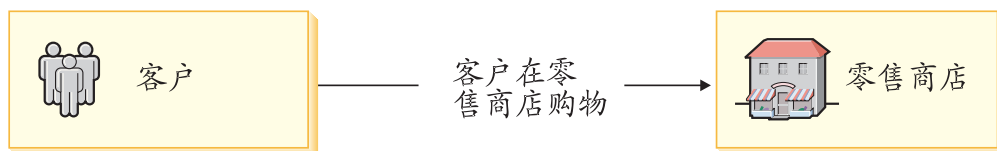
一个或多个其它级别的组织实体可以存在于父组织实体之下。您可以按支持公司所需的数量来添加子组织实体。

## 组织结构如何支持业务模型？

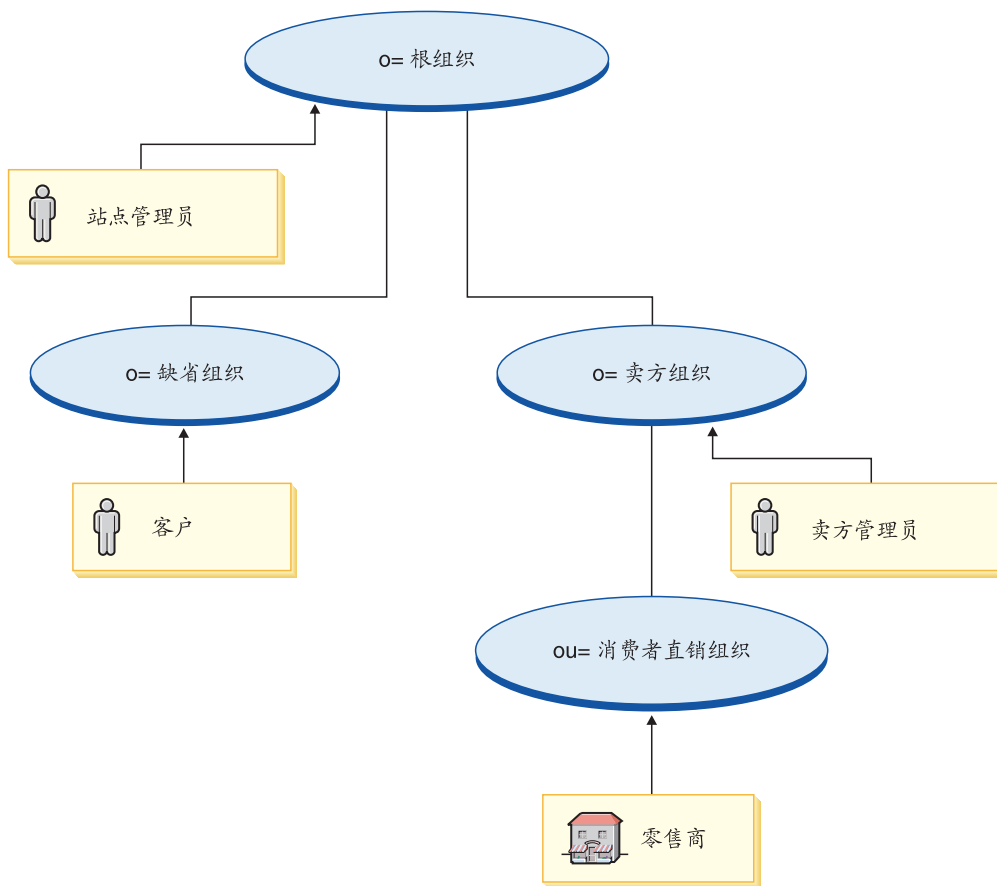
WebSphere Commerce 组织结构非常灵活，可以支持受支持业务模型中的所有实体。以下部分中的图表说明了典型的业务模型示例是如何映射到 WebSphere Commerce 组织结构中的。

### 消费者直销

下图说明了一个典型的消费者直销公司。



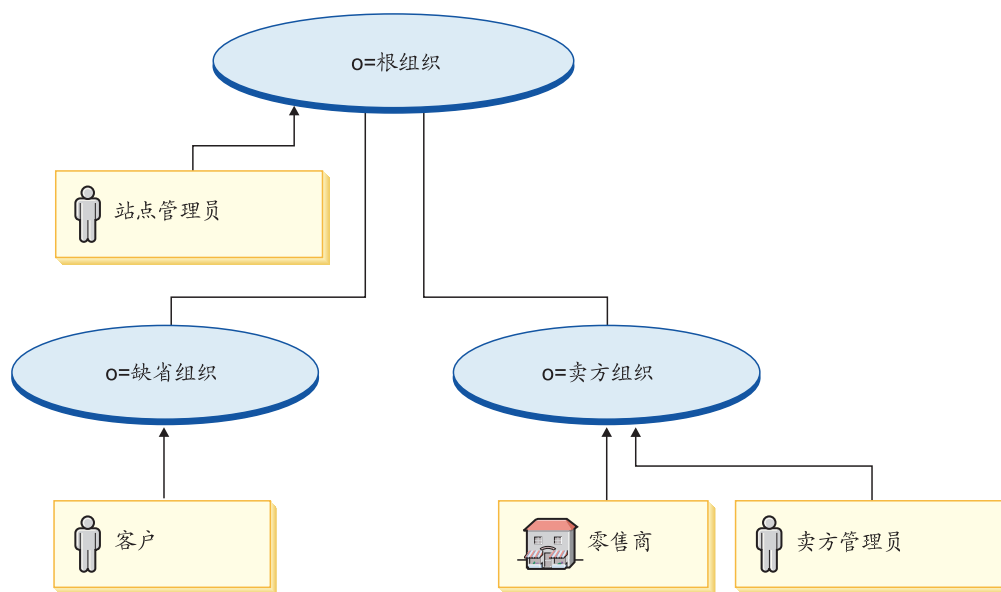
为使该公司与 WebSphere Commerce 在线，上图中的实体必须分配到以下组织中：



- **根组织**: 公司中的所有组织都成为根组织的子代。维护在线站点的站点管理员属于根组织。

- **缺省组织**: 所有的公司客户都属于缺省组织。
- **卖方组织**: 创建卖方组织的目的是拥有所有的卖方组织（包括商店和维护商店的管理员）。维护商店功能的管理员（例如客户服务代表、产品目录和产品经理）称为卖方管理员并直接属于卖方组织。
  - 子组织单位（ou），即消费者直销组织，是创建于卖方组织之下以拥有商店的（即零售商）。

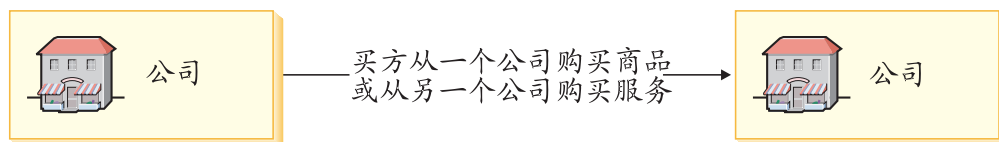
Express WebSphere Commerce - Express 中的组织结构与上述消费者直销组织稍有不同。为使消费者直销公司与 WebSphere Commerce - Express 在线，上述消费者直销图中的实体必须分配到以下组织中：



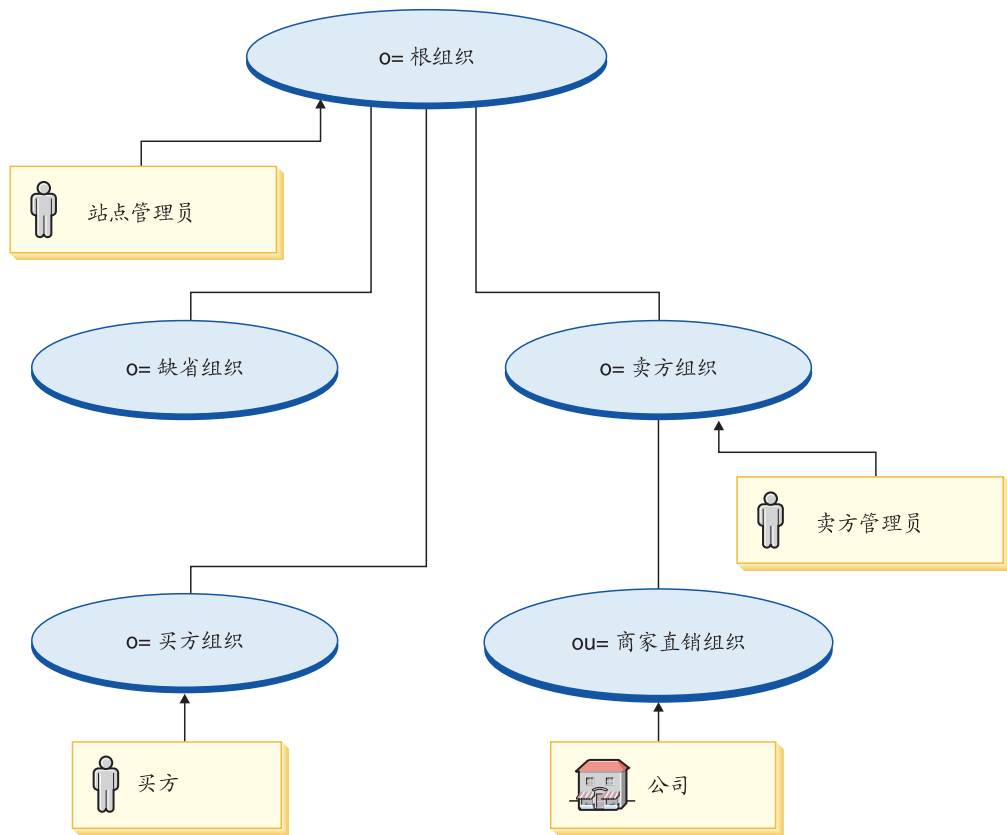
- **根组织**: 公司中的所有组织都成为根组织的子代。维护在线站点的站点管理员属于根组织。
  - **缺省组织**: 所有的公司客户都属于缺省组织。
  - **卖方组织**: 创建卖方组织是为了拥有所有商店（零售商）和维护商店的管理员。维护商店功能的管理员（例如客户服务代表、产品目录和产品经理）称为卖方管理员并直接属于卖方组织。

## B2B 直销

Business 下图例举了一个典型的 B2B 直销公司。



为了在线放置此公司，上述图表中的实体必须分配到以下组织中：

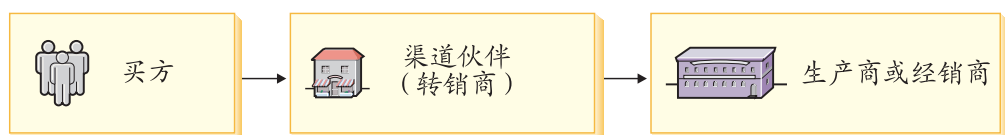


- **根组织：** 公司中的所有组织都成为根组织的子代。维护在线站点的站点管理员属于根组织。
  - **缺省组织：** 与消费者直销组织结构不同的是，客户不属于缺省组织。客户是买方组织所拥有的买方。
  - **买方组织：** 在 B2B 直销组织结构中，为客户（在 B2B 直销公司中称之为买方）指定了他们自己的组织。
  - **卖方组织：** 创建卖方组织的目的是拥有所有拥有商店的组织。维护商店功能的管理员（例如客户服务代表、产品目录和产品经理）被称为卖方管理员并直接属于卖方组织。
    - 子组织单位（ou），即 B2B 直销组织，是创建于卖方组织之下以拥有商店的（即公司）。

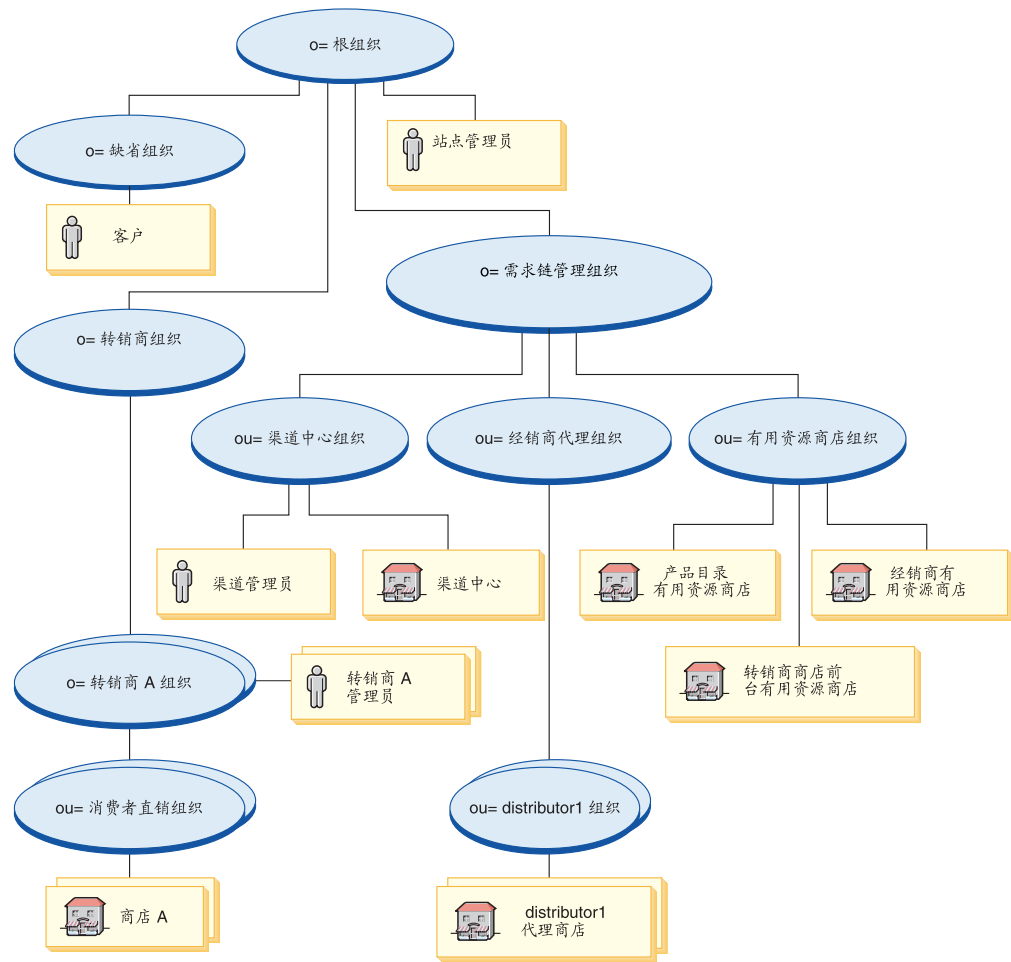
## 需求链

### Business

下图说明了某需求链公司的一个示例。



为了使此公司在线，先前图表中的实体必须分配到以下组织中：

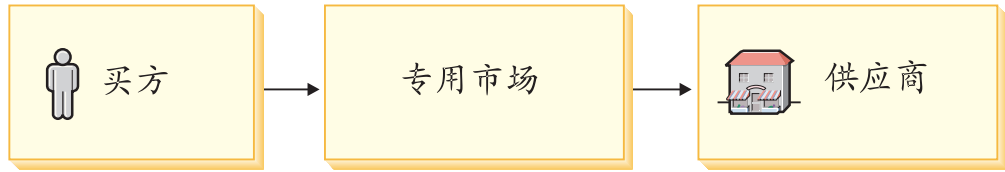


- **根组织：** 公司中的所有组织都成为根组织的子代。同时，维护在线站点的站点管理员，即站点管理员，也直接添加在根组织下。
  - **缺省组织：** 缺省情况下，缺省组织下什么都不放置。转销商商店的客户可以放在该组织下。
  - **需求链管理组织：** 创建需求链管理组织是为了拥有与渠道相关的所有组织（被拥有转销商的组织排除在外的）。需求链管理组织拥有以下子组织单位：
    - **渠道中心组织：** 创建渠道中心组织是为了拥有渠道中心。维护渠道中心功能及管理转销商组织的管理员称为渠道管理员并直接属于渠道中心组织。
    - **经销商代理组织：** 创建经销商代理组织是为了拥有所有与经销商的连接。为组织中的每个经销商代理都创建了子组织单位。
      - **经销商组织：** 为站点中每个经销商代理都创建了新的经销商组织单位。
    - **有用资源商店组织：** 创建有用资源商店组织是为了拥有所有用于为渠道合作伙伴（转销商和经销商）创建商店的有用资源。
  - **转销商组织：** 创建转销商组织是为了拥有需求链中的所有转销商。将为每个转销商创建子组织。
    - **转销商组织 A、B、C：** 父级转销商组织之下为每个转销商商店都创建了新的转销商组织。维护商店功能的管理员（例如客户服务代表、产品目录和产品经理）称为转销商管理员并直接属于相应的转销商组织。

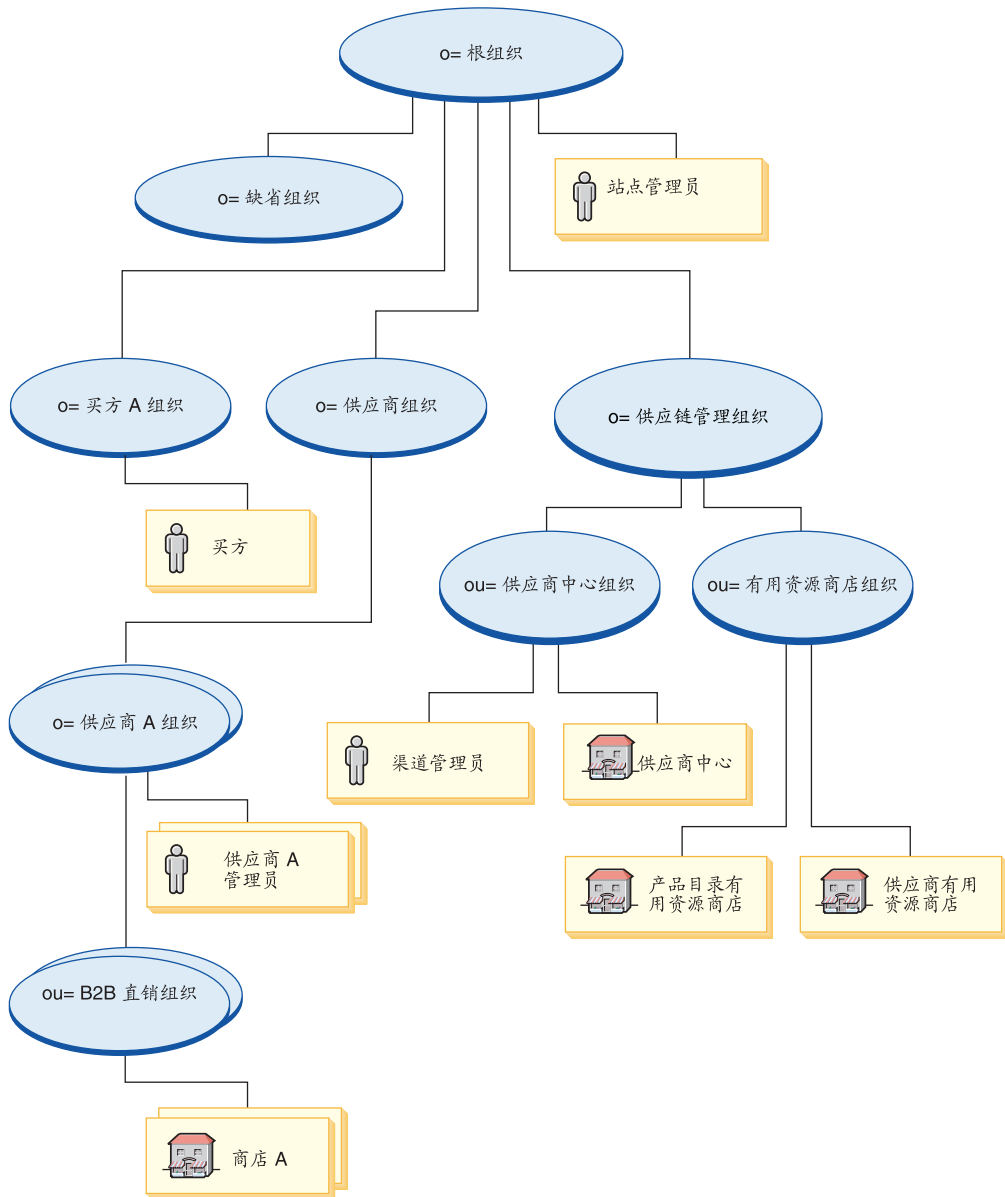
# 供应链

Business

下图说明了一个典型的供应链公司。



为了在线放置此公司，上述图表中的实体必须分配到以下组织中：



- **根组织**：公司中的所有组织都成为根组织的子代。同时，维护在线站点的站点管理员，即站点管理员，也直接添加在根组织下。
  - **缺省组织**：缺省情况下，缺省组织下什么不放置。

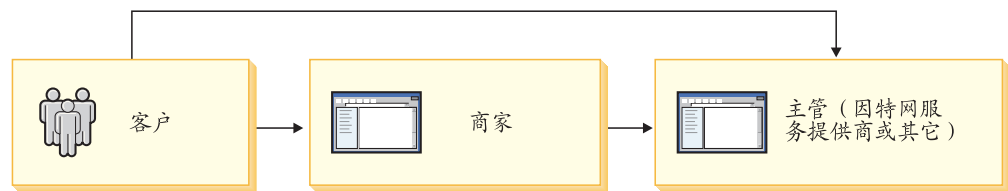


- **供应链管理组织**: 创建供应链管理组织是为了拥有与供应链相关的所有组织（被拥有转销商的组织排除在外的）。供应链管理组织拥有以下子组织单位:
  - **供应商中心组织**: 创建供应商中心组织是为了拥有供应商中心。维护供应商渠道中心功能及管理供应商组织的管理员被称为渠道管理员并直接属于供应商中心组织。
  - **有用资源商店组织**: 创建有用资源商店组织是为了拥有所有用于为供应商创建商店的有用资源。
- **供应商组织**: 创建供应商组织是为了拥有供应链中的所有供应商。将为每个供应商创建子组织。
  - **供应商组织 A、B、C**: 父级供应商组织之下为每个供应商商店都创建了新的供应商组织。维护商店功能的管理员称为供应商管理员并直接属于相应的供应商组织。
- **买方组织**: 在根组织下授予了买方自己的组织。所有买方都属于相应的买方组织。

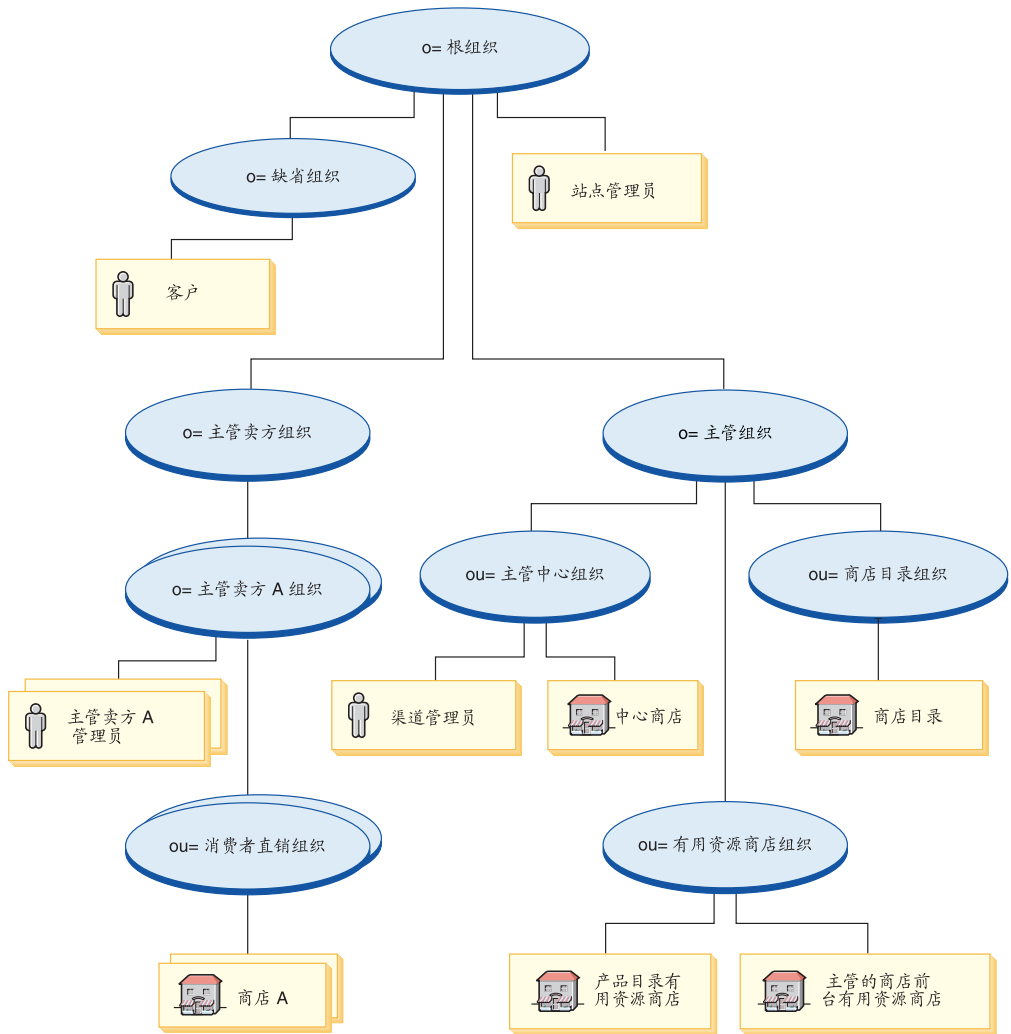
## 主管

### Business

下图说明了一个典型的主管公司。



为了使此公司在线，上述图表中的实体必须分配到以下组织中：



- **根组织：** 公司中的所有组织都成为根组织的子代。同时，维护在线站点的站点管理员，即站点管理员，也直接添加在根组织下。
  - **缺省组织：** 所有公司的客户都属于缺省组织
  - **主管组织：** 创建主管组织是为了拥有与主管相关的所有组织（拥有主管商店的组织除外）。主管组织拥有以下子组织单位：
    - **主管中心组织：** 创建主管中心组织是为了拥有主管中心。维护主管中心功能及管理主管组织的管理员称为渠道管理员并直接属于主管中心组织。
    - **商店目录组织：** 创建商店目录组织是为了拥有商店目录。
    - **有用资源商店组织：** 创建有用资源商店组织是为了拥有所有用于创建主管商店的有用资源。
  - **主管卖方组织：** 创建主管卖方组织是为了拥有所有主管商店。将为每个主管商店创建了子组织单位。
    - **主管商店组织 A、B、C：** 父级主管组织之下为每个主管商店都创建了新的主管商店组织。维护商店功能的管理人员称为主管卖方管理员，并直接属于相应的主管商店组织。

---

## 样本组织结构

WebSphere Commerce 为每个受支持的业务模型都提供了样本组织结构。这些样本组织结构都独立可用（作为组件商店归档文件），允许您把样本组织结构用作自己站点的起点或样本公司的一部分。关于样本组织结构的更多信息，请参阅《*WebSphere Commerce 样本商店指南*》。

---

## 创建组织结构

可以为自己的站点创建新的组织结构，但在此建议的是最好通过发布随 WebSphere Commerce 提供的样本组织结构之一来开始，然后根据需要对组织结构进行更改。关于编辑组织数据的更多信息，请参阅第 115 页的『了解 WebSphere Commerce 中的成员有用资源』。



---

## 第 4 章 WebSphere Commerce 中的访问控制

WebSphere Commerce 允许您通过访问控制来确定特定用户（可以是客户、采购员、管理员、经销商、生产商或供应商）可以执行的、与您的业务有关的任务。

关于 WebSphere Commerce 访问控制模型的详细信息包含在《*WebSphere Commerce 安全性指南*》中。为了了解访问控制是如何影响站点和商店开发的，这里也提供了简单的摘要。

---



### 了解 WebSphere Commerce 中的访问控制

WebSphere Commerce 中的访问控制由以下元素组成：用户、操作、资源和关系。

- 用户是使用系统的人。出于访问控制目的，用户必须分组到相关的访问组中。用来确定访问组成员关系的一个公共属性是角色。角色是按每个组织指定给用户的。关于角色的更多信息，请参阅第 116 页的『角色』。访问组的一些示例包括注册客户、临时客户或象客户服务代表这样的管理组。
- 操作是用户可对资源执行的活动。出于访问控制目的，操作也必须分组到相关的操作组中。例如，商店中使用的常用操作是视图。视图供调用向客户显示商店页面。您商店中使用的视图必须先声明为操作并指定给操作组，然后才能被访问。
- 资源是受到保护的实体。例如，如果操作是视图，则要保护的资源是调用该视图的命令，如 `com.ibm.commerce.command.ViewCommand`。出于访问控制目的，资源被分组到资源组中。
- 关系是指用户和资源之间的关系。访问控制策略可能要求满足用户和资源之间的一种关系。例如，可能只允许用户显示他们已经创建的订单。

### 访问控制策略

访问控制策略授权访问组对 WebSphere Commerce 的资源执行特定操作，只要访问组中的用户满足对于资源的特定关系。

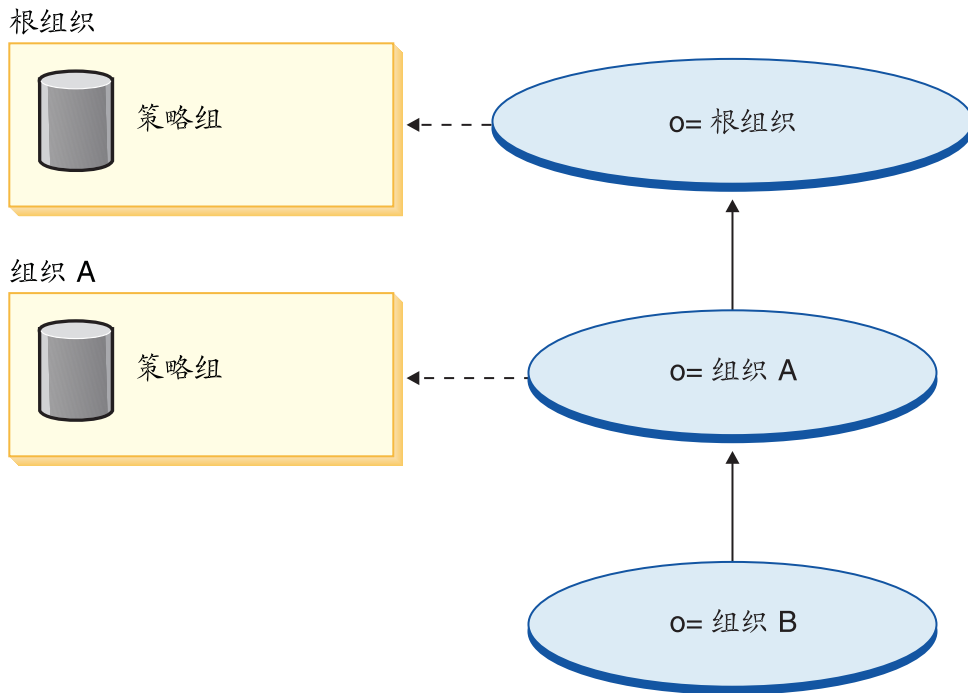
WebSphere Commerce 提供了三百个以上的缺省访问控制策略，这些策略已在实例创建期间装入。这些策略涵盖了范围广泛的普通业务活动，包括订单创建和处理以及贸易，例如对  报价和  合同的请求。《*WebSphere Commerce 安全性指南*》描述了缺省策略。

### 访问控制策略组

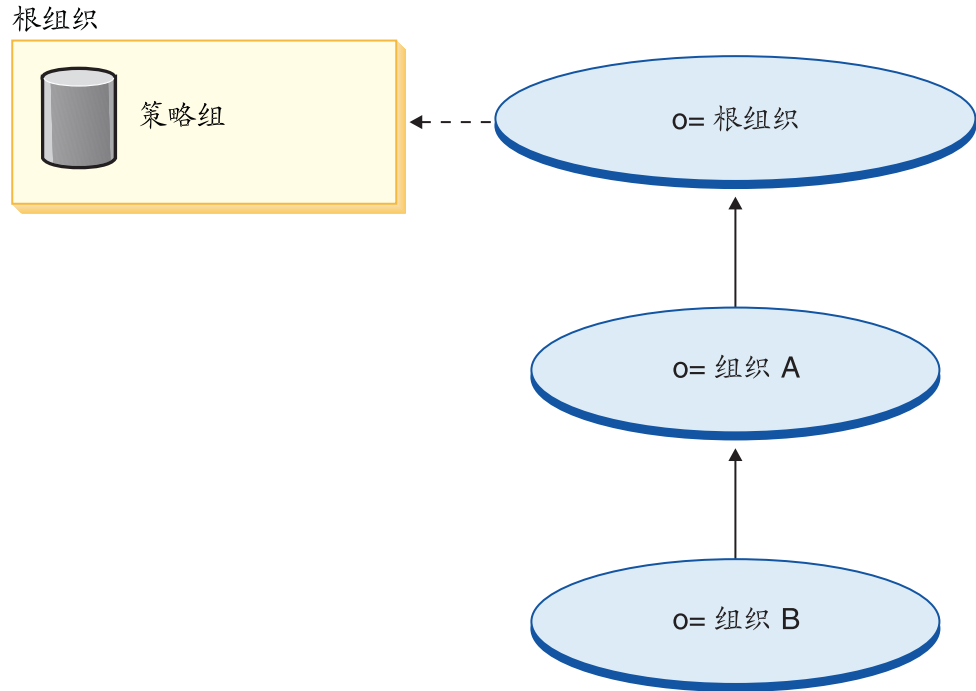
为将访问控制策略应用于您的商店或站点，它必须属于某个访问控制策略组，并且必须由拥有资源的组织预订该策略组。缺省情况下，随 WebSphere Commerce 提供的所有访问控制策略都分配给众多策略组。有关随 WebSphere Commerce 提供的缺省策略的列表，请参阅《*WebSphere Commerce 安全性指南*》。

虽然访问控制策略组为组织所拥有，但它们不会自动应用到组织中。为使访问控制策略应用到组织中，此组织必须预订策略组。如果此组织有子组织，则父组织预订的所有策略组都会自动应用到子组织中。但是如果子组织直接组织了一个策略组，则父组织预订的策略组将不再应用到子组织中。

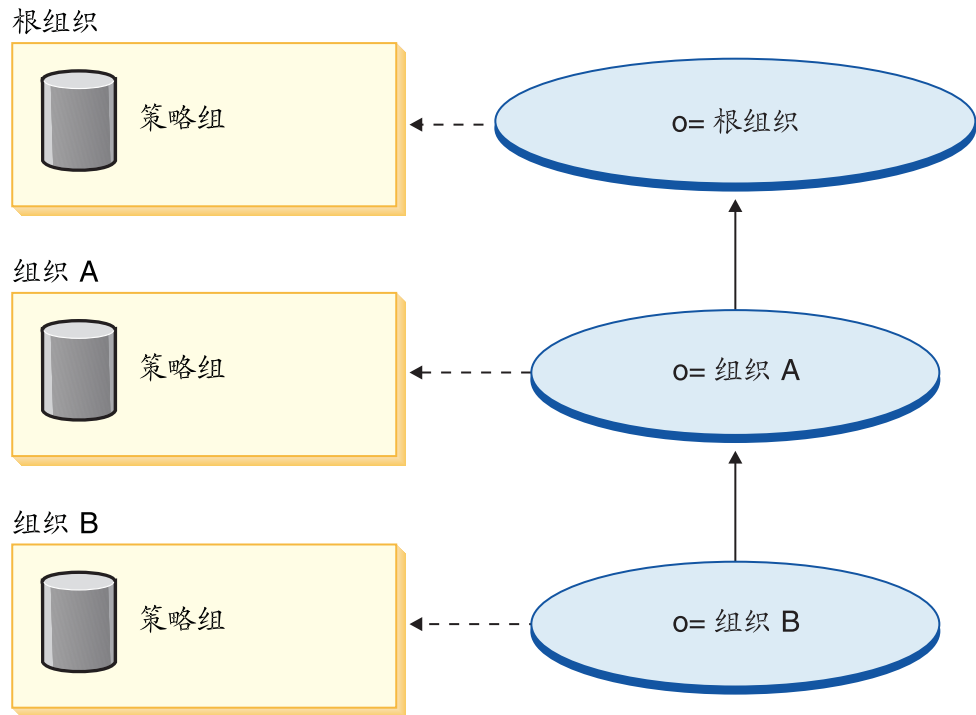
在先前版本的 WebSphere Commerce 中，一个策略适用于该策略的所有者组织的子代所拥有的所有资源。例如，如果组织 A 具有某个策略，并且它是组织 B 的父代，则组织 B 也默示地具有该策略。在 WebSphere Commerce 5.5 中，组织现在可以预订策略组。在 WebSphere Commerce 5.5 中，如果组织 B 不预订任何策略组，访问控制框架将开始搜索组织层次结构，直到遇到预订了至少一个策略组的组织。如果组织 B 的直属父级组织（组织 A）预订了一个策略组，则搜索停止，并且组织 A 的策略组中的这些策略应用于组织 A 和 B。这可以在下图中看出。



如果组织 A 没有预订策略组，搜索将在组织层次结构中继续进行，直至遇到预订了策略组的组织。这在下图中可以看出，其中根组织预订了一个策略组。组织 B 和组织 A 继承了该组中的策略。



如果组织 B 预订了一个策略组，则搜索在组织 B 处停止，且组织 B 只能应用其预订的那些策略，如下图中所示。



注：从访问控制的角度来说，资源的所有权具有特殊的意义。所有资源均必须实现 `com.ibm.commerce.security.Protectable` 接口。此接口上的方法之一是 `getOwner()`，它返回资源所有者的成员标识。例如，Order 实体 bean 是通过让其远程接口扩展 `Protectable` 接口而受到保护的资源。Order 的 `getOwner()` 实现是：特定 Order 资源返回下订单处的商店的所有者。对于命令（例如

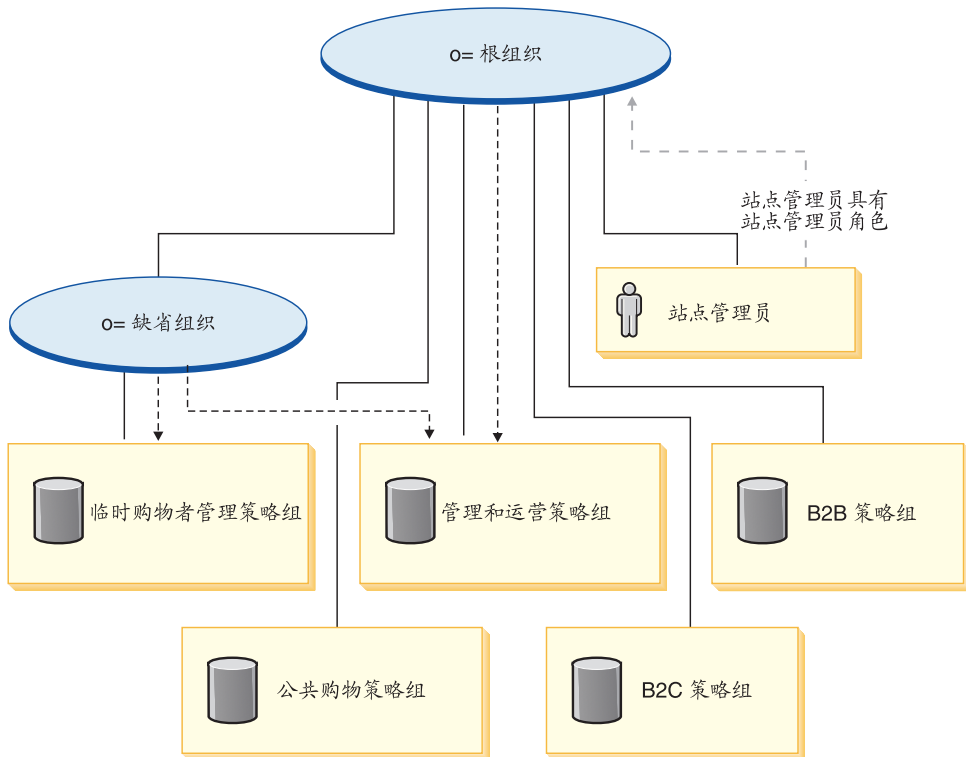
com.ibm.commerce.command.ViewCommand) 作为资源的策略, getOwner() 的缺省实现是返回当前处于命令上下文中的商店的所有者。如果在命令上下文中没有任何商店, 则使用根组织作为所有者。关于更多信息, 请参阅《WebSphere Commerce 编程指南与教程》。

## 了解业务模型中的访问控制

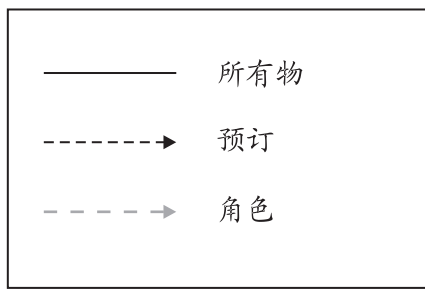
WebSphere Commerce 访问控制结构非常灵活, 可以支持受支持业务模型中的所有实体。以下部分中的图表说明了访问控制是如何应用于每个业务模型的典型示例。

### 基本访问控制结构

不管业务模型如何, 基本访问控制结构是在实例创建期间安装的。



图注



根组织拥有以下缺省策略组:

- 管理和运营
- 公共购物



- B2C
- B2B

然而，根组织仅预订管理和运营策略组。因此，这些策略适用于直接位于根组织下的站点管理员。

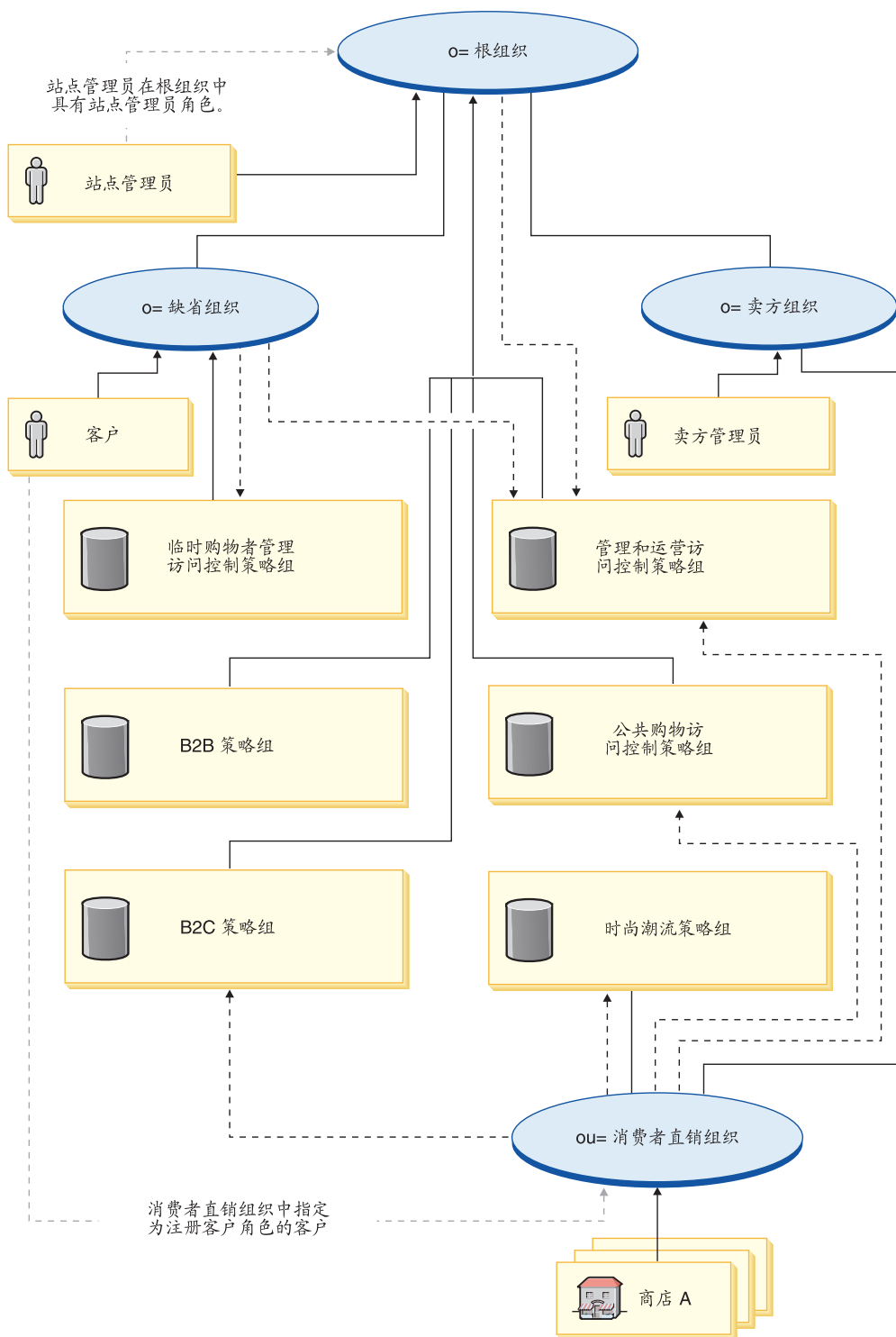
管理和运营策略组中的策略并不通过继承而适用于缺省组织，因为缺省组织预订了临时购物者管理策略组。为应用管理和运营策略，缺省组织必须显式预订管理和运营策略组。

缺省组织拥有临时购物者管理策略组。

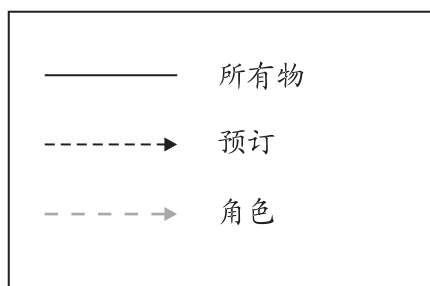
**注：**有关缺省策略组的更详细信息，请参阅《*WebSphere Commerce 安全性指南*》的附录。

## 消费者直销

下图描述了一个基本消费者直销组织和访问控制结构。



## 图注



在这个描述基本消费者直销组织的图中，根组织拥有并预订第 34 页的『基本访问控制结构』中描述的缺省策略组。

消费者直销组织直接预订 B2C 访问控制策略、管理和运营策略组以及公共购物策略组。

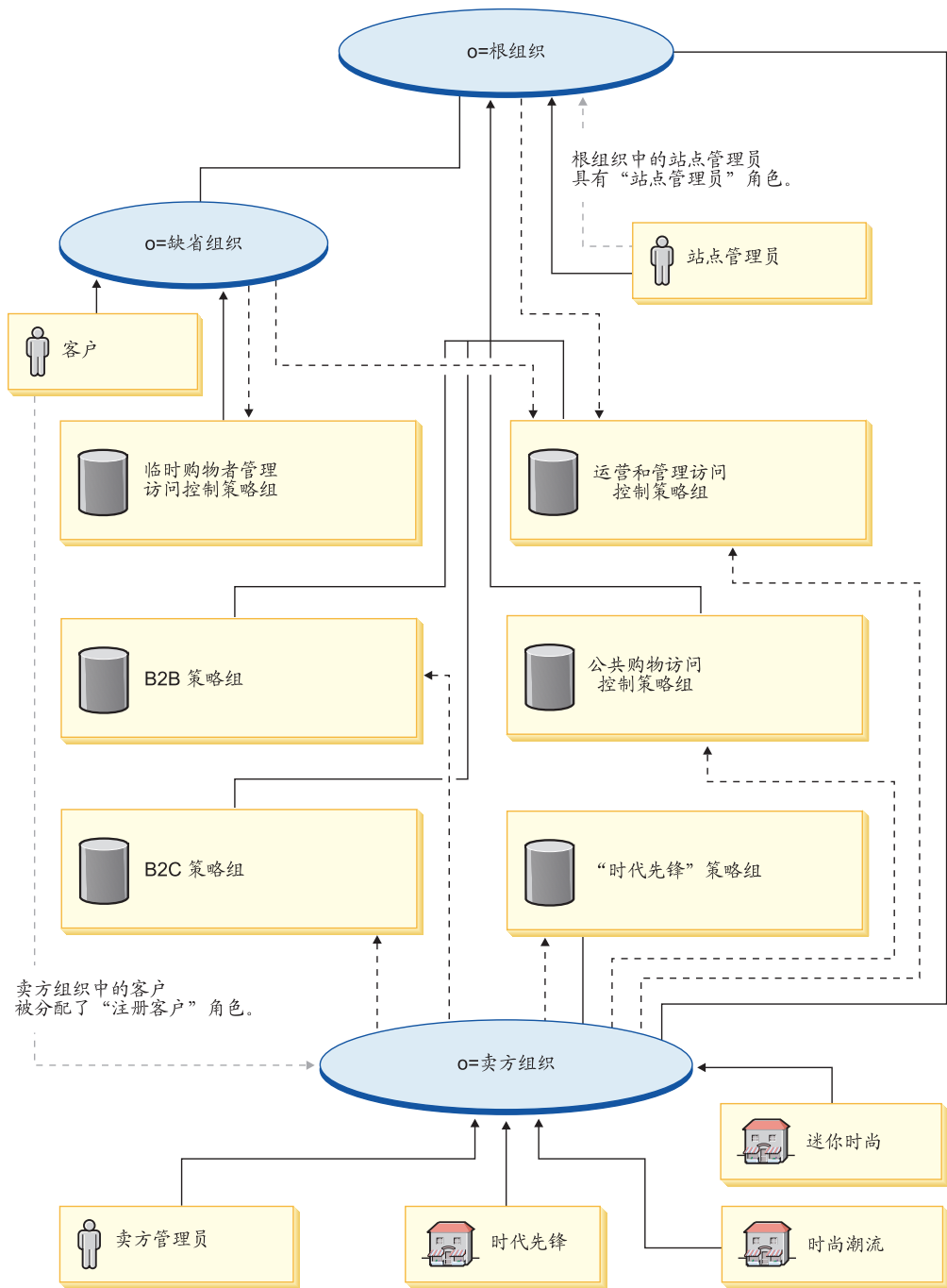
消费者直销组织还拥有和预订时尚潮流策略组。时尚潮流策略组包含以下策略：

`AllUsersExecuteFashionAllUsersViews`

由于访问控制策略组为组织实体所预订，因此如果您正在站点中创建多个商店并希望把不同的访问控制策略组应用于单个商店，则必须创建各自的组织以拥有各个商店。

**Express** WebSphere Commerce - Express 中的访问控制结构与上述消费者直销访问控制结构稍有不同。

下图描述 WebSphere Commerce - Express 中的访问控制结构:



在这个描述 WebSphere Commerce - Express 中消费者直销组织的图中，根组织拥有并预订缺省策略组，如第 34 页的『基本访问控制结构』中所述。

卖方组织直接预订 B2C 访问控制策略、B2B 访问控制策略、管理和运营策略组以及公共购物策略组。

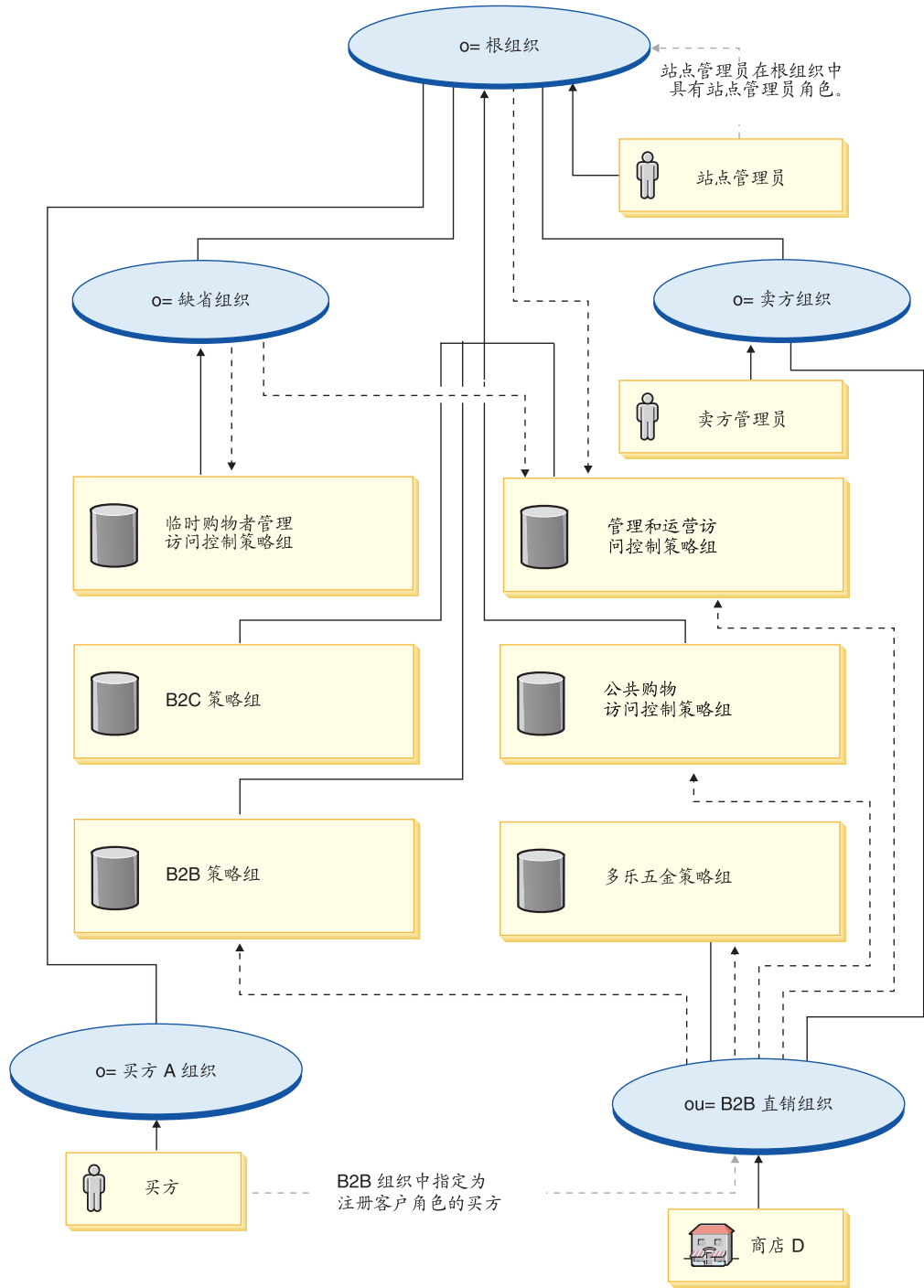
卖方组织还拥有和预订 Express 策略组。Express 策略组包含以下策略：

- AllUsersExecuteExpressAllUsersViews
- RegisteredUsersExecuteExpressAllUsersViews

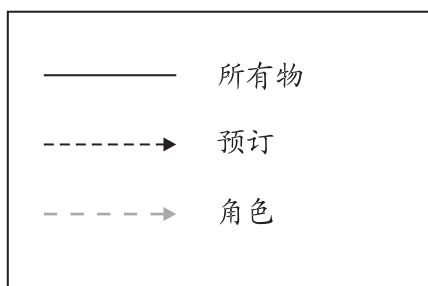
# B2B 直销

Business

下图描述了一个基本的 B2B 直销组织和访问控制结构。



## 图注



在这个描述基本 B2B 直销组织结构的图中，根组织拥有并预订第 34 页的『基本访问控制结构』中描述的缺省策略组。

B2B 直销组织直接预订 B2B、管理和运营以及公共购物策略组。

B2B 直销组织还拥有和预订多乐五金策略组。多乐五金策略组包含以下策略：

- AllUsersForToolTechExecuteToolTechAllUsersViews
- RegisteredCustomersForOrgForToolTechExecuteToolTechRegisteredCustomerViews

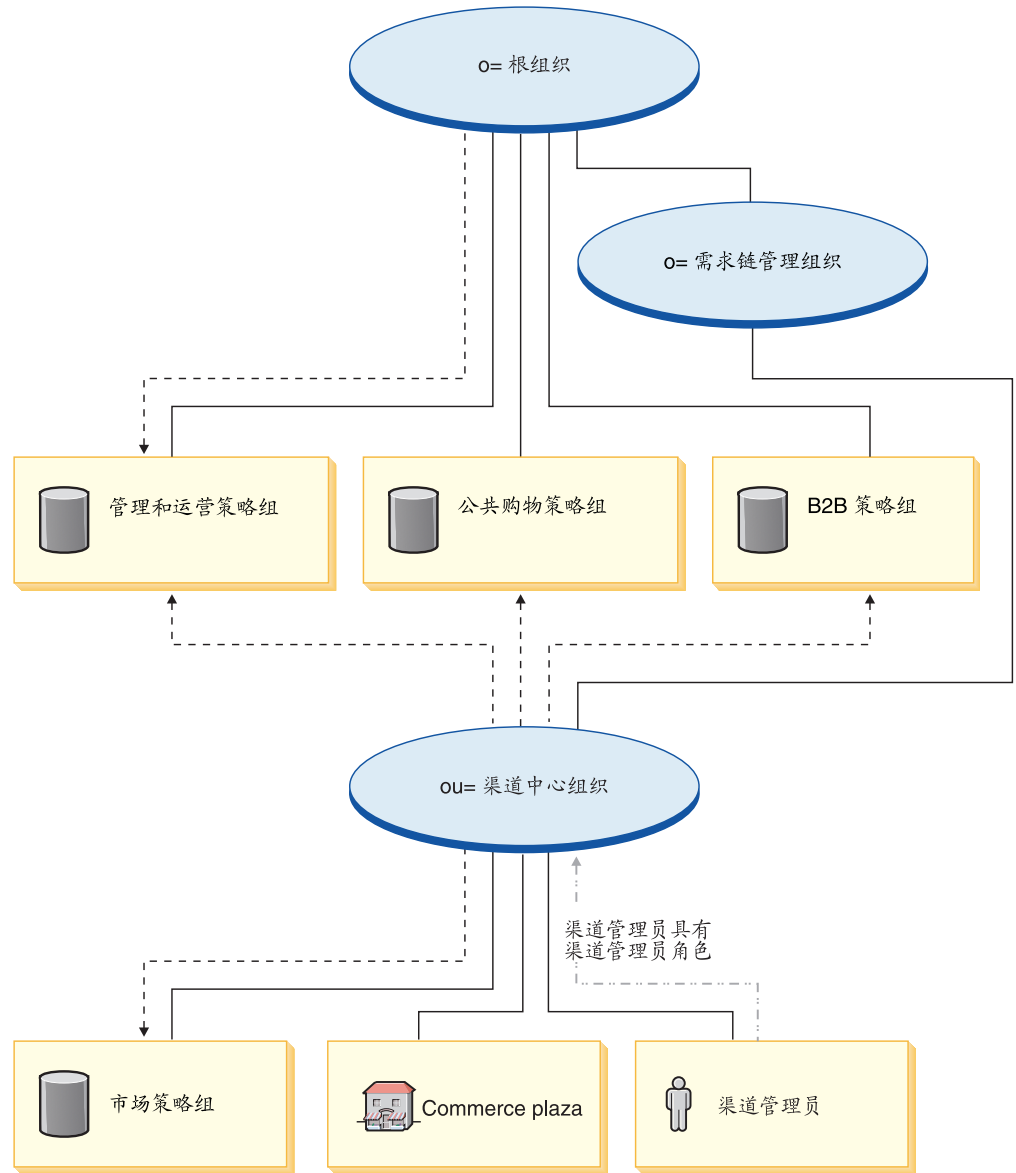
买方就是在 B2B 直销商店中下订单的客户。所有买方必须属于一个买方组织。通常，买方组织不预订任何策略组，因为从根组织继承的管理和运营策略已足够。

由于访问控制策略组为组织实体所预订，因此如果您正在站点中创建多个商店并希望把不同的访问控制策略组应用于单个商店，则必须创建各自的组织以拥有各个商店。

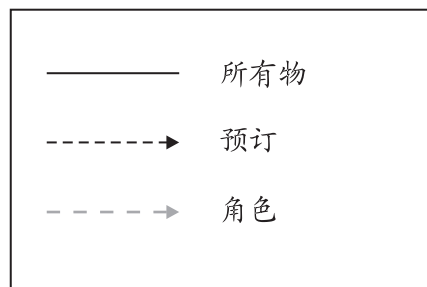
## 需求链

Business

在这些描述需求链组织结构的图中，根组织拥有并预订第 34 页的『基本访问控制结构』中描述的缺省策略组。



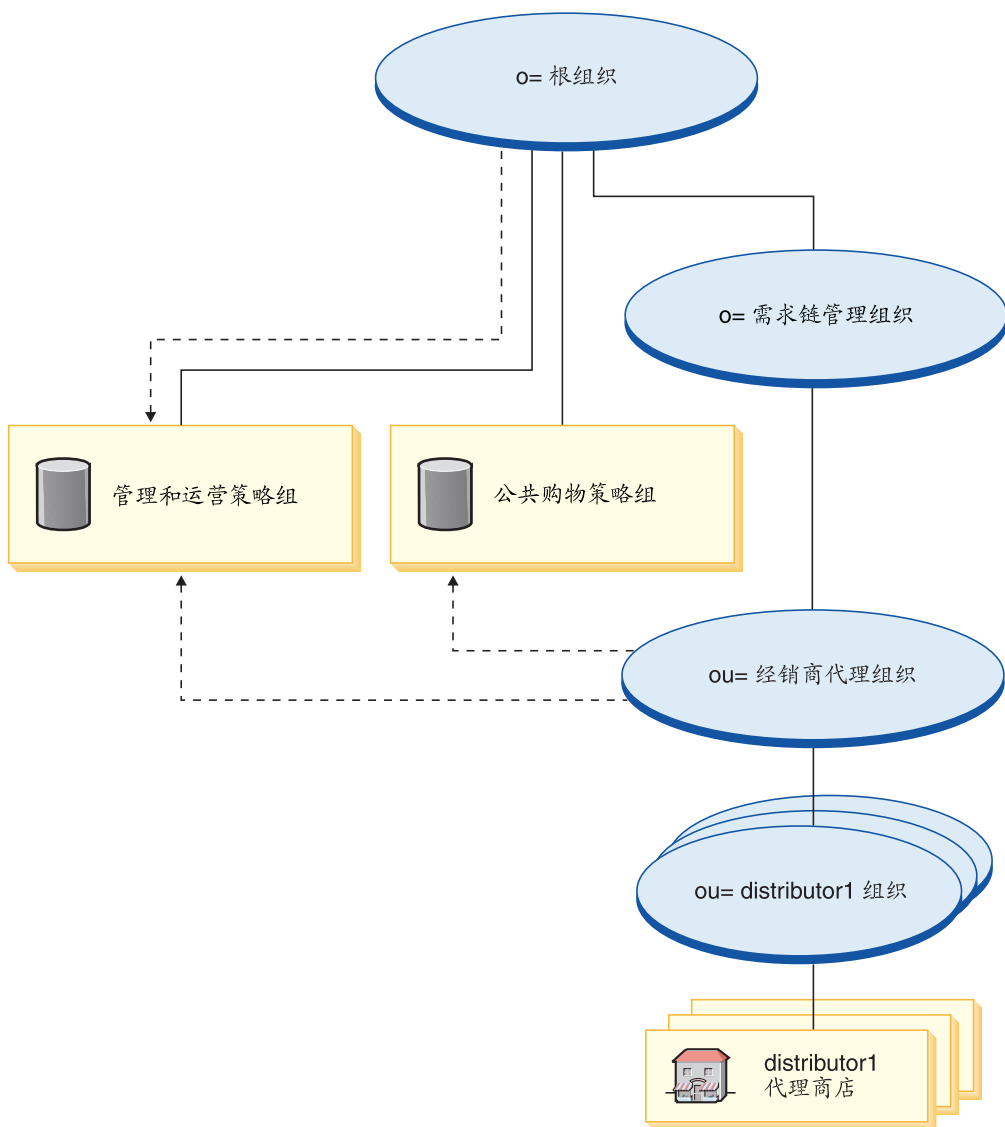
图注



渠道中心组织直接预订管理和运营策略组、公共购物策略组、B2B 策略组，并且拥有及预订市场策略组。因此，这些策略适用于那些直接位于渠道中心组织下的渠道管理员，以及渠道中心（Commerce Plaza）。

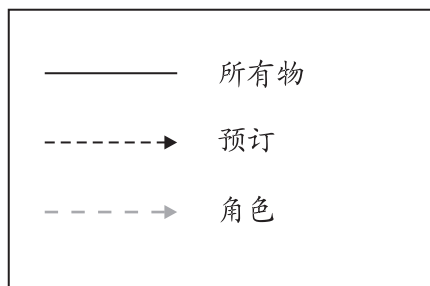
市场策略组包含以下策略:

- AllUsersExecuteMarketplaceAllUserViews
- RegisteredCustomersForOrgExecuteMarketplaceRegisteredCustomerViews
- ContractAdministratorsForChannelOrgExecuteCreateCommandsOnMemberResource
- ContractAdministratorsForChannelOrgExecuteContractDeployCommandsOnContractResource
- ContractAdministratorsForChannelOrgDisplayContractDatabaseResourceGroup

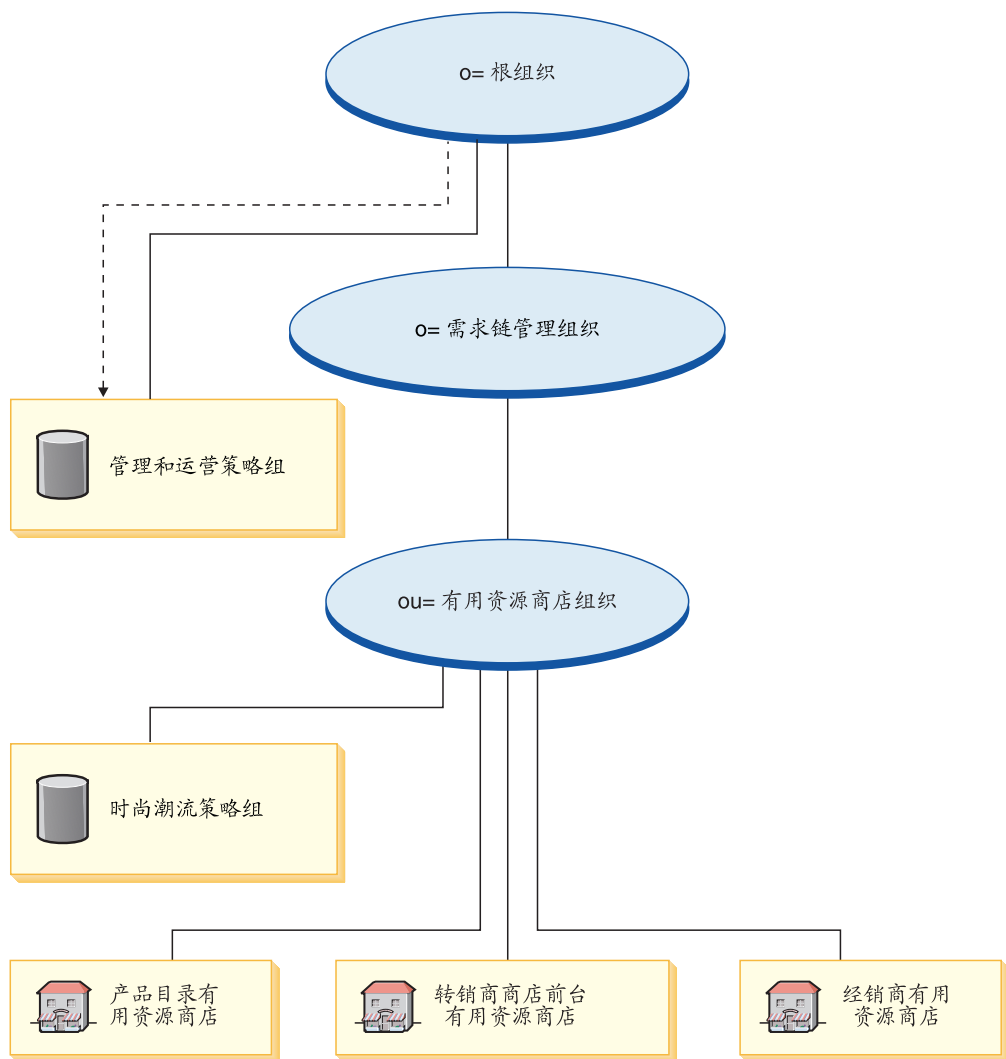




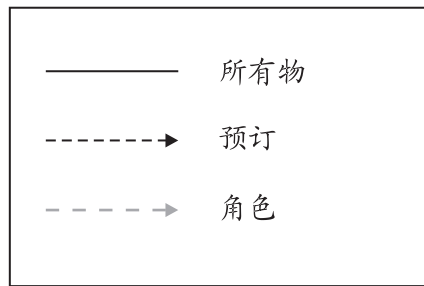
### 图注



经销商代理组织预订管理和运营策略组以及公共购物策略组。因此，这些策略适用于那些直接位于经销商代理组织下的经销商组织。

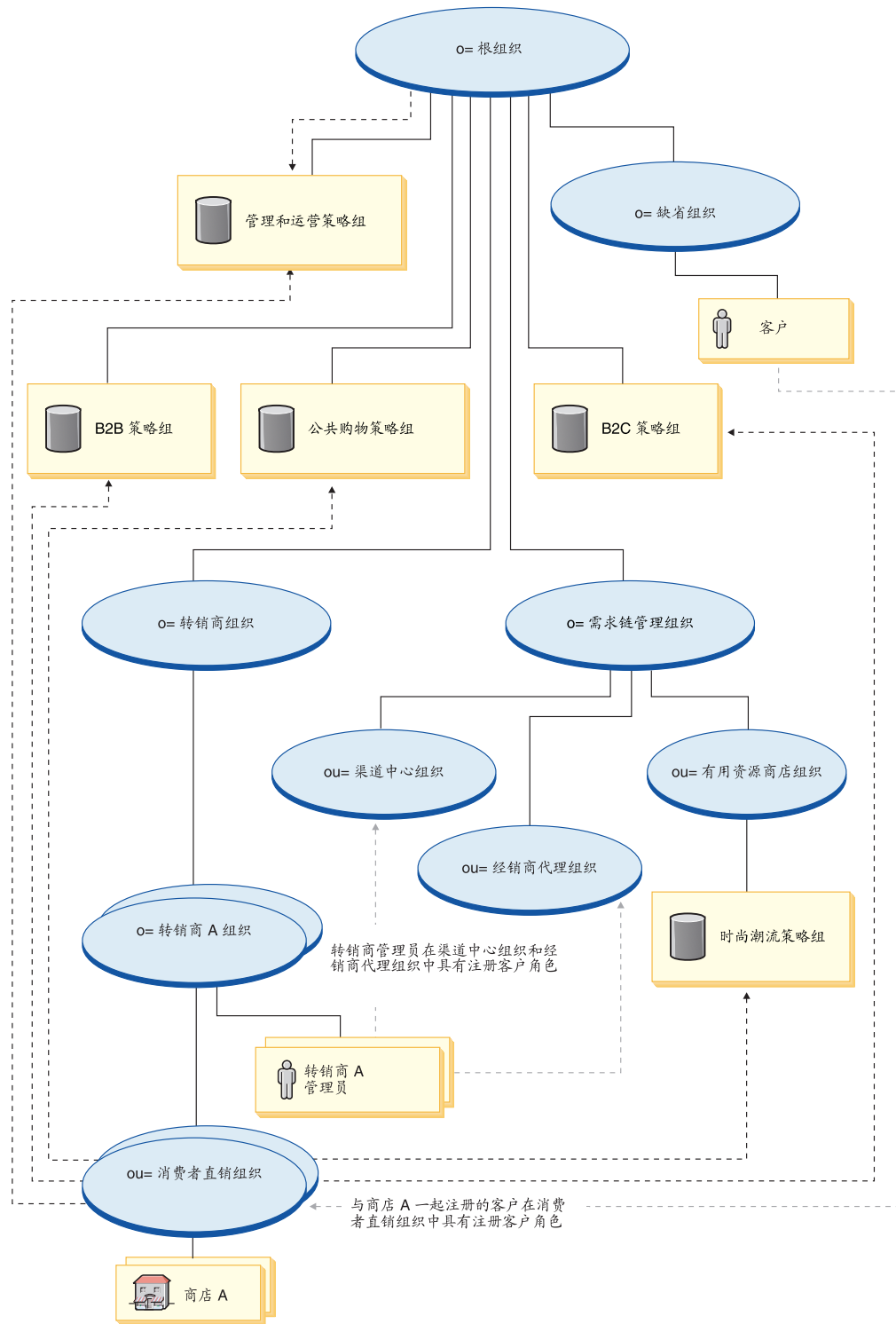


### 图注

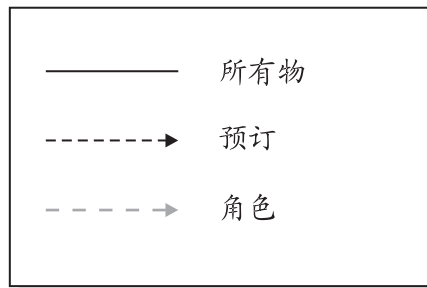


有用资源商店组织不直接预订任何策略组。因此，它从根组织继承管理和运营策略组。这些策略适用于有用资源商店组织及其拥有的有用资源商店。有用资源商店组织拥有“时尚潮流”策略组，但未预订它。

注：当创建转销商商店时，各转销商消费者直销组织将预订“时尚潮流”策略组。



## 图注



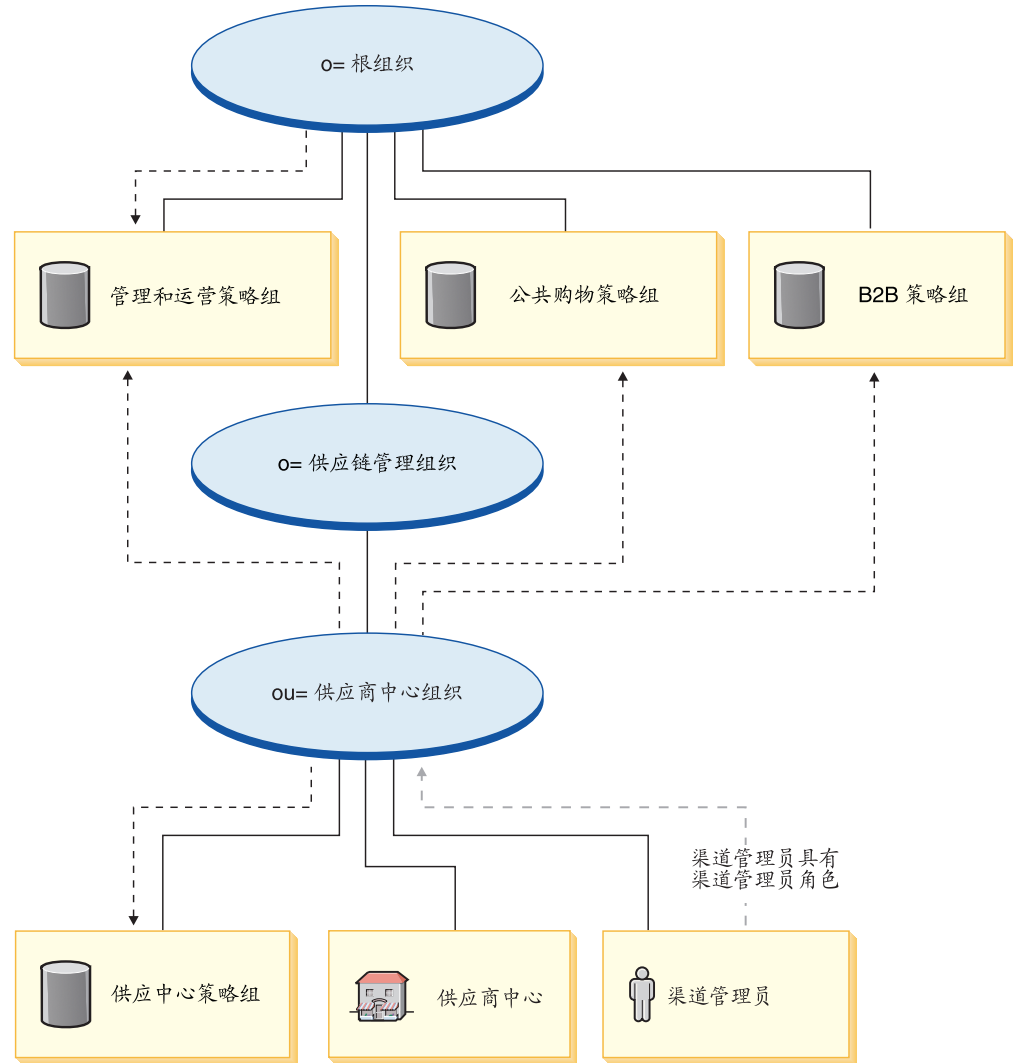
转销商组织不直接预订任何策略组。因此，它从根组织继承管理和运营策略组。这些策略适用于转销商组织及其拥有的转销商 A 组织以及转销商 A 管理员。

消费者直销组织直接预订管理和运营策略组、公共购物策略组、B2C 和 B2B 策略组以及“时尚潮流”策略组。这些策略适用于消费者直销组织拥有的所有商店。

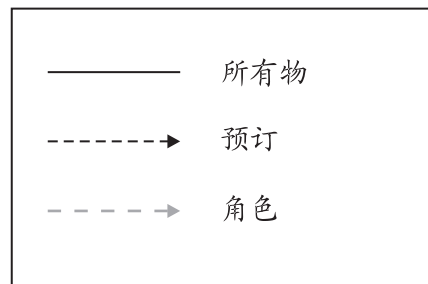
## 供应链

Business

在这些描述基本供应链组织结构的图中，根组织拥有并预订第 34 页的『基本访问控制结构』中描述的缺省策略组。



图注

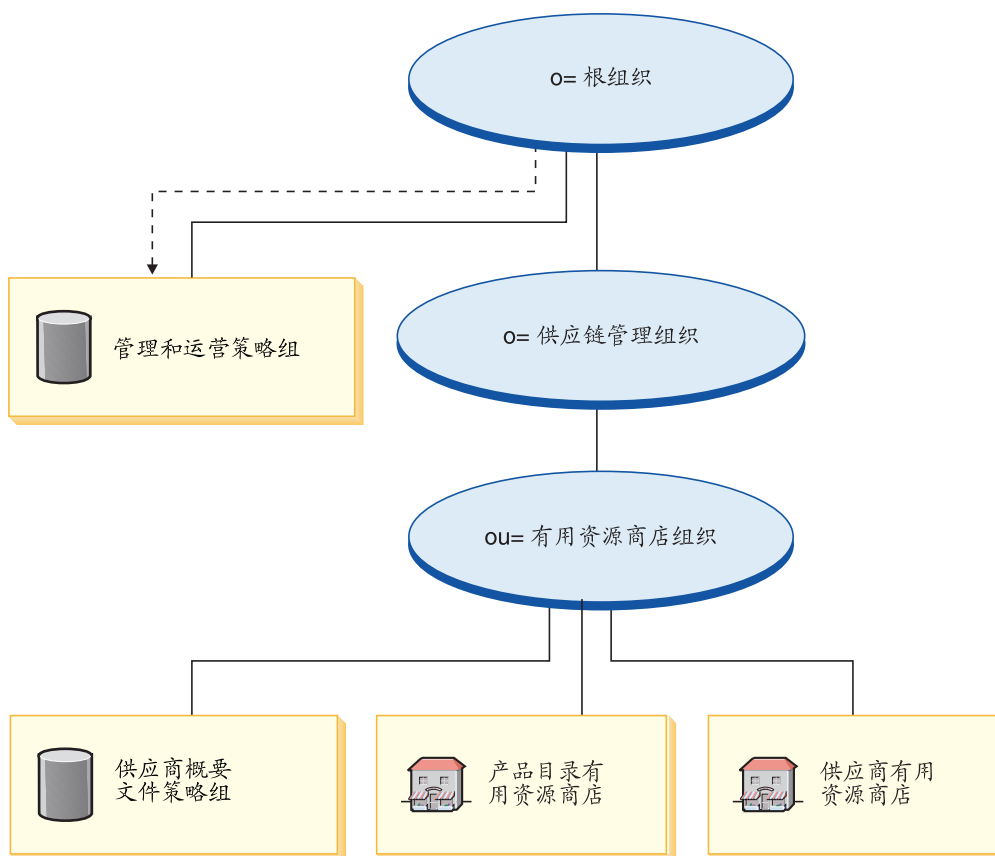


供应商中心组织直接预订管理和运营策略组、公共购物策略组、B2B 策略组，并且拥有及预订供应中心策略组。结果，这些策略应用于那些直接位于供应商中心组织下的渠道管理员，以及供应商中心。

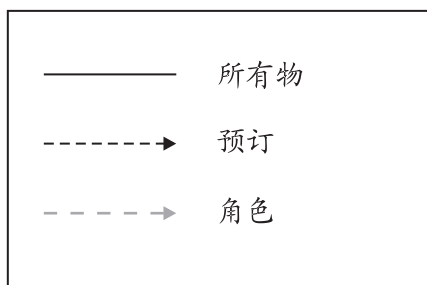
供应商中心策略组包含以下策略：

- AllUsersForSupplierHubExecuteSupplierHubAllUsersViews

- RegisteredCustomersForOrgForSupplierHubExecuteSupplierHubRegisteredCustomerViews
- ContractAdministratorsForChannelOrgExecuteCreateCommandsOnMemberResource
- ContractAdministratorsForChannelOrgExecuteContractDeployCommandsOnContractResource
- ContractAdministratorsForChannelOrgDisplayContractDatabaseanResourceGroup

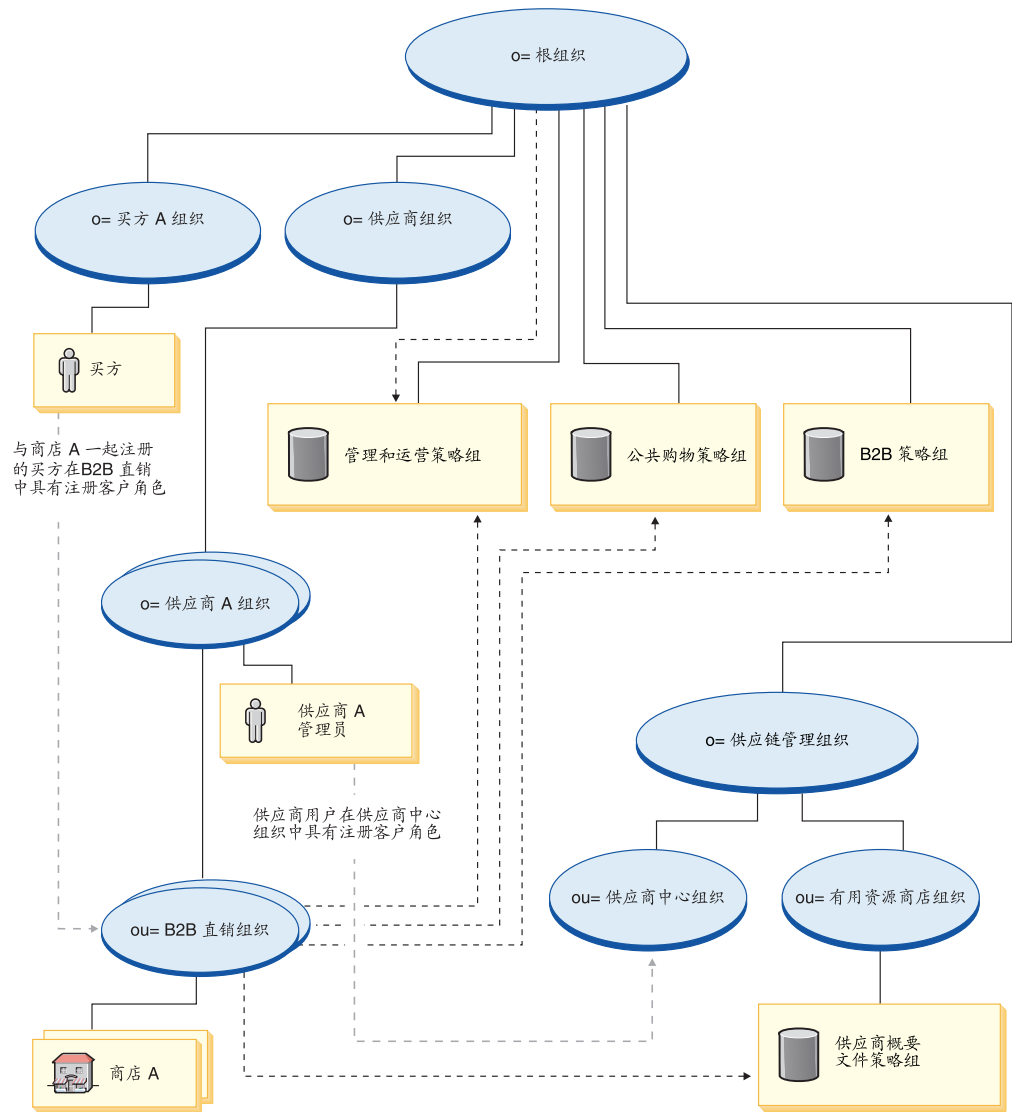


### 图注

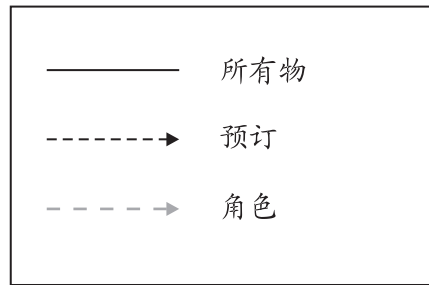


有用资源商店组织不直接预订任何策略组。因此，它从根组织继承管理和运营策略组。这些策略适用于有用资源商店组织及其拥有的有用资源商店。有用资源商店组织拥有供应商概要文件策略组，但未预订它。

注：当创建供应商商店时，单个供应商的 B2B 直销组织将预订供应商概要文件策略组。



图注



供应商组织不直接预订任何策略组。因此，它从根组织继承管理和运营策略组。这些策略适用于供应商组织及其拥有的供应商 A 组织以及供应商 A 管理员。

B2B 直销组织直接预订管理和运营、公共购物、B2B 和供应商概要文件策略组。这些策略适用于 B2B 直销组织拥有的所有商店。

供应商概要文件策略组包含以下策略:

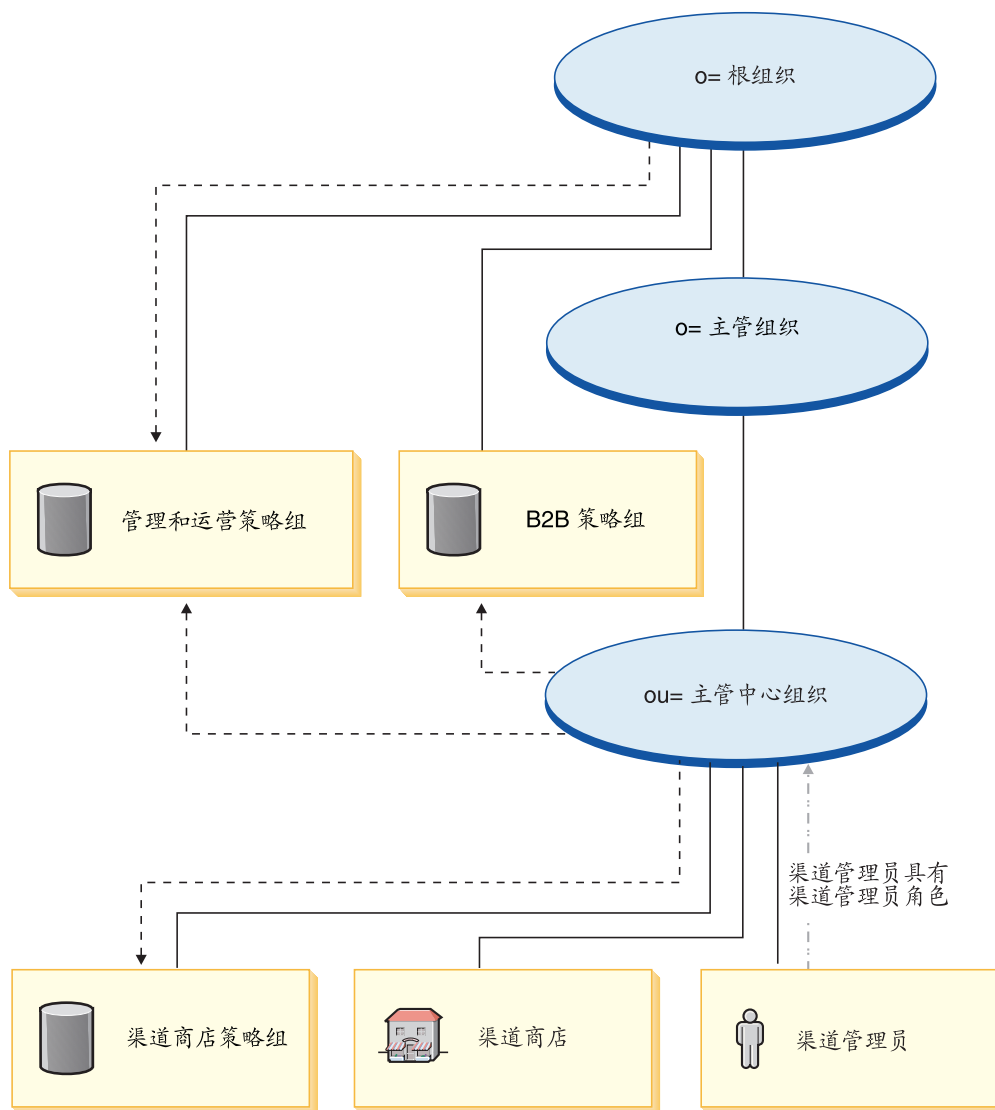
- AllUsersForSupplierExecuteSupplierAllUsersViews
- RegisteredCustomersForOrgForSupplierExecuteSupplierRegisteredCustomerViews

买方就是在 B2B 商店中下订单的客户。所有买方必须属于一个买方组织。通常, 买方组织不预订任何策略组, 因为从根组织继承的管理和运营策略已足够。

## 主管

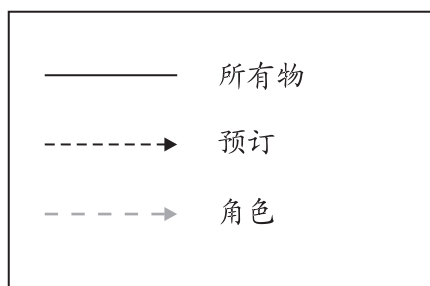
### Business

在这些描述基本主管组织结构的图中, 根组织拥有并预订第 34 页的『基本访问控制结构』中描述的缺省策略组。





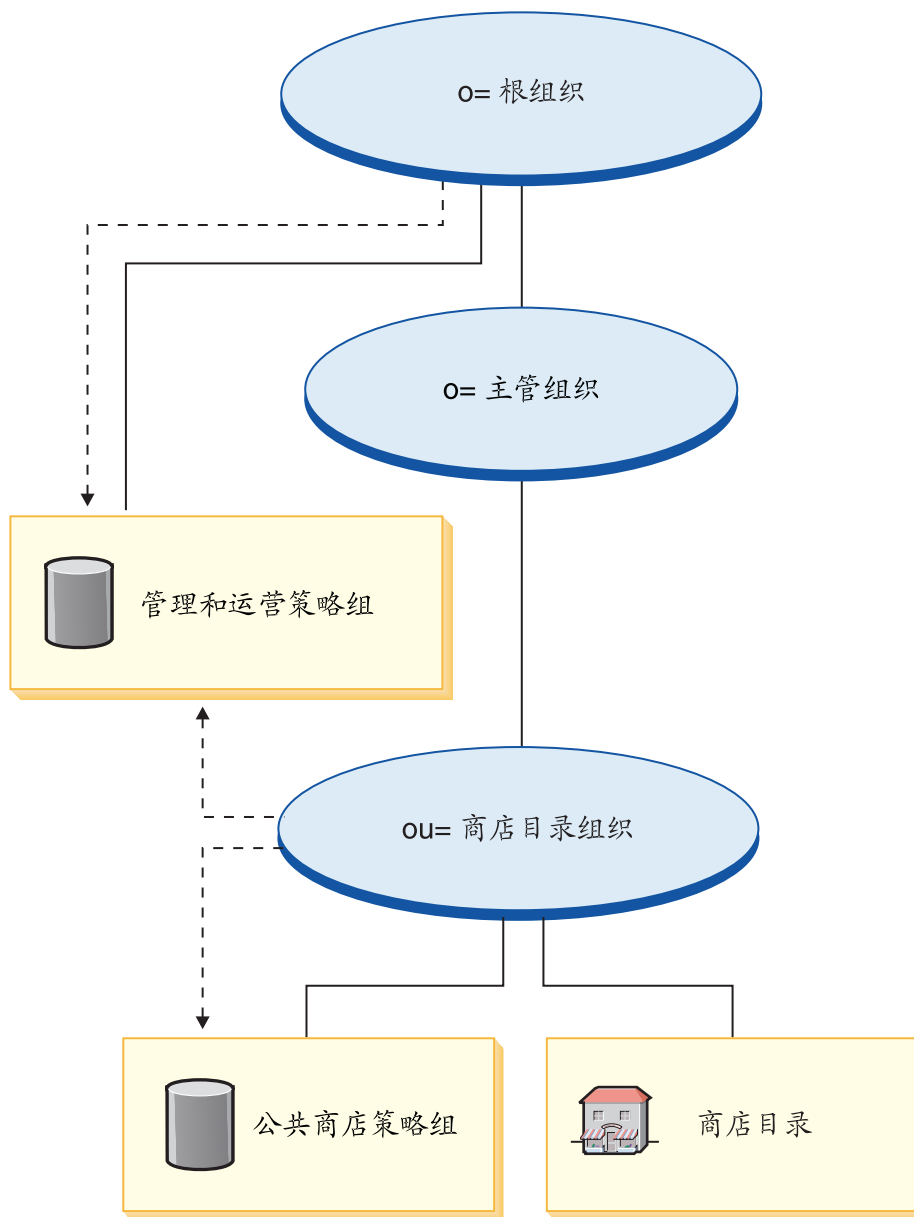
## 图注



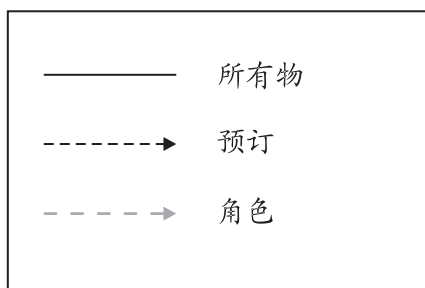
主管中心直接预订管理和运营策略组、B2B 策略组，并且拥有及预订渠道商店策略组。因此，这些策略适用于那些直接位于主管中心组织下的渠道管理员，以及渠道商店（主管中心）。

主管中心策略组包含以下策略：

- AllUsersExecuteChannelStoreAllUsersViews
- ContractAdministratorsForChannelOrgExecuteCreate  
CommandsOnMemberResource
- ContractAdministratorsForChannelOrgExecuteContract  
DeployCommandsOnContractResource
- ContractAdministratorsForChannelOrgDisplayContract  
DatabaseanResourceGroup



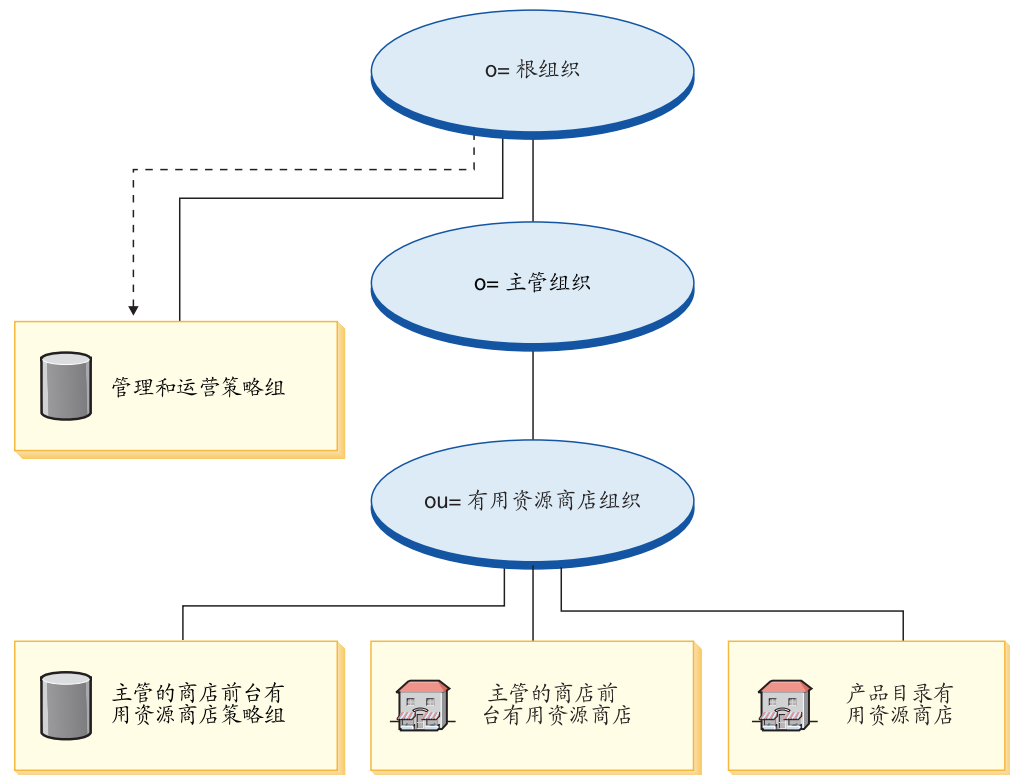
图注



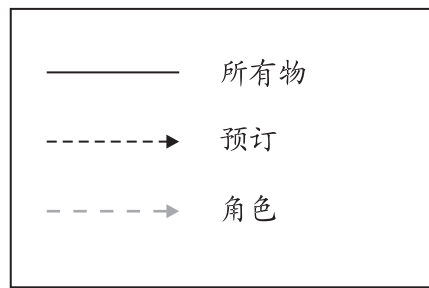
商店目录组织直接预订管理和运营策略组，并且拥有及预订商店目录策略组。结果，这些策略应用于那些直接位于商店目录组织下的商店目录。

商店目录策略组包含以下策略：

- AllUsersExecutePublicStoreAllUsersViews

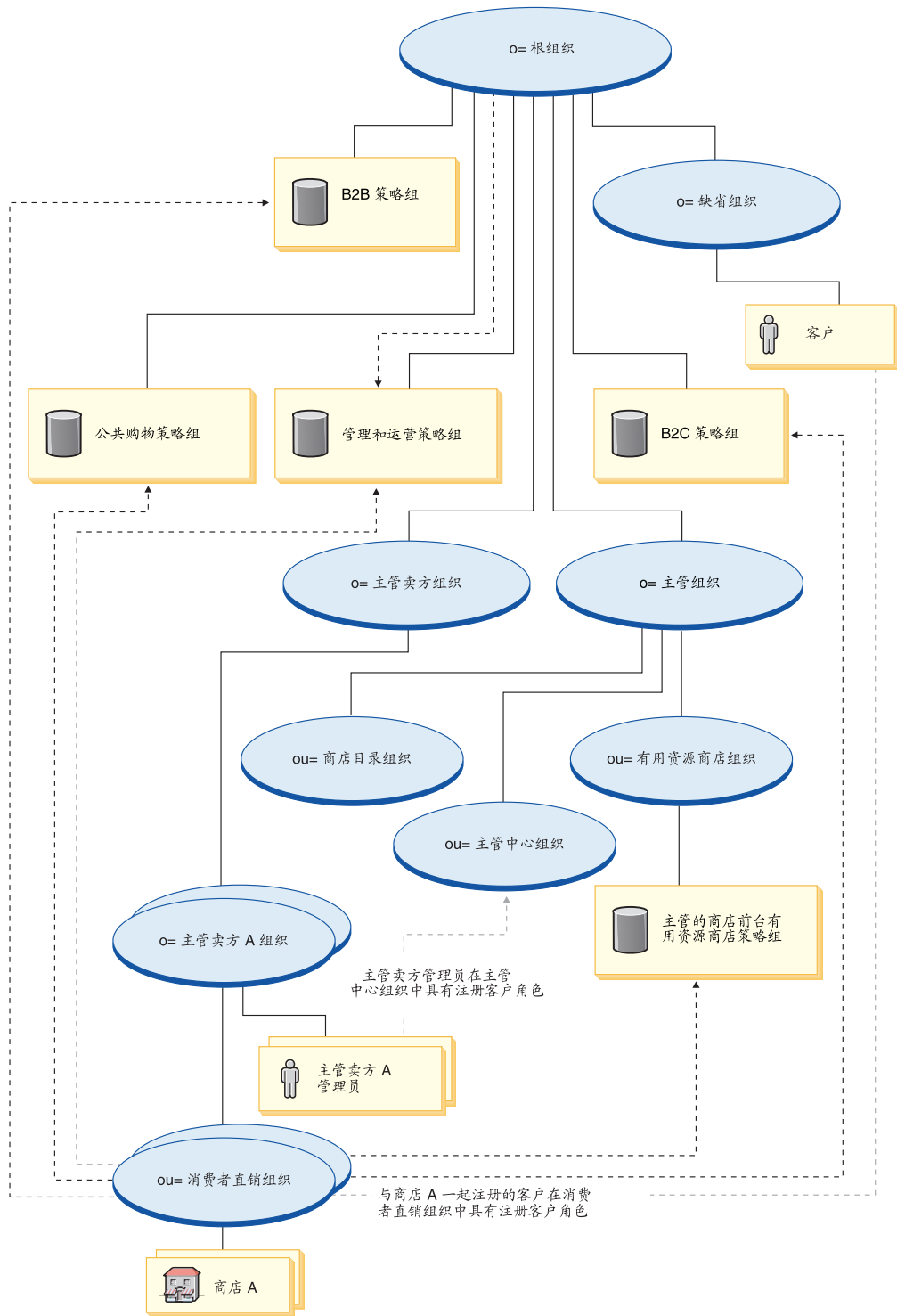


图注

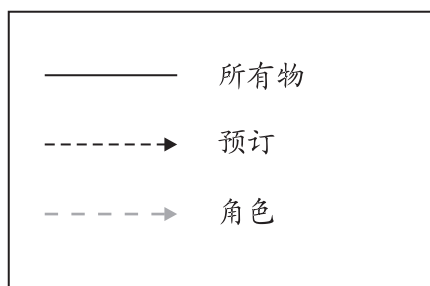


有用资源商店组织不直接预订任何策略组。因此，它从根组织继承管理和运营策略组。这些策略适用于有用资源商店组织及其拥有的有用资源商店。有用资源商店组织拥有主管商店前台有用资源商店策略组，但未预订它。

注：当创建主管商店时，各主管卖方组织将预订主管商店前台有用资源商店策略组。



## 图注



主管卖方组织不直接预订任何策略组。因此，它从根组织继承管理和运营策略组。这些策略适用于主管卖方组织及其拥有的主管卖方 A 组织以及主管卖方 A 管理员。

消费者直销组织直接预订管理和运营、公共购物、B2C 和 B2B 策略组以及主管商店前台有用资源商店策略组。这些策略适用于消费者直销组织拥有的所有商店。

---

## 样本公司中的访问控制

WebSphere Commerce 中的每个样本公司都含有访问控制框架。关于在这些公司中如何实现访问控制框架的更多详细信息，请参阅《*WebSphere Commerce 样本商店指南*》

---

## 向商店添加访问控制

关于向商店添加访问控制的更多信息，请参阅第 277 页的第 33 章，『商店中的访问控制』。



---

## 第 5 章 WebSphere Commerce 业务策略框架

**Business** 业务策略是商店或商店组所遵循的规则集合，它们定义业务流程、行业惯例、商店或商店组出售物的范围和特征，以及商店或站点与客户及其它业务合作伙伴的交互方式。例如，您的站点可能具有决定允许客户在何时、以何种方式将产品退回到商店的业务策略或决定商店接受何种支付方法的业务策略。

---

### 了解 WebSphere Commerce 业务策略框架

WebSphere Commerce 提供了一个框架，允许您在网上商店或站点实现您的商店的业务策略。业务策略框架由以下几部分构成：

- 业务策略
- **Business** 商业帐户
- 合同和 **Business** 服务协议
- 条款和条件

#### 业务策略

在多数实例中，您得为要在网上商店或站点中实现的业务预定义业务策略。WebSphere Commerce 提供了一套业务策略可以直接使用或按您的需要加以更改后使用。关于随 WebSphere Commerce 提供的缺省业务策略的更多信息，请参阅 WebSphere Commerce 产品和开发联机帮助。关于如何编辑这些业务策略的信息，请参阅 WebSphere Commerce 产品和开发联机帮助。

#### 商业帐户

**Business**

商业帐户定义客户和您公司之间的关系。商业帐户跟踪客户组织的合同和订单并配置客户组织的买方如何在商店中购物。

#### 合同和服务协议

在客户或业务合作伙伴（例如转销商或经销商）可以访问您的商店之前，必须首先创建一个合同或服务协议来定义客户或业务合作伙伴对商店的访问权限。在 WebSphere Commerce 业务策略框架中，创建客户的合同和其它类型业务合作伙伴的服务协议。

- 合同：与客户的合同定义了您的商店中客户可以访问的区域、可以看到的价格以及客户对站点和那些价格有访问权的时间。所有商店必须至少包含一个合同，如果没有合同，则除了内部管理员之外，无人能够访问您的商店。WebSphere Commerce 提供了一个缺省合同，它应用于在商店购物的所有客户。在 WebSphere Commerce Professional Edition 中，缺省合同是唯一受支持的合同。
- **Business** 服务协议：与业务合作伙伴（业务合作伙伴可能是转销商、经销商、生产商、供应商或其他合作伙伴）达成的服务协议定义了您与业务合作伙伴的安排。例如与转销商的服务协议可能定义了转销商对您的站点拥有何种控制权，是否可以共

享您的产品目录，以及您是否为他们主管商店。与经销商的服务协议可能定义了您站点的客户如何才能从经销商那里接收报价，以及如何才能从您的站点访问经销商。

## 条款和条件

条款和条件定义了合同和服务协议对于特殊的客户或业务合作伙伴是如何实现的。对于合同，条款和条件可能定义了正依据合同销售的商品；正在销售的商品的价格；如何把商品发送给客户以及客户如何支付订单。对于与业务合作伙伴的服务协议，条款和条件可能限制了允许业务合作伙伴出售的产品。

因为站点或商店运作的多数方面都是由业务策略所定义，所以条款和条件通常参考业务策略。条款和条件为它们所参考的业务策略提供了标准的参数。向业务策略提供参数使您可以对每个合同的业务策略的行为进行修改。

---

## 样本公司中的业务策略

WebSphere Commerce 中的每个样本公司中都含有业务策略框架。关于在这些业务中如何实现公司策略框架的更多详细信息，请参阅《*WebSphere Commerce 样本商店指南*》。

---

## 向站点添加业务策略

关于在您的站点实现业务策略框架的更多信息，请参阅第 177 页的第 18 章，『[合同有用资源](#)』。



---

## 第 6 章 实例体系结构

本章简介了 WebSphere Commerce Server 实例体系结构。

---

### WebSphere Commerce Server

WebSphere Commerce Server 是处理电子交易解决方案的与商店及贸易相关的功能的 WebSphere Application Server 应用程序。商店前台有用资源和业务逻辑驻留在 WebSphere Commerce Server 内的 Web 应用程序中。WebSphere Commerce 提供了缺省 Web 应用程序 (Stores.war) 供您使用, 您也可以创建自己的 Web 应用程序。

Web 应用程序可以包含一个商店的有用资源, 或者多个商店的有用资源。当 Web 应用程序包含多个商店前台和业务逻辑时, 每个商店的有用资源按商店目录 (storedir) 分开。

---

### WebSphere Commerce Server 实例

WebSphere Commerce Server 实例是带有关联数据库的已部署的 WebSphere Application Server 应用程序。一个实例可以支持多个商店。实例中的所有商店共享同一个数据库, 并且可以共享某些类型的数据, 例如产品目录、供货或接收。实例中的所有商店还共享同一个 EJB 容器。

在一个实例中可以创建一个商店, 也可以创建多个商店。关于一个实例中多个商店的信息, 请参阅第 62 页的『单个实例中的多家商店』。



---

## 第 7 章 商店体系结构

为了支持网上商店的创建，WebSphere Commerce 提供了商店体系结构。本章中将讨论此体系结构以及可使用它来实现的一些商店示例。

---

### 了解 WebSphere Commerce 商店体系结构

为了支持您的站点中的商店，WebSphere Commerce 提供了商店体系结构，使您能够创建网上商店。商店体系结构由以下组件构成：

- 商店有用资源
- 对单个实例中的多家商店的支持
- 商店之间的关系

#### 商店有用资源

在 WebSphere Commerce 中，网上商店就是网上业务的所有交易开展的地方。所有通过 WebSphere Commerce 创建的网上商店都包括以下类型的有用资源中的至少一项：

- 商店前台：商店的外部部分或向客户显示的部分被称为商店前台。商店前台由 Web 有用资源（如 HTML 页面、JSP 文件、样式表、图像、图形和其它多媒体文件类型）组成。本指南讨论了创建用于构建商店页面的 JSP 文件时涉及到的概念和任务。关于更多信息，请参阅第 73 页的第 4 部分，『开发商店前台』。
- 业务逻辑：商店处理客户请求的部分（包括命令和定制代码）称为业务逻辑。关于创建业务逻辑或定制代码的更多详细信息，请参阅《WebSphere Commerce 编程指南与教程》。
- 商店数据：组成商店的数据有用资源。为了正常运作，商店必须准备好数据以支持所有客户活动。例如，为让客户能够进行购买，商店必须包含待售商品的产品目录、处理订单的过程、实现请求的库存以及送货过程。您的商店还必须具有处理和收取支付款项的方法。创建商店数据中涉及到的概念和任务在第 107 页的第 6 部分，『开发商店数据』中进行讨论。

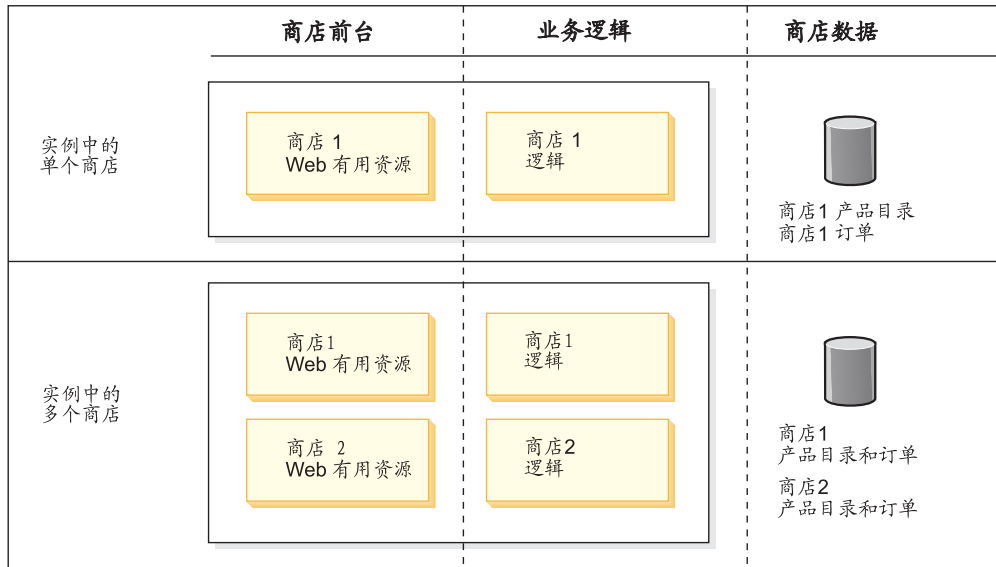
如果一家商店包含所有三种有用资源（即商店前台有用资源、业务逻辑和商店数据），该商店将是一家完全可运作的商店。如果一家商店仅包含这些有用资源的一部分，也就是说它包含商店前台有用资源和业务逻辑，或商店数据和业务逻辑，或仅包含商店数据，则在 WebSphere Commerce 中称为有用资源商店。

#### 有用资源商店

有用资源商店是可以在其它商店中利用的可共享资源集合（业务助诊文件、业务过程和商店前台有用资源）。例如，中心商店可以利用产品目录有用资源商店（可以由中心的渠道或合作伙伴共享），而不是将产品目录作为中心商店的一部分来创建。有用资源商店通常由可以为多家商店使用的有用资源组成。关于更多信息，请参阅第 64 页的『商店之间的关系』。

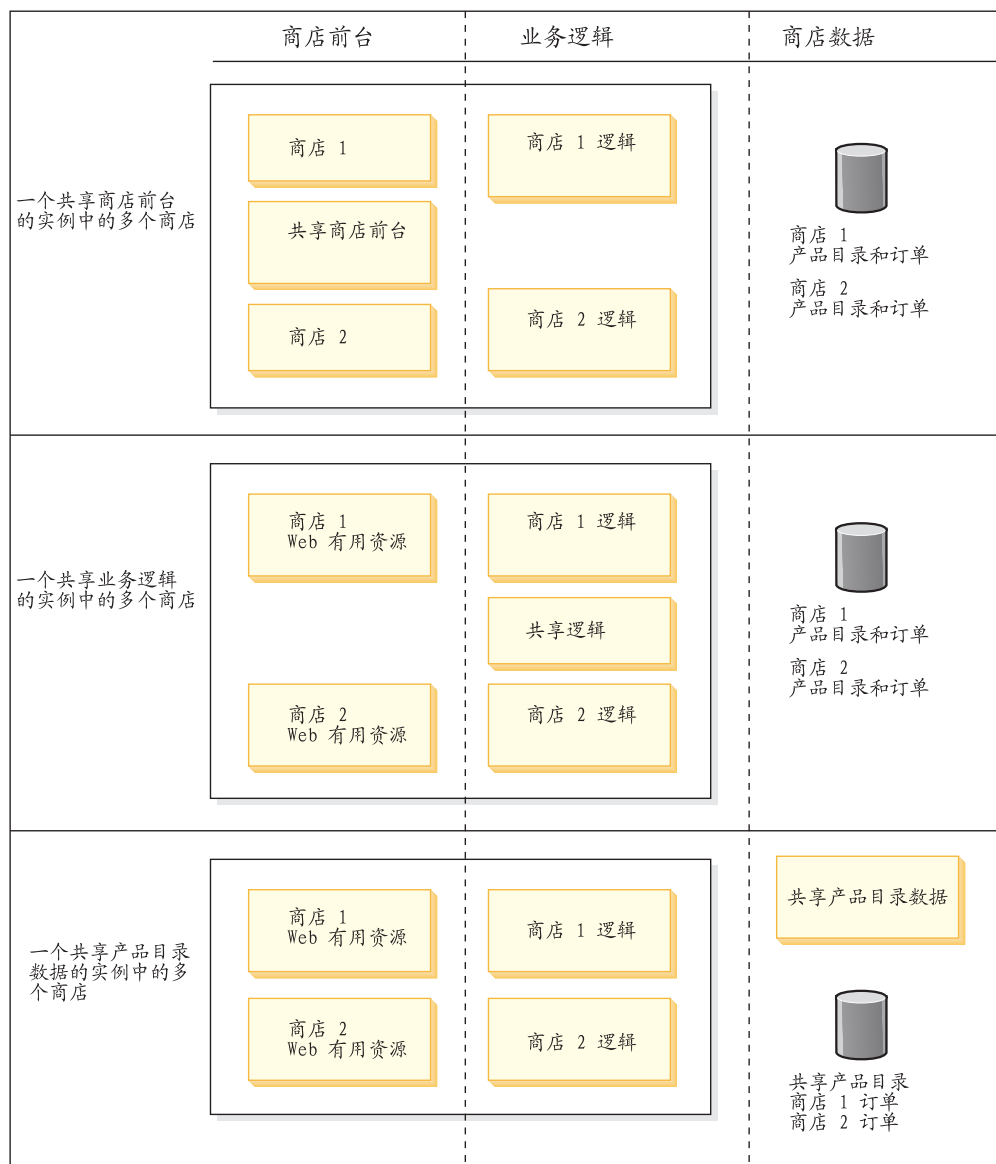
## 单个实例中的多家商店

WebSphere Commerce 使您能够支持 WebSphere Commerce Server 实例中的多家网上商店。下图说明了几种可能的商店配置:



上图中详细描述的是独立商店。也就是说，尽管它们处在同一个实例中，但不共享任何数据，相互之间也没有关系。它们有独立的商店前台、业务逻辑和商店数据。

也可以在一个实例中创建多家商店，它们共享同一个商店前台、同一个业务逻辑或相同的商店数据（包括产品目录），或者这三者的任意组合。下图说明了一些可能的配置，在这些配置中商店共享有用资源:



**注：** 上图仅列出一个实例中多家商店之间几种可能的配置。商店可以共享多个有用资源类型，例如一个站点中的多家商店可共享商店前台、业务逻辑和数据，或者三者的任意组合。

关于一个实例中的多家商店如何共享公共商店有用资源的更多信息，请参阅第 64 页的『商店之间的关系』。

多家商店可以存在于单个商店 Web 模块中。如果是这样，则商店有用资源可使用以下方法分开：

- 商店前台有用资源：商店 Web 模块中的每家商店的商店前台有用资源存储在独立的商店目录（*storedir*）中。例如，MyStore 的所有商店前台有用资源都在 MyStore 目录中。
- 业务逻辑：商店标识用于为每家商店选择命令实现（如在命令注册表中指定的那样）。
- 商店数据：每家商家的数据有用资源通过唯一索引标识。

## 商店之间的关系

**Business** 为了支持一个站点中多家具有相同商店前台、业务逻辑或商店数据或者共享有用资源的任意组合的商店，以及支持一个站点中各商店之间其它类型的关系（例如一家商店主管另一家，或者将购物车从一家商店转移到另一家），WebSphere Commerce 现在为商店间的多种关系提供了体系结构。

商店之间的关系允许一家商店向另一家商店提供服务。例如，商店 A 可以主管商店 B，或者商店 C 可以使用来自商店 D 的产品目录数据。

为了实现这些商店关系，支持每种商店关系的代码是必需的。WebSphere Commerce 包含许多商店关系和支持代码。这些商店关系可以粗略地分为以下几类：

- 一家商店向另一家商店提供有用资源的情况下的关系。这些类型的商店关系包括一家商店向另一家商店提供 URL、命令、业务策略、属性文件和货币。
- 一家商店与另一家商店具有“业务关系”的情况下的关系。这些类型的商店关系包括一家商店主管另一家，或者一家商店向另一家参考订单。

**注：**关于随 WebSphere Commerce 提供的缺省商店关系的详细列表，请参阅第 129 页的第 14 章，『商店间的关系』。

---

## 了解商店体系结构如何支持业务模型

为了支持业务模型所需的商店，WebSphere Commerce 使用商店体系结构来创建以下类型的商店：

- 面向客户的商店
- 代理商店
- 有用资源商店

**注：**推荐使用这些特定的商店来实现 WebSphere Commerce 支持的业务模型。您也可以使用商店体系结构创建您自身类型的商店。

## 面向客户的商店

面向客户的商店是客户可以直接访问的商店。这些商店是您站点的主要组件。WebSphere Commerce 支持以下类型的面向客户的商店：

- 直销商店：支持涉及产品、服务或信息的商业交易的商店，这些交易可以直接在公司和消费者之间进行，也可以在两个公司或不同方之间进行。WebSphere Commerce 支持两类直销商店：
  - 消费者直销
  - **Business** B2B 直销
- **Business** 中心商店：一种商店，该商店使自己的客户或合作伙伴可通过使用站点上的其它商店来访问中心所有者的一个或多个合作伙伴或客户提供的产品或服务。
- **Business** 受托管商店：由站点操作员为商店所有者主管的商店。商店所有者可以选择管理商店。

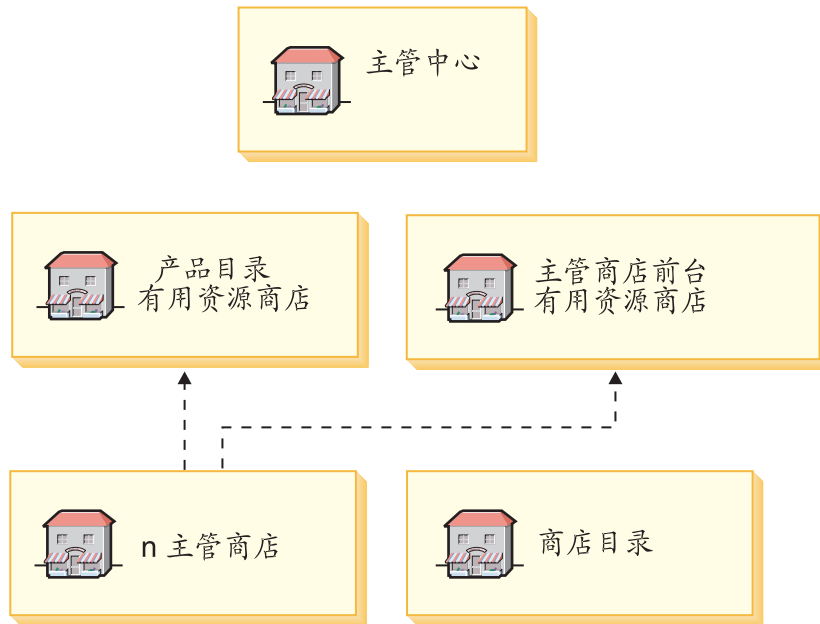
## 创建直销商店和中心商店

直销和 **Business** 中心商店在 WebSphere Commerce 中是商店创建方面最传统的商店。也就是说，您需要为每个商店创建商店前台有用资源、业务逻辑和商店数据。您可以选择按传统方式创建这些有用资源，方法是创建只用于该商店的有用资源。不过，您还可以选择创建由其它商店使用的有用资源，方法是在某个有用资源商店中创建商店前台和业务逻辑有用资源，或者将它们作为可以跨商店使用的数据来创建。您可能还想使用来自其它商店的有用资源创建您的直销或中心商店部分。

关于创建商店前台有用资源的信息，请参阅第 73 页的第 4 部分，『开发商店前台』。关于创建业务逻辑或定制代码的更多信息，请参阅《WebSphere Commerce 编程指南与教程》。关于创建商店数据的更多信息，请参阅第 107 页的第 6 部分，『开发商店数据』。关于商店间共享有用资源的更多信息，请参阅第 129 页的第 14 章，『商店间的关系』。

## 创建主管商店

**Business** 在随 WebSphere Commerce 提供的样本中，受托管商店大部分是通过共享现有有用资源商店的有用资源来创建的。例如，您使用另一家商店的商店前台和（取决于您的企业）产品目录，而不是为您在主管的每家商店均创建商店前台或产品目录有用资源。为了便于创建主管商店，WebSphere Commerce 使用了有用资源商店。下图举例说明主管商店如何使用主管商店前台有用资源商店和产品目录有用资源商店的有用资源。



然后，您的主管公司管理员可以选择进行装饰性的更改（例如新的外观、他们自己的新徽标和他们自己的一些文本）来定制他们的商店，以及更改某些数据（过滤产品目录、更改价格等等）。

您也可以按传统方式创建主管商店，方法是每家主管商店分别创建商店前台有用资源、业务逻辑和商店数据。关于创建商店前台有用资源的信息，请参阅第 73 页的第 4 部分，『开发商店前台』。关于创建业务逻辑或定制代码的更多信息，请参阅《*WebSphere Commerce 编程指南与教程*》。关于创建商店数据的更多信息，请参阅第 107 页的第 6 部分，『开发商店数据』。

**“商店创建”向导：** 随 WebSphere Commerce 提供的“商店创建”向导使您能够方便快捷地创建主管商店。该向导要求客户提供一些关于他们的商店的基本数据（名称、描述等等），允许客户选择他们想要使用的商店前台或产品目录，然后为他们创建商店。生成的商店具有一些独特数据（使其成为独特商店的基本商店数据），但使用的商店前台和产品目录数据却是来自现有的有用资源商店。

“商店创建”向导的行为受控于模板，该模板确定哪些选项可用于创建主管商店，其中包括商店关系、送货方式、消息和共享供货中心。WebSphere Commerce 提供几个模板用于“商店创建”向导，每个模板都对应一个受支持的业务模型。这些模板位于以下目录：

`WC_installdir/xml/trading/xml`

基于在向导中选择的商店前台有用资源商店类型，模板与“商店创建”向导相关联。例如，如果您选择使用转销商商店前台有用资源商店（在 STORE 表的 STORETYPE 字段中标识为 RPS）的有用资源，则“商店创建”向导使用 `TemplateHostingContractRPS.xml`。

关于使用“商店创建”向导创建主管商店的信息，请参阅 WebSphere Commerce 产品联机帮助。



注：如果不愿意使用“商店创建”向导创建主管商店，您可以基于其中一个模板创建服务协议，然后将其导入到 WebSphere Commerce。有关信息请参阅 WebSphere Commerce 产品联机帮助。

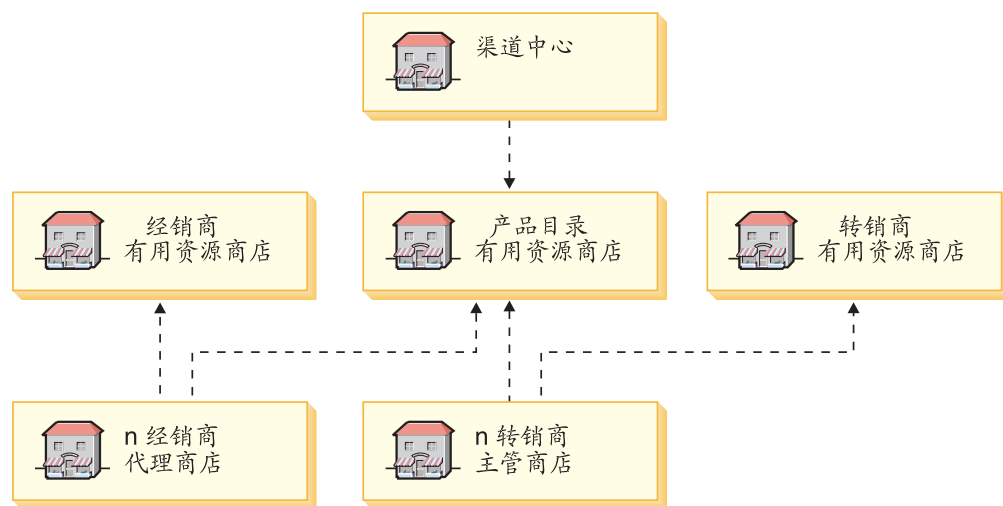
为了更改主管商店共享的有用资源，必须更改有用资源商店。关于更多信息，请参阅第 68 页的『创建有用资源商店』。

## 代理商店

WebSphere Commerce 也支持被称为代理商店的实体。代理商店是代表业务合作伙伴的可操作有用资源的商店，它提供的业务逻辑使 WebSphere Commerce 站点能够与外部业务合作伙伴相互作用。例如，代理商店可以捕获传送到远程订单捕获系统的订单，以及捕获供应商的库存信息或发送到供应商供货中心的信息。与面向客户的商店不同的是，代理商店不包括商店前台也无法由用户访问。

### 创建代理商店

创建代理商店与创建受托管商店非常类似，因为代理商店有用资源大部分由现有商店（包括有用资源商店）提供。正如随 WebSphere Commerce 提供的样本中所实现的那样，代理商店不包括商店前台。因此，仅共享另一家商店的产品目录的有用资源。下图举例说明经销商代理商店使用经销商有用资源商店和产品目录有用资源商店的有用资源。



WebSphere Commerce 通过随后将要导入到 WebSphere Commerce 中的服务协议创建代理商店，从而实现代理商店，而不是提供用户界面来创建代理商店。服务协议受控于模板，该模板确定您需要创建哪些信息。用于创建代理商店的模板（TemplateReferralContract.xml）在以下目录中提供：

`WC_installdir/xml/trading/xml`

要创建代理商店，请按照模板创建新的服务协议，然后将其导入到 WebSphere Commerce 中。关于更多信息，请参阅 WebSphere Commerce 产品联机帮助。

## 有用资源商店

为了便于创建面向客户的商店和代理商店，WebSphere Commerce 实现了有用资源商店。有用资源商店是可以在其它商店中利用的可共享资源集合（业务助诊文件、业务过程和商店前台有用资源）。例如，中心商店可以利用产品目录有用资源商店（可以由中心

的渠道或合作伙伴共享），而不是将产品目录作为中心商店的一部分来创建。有用资源商店通常由可以为多家商店使用的有用资源组成。关于更多信息，请参阅第 64 页的『商店之间的关系』。

WebSphere Commerce 提供了样本产品目录有用资源商店和商店前台有用资源商店。

## 创建有用资源商店

有用资源商店是向其它商店提供有用资源的商店。正如为随 WebSphere Commerce 提供的样本所实现的那样，有用资源商店由有用资源集合组成，但不是完全可运作商店。要创建有用资源商店，您要遵循与在直销或中心商店创建有用资源相同的方法。也就是说，如果希望有用资源商店包含产品目录有用资源，您要按照第 107 页的第 6 部分，『开发商店数据』中的指示信息来创建产品目录数据。如果有用资源商店将包含商店前台有用资源，请参阅第 73 页的第 4 部分，『开发商店前台』。如果有用资源商店将包含业务逻辑，请参阅《WebSphere Commerce 编程指南与教程》。

## 受支持业务模型中的商店

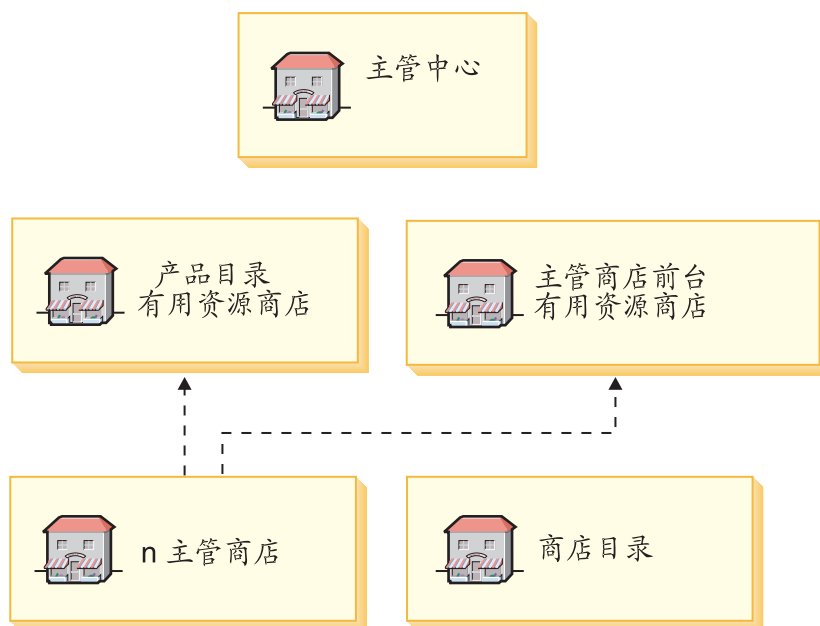
下面几节举例说明如何在样本企业中实现商店。

注：由于消费者直销样本和  B2B 直销样本均包含一个直销商店，故此处不讨论它们。

### 主管

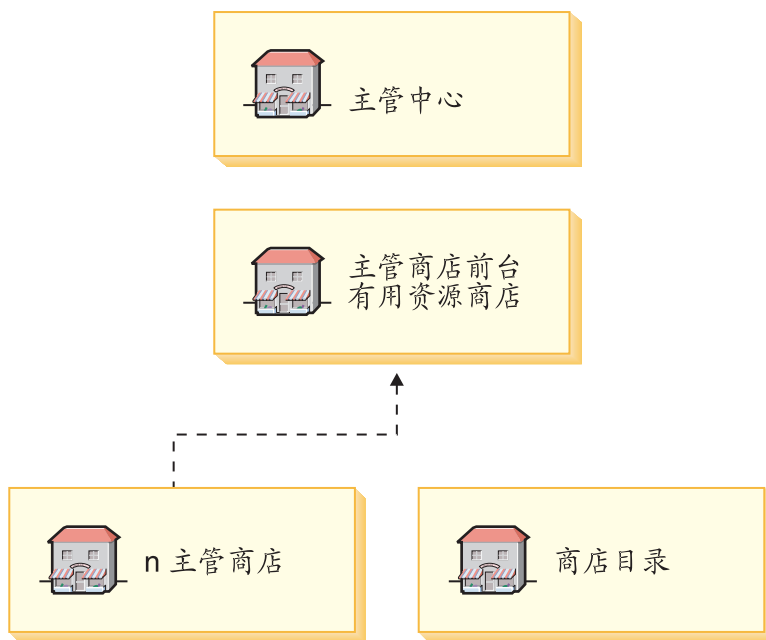
 Business

下图举例说明组成主管样本的商店的类型。



样本主管站点包含一家中心商店（主管中心）、两家有用资源商店（产品目录有用资源和主管商店前台有用资源商店）以及商店目录。商店目录是站点中所有主管商店的列表，并充当它们的网关。主管商店是使用这两家有用资源商店的有用资源创建的。

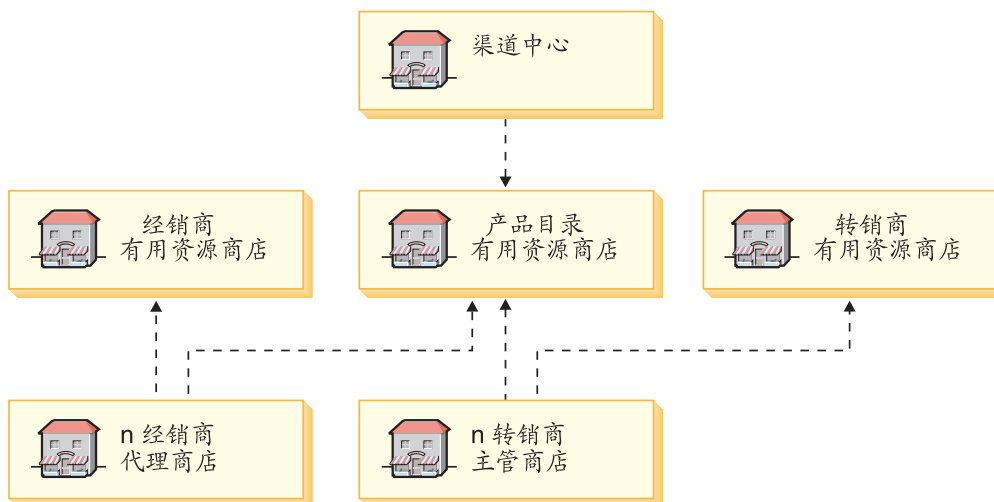
请注意，客户可以选择创建他们自己的产品目录数据，而不是使用在产品目录有用资源商店中定义的产品目录。这种变化造成主管站点的第二次实现，如下图中所说明的那样：



## 需求链

Business

下图举例说明组成需求链样本的商店的类型。

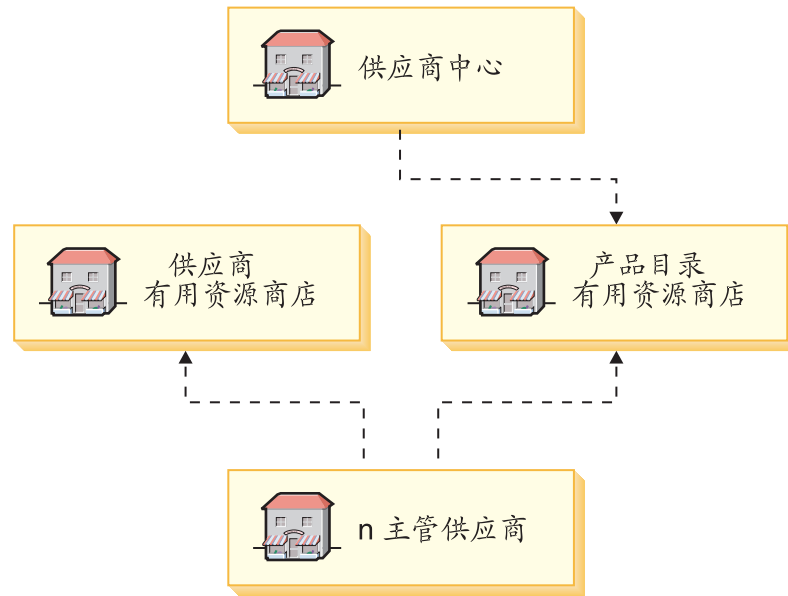


需求链样本站点包含一家中心商店（渠道中心）和三家有用资源商店（经销商有用资源商店、产品目录有用资源商店和转销商商店前台有用资源商店）。请注意，渠道中心使用产品目录有用资源商店中定义的产品目录有用资源。经销商代理商店是使用经销商有用资源商店的有用资源创建的，而转销商主管商店是使用产品目录有用资源商店和转销商商店前台有用资源商店的有用资源创建的。

## 供应链

Business

下图举例说明组成供应链样本的商店的类型。



供应链样本站点包含一家中心商店（供应商中心）和两家有用资源商店（产品目录有用资源商店和供应商有用资源商店）。请注意，供应商中心使用产品目录有用资源商店中定义的有用资源。主管供应商是使用产品目录有用资源商店和供应商有用资源商店的有用资源创建的。

**注：** 供应商中心所有者定义主管供应商将在产品目录有用资源商店中使用的产品目录分类（例如，产品目录结构、以及可能共享的产品和商品）。



---

## 第 4 部分 开发商店前台





---

## 第 8 章 开发商店前台

本章提供了 WebSphere Commerce 商店前台体系结构的概述，其中包括商店的外部部分、Web 有用资源（如 HTML 页面、JSP 文件、样式表、图像、图形或其它多媒体文件类型）如何显示给客户的信息。

---

### 商店前台体系结构

WebSphere Commerce 使用一个命令和视图系统将商店前台中的 Web 有用资源显示给客户。

- 命令执行特定的业务过程，如将产品添加到购物车中、处理订单、更新客户的通讯录或显示特定的产品页面。操作完成时，命令返回一个视图。
- 视图显示命令和用户操作的结果，也就是将商店页面（JSP 文件）呈现给客户的视图。为使视图调用 JSP 文件，必须向视图注册表（VIEWREG）中的视图注册 JSP 文件名。相应的 JSP 文件通过使用 JSP 文件名存储在 WebSphere Commerce 商店 Web 应用程序下的商店子目录（stordir）中。

命令和视图都是使用 URL 调用的。例如，当客户在样本商店中单击**购物车**时，客户调用 URL `https://hostname/path/OrderItemDisplay?`，此 URL 将传递到 WebSphere Commerce Server 中。WebSphere Commerce Server 调用 `OrderItemDisplay` 命令，购物车页面显示给客户。

当客户在样本商店中单击**帮助**时，客户调用 URL `https://hostname/path/HelpView?`，此 URL 传递到 WebSphere Commerce Server 中。WebSphere Commerce Server 调用 `HelpView`，返回“帮助”页面。

WebSphere Commerce Server 也可以将多条命令映射到一个 URL，此操作允许每个商店具有各自的命令实现（可选）。

同样，WebSphere Commerce Server 也允许将多个 JSP 文件映射到单个视图，这样每个商店都可以为不同的设备类型注册不同的 JSP 文件名（可选）

### 缺省命令和视图

WebSphere Commerce 提供了可以在商店中使用的缺省命令和视图。这些缺省命令和视图列在 `wcs.bootstrap.xml` 文件中。引导程序文件位于以下目录中：

- `WC_installdir/schema/xml`

如果未提供所需的命令或视图，可以创建自己的命令或视图。关于创建命令和视图的信息，请参阅《*WebSphere Commerce 编程指南与教程*》。

---

### 创建商店页面

创建商店前台中最大的任务是创建实际的商店页面。开始商店页面的开发工作之前，您应当完成以下规划活动：

- 列出所需商店页面的列表
- 制订一系列命令和视图 URL

- 使 JSP 文件名与视图关联
- 制订一系列访问控制策略。关于更多信息，请参阅第 277 页的第 33 章，『商店中的访问控制』。

**注：**规划商店页面时，还应创建高速缓存战略。关于高速缓存的更多信息，请参阅第 83 页的第 9 章，『高速缓存商店页面』。

## 列出商店页面的列表

为了列出创建商店所需页面的列表，您需要了解商店的业务和功能需求以及任何已定义的业务过程。

### 从使用案例开始

很多人以使用案例的形式收集需求。使用案例以客户和建议的系统之间交互的形式定义商店中的业务过程。对网上商店而言，使用案例可以定义客户如何在商店注册、如何浏览产品目录或如何订购商品。

联机帮助中提供了一组详细描述样本商店业务过程的使用案例。这些使用案例可以帮助您更全面地了解样本商店的流程，而且可以用作指南（如果希望创建自己商店的使用案例）。

以下是“注册”使用案例的示例：

**“注册”使用案例：** 注册过程允许客户在数据库中输入个人信息。

**操作者：**

- 客户

**主要流程：** 客户从辅助栏选择**注册**。系统然后显示包含以下字段的页面：

- 电子邮件
- 密码
- 验证密码
- 名字
- 姓氏
- 年龄（可选）
- 性别（可选）

客户在以上字段中输入相应的信息，并选择**提交**。系统在系统中创建一个新客户并保存该客户的信息（E1、E2、E3）。系统提示客户按照“管理个人帐户”使用案例中的过程管理他们的帐户。

**备用流程：** 无。

**异常流程：**

**E1：** 电子邮件地址已经存在：

- 如果电子邮件地址在系统中已经存在，则系统显示一条错误消息，要求用户输入另一个电子邮件地址。使用案例从头开始恢复。

**E2：** 遗漏必填字段：

- 如果以下任何字段（电子邮件、密码、验证密码、名字或姓氏）中有未填写的，则系统发出一条错误消息。使用案例从头开始恢复。

E3: 无效密码:

- 如果密码无效或与验证密码不匹配，则系统发出警告。

**确定商店购物流程:** 不管是开发使用案例来说明商店的业务过程，还是使用另一种方法，一旦业务过程可用，就可以创建商店的购物流程。

**注:** 由于使用案例通常包含流程信息，如“如果客户选择**提交**，则显示‘订单’页面”，因此使用案例可以为创建购物流程图提供有用信息。

购物流程反映为商店定义的需求和业务过程，说明客户将如何在商店中移动。例如，客户可以通过主页进入站点并在浏览产品目录之前被要求进行注册，或者您可以选择允许客户作为临时客户查看产品目录而不必注册。一些购物流程允许客户完成“快速结帐”，而其它流程则需要客户在每次进行购买时完成所有结帐步骤。或者，您的购物流程可以允许客户在两种结帐中进行选择。

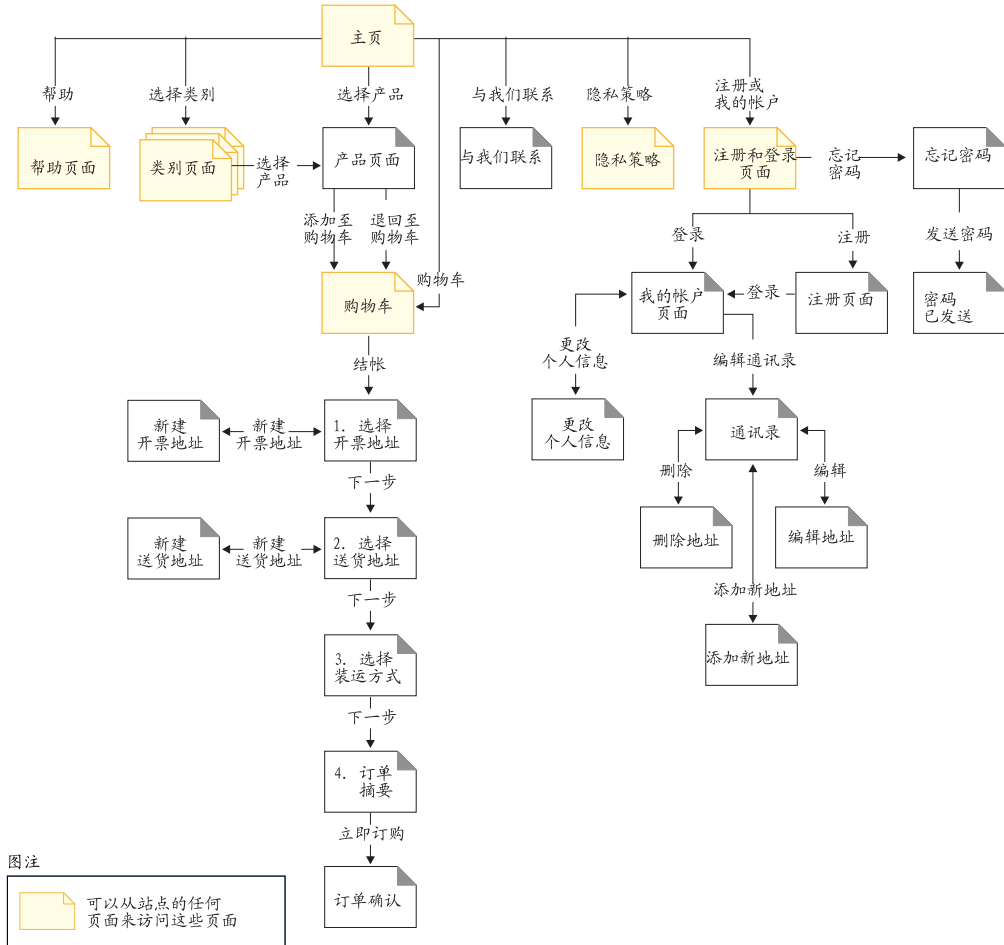


要验证商店流程图是否完整，请确保商店使用案例中的所有步骤在商店流程图中都已说明。

---

绘制购物流程图（就象下面的“时尚潮流”样本商店购物流程图那样），这样就允许您查看客户如何浏览商店。

注：该图仅包含“时尚潮流”商店流程的一部分。关于完整的流程，请参阅《WebSphere Commerce 样本商店指南》。



“时尚潮流”购物流程图十分简单。尽管其中包括客户在商店中移动的主要流程，但不包括任何错误方案。例如，客户使用错误密码登录或输入无效信用卡号时会发生什么情况？但是，即使类似此图的简单图也能够让您列出商店所需的页面的列表。要开始此操作，您需要为购物流程图中列出的每个页面创建一个视图。

例如，如果您要创建的商店具有与“时尚潮流”图中相同的购物流程，就必须创建以下页面：

注：下表列出了“时尚潮流”商店中使用的视图名称

“时尚潮流”购物流程图页面（如客户所见）	相应视图
主页	StoreCatalogDisplayView
帮助页面	HelpView
与我们联系	ContactView
隐私策略	PrivacyView
注册或登录页面	LogonForm
忘记密码	LogoffView
密码已发送	ResetPasswordForm
我的帐户页面	LogonForm

“时尚潮流”购物流程图页面（如客户所见）	相应视图
更改个人信息	UserRegistrationForm
通讯录	AddressBookForm
添加新地址	AddressForm
删除地址	AddressBookForm
编辑地址	AddressForm
注册页面	UserRegistrationForm
购物车	OrderItemDisplayViewShiptoAssoc
选择开票地址	BillingAddressView
新建开票地址	AddressForm
选择送货地址	MultipleShippingAddressView
新建送货地址	AddressForm
选择送货方式	MultipleShippingMethodView
订单摘要	AllocationCheck
订单确认	OrderOKView

**注：**“时尚潮流”中使用的很多视图都是专为“时尚潮流”创建的。这些视图在“时尚潮流”商店归档文件中的 `command.xml` 文件中列出。关于更多信息，请参阅第 137 页的『在 WebSphere Commerce 中注册命令、视图和 URL』。

上表仅指出需要创建的基本页面集。要确定需要创建的其它页面，您可以更仔细地查看用于定义业务过程的使用案例或其它方法。

**错误页面：** 使用案例中的异常流程也可以帮助您确定需要为商店创建的错误页面。“时尚潮流”的注册使用案例指定了以下异常流程：

- 电子邮件地址已经存在：如果电子邮件地址在系统中已经存在，则系统显示一条错误消息，要求用户输入另一个电子邮件地址。使用案例从头开始恢复。
- 遗漏必填字段：如果以下任何字段（电子邮件、密码、验证密码、名字或姓氏）中有未填写的，则系统发出一条错误消息。使用案例从头开始恢复。
- 无效密码：如果密码与验证密码不匹配，则系统发出一条警告。

结果，您将需要为每个异常流程创建一个错误页面或一条错误消息。

## 列出命令和视图 URL 的列表

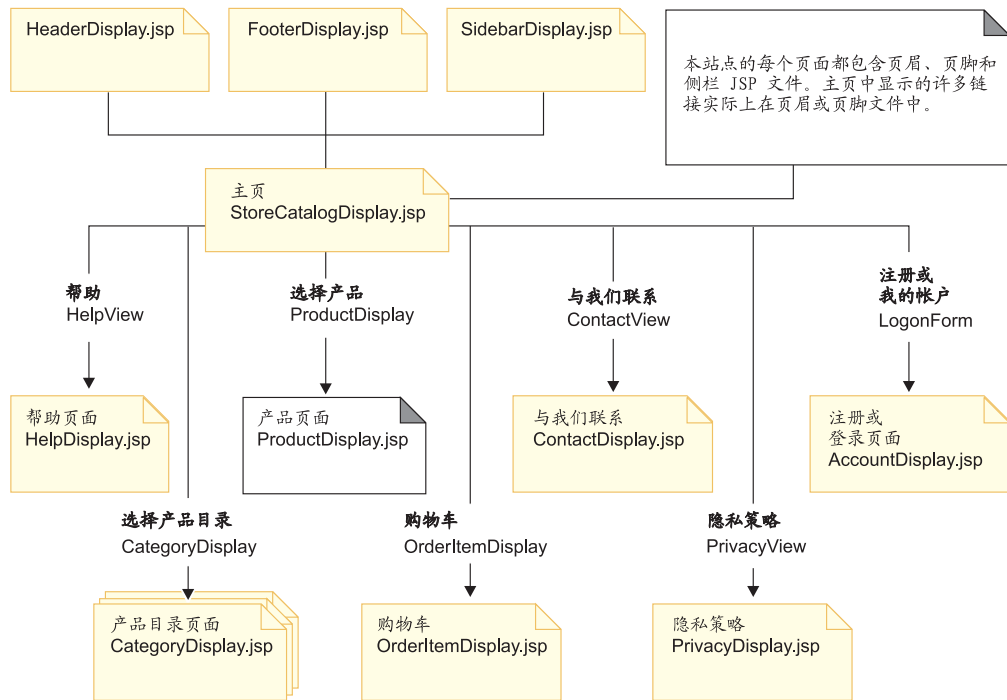
如“时尚潮流”购物流程图所示，业务流程（如结帐和注册）可能需要几个页面。要将这些页面结合到一个正常工作的业务过程或流程中（而不是仅仅一个页面集），您必须在页面中包含命令和视图。

### 列出所需 URL 的列表

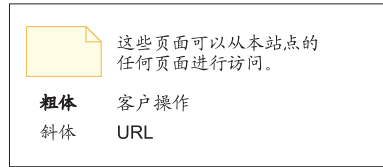
就如同列出创建商店所需页面的列表，您还需要列出实现商店业务过程所需的命令和视图 URL 的列表。使用商店的购物流程图以及缺省命令和视图的列表确定完成每个操作所需的 URL。

了解样本商店中使用的命令和视图 URL 也可以帮助您确定商店中所需的 URL。下图标识了“时尚潮流”购物流程图中某些操作的 URL。关于更多详细信息，请参阅

《WebSphere Commerce 样本商店指南》中关于样本商店的信息。



图注



## 使 JSP 文件名与视图关联

WebSphere Commerce Server 使用视图命令编辑作为请求响应的视图。WebSphere Commerce Server 提供了以下视图命令：

- `HttpForwardViewCommandImpl`：此视图命令将视图请求转发到 JSP 文件。
- `HttpRedirectViewCommandImpl`：此视图命令将视图请求重定向到另一个 URL。
- `HttpDirectViewCommandImpl`：此类型的视图命令将响应视图直接发送到客户机。它不调用 JSP 文件。直接视图允许控制器命令产生输出响应（而非视图命令）。

使用 `HttpForwardViewCommandImpl` 视图命令直接提供 JSP 文件。例如，在说明“时尚潮流”中所用 URL 的图中，为显示帮助页面（`Help.jsp`），`HelpView` 在视图注册表中注册并与 `Help.jsp` 和 `HttpForwardViewCommandImpl` 命令相关联。这在以下示例中演示：

```
<viewreg
viewname="HelpView"
devicefmt_id="-1"
storeent_id="@storeent_id_1"
interfacename="com.ibm.commerce.command.ForwardViewCommand"
classname="com.ibm.commerce.command.HttpForwardViewCommandImpl"
```

```
properties="docname=Help.jsp"
internal="0"
https="0"
/>
```

请注意使用接口和实现类的全限定类名。

使用 `HttpForwardViewCommandImpl` 视图命令提供从显示命令返回的视图。显示命令从数据库读取数据，但不对其进行更改。例如，在说明“时尚潮流”中所用 URL 的图中，`OrderItemDisplay` 命令返回 `OrderItemDisplayViewShiptoAssoc` 视图。在视图注册表中注册此视图时，`OrderItemDisplay.jsp` 和 `HttpForwardViewCommandImpl` 与其相关联。这在以下示例中演示：

```
<viewreg
viewname="OrderItemDisplayViewShiptoAssoc"
devicefmt_id="-1"
  storeent_id="@storeent_id_1"
interfacename="com.ibm.commerce.command.ForwardViewCommand"
classname="com.ibm.commerce.command.HttpForwardViewCommandImpl"
properties="docname=OrderItemDisplay.jsp"
internal="0"
https="0"
/>
```

您必须使每个关联视图的 JSP 文件名与所使用的每条显示命令（例如 `OrderItemDisplay`）关联。关于使 JSP 文件名与视图关联的更多信息，请参阅第 137 页的『在 WebSphere Commerce 中注册命令、视图和 URL』。

**注：** 产品显示和产品类别显示命令返回视图和 JSP 文件名。这些显示产品和类别的 JSP 文件名存储在产品目录数据中。关于更多信息，请参阅第 161 页的『显示商店产品目录有用资源』。可以指定不同的 JSP 文件名以显示商店支持的每个成员组或每种语言的产品和类别（可选）。

`HttpRedirectViewCommandImpl` 视图命令用于生成非显示命令（更改数据库的命令）的输出。非显示命令必须与显示命令相关联以防在客户重新装入该页面或单击“上一步”按钮情况下意外地重新执行该命令。

要重定向显示命令，请在非显示命令的 URL 上使用 `&URL=` 参数来指定该显示命令。例如，在“时尚潮流”样本商店地址表单中添加地址信息并单击**提交**时，它将调用 `AddressAdd` 命令。用于调用 `AddressAdd` 命令的 URL 用 `&URL=` 参数指定 `AddressBookForm` 命令。这会导致重定向到 `AddressBookForm` 显示命令，该命令将返回 `AddressBookForm` 视图。在视图注册表中注册 `AddressBookForm` 视图时，`AddressBookForm.jsp` 和 `HttpForwardViewCommandImpl` 与其关联。

必须对所有非显示命令使用 `URL=parameter` 技术。非显示命令是导致数据库中的数据发生更改的命令。





---

## 第 9 章 高速缓存商店页面

开发商店前台时，您还需要确定如何对商店页面进行高速缓存。本章讨论为您的商店创建和实现高速缓存战略。

---

### 规划高速缓存战略

确定高速缓存战略时，您首先需要考虑以下问题：

- 应对哪些页面进行高速缓存
- 应将页面高速缓存为完整页面还是页面片段

#### 应对哪些页面进行高速缓存

创建高级高速缓存战略时，您首先需要确定应对商店中的哪些页面进行高速缓存。很适合高速缓存的页面是具有以下特点的页面：它们被频繁访问，但在一段时间内还是稳定的，而且所包含的内容可以由各种用户重新使用。例如，产品目录显示页通常是启用高速缓存的良好候选。

#### 应将页面高速缓存为完整页面还是页面片段

在 V5.5 中，WebSphere Commerce 使用 WebSphere Application Server 动态高速缓存服务，该服务允许 WebSphere Commerce 既支持对 Web 页面进行整体的高速缓存，也支持对页面的片段进行高速缓存。对 Web 页面进行整体的高速缓存只是将整个页面作为一个实体进行高速缓存，即使该页面由几个较小的片段组成。页面片段可能包括独立的页眉、辅助栏或页脚。甚至页面的主体也可以分成几个片段。例如，主体页面上的一个片段可能显示一个产品，而第二个片段显示其价格。对页面分段则使您能显示为各个用户个性化的内容。随 WebSphere Commerce 提供的样本商店页面由几个片段组成（页眉、辅助栏、页脚和主内容）。

如果您的商店页面由片段组成，则您还有机会按片段对页面进行高速缓存。对各个片段进行高速缓存使您能够对页面中更多用户可重新使用的部分进行高速缓存。如果某页仅包含小部分用户的个性化信息，则将该页面高速缓存为完整页面将不会使该页面被频繁重新使用，因为只有那部分用户可以重新使用该高速缓存页面。例如，如果某页在页眉中显示对每个客户的欢迎消息，并根据用户标识进行高速缓存，则只有该特定用户可以重新使用该高速缓存页面。然而，如果将该页面分解成若干片段，则可以对大多数用户将重新使用的那些片段进行高速缓存。例如，页脚、侧栏和产品显示片段可能适用于所有用户，而价格和页眉片段则可能是个性化的。

当页面受到请求时，会将各个片段重新聚合，以生成页面。

可以使用完整页面高速缓存或片段高速缓存或两种方法的组合来对商店页面进行高速缓存。

---

## 制订更详细的高速缓存战略

在确定应对哪些页面和页面片段进行高速缓存之后，您需要确定更详细的高速缓存战略。对于您计划高速缓存的每一个页面或片段，您需要确定以下几点：

- 如何请求页面或片段
- 页面或片段是否依赖于商店关系
- 如何使高速缓存数据无效

### 如何请求页面或片段

如何请求 JSP 文件（是单个页面还是页面片段）确定了 WebSphere Application Server 将如何对其进行高速缓存。例如，WebSphere Application Server 需要知道是否 JSP 文件作为对 servlet、对象、EJB 或命令的响应而显示。结果，您需要编辑一个列表，以说明将如何请求您计划进行高速缓存的每一个页面或片段。

### 页面或片段是否依赖于商店关系

**Business** 如第 129 页的第 14 章，『商店间的关系』和第 61 页的第 7 章，『商店体系结构』中所讨论的那样，商店与其它商店的关系可允许那些商店相互使用数据。例如，商店 A 可以使用商店 B 中定义的产品目录数据。商店还可以与多家商店有关系，从而使它们能使用几个不同来源的数据。作为详细高速缓存规划的一部分，您需要确定每个页面或片段中显示的数据是否依赖于与其它商店的关系。如果某个页面的确显示其它商店的信息，则每次更新其它商店的数据时，您的已高速缓存的页面也需要更新。关于对商店关系进行高速缓存的信息，请参阅第 89 页的『实现对使用商店关系的商店页面的高速缓存』。

### 如何使高速缓存数据无效

对于您计划进行高速缓存的每个页面或页面片段，您还需要确定高速缓存的页面或片段何时不再有效，并从高速缓存中除去相应的高速缓存条目。此过程称为失效。为了确定高速缓存页面何时被更改（因此而不再有效），您需要确定可能使高速缓存页面过时的因素。例如，当客户向购物车添加新的商品时，高速缓存过的购物车页面则失效。当管理员使用 WebSphere 贸易加速器来更新商店时，或者当使用装入程序软件包或 WebSphere 贸易加速器中的工具调用来添加新的产品目录数据时，高速缓存过的页面也可能失效。

在编辑了可使高速缓存页面或片段失效的所有可能的方法的列表之后，您需要确定哪些事件用于促成失效。导致失效的事件可能包括 servlet 请求、控制器命令或任务命令等等。例如，如果使用 WebSphere 贸易加速器中的产品管理工具更新产品描述，WebSphere Commerce 会在内部调用命令 `AddCatalogEntryDescCmd` 或 `UpdateCatalogEntryDescCmd` 进行更改。如果您希望使这些命令更改的高速缓存页面无效，则需要向 `cachespec.xml` 文件添加失效策略，这些策略将拦截命令执行并触发失效。有关实现失效的信息，请参阅以下：

- 请参阅《WebSphere Commerce 管理指南》的『动态高速缓存』一章以获取关于设置新的失效策略的指示信息，以及高速缓存失效的示例。
- 请参阅第 88 页的『使 `cachespec.xml` 文件中的高速缓存数据失效』以获取关于如何合并 WebSphere Commerce 在商店的 `cachespec.xml` 文件中提供的样本失效策略的指示信息。

## 实现高速缓存战略

在您已收集了高速缓存战略所需的所有详细信息之后，可以通过创建高速缓存策略文件来实现该战略，此高速缓存策略文件定义您已收集的信息，包括哪些内容需要进行高速缓存，如何进行高速缓存，以及如何使高速缓存过的页面失效。WebSphere Application Server 动态高速缓存服务使用此高速缓存策略文件（称为 `cachespec.xml`）在商店中实现高速缓存。

随 WebSphere Commerce 提供的每个样本商店均包含 `cachespec.xml` 文件，该文件定义该商店的高速缓存战略。这些文件位于以下目录中：

`WC_install_dir/samples/dynacache/BusinessModel`

如果您的商店基于一个样本，您可以选择更改这些文件，或者使用这些文件中的一个作为您的商店的 `cachespec.xml` 文件的基础。

### 了解 `cachespec.xml` 文件

为了对 WebSphere Commerce 的商店页面进行高速缓存，您必须在 `cachespec.xml` 文件中定义可高速缓存的对象。WebSphere Commerce 仅使用 `cachespec.xml` 文件中定义的元素子集。该元素子集将在本节中说明。有关 `cachespec.xml` 文件的更详细信息，请参阅 WebSphere Application Server 信息中心

（<http://www.ibm.com/software/webservers/appserv/infocenter.html>）主题“Cachespec.xml 文件”。有关更多信息，请参阅《WebSphere Commerce 管理指南》的『动态高速缓存』一章。

### 了解 WebSphere Commerce 使用的元素

WebSphere Commerce 使用 `cachespec.xml` 中的以下元素：

- Class
- Name
- Property

这四个元素的使用在以下示例中说明：

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

**Class:** class 元素是必需的元素。它确定 WebSphere Application Server 将如何解释剩余的高速缓存策略定义。WebSphere Commerce 使用以下 class 值：

- command
- servlet

值 `command` 指的是使用 WebSphere Commerce 程序设计模块的 `class`。

值 `servlet` 指的是部署在 WebSphere Application Server servlet 引擎中的 `servlet` 或 `JSP` 文件。

**注：**对于 WebSphere Commerce v5.5，仅支持命令失效。

**Name:** Name 是 `servlet` 或 `command` 的全限定类名。Name 是必需的元素。

command 的 name 值必须包含软件包名称。例如  
`com.ibm.commerce.dynacache.commands.MemberGroupsCacheCmdImpl`

servlet 和 JSP 文件的 name 值必须包括要进行高速缓存的 JSP 文件或 servlet 的完整 URL。例如 `com.ibm.commerce.server.RequestServlet.class`

`/ToolTech/ShoppingArea`  
`/CatalogSection/CategorySubsection/StoreCatalogDisplay.jsp。`

**Property:** property 元素采取以下形式: `<property name=key>value</property>`, 其中 *key* 是正在定义的属性的名称, 而 *value* 是相应的值。可以在可高速缓存的对象上设置可选属性。例如 `<property name="consume-subfragments">true</property>`

对 WebSphere Commerce 商店页面进行高速缓存时, 使用以下属性:

属性	值	有效类	描述
EdgeCacheable	True 或 False, 缺省值为 False。	Servlet	如果该属性为 true, 则给定的 servlet 或 JSP 文件可以在外部从 Edge Server 请求。是 servlet 可高速缓存还是 JSP 文件可高速缓存, 这取决于高速缓存规范的其余部分。
consume-subfragments	True 或 False。缺省值为 false	Servlet	当高速缓存 servlet 时, 仅存储该 servlet 的内容。创建要包含或转发的任何其它片段的占位符。 Consume-subfragments (CSF) 指示高速缓存当通过包含遇到子 servlet 时继续保存内容。父条目 (标记了 CSF 的条目) 将在其高速缓存条目中包含来自所有片段的所有内容, 这会生成一个大的高速缓存条目, 该条目不具有包含或转发, 而是来自整个条目树的内容。此方法可以节省大量的应用程序服务器处理, 但通常仅在外部的 HTTP 请求包含了确定整个树的已包含片段所需的全部信息时才有用。

save-attributes	True 或 False。缺省值为 true。	Servlet	当 save-attributes 设置为 false 时，请求属性不与高速缓存条目一同保存。
store-cookies	True 或 False。缺省值为 true。	Servlet	当 store-cookies 设置为 false 时，请求 cookie 不与高速缓存条目一同保存。

缺省情况下，DynaCache 将 cookie（按 servlet 类进行高速缓存时）和所有请求属性（servlet 和 JSP）以及高速缓存条目进行高速缓存。但是，WebSphere Commerce cookie 和请求属性包含不应高速缓存的特定于用户的信息。因此，对完整页面进行高速缓存时，以下属性名和值是必填的：

```
<property name="save-attributes">false</property>
<property name="store-cookies">false</property>
```

对于所有为 JSP 文件定义的高速缓存条目，以下属性名和值是必填的：

```
<property name="save-attributes">false</property>
```

## 了解高速缓存标识规则

高速缓存标识唯一标识高速缓存条目。为了让 WebSphere Application Server 对某个对象进行高速缓存，它必须知道如何为该对象的不同调用生成唯一标识。这些标识是从用户所写的定制 Java 代码或高速缓存条目的高速缓存策略中定义的规则创建的。

在 cachespec.xml 文件中，cache-id 元素定义生成标识的规则。每个高速缓存条目都可以有多个高速缓存标识规则，它们将以定义的顺序执行，直到某个规则返回非空的高速缓存标识，或者没有剩下任何要执行的规则。如果没有任何高速缓存标识生成规则生成有效的高速缓存标识，则不对对象进行高速缓存。

这些标识是使用以下方法之一制订的：

- 使用高速缓存条目的高速缓存策略中定义的组件元素
- 编写定制 Java 代码以根据输入变量和系统状态构建标识

## 了解相关性标识规则

相关性标识元素指定附加高速缓存组标识，这些标识将多个高速缓存条目关联到同一个组标识。生成相关性标识的方法是将相关性标识基本字符串与其组件元素返回的值连接在一起。如果某个必需的组件返回空值，则整个相关性标识既不会生成也得不到使用。

您可以通过 WebSphere 动态高速缓存 API 显式地验证相关性标识，或者通过使用另一个高速缓存条目失效元素进行验证。每个高速缓存条目可以存在多个相关性标识规则。所有相关性标识规则都独立执行。有关如何定义相关性标识规则的更多信息，请参阅《WebSphere Commerce 管理指南》的『动态高速缓存』一章。

## 了解失效规则

失效规则可以采用与相关性标识完全一样的方式进行定义。不过，由失效规则生成的标识是用来使具有相同相关性标识的高速缓存条目失效的。生成失效标识的方法是将失效标识基本字符串与其组件元素返回的值连接在一起。如果某个必需的组件返回空

值，则整个失效标识不会生成，也不会发生任何失效。每个高速缓存条目可以存在多个失效规则。所有失效规则都独立执行。有关如何定义失效规则的更多信息，请参阅《WebSphere Commerce 管理指南》。

## 使 `cachespec.xml` 文件中的高速缓存数据失效

缺省情况下，随样本商店归档文件提供的 `cachespec.xml` 文件不包含失效策略。如果要在样本商店或在基于样本的商店中使用 DynaCache 来使高速缓存失效自动化，则必须向商店的 `cachespec.xml` 文件添加失效策略。以下目录中的几个 `cachespec.xml` 文件中提供了样本失效策略：

`WC_installdir/samples/dynacache/invalidation`

此目录包含功能区域的独立 `cachespec.xml` 文件，这些功能区域包括产品目录、购物车、商店等等。每个文件都包含该特定区域的失效策略。

如果计划对商店中的产品目录页面进行高速缓存，您应该将以下文件的失效策略添加到您的商店中：

- `WC_installdir/samples/dynacache/invalidation/catalog/cachespec.xml`
- `WC_installdir/samples/dynacache/invalidation/membergroup/cachespec.xml`

**注：**对于这些成员组失效规则，您需要向高速缓存条目添加附加的相关性标识。请参阅该 `cachespec.xml` 文件的内容以获取更多详细信息。

- `WC_installdir/samples/dynacache/invalidation/store/cachespec.xml`

### 将样本失效策略添加到商店的 `cachespec.xml` 文件中

为了将样本失效文件中提供的失效策略添加到商店中，请执行以下操作：

1. 打开商店的 `cachespec.xml` 文件。

- `WAS_installdir/installedApps/cell_name/WC_instanceName.ear/Stores.war/WEB-INF directory`

如果商店没有定义高速缓存策略，且基于随 WebSphere Commerce 提供的样本，则可以从以下目录使用样本 `cachespec.xml` 文件：

- `WC_installdir/samples/dynacache/BusinessModel`

2. 打开样本失效文件 `cachespec.xml`。样本失效文件 `cachespec.xml` 位于以下目录：

- `WC_installdir/samples/dynacache/invalidation`

3. 将失效策略从样本失效文件复制到商店的 `cachespec.xml` 文件。可以将失效策略放在商店的 `cachespec.xml` 文件末尾的最后一个元素的后面。
4. 请确保失效标识匹配高速缓存策略中相应的相关性标识。如果匹配的相关性标识不存在，则失效策略将不会执行，而您应更改失效规则的标识或相关性标识规则的标识以使它们能匹配。

**注：**商店可能有附加的或不同的业务需求，需要您添加附加的失效策略和相关性标识。

5. 如有必要，请在从样本失效文件复制的部分中更改 JSP 文件的名称和目录，以匹配商店的 `cachespec.xml` 文件的剩余部分中的信息。
6. 保存该文件。

## 实现对使用商店关系的商店页面的高速缓存

**Business** 如果商店正在通过商店关系使用其它商店中定义的数据，则您必须使用高速缓存过滤器指定的请求属性来定义关系。高速缓存过滤器是一个 servlet 过滤器，它从会话和 WebSphere Application Server DynaCache 可以使用的商店关系信息来定义请求属性。然后 DynaCache 使用该信息来构造要用于高速缓存失效的高速缓存标识和相关性标识。有关会话信息的请求属性设置的列表，请参阅《WebSphere Commerce 管理指南》的『动态高速缓存』一章。

高速缓存过滤器通过从 StoreAccessBean 调用 `getStorePath()` 和 `getStoresForRelatedStore()` 方法，来创建商店关系信息。相应的信息列于下表中：

表 3.

商店关系类型	商店关系标识	<code>getStorePath()</code> 的请求属性名称	<code>getStoresForRelatedStore()</code> 的请求属性名称
IBM 商业业务策略	-1	DC_busN	DC_bus_RS_N
IBM 商业业务竞销	-3	DC_campN	DC_camp_RS_N
IBM 商业业务产品目录	-4	DC_catN	DC_cat_RS_N
IBM 商业业务命令	-5	DC_cmdN	DC_cmd_RS_N
IBM 商业主管商店	-6	DC_hostN	DC_host_RS_N
IBM 商业价格	-7	DC_prcN	DC_prc_RS_N
IBM 商业参照	-8	DC_refN	DC_ref_RS_N
IBM 商业分段	-9	DC_segN	DC_seg_RS_N
IBM 商业 URL	-10	DC_urlN	DC_url_RS_N
IBM 商业视图	-11	DC_viewN	DC_view_RS_N
IBM 商业库存	-13	DC_invN	DC_inv_RS_N
IBM 商业基本商品	-14	DC_baseItemN	DC_baseItem_RS_N
IBM 商业渠道商店	-15	DC_chsN	DC_chs_RS_N
IBM 商业货币转换	-17	DC_currConvN	DC_currConv_RS_N
IBM 商业货币格式	-18	DC_currFmtN	DC_currFmt_RS_N
IBM 商业支持货币	-19	DC_supCurrN	DC_supCurr_RS_N
IBM 商业对应价值货币	-20	DC_cterCurrN	DC_cterCurr_RS_N
IBM 商业评估格式	-21	DC_meaFmtN	DC_meaFmt_RS_X

**注：**在由于 DynaCache 不支持请求属性数组而返回多个商店标识的情况下，高速缓存过滤器设置多个请求属性。例如，如果 `getStorePath()` 为资源标识 -4（IBM 商业业务产品目录）返回数组 [10051, 10002]，则请求属性设置将如下：

- DC\_cat0 为 10051
- DC\_cat1 为 10002

### 商店关系高速缓存示例

**Business** 要了解使用商店关系的高速缓存页面如何工作，请考虑以下示例。

发布样本组合商店归档文件  DemandChain.sar, 然后在该站点中创建受托管商店 (例如 ResellerOne), 则创建以下商店。

表 4.

商店标识	目录	商店类型
10001	CommercePlaza	渠道中心
10002	CommercePlazaCatalog	产品目录有用资源商店
10003	CommercePlaza	经销商代理
10004	ConsumerDirectResellerProfile	主管商店前台有用资源商店
10051	ResellerOne	转销商主管商店

转销商主管商店 ResellerOne (10051) 使用主管商店前台有用资源商店 (10004) 和产品目录有用资源商店 (10002) 中定义的有用资源。

为了设置高速缓存关系, 高速缓存过滤器获取以下信息:

表 5.

商店标识	关系类型	getStorePath()	getStoreFor RelatedStore()
10001	-1 (业务策略) -4 (产品目录) -7 (价格) -17 (货币格式) -19 (支持的货币)	10002	不适用
10001	-6 (主管商店)	10051	不适用
10051	-1 (业务策略) -14 (基本商品)	10051, 10002, 10004	10051
10051	-3 (竞销) -5 (命令) -10 (URL) -11 (视图)	10051, 10004	10051
10051	-4 (产品目录) -7 (价格) -17 (货币转换) -18 (货币格式) -19 (支持的货币) -20 (对应值币种) -21 (计量格式)	10051, 10002	10051

然后高速缓存过滤器设置以下请求属性:

表 6.

商店关系	商店标识 10051	商店标识 10051	商店标识 10001
-1 (业务策略)	DC_bus0=10051 DC_bus1=10002 DC_bus2=10004	DC_bus_RS_0=10051	DC_bus0=10002



表 6. (续)

-2 (税款)	DC_tax0=10051 DC_tax1=10004	DC_tax_RS_0=10051	
-4 (产品目录)	DC_cat0=10051 DC_cat1=10002	DC_cat_RS_0=10051	DC_cat0=10002
-6 (主管商店)	DC_host0=10051	DC_host_RS_0=10001	DC_host0=10051

无论何时产品目录有用资源商店 (10002) 的产品目录发生更改, ResellerOne 商店 (10051) 的产品目录页面也必须在使用产品目录有用资源商店 (10002) 中的信息之前失效。为了使 10051 中的页面失效, 必须为此商店关系设置额外的相关性标识。在以下示例中说明了设置 StoreCatalogDisplay 的额外相关性标识:

```
<!-- Start Store Relationship Dependency Ids -->
<!-- DC_cat1 is the catalog Profile Store ID -->
<dependency-id>storeId
<component id="DC_cat1" type="attribute">
<required>true</required>
</component>
</dependency-id>

<dependency-id>storeId:catalogId
<component id="DC_cat1" type="attribute">
<required>true</required>
</component>
<component id="catalogId" type="attribute">
<required>true</required>
</component>
</dependency-id>

<dependency-id>StoreCatalogDisplay:storeId
<component id="DC_cat1" type="attribute">
<required>true</required>
</component>
</dependency-id>
<!-- Ends Store Relationship Dependency Ids -->
```

创建的额外相关性标识如下:

- storeId:10002
- storeId:catalogId:10002:10051
- StoreCatalogDisplay:storeId:10002

一旦定义了这些额外相关性标识, 则无论何时导致产品目录有用资源商店页面失效的产品目录有用资源商店 10002 发生更改, 主管商店 (10051) 页面也将失效。

## 以动态高速缓存来替换高速缓存命令功能

先前版本的 WebSphere Commerce 使用 CacheCommand

( com.ibm.commerce.cache.commands.CacheCommandImpl ) 实现更高级的高速缓存配置, 例如, 按客户概要文件中确定的用户状态和类型对页面进行高速缓存。

在 V5.5 中, 就象使用高速缓存命令一样, 您可以通过将高速缓存命令逻辑添加到 JSP 文件从而使用动态高速缓存对 servlet 或 JSP 文件结果进行高速缓存。

请看以下示例:

StoreCatalogDisplay 命令可以根据用户状态和类型属性而显示不同页眉。要对页眉 JSP 文件进行高速缓存, 请创建新的包含用户状态和类型属性的 JSP 文件 CacheParametersSetup.jsp。例如:

```
<%@ page import="com.ibm.commerce.command.CommandContext" %>
<%
    String userState = null;
    String userType = null;

    CommandContext cmdcontext = (CommandContext) request.getAttribute
(ECConstants.EC_COMMANDCONTEXT);
    if (cmdContext != null) {
        userState = cmdcontext.getUser().getState();
        userType = cmdcontext.getUser().getRegisterType();
    }
%>
```

然后 StoreCatalogDisplay.jsp 使用 userState 和 userType 作为输入参数静态包含 CacheParametersSetup.jsp 并动态包含 CachedHeaderDisplay.jsp:

```
<%@ include file="CacheParametersSetup.jsp"%>
<jsp:include page="CachedHeaderDisplay.jsp" flush="true">
    <jsp:param name="storeId" value="<%= storeId %%" />
    <jsp:param name="catalogId" value="<%= catalogId %%" />
    <jsp:param name="langId" value="<%= languageId %%" />
    <jsp:param name="userState" value="<%= userState %%" />
    <jsp:param name="userType" value="<%= userType %%" />
</jsp:include>
```

CachedHeaderDisplay.jsp 文件包含根据输入参数而显示不同信息的逻辑。

```
<%
    if (userType.equals("G")) {
%>
        <table cellpadding="0" cellspacing="0" border="0" width="100%" height="28">
            .
            .
            .
        </table>
<%
    }
    else {
%>
        <table cellpadding="0" cellspacing="0" border="0" width="100%" height="28">
            .
            .
            .
        </table>
<%
    }
%>
```

为了完成高速缓存, 输入参数必须按高速缓存标识规则来标识。

```
<cache-entry>
    <class>servlet</class>
    <name>../CachedHeaderDisplay.jsp</name>
    <property name="save-attributes">>false</property>

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

```
<required>true</required>
</component>
  < /cache-id>
  . . .
</cache-entry>
```



---

## 第 5 部分 商店数据概述



---

## 第 10 章 商店数据

本章提供了 WebSphere Commerce Server 商店数据体系结构和创建商店的数据有用资源两者的概述。本章中还介绍了 WebSphere Commerce Server 信息模型。

---

### 什么是商店数据？

商店数据是装入 WebSphere Commerce Server 数据库的信息，使您的商店可以正常运作。为了正常运作，商店必须准备好数据以支持所有客户活动。例如，要使客户可以进行购买，商店必须包含待售商品的产品目录（产品目录数据）、与处理订单关联的数据（税款和送货数据）和用于实现请求的库存（库存和供货数据）。

### 商店数据信息模型

本指南使用信息模型说明商店数据在 WebSphere Commerce Server 中是如何组织的。WebSphere Commerce Server 信息模型是关于 WebSphere Commerce Server 数据模型中所包含信息的高级抽象。信息模型突出显示数据模型的最重要功能，但不包括特定于模式和对象实现的较低级详细信息。

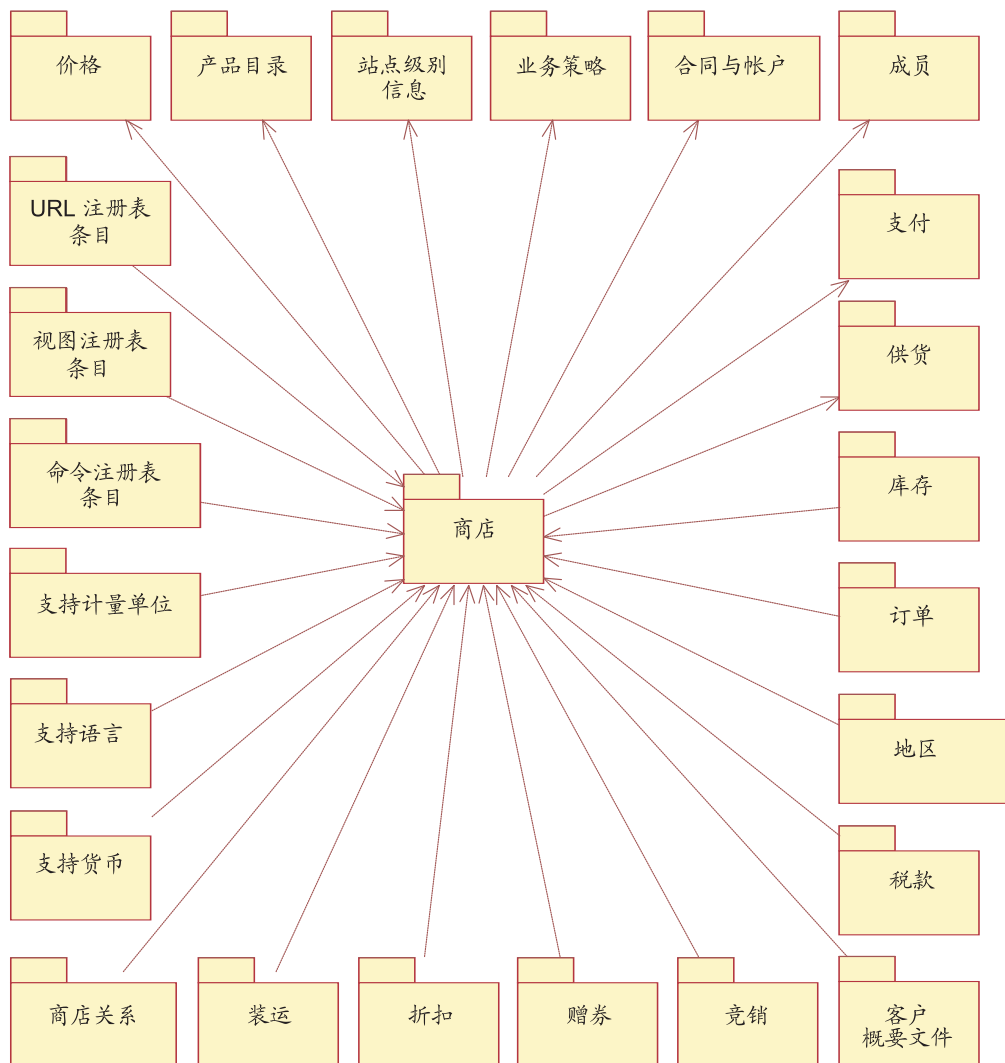
例如，包含实体关系数据（如外键对）的数据模型中某些表和对象不作为实体在信息模型中显示。这些实体关系由信息模型中实体之间的关系线指出。这些信息模型也不同于数据模型，在数据模型中每个条目代表一个表，而在信息模型中任何描述对象都可以映射到相同的数据库表，或单个对象可以映射到几个数据库表。此信息模型也不说明 *详细信息扩展*（作为涉及实现的结果而存储于单独表中的实体附加数据属性：例如，产品描述是产品实体单独存储的扩展）。最后，和数据模型不同的是，信息模型可能也说明继承的概念。关于实体关系数据和详细信息扩展的更多信息，请参阅 WebSphere Commerce 产品和开发联机帮助中的数据模型。



关于 WebSphere Commerce 对象和数据模型的更多信息，请参阅 WebSphere Commerce 联机帮助。

---

下图说明 WebSphere Commerce 商店的数据有用资源。



本图和商店数据部分中的其它所有图都是 WebSphere Commerce 信息模型的一部分。关于信息模型的更多信息，请参阅第 97 页的『商店数据信息模型』。关于本图中所使用的约定的更多信息，请参阅第 415 页的附录 A，『UML 图注』。

以上这些图中例举的每个数据有用资源均在第 107 页的第 6 部分，『开发商店数据』中的章节中进行更详细的讨论。

#### 注:

在 UML 符号表示法中，带箭头的点划线从某对象开始伸展并指向另一个对象，表示第一个对象具有对第二个对象的相关性。在该图中，显式的对象均称为数据包。请注意，某些数据包中的数据（如“支持货币”列表）是特定于某特定“商店”的，因此该数据包显示为从属于“商店”数据包。其它数据包（如“产品目录”）并不特定于任何特定“商店”，而是每个“商店”可以使用“产品目录”，因此“商店”对象显示为从属于“产品目录”数据包。因此，“支持货币”列表组成“商店”的一部分，而“商店”使用“产品目录”。



**Business** 应特别关注的一个“商店”部分是它与其它“商店”的“商店关系”。每个商店关系均指示“商店”依赖另一个“商店”来提供某些服务或信息。可定义关系以简化一个“商店”的数据（如其“支持货币”列表）由另一个“商店”使用。在这种情况下，第一个“商店”充当第二个（客户机）“商店”数据的提供程序或容器。当创建了更多客户机“商店”时，它们也可以定义表明它们从某些其它“商店”获取某些数据的关系。“商店关系”通过这种方式帮助数据共享；数据可以由提供程序“商店”创建并维护一次，而由几个客户机“商店”使用。关于商店关系的更多信息，请参阅第 129 页的第 14 章，『商店间的关系』。

信息模型中的数据可按以下方式分类：

- 按子系统
- 按数据类型

## 按子系统查看商店数据信息模型

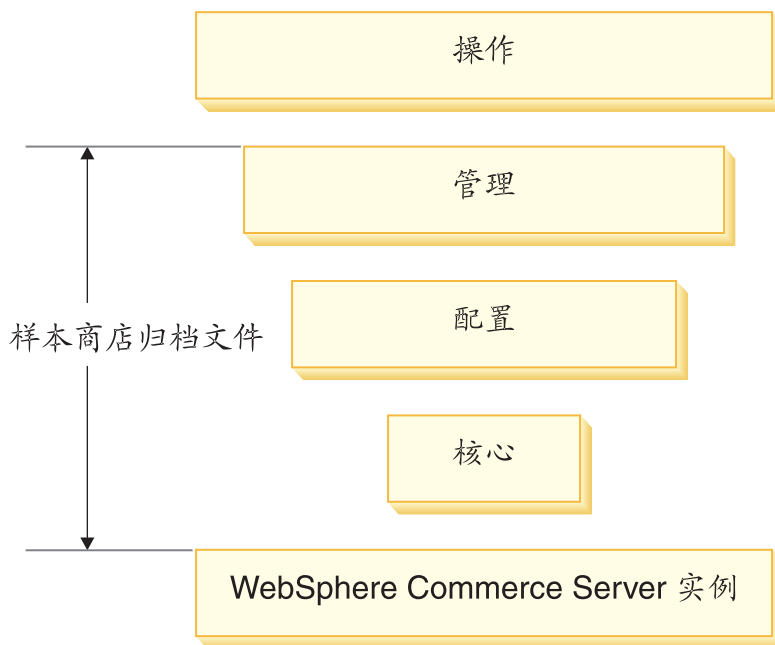
商店数据信息模型中的每个数据有用资源均可分组到以下功能区域：

表 7.

销售策略	市场营销	贸易	订单管理	产品目录	成员	运行时
折扣	竞销	合同	送货	产品目录	组织	组织
供应商	客户概要文件	帐户	税款	价格	组	URL、命令和视图注册表
拍卖	电子邮件活动	<b>Business</b> RFQ	地区		用户	支持语言
	赠券		订单			支持的计量单位
			库存			支持货币
			供货			站点
			支付			商店
						商店关系
						业务策略

## 按数据类型查看商店数据信息模型

WebSphere Commerce 商店中的数据符合下图中描述的类型。『按子系统查看商店数据信息模型』的图中说明的每个商店数据有用资源可以分类为属于以下说明的一种或多种类型的商店数据。



## WebSphere Commerce Server 实例

数据的基本级别包含在 WebSphere Commerce Server 实例中。创建实例时，以 XML 格式装入的引导程序文件将用信息填充数据库。引导程序文件创建以下类型的数据：

- 计算用法类型、设备类型（浏览器、电子邮件、I 方式等等）、消息类型、角色和地址
- 缺省管理标识 WCSADMIN
- 缺省命令、视图和 URL
- 缺省业务策略
- 缺省访问组和访问控制策略
- 实例支持的语言和货币
- 缺省数量单位和数量单位转换
- 缺省调度作业和状态代码
- 缺省条款和条件
- 可以用作商店所有者的缺省组织
- 缺省站点组织
- 缺省商店组
- 缺省登台信息

此信息可用于该实例中存在的所有商店，并且在第 99 页的『按子系统查看商店数据信息模型』的图中标识为站点级别信息。

关于引导程序文件及其填充的数据库表的更多详细信息，请参阅 WebSphere Commerce 联机帮助。

## 核心数据

下一级别的商店数据是核心数据。核心数据分为两级：

- 组织

- 商店

组织核心数据创建业务模型特定环境的最小数据，包括：

- 组织结构。
- 预定用户角色。
- 必要访问控制策略。

组织核心数据在样本组合商店归档文件中与样本组织结构组合商店归档文件中都是可用的。

核心数据创建环境内商店的最小数据，包括：

- STOREENT 表中的商店标识。这在数据库中创建商店。
- 缺省合同。
- 合同数据库表中的存储标识。
- 拥有合同数据库表中商店的组织的成员标识。
- STORE 表中的商店目录。商店目录是商店的 Web 有用资源所在的目录。
- STADDRESS 表中商店地址的别名或标识。别名对于每个商店是唯一的。

商店核心数据在样本组合商店归档文件中与样本组件商店归档文件中都是可用的。

如果您在管理控制台中通过使用发布实用程序发布过以上指出的任何样本商店归档文件，则本信息为您创建。发布实用程序使您能够选择可作为商店所有者的缺省组织，或者您可以使用组织管理控制台创建另一个组织作为所有者。如果没有发布样本组合商店归档文件以用作商店的基础，则您将必须使用装入程序软件包将此信息装入到数据库，或者直接编辑数据库。有关使用装入程序软件包的更多信息，请参阅第 323 页的第 37 章，『装入商店数据的概述』。

第 99 页的『按子系统查看商店数据信息模型』的图中的商店数据是核心数据。

## 配置数据

配置数据控制贸易服务器运行时。贸易服务器运行时提供一种在其中部署和执行贸易应用程序的框架。该框架由命令执行、异常处理、事务控制、数据访问和持久性组成。贸易服务器运行时利用 WebSphere Application Server 提供的运行时服务来支持 WebSphere Commerce Server 应用程序。配置数据确定商店将用于显示商店页面的命令、视图和 JSP 文件。

第 99 页的『按子系统查看商店数据信息模型』的图中标识的以下数据有用资源分类为配置数据：

- 命令注册表项
- 视图注册表项
- URL 注册表项

## 受管数据

受管数据是卖方创建的数据，对卖方站点的客户而言是只读的。由于卖方完全控制此数据的状态，所以可以通过内容管理系统管理受管数据。

第 99 页的『按子系统查看商店数据信息模型』的图中标识的以下数据有用资源分类为受管数据：


- 业务策略
- 竞销
- 产品目录
- 合同
- 赠券
- 货币
- 客户概要文件
- 折扣
- 电子邮件活动
- 供货中心
- 库存（产品目录商品的配置信息）
- 地区
- 语言
- 成员
- 支付
- 价格
- 卖方
- 送货
- 税款
- 计量单位
- 供应商

## 操作数据

操作数据是站点客户通过与站点的交互创建或更改（直接或间接）的数据。例如，客户订单被视为库存级别（随商店操作上升或下降）的操作数据。客户也被视为操作数据。卖方创建的数据也是可操作的。

由于对操作数据的更改不完全受卖方的控制，故这些数据不是使用内容管理系统来管理的。

第 99 页的『按子系统查看商店数据信息模型』的图中标识的以下数据有用资源分类为操作数据：

- 拍卖
- 合同
- 客户
- 电子邮件活动
- 供货
- 库存（接收货物、预期接收货物、库存分配）
- 订单
-  报价请求（RFQ）

**注：**在某些情况下，操作数据和受管数据之间的界线可能很难确定。例如，在一个商店中，客户和合同数据可能被视为受管数据，而在另一个商店中，同一类型的数

据可能就被视为操作数据。第一个商店可以管理其客户数据与相关合同，因为它们拥有一组特定的客户（即客户无法在线注册）。但是，第二个商店允许客户在线注册并在线创建合同信息。

另一个示例涉及产品目录数据。在单个卖方站点中，产品目录被视为受管数据。在价值链站点中，产品目录数据可能被视为操作数据。

在一些站点中，相同数据类型的某些记录可能被视为受管数据，而另一些记录则被视为操作数据。例如，缺省合同可能是受管数据，而在线协商的特定合同则是操作数据。另一个示例是电子邮件活动。电子邮件活动信息和模板被视为受管数据，但从模板生成并发送到客户的实际电子邮件活动被视为操作数据，正如发送邮件所导致的任何事件一样，例如客户打开电子邮件，或单击电子邮件的任何可单击的内容。

## 商店数据类型和样本公司

随 WebSphere Commerce 提供的样本公司包含商店数据体系结构中大多数类型的商店数据。例如，WebSphere Commerce Server 实例必须存在，才能使用样本商店创建商店或发布样本商店。这样，当您通过在管理控制台中使用发布实用程序中的工具基于样本商店创建商店时，就创建了核心数据。样本商店包含所有必需的配置以及正常运作的商店所需的大多数受管数据。基于特定样本商店创建商店时，可能会指示您使用 WebSphere 贸易加速器中的工具完成数据的一些设置。

---

## 创建数据的工具

WebSphere Commerce 提供了创建和操作商店数据的多种工具。下面列出了这些工具：

### WebSphere Commerce 装入程序软件包

装入程序软件包主要由准备数据并将数据装入 WebSphere Commerce 数据库的实用程序组成。关于更多信息，请参阅第 309 页的第 10 部分，『发布商店』。

### 管理控制台

管理控制台使您可以通过完成管理操作和配置任务来控制站点和商店。您还可以使用管理控制台创建新的组织和用户，以及向角色分配用户。管理控制台还使您可以确定将在商店中可用的通知和消息传递类型。管理控制台包含发布实用程序，该程序使您能够发布样本企业和商店。

### WebSphere 贸易加速器

WebSphere 贸易加速器是使您可以创建和维护各种商店有用资源的在线工具的工作台。大部分商店数据可以使用 WebSphere 贸易加速器中的工具创建和管理。关于更多信息，请参阅第 104 页的『工具和商店数据摘要图表』。

### 组织管理控制台

组织管理控制台使您能够创建和管理访问您的站点或商店的组织。组织管理控制台还使买方管理员能够管理其组织内的买方。

## 工具和商店数据摘要图表

以下图表列出了可用于创建每种类型的数据的工具。

创建数据的工具	核心数据	配置数据	受管数据	操作数据
WebSphere Commerce 装入程序软件包	使用装入程序软件包以 XML 文件形式装入核心数据。关于更多信息，请参阅第 124 页的『在 XML 文件中创建商店数据有用资源』。	使用装入程序软件包以 XML 文件形式装入配置数据。关于更多信息，请参阅第 137 页的『创建 XML 文件以注册命令、视图和 URL』。	使用装入程序软件包以 XML 文件形式装入受管数据。关于更多信息，请参阅关于受管数据有用资源的相应章节。	通常，操作数据不能使用装入程序软件包装入。但是，所选的客户数据可以使用装入程序软件包来装入。
管理控制台	使用管理控制台发布商店归档文件时，就为您创建了核心数据。关于使用发布的更多信息，请参阅 WebSphere Commerce 产品联机帮助。	未提供。	未提供。	未提供。
WebSphere 贸易加速器	未提供。	未提供。	<p>使用 WebSphere 贸易加速器创建或编辑以下数据：</p> <ul style="list-style-type: none"> <li>• 竞销</li> <li>• 合同（缺省合同必须存在于数据库中，您才可以使用 WebSphere 贸易加速器中的“业务关系管理”工具创建附加合同或更改现有合同。使用装入程序软件包（商店创建向导）或发布商店归档文件在数据库中创建缺省合同。</li> </ul> <ul style="list-style-type: none"> <li>• 地区</li> <li>• 税款</li> <li>• 送货</li> <li>• 货币</li> <li>• 语言</li> </ul>	<p>客户在向商店注册或从商店进行购买时创建操作数据。但是，在某些情况下，可以使用 WebSphere 贸易加速器为客户下订单或者创建退货。</p> <p>WebSphere 贸易加速器还使您可以管理库存（接收货物或预期接收货物）。</p>

创建数据的工具	核心数据	配置数据	受管数据	操作数据
WebSphere 贸易加速器 (续)	未提供。	未提供。	<ul style="list-style-type: none"> <li>• 供货</li> <li>• 折扣</li> <li>• 产品目录 (主产品目录必须存在于数据库中, 您才可以使用 WebSphere 贸易加速器中的“产品管理”工具添加或更改产品信息。要获取销售策略和服务的不同视图, 请通过更改 XML 源来创建销售产品目录。使用装入程序软件包或发布商店归档文件, 在数据库中创建主产品目录。)</li> <li>• 价格</li> </ul>	未提供。
组织管理控制台	使用组织管理控制台来创建和管理组织。	未提供。	未提供。	客户和买方是在他们进入商店时创建的。但是, 通过使用组织管理控制台, 您还可以管理用户和核准买方, 或者创建新的用户和买方。





---

## 第 6 部分 开发商店数据

本部分中的章节更详细地解释了每个商店数据有用资源。本部分中的商店数据有用资源按照 WebSphere Commerce 商店数据体系结构进行组织:

- WebSphere Commerce Server 实例

- 站点

- 核心数据

- 组织

- 商店

- 商店之间的关系

- 配置数据

- 命令注册表

- 视图注册表

- URL 注册表

- 受管数据

- 产品目录

- 价格

- 合同 (包括业务策略)

- 供货

- 竞销

- 支付

- 支持的语言

- 支持的货币

- 支持的计量单位

- 地区

- 送货

- 税务

- 折扣

- 操作数据

- 库存

- 订单

- 客户

- 拍卖

-  RFQ

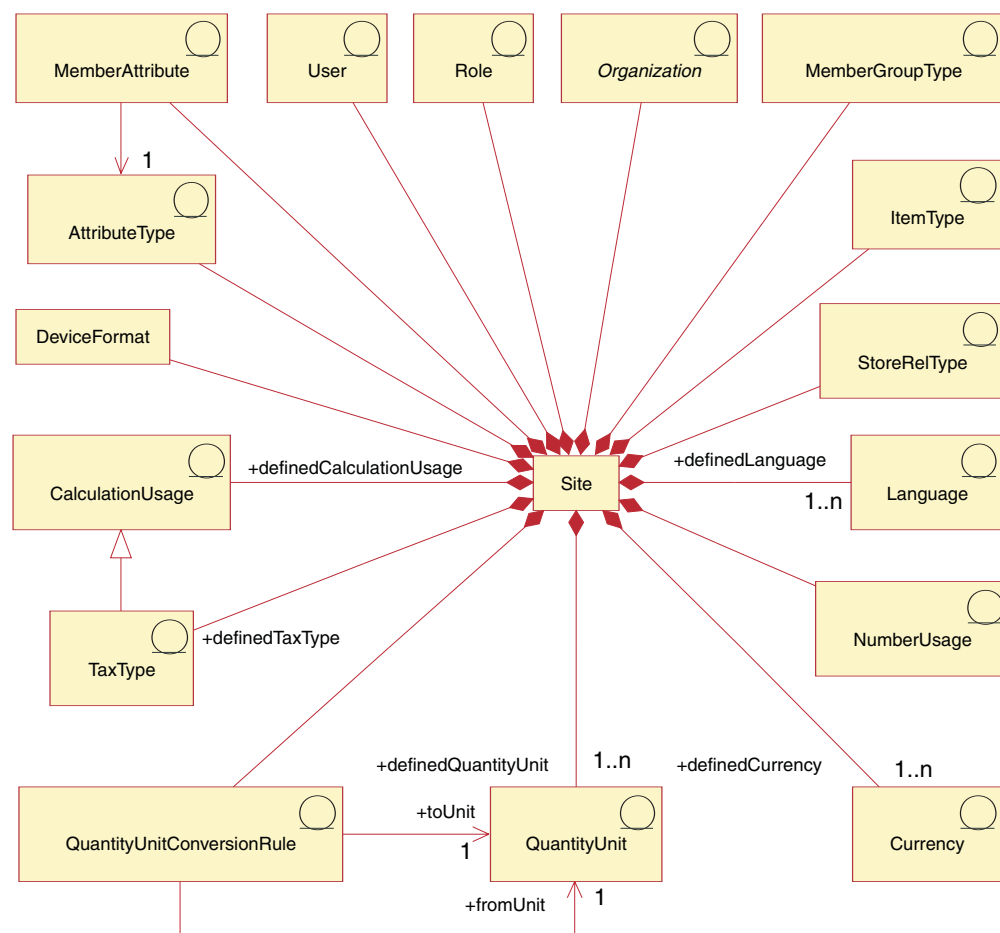


## 第 11 章 站点有用资源

每个 WebSphere Commerce Server 实例都具有其自己的关系信息数据库。实例由引导程序文件创建，这些文件在创建模式之后用信息填充数据库表。一旦装入数据，您就可以在相应的数据库表中查看预装入的信息。许多数据库表包含特定于商店或商店组的商店或商店组级别信息。一些表包含的信息表示可由实例中的所有商店使用的 WebSphere Commerce 站点级别功能。所有这些信息由 WebSphere Commerce 站点管理员管理。这些功能在本章中进行讨论。关于引导文件的更多信息，请参阅 WebSphere Commerce 产品和开发联机帮助。关于特定于商店的有用资源信息的更多信息，请参阅第 123 页的第 13 章，『商店有用资源』。

### 了解 WebSphere Commerce 中的站点有用资源

下图说明了站点包含的数据类型及其与站点的关系。



关于本图中使用的约定的更多信息，请参阅第 415 页的附录 A，『UML 图注』。本图和商店数据部分中的其它所有图都是 WebSphere Commerce 信息模型的一部分。关于信息模型的更多信息，请参阅第 97 页的『商店数据信息模型』。

## 语言

站点可以在 LANGUAGE 表中定义许多语言，在 LANGUEDS 表中对其进行描述。每个商店通常通过向 STORELANG 表添加行支持这些语言的子集。预定义的十种语言是：德语、繁体中文和简体中文、日语、韩国语、意大利语、法语、西班牙语、巴西葡萄牙语和英语。

## 成员属性

成员属性存储在 MBRATTR 表中，代表一组已定义的属性名称，其值可以为组织或用户存储。此类属性名称的示例包括 JobFunction、ProcurementCard、SpendingLimit、ReferredBy 和 CountryOfOperation。特定组织或用户的属性值存储在 MBRATTRVAL 表中，且这些值对不同的商店或商店组可以不同。

## 属性类型

属性类型存储在 ATTRTYPE 表中，代表可以用于代表属性值的已定义数据类型。数据类型的示例包括：INTEGER、STRING 和 FLOAT。

## 成员组类型

成员组类型存储在 MBRGRPTYPE 表中，代表一组已定义的成员组用法。通过向 MBRGRPUSG 表添加行为成员组指定用法。成员组用法的示例包括 AccessGroup（供访问控制策略使用）和 UserGroup（供一般用途使用，如客户组）。

## 用户

用户代表已认证的用户身份。用户通常表示以买方组织名义下订单或核准订单的客户、为销售组织处理订单或维护商店级别有用资源的销售代理，或者维护 WebSphere Commerce Server 实例的站点管理员。每个用户与一个站点关联，并且在 USERS 表中定义。

## 组织

组织代表组织和组织内的组织单位。组织通常代表负责购买或销售的商业实体。客户在 B2B 直销购买组织中下的订单被记录为代表买方组织下的订单。商店、产品目录和供货中心的所有者是负责销售的特定方面的组织。组织在 ORGENTITY 表中定义。

## 角色

角色代表可在组织内为用户指定的一组已定义角色。例如，可以在销售组织内为用户指定客户服务代表角色，或者可以在买方组织内为用户指定买方核准员角色。缺省角色的名称和描述填充在 ROLE 表中。关于特定角色的更多信息，请参阅 WebSphere Commerce 联机帮助。

## 数量单位转换

每个站点都有数量转换。这些转换代表用于在不同计量单位之间进行转换的乘法或除法运算。这些规则填充在 QTYCONVERT 表中。

## 数量单位

数量单位代表站点的一组计量单位。这些单位在 QTYUNIT 表中定义，QTYUNITDSC 表中对其进行了描述。每个商店可以通过向 QTYFORMAT 表添加行指定如何舍入和格式化以每种计量单位表示的数量以供显示（根据其预期用法）。

## 税款类型

税款类型代表计算税款的计算用法。销售税和送货税是计算税款的两种不同计算用法。税款类型在 TAXTYPE 表中定义。

## 计算用法

计算用法代表可由 OrderPrepare 命令执行的不同类型的计算。为折扣、送货费用、销售税、送货税和电子赠券定义了计算用法。计算用法在 CALUSAGE 表中定义。

## 货币

每个站点都在 SETCURR 表中定义许多货币并在 SETCURRDSC 表中对这些货币进行描述。每个站点都通过向 CURLIST 表添加行（每行对应一种支持的货币）支持这些货币的子集。

**注：**对于一些站点有用资源（如语言、货币、数量单位和数量单位转换规则），站点管理员可以通过向适当的表添加行来扩展站点级别功能。对于其它站点有用资源，扩展其代表的站点级别功能可能还需要相关定制。例如，如果站点管理员添加了一种新的数值用法以定制的货币符号显示小计，则必须定制显示小计的程序，以便指定在格式化小计金额以供显示时使用新的小计数值用法。

## 数值用法

数值用法代表数值的预期用法。商店可以根据使用数值的方式来为显示的数值指定不同的舍入和格式化规则。例如商店可以通过指定“单价”用法将单价舍入到四位小数，而通过指定“缺省”用法将其它货币金额舍入到两位小数。数值用法在 NUMBRUSG 表中定义，NUMBRUSGDS 表中对其进行了描述。

## 商品类型

商品类型代表不同类型的基本商品。WebSphere Commerce 中两种类型的基本商品是可改装的成套商品和一般商品。商品类型在 ITEMTYPE 表中预定义。关于基本商品的更多信息，请参阅第 259 页的第 29 章，『库存有用资源』。

## 设备格式

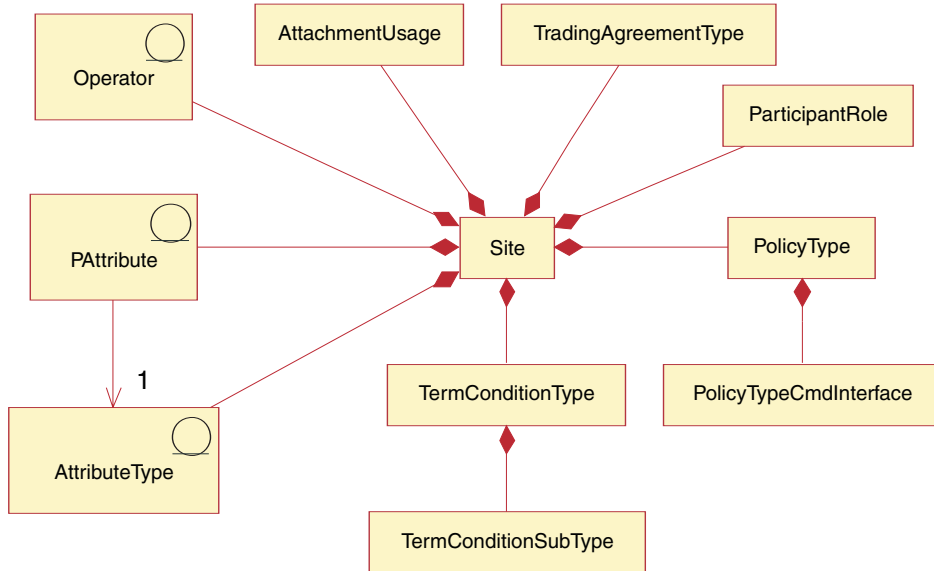
设备格式存储在 DEVICEFMT 表中，并代表站点使用的诸多设备格式，如浏览器、I\_MODE、电子邮件、XMLMQ 和 XMLHTTP。所有这些设备类型都允许用户通过各种介质与站点交互。

## 商店关系类型

**Business** 商店关系类型（StoreRelType）定义两家商店之间关系的类型。每个商店关系类型均定义自己的关系，也就是说，关系中的每个伙伴将起什么作用，以及两个伙伴之间的关系是什么。商店关系类型是在 STRELTYT 表中定义和在 STRELTYT PDS 表中描述的。

## 站点级贸易协议数据

下图举例说明了站点包含的贸易协议数据的类型及其与站点的关系。



关于本图中使用的约定的更多信息，请参阅第 415 页的附录 A，『UML 图注』。本图和商店数据部分中的其它所有图都是 WebSphere Commerce 信息模型的一部分。关于信息模型的更多信息，请参阅第 97 页的『商店数据信息模型』。

## 贸易协议类型

WebSphere Commerce 提供了很多贸易机制管理买方与卖方之间的交互。贸易协议代表贸易机制的一个实例，并记录贸易机制的该实例的属性。WebSphere Commerce 中的每个合同、商业帐户和 RFQ 都由一个贸易协议表示。有一个管理 WebSphere Commerce 中所有拍卖的贸易协议。WebSphere Commerce 支持几个贸易协议类型，包括帐户、合同、RFQ、交换和拍卖。贸易协议类型是在 TRDTYPE 表中定义的。关于贸易协议的更多信息，请参阅第 177 页的第 18 章，『合同有用资源』。

## 参与方角色

贸易协议的参与方在每份贸易协议中均扮演特定的角色。WebSphere Commerce 支持几种参与方角色，包括创建者、卖方、买方、供应商、核准员、管理员、经销商、服务供应商、转销商、主管和接收方。参与方角色是在 PARTROLE 表中定义的。

## 策略类型

WebSphere Commerce 支持几种业务策略，包括价格、产品集、送货方式、送货费用、支付以及其它几种策略。策略类型是在 POLICYTYPE 表中定义的。关于业务策略的更多信息，请参阅第 177 页的第 18 章，『合同有用资源』。

### 策略类型命令接口

策略类型命令接口是业务策略对象的 Java 命令接口。每个策略实例的命令必须实现该接口。每个业务策略对象可以有 0 条或多条命令。

## 条款和条件类型

条款和条件定义贸易协议的行为和属性。WebSphere Commerce 支持几个条款和条件类型，包括定价、支付和送货。条款和条件类型是在 TCTYPE 表中定义的。关于条款和条件的更多信息，请参阅第 177 页的第 18 章，『合同有用资源』。

### 条款和条件子类型

每个条款和条件类型均可以包含几个条款和条件子类型。条款和条件子类型是在 TCSUBTYPE 中定义的。

## 个性化属性

个性化属性允许您创建产品的属性。个性化属性是在 PATTRIBUTE 表中定义的。每一个个性化属性均有且仅有一个属性类型。

## 属性类型

属性类型定义属性的类型。属性类型是在 ATTRTYPE 表中定义的。

## 运算符

站点中使用的运算符包括简单运算符（允许单个值）、复合运算符（范围 - 连续）和复合运算符（集合）。运算符是在 OPERATOR 表中定义的。

## 附件用法

附件是贸易文档的支持文档。例如，它可以是产品的说明书，或者是价目表电子表格。附件用法描述将在何处如何使用附件。附件用法是在 ATTACHUSG 表中定义的。

---

## 在 WebSphere Commerce 中创建站点有用资源

站点有用资源是当您在 WebSphere Commerce Server 中创建实例时创建的。关于在 WebSphere Commerce Server 中创建实例的更多信息，请参阅《WebSphere Commerce 安装指南》中的『创建 WebSphere Commerce 实例』。










本图和商店数据部分中的其它所有图都是 WebSphere Commerce Server 信息模型的一部分。关于信息模型的更多信息，请参阅第 97 页的『商店数据信息模型』。关于本图中所使用的约定的更多信息，请参阅第 415 页的附录 A，『UML 图注』。

## 成员

WebSphere Commerce 中的成员可以是以下任意一个：

- **组织实体**。这可以是组织（例如 IBM）或大型组织中的组织单位（例如 IBM 中的“电子交易分部”）。
- **用户**（已注册的或未注册的）。注册用户具有唯一的标识和密码，且为了注册的目的，需要提供概要文件数据。注册用户可以根据他们的概要文件类型分类：类型“B”表示商业用户（或  B2B 直销客户），类型“C”表示零售用户（或消费者直销客户）。关于注册用户和非注册用户的更多信息，请参阅 WebSphere Commerce 开发联机帮助中的“成员”。
- **成员组**。这是为了不同商业原因分类而成的用户组。分组可以用于访问控制目的、核准目的，以及市场营销目的（例如计算折扣、价格和显示产品）。

每个商店实体（即，商店或商店组）由一个成员拥有。

## 成员属性

WebSphere Commerce 成员具有一系列属性，每个属性具有一个与其关联的值。成员的基本用户概要文件包含了注册信息、情况调查、地址信息、购买历史和其它杂项属性。

商业用户概要文件包含与基本用户概要文件相同的信息，另外还有职业信息（例如员工编号或职务，或者工作描述）。在注册期间，商业用户应当标识他们所属的商业组织。组织实体的概要文件包含诸如组织名称和业务类别之类的附加信息。

访问控制规则强制实施了执行概要文件管理的用户权限。成员概要文件可以包含各种个人和业务相关的属性（例如角色、支付信息、地址、首选语言和货币以及普及计算设备）。属性可以是区别商店的。对于用户和组织实体（不是成员组），这些属性是受支持的。

## 角色

每个用户可以在组织中执行一个或多个角色。站点管理员对每个成员指定一个或多个角色。例如，作为 IBM 组织的一个成员，John Smith 的角色是客户服务代表，这意味着 John 代表 IBM 客户来执行任务，并帮助他们查询或关注其注册信息、订单或退货。John 还可能具有客户服务主管的角色，该角色对上述任务负全责，并具有核准和监督其它客户服务代表的权力。

WebSphere Commerce 系统提供了以下一组缺省角色类型：

- 业务关系角色
- 客户服务角色
- 市场营销角色
- 运营角色

- 组织管理角色
- 产品管理和销售策略角色
- 技术操作角色

关于这些角色中的每一个角色的详细信息，请参阅 **WebSphere Commerce** 联机帮助的主题“角色”。站点管理员可以按组织实体指定这些角色以及任何由该站点管理员创建的新角色；即，属于一个组织实体的用户可以担当指定给该组织实体的角色。

当为用户指定了角色时，该角色将在组织实体范围内起作用。这可以是任何组织实体；不必是用户的上级之一。然而，因为角色是继承的，所以用户将在为其分配角色所在组织的所有子代中扮演已分配的角色。例如，如果授予用户根组织中的角色，则用户将对所有组织实体扮演该角色。

**WebSphere Commerce** 角色可以通过组织管理控制台手工分配，也可以通过注册和会话管理命令自动分配。该自动角色分配基于 `MemberRegistrationAttributes.xml` 文件中指定的配置。**WebSphere Commerce 5.5** 提供 `MemberRegistrationAttributes.xml` 文件，可以修改该文件使其符合特定注册的要求。关于自动角色分配以及 `MemberRegistrationAttributes.xml` 文件的更多信息，请参阅 **WebSphere Commerce** 开发联机帮助的主题“`MemberRegistrationAttributes XML` 和 `DTD` 文件”。



关于 **WebSphere Commerce** 中成员有用资源的结构更多详细信息，请参阅 **WebSphere Commerce** 开发联机帮助中的成员对象和数据模型。

---

## 了解 **WebSphere Commerce** 中的客户有用资源

客户是 **WebSphere Commerce** 中的用户。客户可以浏览商店的在线产品目录、下订单、创建兴趣列表、建立地址（如一般联系地址、帐单地址以及送货地址）以及从商店或卖方处购物。客户也是用户。下图说明了客户从商店中下订单所需要的有用资源。

如前图中所示，**WebSphere Commerce** 系统包含成员。可以为每个用户和组织实体成员分配一个角色。

**注：**在 **WebSphere Commerce** 中，成员可以是组织实体、用户或成员组。请参阅第 116 页的『成员』以获取更多详细信息。

在这种情况下，用户是客户。客户必须提供地址信息，并可以拥有兴趣商品列表。该图说明了成员（客户）和与其关联的客户有用资源之间的双向关系：客户必须拥有并提供一个地址且可以拥有一个兴趣列表以在商店购物；而地址和兴趣列表取决于客户的存在。

### 地址信息

当客户从商店购买商品时，他必须提供三种类型的地址信息：联系地址、开票地址和送货地址。下面描述了这三种地址类型；每个地址可以是唯一的或者是相同的：

- 联系地址用于为了不同的目的通知客户，例如关于订单的状态或对订单的更改，以及关于即将进行的商店事件（例如促销或商店维护）的通知。客户的联系地址包含街道名称和门牌号码、市/县/区、省/直辖市、邮政编码、国家或地区、电子邮件地址、电话号码和传真号码。通常情况下，联系地址是最容易联系上客户的地址，例如工作地址。

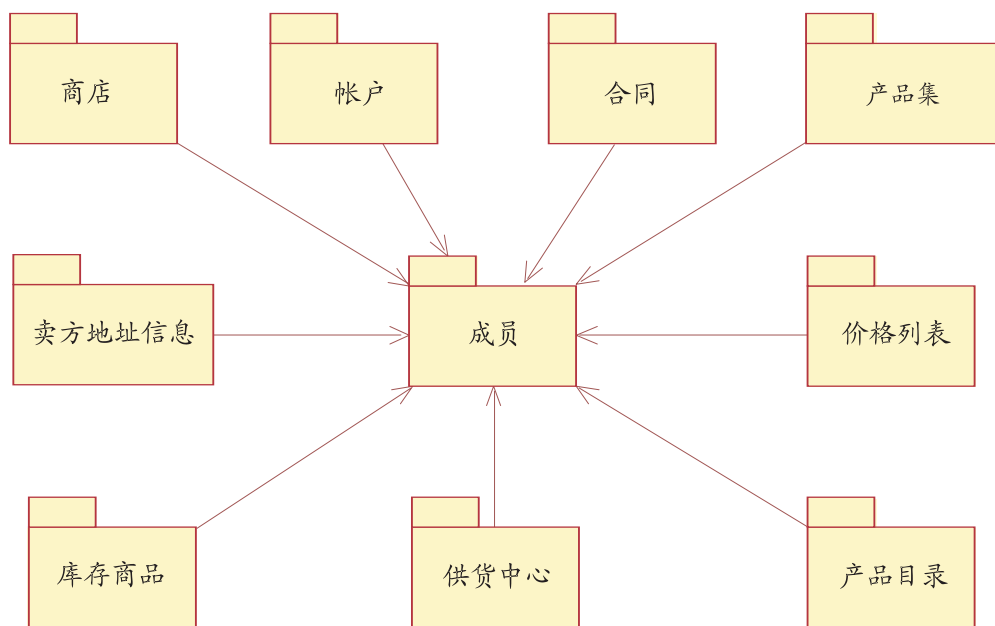
- 开票地址用于发送所购买商品的帐单或发票。开票地址包含街道名称和门牌号码、市/县/区、省/直辖市、邮政编码、国家或地区、电话号码和电子邮件地址。开票地址可以与联系地址或送货地址相同，也可以不同。
- 送货地址用于交付所购买的商品。送货地址包含街道名称和门牌号码、市/县/区、省/直辖市、邮政编码、国家或地区、电话号码和电子邮件地址。送货地址可以与联系地址或开票地址相同，也可以不同。

## 兴趣列表

商店可以支持兴趣列表。即，客户将未来可能希望订购的产品添加到他们的兴趣列表。兴趣列表不是购物车；兴趣列表可以包含来自多个商店的商品，并且不包含价格、送货地址、送货方式、库存可用性信息或计算的金额（例如折扣、送货费用和税款）。

## 了解 WebSphere Commerce 中的卖方有用资源


卖方是 WebSphere Commerce 中的用户。卖方除了跟踪商店销售情况之外，还监督商店的整体目标和管理。卖方向客户销售商品和服务。卖方角色等同于商家，他对所有 WebSphere 贸易加速器功能都有访问权。下图说明了卖方维护商店以及向客户销售时所需要的有用资源。



本图和商店数据部分中的其它所有图都是 WebSphere Commerce Server 信息模型的一部分。关于信息模型的更多信息，请参阅第 97 页的『商店数据信息模型』。关于本图中所使用的约定的更多信息，请参阅第 415 页的附录 A，『UML 图注』。

如前图中所示，WebSphere Commerce 系统包含成员。每个成员被分配一个角色，例如商店的客户服务代表或仓库的收货员。为了向客户销售，卖方角色可以维护以下有用资源：

- 商店

-  帐户（可选）
- 合同（或至少是 WebSphere Commerce 缺省合同）
- 产品集
- 价格列表
- 产品目录
- 供货中心
- 库存商品

前图说明了成员（卖方）和卖方有用资源之间的关系，即：卖方可以具有以上列出的有用资源以维护商店，而这些有用资源需要有一个卖方来部署它们。

## 商店

WebSphere Commerce 网上商店包括一系列 HTML 和 JavaServer Pages 文件，以及税款、送货费用、支付、产品目录和其它数据库有用资源（它们包含在商店归档文件中）。商店还包含商店数据，此数据是填充到 WebSphere Commerce 数据库中以使商店能运作的信息。

关于 WebSphere Commerce 商店的更多信息，请参阅第 123 页的第 13 章，『商店有用资源』和第 107 页的第 6 部分，『开发商店数据』。

## 帐户

 Business

商店可以设置客户的商业帐户以允许他们从商店中购买商品。帐户包含以下信息：

- 帐户名称，它通常是此客户关联的组织名称。此组织已与商店之间签订了合同，规定了有关客户在商店中购物的条款。例如，组织 IBM 可能具有与“ABC 办公用品公司”的合同。
- 代表名称，这是在负责此帐户的卖方组织内部的代表组织名称。
- 属于此帐户的合同的数目。

关于 WebSphere Commerce 帐户的更多信息，请参阅第 178 页的『帐户（商业帐户）』和 WebSphere Commerce 联机帮助。

## 合同

通常情况下，在 WebSphere Commerce 中，所有的客户都必须在合同下进行购物。客户和卖方之间的每个帐户都必须与一个或多个合同关联（为了使未注册客户或客户能在商店购物，或为了客户能购买其它合同未涵盖的产品，至少要关联一个缺省合同）。合同让客户能够在合同中规定的条款和条件及业务策略下，在指定的时间内以指定的价格从商店购买产品。卖方部署合同，以便客户可以从商店购买商品。

合同中的买方可以是用户、组织或成员组。如果是用户，则买方看作是客户。如果是在合同中定义为买方的组织，则该组织的任何子对象都可以作为合同的买方。如果是成员组，则该成员组中的任何用户都可以作为合同的买方。

关于卖方可以使用的 WebSphere Commerce 合同和缺省合同的更多信息，请参阅第 179 页的『合同』。

## 产品集

产品集为卖方提供了一种机制将在线产品目录分类成逻辑子集，以便卖方可以让不同的客户使用不同的产品目录视图。此外，卖方还可以为客户创建合同，并规定客户仅可以在预定义的产品集下购买产品。

关于 WebSphere Commerce 产品集的更多信息，请参阅第 144 页的『产品集』。

## 价格列表

价格列表与卖方提供的价格关联，或与向客户显示的价格关联。对于同样的产品，卖方可以对不同的客户列出不同的价格。在 WebSphere Commerce 中，报价也称为贸易状态，它代表一个产品目录条目的价格以及客户为了有资格使用该价格而必须满足的条件。

在 WebSphere Commerce 中，报价对象是成员所拥有的 TradingPositionContainer 的组成部分。TradingPositionContainer 包含 TradingPosition，并可以通过贸易协议或合同对所有的客户都可用，或者仅对某组中的客户可用。有时候，TradingPositionContainer 称为价格列表。有两种价格列表：标准价格列表（它包含商店产品目录中产品的基本价格）或定制价格列表（它指定了产品的列表及其定制的价格）。

关于 WebSphere Commerce 价格列表的更多信息，请参阅第 169 页的第 17 章，『对有用资源进行定价』。

## 产品目录

WebSphere Commerce 商店至少使用一个在线产品目录来陈列卖方为销售而提供的商品和服务。通常情况下，一个在线产品目录包含待售的商品的价格、图像和描述。在线产品目录还可将商品显示为不同的类别以为浏览提供便利。

WebSphere Commerce 系统中的每个商店都必须具有主产品目录（它用于产品目录的管理）。主产品目录是管理卖方商品的中心位置；它是一个单一产品目录，包含所有产品、商品和关系以及商店中待售的一切物品的标准价格。如果卖方有多个商店，则主产品目录可以在这些商店之间共享。

关于 WebSphere Commerce 产品集的更多信息，请参阅第 141 页的第 16 章，『产品目录有用资源』。

## 供货中心

商店将供货中心同时作为库存仓库以及送货和接收中心来使用。卖方可能具有一个或多个供货中心。

从 WebSphere Commerce 服务器的角度看，FulfillmentCenter 对象是独立于 Store 对象的。它管理产品库存和送货。要根据一个订单送货，供货中心依靠由客户指定的 ShippingMode 对象。ShippingMode 对象为了对订单供货而指示递送者和送货的方法。在供货中心中，ShippingArrangement 对象指示 Store 对象已安排好与 FulfillmentCenter 对象一起使用某个 ShippingMode 来发送产品。

关于 WebSphere Commerce 供货中心的更多信息，请参阅第 193 页的第 19 章，『供货有用资源』和第 223 页的第 26 章，『送货有用资源』。

## 库存商品

库存商品包含了可以在卖方供货中心中实际计算的任何东西。WebSphere Commerce 系统定义了可以提供的特定库存类型，例如商品、产品、库存标识、捆绑销售商品以及打包销售商品；但是这些是所有被考虑在内的库存。产品通过使用 WebSphere 贸易加速器的“产品管理”工具进行配置来供货。

关于 WebSphere Commerce 库存商品的更多信息，请参阅 WebSphere Commerce 开发联机帮助和 第 259 页的第 29 章，『库存有用资源』。

---

## 了解 WebSphere Commerce 中的管理员有用资源

管理员只是具有已分配角色的用户或成员，这些角色使他们能够执行某些管理活动。关于可以与管理员相关联的有用资源的更多详细信息，请参阅第 115 页的『了解 WebSphere Commerce 中的成员有用资源』。

---

## 在 WebSphere Commerce 中创建成员有用资源

要创建卖方（作为商店所有者的组织）并保存有关卖方的信息，请使用 WebSphere Commerce 管理控制台。关于更多信息，请参阅 WebSphere Commerce 开发联机帮助的主题“创建组织”。

要创建管理员，请使用 WebSphere Commerce 管理控制台来创建用户，然后将期望的角色分配给该用户。关于更多信息，请参阅 WebSphere Commerce 开发联机帮助的主题“创建用户”和“按用户专有名称分配角色”。

客户不是由商店开发者创建的；当客户在商店中注册时，注册信息将由 WebSphere Commerce 系统收集并保存。

随 WebSphere Commerce 提供的每个样本商店都包含其自身版本的 MemberRegistrationAttributes.xml 文件，该文件用于为注册和会话管理命令配置自动角色分配。如果您选择修改组织结构或对角色分配有特定的要求，则您必须要修改该文件。关于该文件以及如何配置该文件使其符合需要的更多信息，请参阅 WebSphere Commerce 开发联机帮助中的主题“MemberRegistrationAttributes XML 和 DTD 文件”。





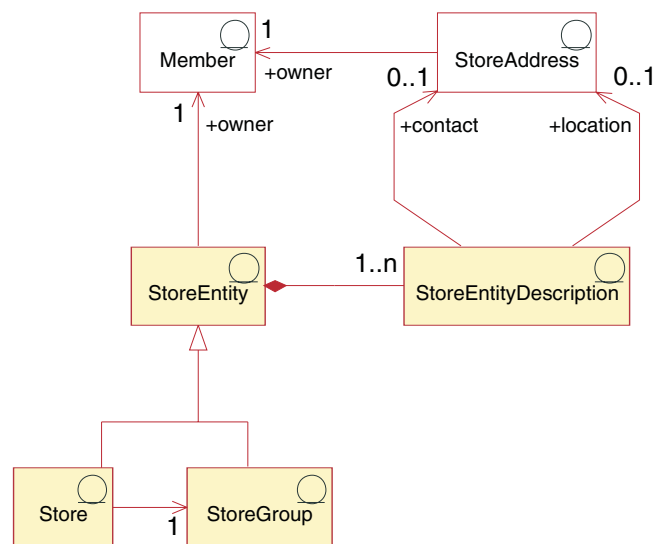
## 第 13 章 商店有用资源

要在 WebSphere Commerce 中创建商店，必须首先在数据库中创建以下内容：

- 商店
- 商店所属的组
- 双重代表商店或商店组的抽象商店实体对象

### 了解 WebSphere Commerce 中的商店有用资源

下图说明了 WebSphere Commerce Server 中的商店有用资源。



本图和商店数据部分中的其它所有图都是 WebSphere Commerce Server 信息模型的一部分。关于信息模型的更多信息，请参阅第 97 页的『商店数据信息模型』。关于本图中所使用的约定的更多信息，请参阅第 415 页的附录 A，『UML 图注』。

### 商店实体

商店实体是可以代表商店或商店组的抽象超类。

商店实体有一个所有者（成员）。关于成员的更多信息，请参阅第 115 页的『了解 WebSphere Commerce 中的成员有用资源』。

#### 商店实体描述

商店实体描述描述了商店实体。商店实体可能包含描述。如果您的商店支持多种语言，则商店实体描述可能使用多种语言。描述可以包括商店实体的联系地址以及商店实体的位置地址。

## 商店

商店是商店实体。商店必须属于某个商店组。

## 商店组

商店组是商店的集合。商店组是商店实体。商店组作为公共信息的容器，可以商店组级别存储并由商店组中的所有商店共享。例如，同一商店组中的商店可以共享诸如税类别、支持的语言、支持的货币、计算代码和送货地区等信息。

当前，WebSphere Commerce Server 中在站点管理级别上仅可存在一个商店组并对其进行维护。



---

关于 WebSphere Commerce Server 中商店有用资源结构的更多详细信息，请参阅 WebSphere Commerce 联机帮助中的商店对象和数据模型。

---

## 在 WebSphere Commerce 中创建商店有用资源

WebSphere 贸易加速器中的商店工具使您可以创建或编辑以下商店有用资源：

- 合同有用资源中的商店标识和成员标识
- STOREENT 表中的商店标识
- STORE 表中的商店目录
- STADDRESS 表中的地址昵称
- 商店描述
- 商店地址

结果，您有两个选项可用于创建商店有用资源：

- 编辑随 WebSphere Commerce 提供的其中一个样本商店的现有商店有用资源。
- 以 XML 文件形式创建商店有用资源，这些资源可以作为商店归档文件的一部分发布，或者可以使用装入程序软件包装入。

关于以 XML 文件形式创建商店有用资源的信息，请参阅『在 XML 文件中创建商店数据有用资源』。关于使用 WebSphere 贸易加速器编辑商店的更多信息，请参阅 WebSphere Commerce 产品联机帮助。

## 在 XML 文件中创建商店数据有用资源

以 XML 文件（可以使用装入程序软件包装入数据库）格式创建商店有用资源。如果正在创建全球化商店，则可能要为商店支持的每种语言环境创建各自的 XML 文件。特定于语言环境的文件应当指定所有描述信息，以便比较容易进行翻译。关于装入程序软件包的更多信息，请参阅第 309 页的第 10 部分，『发布商店』。

样本商店（这些任务中的很多示例是从其中获取的）对不需要翻译的所有信息使用一个 store.xml 文件，对需要翻译的信息为商店支持的每种语言环境使用另一个 store.xml 文件。特定于语言环境的文件包含所有描述信息。

要创建商店有用资源，请执行以下操作：

1. 请复查第 367 页的第 38 章，『装入 WebSphere Commerce 数据库有用资源组』中的信息。

2. 复查用于为样本商店创建商店有用资源的 XML 文件。样本商店的所有文件都位于相应的商店归档文件中。

商店归档文件位于以下目录中:

- `WC_installdir/samplestores`

**注:** 《WebSphere Commerce 样本商店指南》包含关于样本商店中所包含的每个数据有用资源的信息。

每家样本商店包括几个 `store.xml` 文件, 这些文件包括按语言提供的商店信息。由于样本商店已翻译成多种语言, 所以每家商店中将存在多个 `store.xml` 文件。要查看商店归档文件中的 `store.xml` 文件, 请使用 ZIP 程序将商店归档文件解压缩。`store.xml` 文件位于数据目录中。特定于语言的 `store.xml` 位于数据目录的特定于语言环境的子目录中。

3. 请复查第 417 页的附录 B, 『创建数据』中的信息。
4. 通过复制样本商店归档文件中的 `store.xml` 文件之一或通过创建新文件创建 `store.xml` 文件。关于更多信息, 请参阅 `wcs.dtd` 文件。DTD 文件在以下目录中:

- `WC_installdir/schema/xml`

5. 创建商店实体。

- a. 使用以下示例作为指南, 在您的 XML 文件中为 `STOREENT` 表定义商店实体。

```
<storeent
  storeent_id="@storeent_id_1"
  member_id("&MEMBER_ID"
  type="S"
  identifier="ToolTech"
  setccurr="USD"
/>
```

其中

- `storeent_id` 是生成的唯一键。
- `member_id` 是商店实体的所有者。
- `type` 是商店实体的类型: G = 商店组, S = 商店。
- `identifier` 是与所有者一起唯一标识商店实体的字符串。
- `setccurr` 是商店实体的缺省货币, 换句话说, 是将由尚无首选货币的客户使用的货币。如果其对某商店为 `NULL`, 则缺省货币从其商店组中获取。

6. 创建商店地址。

- a. 使用以下示例作为指南, 在您的 XML 文件中为 `STADDRESS` 表创建商店地址。如果您正创建全球化商店, 则应当将此信息包含在特定于语言环境的 XML 文件中。

```
<staddress
  staddress_id="@staddress_id_en_US_1"
  member_id("&MEMBER_ID"
  nickname="storeaddress_English"
  address1="12xx Martindale Avenue"
  address2="Suite 9xx"
  businesstitle="ToolTech"
  city="Toolsville"
  state="Ontario"
  zipcode="Lxx 1xx"
  country="Canada"
```

```
phone1="1-800-555-1234"  
fax1="1-800-555-4321"  
email1="info@tooltech.xxx"  
</>
```

其中

- staddress\_id 是生成的唯一键。
- member\_id 是商店实体的所有者。

## 7. 创建商店实体的描述。

- a. 使用以下示例作为指南，在您的 XML 文件中为 STOREENTDS 表创建商店实体的描述。如果您正创建全球化商店，则应当将此信息包含在特定于语言环境的 XML 文件中。

```
<storeentds  
  description="Commerce Models Store entity"  
  language_id="@en_US"  
  displayname="ToolTech"  
  storeent_id="@storeent_id_1"  
  staddress_id_cont="@staddress_id_en_US_1"  
  staddress_id_loc="@staddress_id_en_US_1"
```

其中

- description 是商店实体的详细描述，适合于对客户显示。
- language\_id 是对在商店中购物的客户显示信息时使用的缺省语言。
- displayname 是商店实体的简短描述，适合于对客户显示。
- storeent\_id 是商店实体。
- staddress\_id\_cont 是商店实体的联系地址。
- staddress\_id\_loc 是商店实体的物理位置。

## 8. 在数据库中创建商店。

- a. 使用以下示例作为指南，在您的 XML 文件中为 STORE 表定义商店。

```
<store  
  store_id="@storeent_id_1"  
  directory="ToolTech"  
  ffmcenter_id="@ffmcenter_id_1"  
    language_id="@en_US"  
  storegrp_id="-1"  
  allocationgoodfor="43200"  
  bopmpadfactor="0"  
  defaultboffset="2592000"  
  ffmselectionflags="0"  
  maxboffset="7776000"  
  rejectedordexpiry="259200"  
  rtnffmctr_id="@ffmcenter_id_1"  
  pricerefflags="0"  
  storetype="B2B"  
</>
```

其中

- store\_id 是生成的唯一键。
- directory 是特定于商店的 Web 有用资源所在的目录。该目录位于 Store.war Web 模块的文档根路径之下。
- ffmcenter\_id 是商店的缺省供货中心。
- language\_id 是对在商店中购物的客户显示信息时使用的缺省语言。

- storegrp\_id 是商店与之关联的商店组。此数字在 STOREGRP 表中生成。
- allocationgoodfor 意味着 ReleaseExpiredAllocations 调度程序作业可以在进行分配后很多秒用于逆转 ATP 库存分配。
- bopmpadfactor 意味着如果该商店在计算订单金额（如税款或送货费用）时对不同供货中心使用不同方法，则先前所提交订单的订单金额可能在最终将供货中心分配到延迟交货的订购商品时发生更改。此填充因子代表提供给 Payment Manager 的订单金额可以增加的百分比（如有必要）。例如，指定 5 允许最多增加百分之五。
- defaultboffset 是无法为延迟交货的 OrderItem 确定估计可提供时间之后的时间，以后将设置为这一秒数。
- maxboffset 意味着如果延迟交货的 OrderItem 的估计可提供时间将来通常超出这一秒数，则将来它会被设置为这一秒数。
- rejectedordexpiry 是支付处于“拒绝”状态的时间长于此秒数的订单，是可以考虑取消的对象。
- rtnffmctr\_id 是将商品退回商店的缺省供货中心。
- pricerefflags 包含位标志，这些位标志控制在 GetContractUnitPrices 任务命令的缺省实现刷新价格时，搜索的是哪些 TradingAgreements 和 Offers:
  - 1 = usePreviousOnly - 使用 OrderItems 引用的内容。如果其再也无法使用则失败。
  - 2 = usePreviousOrSearchAgain - 与 usePreviousOnly 相同，但是再也无法使用时不会失败，而是搜索 ORDIOFFER 和 ORDITRD 表中保存的内容
  - 4 = alwaysSearchAgain - 总是搜索 ORDIOFFER 和 ORDITRD 表中保存的内容。
- storetype 指示以下商店类型之一，以供基于商店类型来提供适当功能的用户界面使用:
  - B2B = B2B 直销
  - B2C = 商家到消费者（消费者直销）
  - CHS = 转销商中心（Commerce Plaza）
  - CPS = 主产品目录概要文件商店（产品目录有用资源商店）
  - RHS = 转销商主管商店
  - RPS = 转销商概要文件商店（转销商商店前台有用资源商店）
  - DPS = 经销商概要文件商店（经销商有用资源商店）
  - DPX = 经销商代理商店
  - HCP = 商业主管中心（主管中心）
  - PBS = 商店目录
  - MPS = 商家概要文件商店（主管商店前台有用资源商店）
  - MHS = 商家主管商店
  - SCP = 供应商中心
  - SPS = 供应商概要文件商店（供应商有用资源商店）
  - SHS = 供应商主管商店

注：以方括号表示的名称是随 WebSphere Commerce 提供的相应样本的名称。

## 9. 定义商店的支持语言。

- a. 使用以下示例作为指南，在 XML 文件中定义商店的支持语言以向 STORELANG 表添加信息。如果您的商店支持多种语言，则应将此信息包含在特定于语言环境的 XML 文件中（商店支持的每种语言各一个）。

```
<storelang
  language_id="&en_US"
  storeent_id="@storeent_id_1"
/>
```

其中

- language\_id 是商店实体支持的语言。
  - storeent\_id 是商店实体。
- b. 使用以下示例作为指南，将关于语言的信息添加到 STORELANGDS 表中。如果您的商店支持多种语言，则应将此信息包含在特定于语言环境的 XML 文件中（商店支持的每种语言各一个）。

```
<storlangds
  description="United States"
  language_id="&en_US"
  storeent_id="@storeent_id_1"
  language_id_desc="& en_US"
/>
```

其中

- description 是语言的简短描述，适合于对选择列表中的客户显示。
- language\_id 是描述的语言。
- storeent\_id 是支持语言的商店实体。
- language\_id\_desc 是被描述的语言。



关于 @ 和 & 用法的更多信息，请参阅第 417 页的附录 B，『创建数据』。

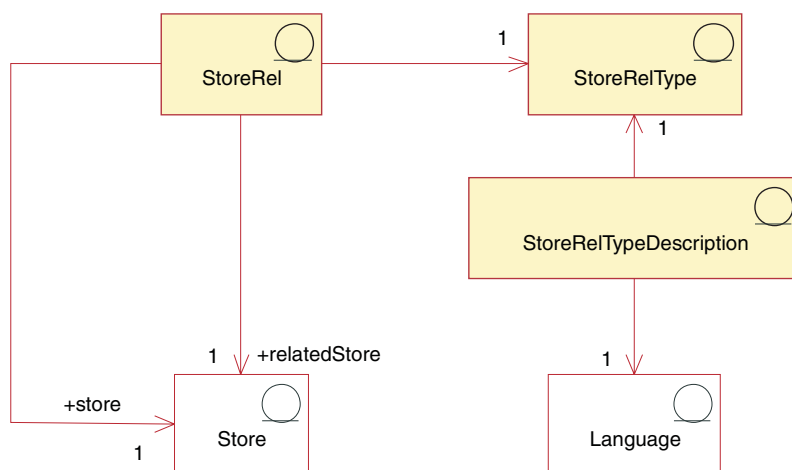
---

## 第 14 章 商店间的关系

**Business** WebSphere Commerce 支持一个站点内商店之间的几种关系。例如，一家商店可能为另一家商店提供主管服务，或者一家商店可能使用由另一家商店所提供的产品目录或货币有用资源。

### 了解 WebSphere Commerce 中商店间的关系

下图举例说明了 WebSphere Commerce Server 中的商店关系。



### 商店关系

商店关系（在 StoreRel 表中捕获）是两家商店之间的关系。所有商店关系都是定向的，也就是说，在每个商店关系中，一家商店提供服务，而该关系中的另一家商店则使用那些服务。例如，商店 A 使用由商店 B 提供的产品目录。

每个商店关系均有一个商店关系类型（StoreRelType）。

### 商店关系类型

商店关系类型（StoreRelType）定义两家商店之间关系的类型。每个商店关系类型均定义自己的关系，也就是说，关系中的每个伙伴将起什么作用，以及两个伙伴之间的关系是什么。

#### WebSphere Commerce 支持的商店关系类型

WebSphere Commerce 支持两家商店之间的多种关系类型。WebSphere Commerce 提供的缺省关系类型可以粗略地分为两类：

- 一家商店向另一家商店提供数据有用资源的情况下的关系。例如，商店 A 提供在商店 B 中使用的产品目录数据。
- 一家商店与另一家商店具有“业务关系”的情况下的关系，也就是说，一家商店可能主管另一家商店，或者一家商店可能将购物车传送到另一家商店。

一家商店向另一家商店提供数据有用资源的情况下的关系:

表 8.

关系类型	描述	关于更多信息, 请参阅
com.ibm.commerce.businessPolicy	一家商店使用在另一家商店中定义的业务策略。	第 177 页的第 18 章, 『合同有用资源』
com.ibm.commerce.campaigns	一家商店使用在另一家商店中定义的竞销。	第 199 页的第 20 章, 『竞销有用资源』
com.ibm.commerce.catalog	一家商店使用在另一家商店中定义的产品目录数据。	第 141 页的第 16 章, 『产品目录有用资源』
com.ibm.commerce.command	一家商店使用在另一家商店中定义的命令。	第 135 页的第 15 章, 『命令、视图和 URL 注册表数据』
com.ibm.commerce.price	一家商店使用在另一家商店中定义的价格数据。	第 169 页的第 17 章, 『对有用资源进行定价』
com.ibm.commerce.segmentation	一家商店使用在另一家商店中定义的客户概要文件数据。	第 273 页的第 32 章, 『客户概要文件』
com.ibm.commerce.URL	一家商店使用在另一家商店中定义的 URL。	第 135 页的第 15 章, 『命令、视图和 URL 注册表数据』
com.ibm.commerce.view	一家商店使用在另一家商店中定义的视图。	第 135 页的第 15 章, 『命令、视图和 URL 注册表数据』
com.ibm.commerce.storeitem	一家商店使用在另一家商店中定义的商品。	第 259 页的第 29 章, 『库存有用资源』
com.ibm.commerce.propertyFiles	一家商店使用在另一家商店中定义的属性文件。	
com.ibm.commerce.currency.conversion	一家商店使用在另一家商店中定义的货币兑换率。	第 211 页的第 23 章, 『货币有用资源』
com.ibm.commerce.currency.supported	一家商店使用在另一家商店中支持的货币。	第 211 页的第 23 章, 『货币有用资源』
com.ibm.commerce.currency.format	一家商店使用在另一家商店中定义的货币格式。	第 211 页的第 23 章, 『货币有用资源』
com.ibm.commerce.currency.countervalue	一家商店使用在另一家商店中定义的货币对应值。	第 211 页的第 23 章, 『货币有用资源』
com.ibm.commerce.measurement.format	一家商店使用在另一家商店中定义的计量单位。	第 217 页的第 24 章, 『计量单位有用资源』

一家商店可以与多家商店有关系。也就是说, 商店 A 可能想要使用来自商店 B、C 和 D 的产品目录资源。为了便于实现这种多家商店之间的关系, 您必须在某家商店使用有用资源时提供这些有用资源所属的商店的序列顺序。在商店之间的关系中排序按以下方式进行:

- 覆盖: 如果商店关系遵循覆盖排序方法, 则序列号最小的商店关系就是所使用的商店关系。以下商店关系使用覆盖方法:
  - 命令
  - 货币



- 计量
- 价格
- 属性文件
- 商店商品
- URL
- 视图
- 合并: 如果商店关系遵循合并排序方法, 则 WebSphere Commerce 寻找与该商店相关的所有商店关系, 并合并来自所有关联商店的数据。以下商店关系使用合并方法:
  - 业务策略
  - 竞销
  - 产品目录
  - 分段

所有缺省商店关系类型均指定为使用覆盖排序方法或合并排序方法。

**注:** 虽然对于合同不存在商店关系类型, 但单个合同可以对多家商店部署。关于更多信息, 请参阅 WebSphere Commerce 产品和开发联机帮助。

**一家商店与另一家商店具有“业务关系”的情况下的关系:**

表 9.

关系类型	描述
com.ibm.commerce. hostedStore	中心商店主管转销商、供应商或主管商店。
com.ibm.commerce. referral	中心商店与经销商之间存在分派关系。中心商店可以将购物车转移经销商商店。通常接收购物车的商店是外部系统的代理商店。
com.ibm.commerce. channelStore	一家商店担当另一家商店的中心商店。这种关系定义商店目录和主管中心之间的关系。

## 商店关系类型描述

商店关系类型描述是要描述关系的类型。每个商店关系类型描述仅描述一个关系类型。商店关系类型描述可以一种以上语言提供。

---

## 创建 WebSphere Commerce 中的商店关系

以可使用装入程序软件包装入到数据库中的 XML 文件格式创建您的商店关系。关于装入程序软件包的更多信息, 请参阅第 309 页的第 10 部分, 『发布商店』。

**注:** 如果您使用“商店创建”向导创建主管商店(关于更多信息, 请参阅第 66 页的『“商店创建”向导』)或使用服务协议创建经销商代理商店(关于更多信息, 请参阅第 67 页的『创建代理商店』), 则将为您创建许多这些商店关系。

要创建商店关系有用资源, 请执行以下操作:

1. 请复查第 367 页的第 38 章,『装入 WebSphere Commerce 数据库有用资源组』中的信息。
2. 复查用于为样本商店创建商店有用资源的 XML 文件。样本商店的所有文件都位于相应的商店归档文件中。

商店归档文件位于以下目录中:

- `WC_installdir/samplestores`

**注:** 《WebSphere Commerce 样本商店指南》包含关于样本商店中所包含的每个数据有用资源的信息。

3. 请复查第 417 页的附录 B,『创建数据』中的信息。
4. 创建 `storerelation.xml` 文件,创建方法可以是复制样本商店归档文件中的其中一个 `storerelation.xml` 文件,也可以是创建新文件。关于更多信息,请参阅 `wcs.dtd` 文件。DTD 文件在以下目录中:
  - `WC_installdir/schema/xml`
5. 创建商店关系。
  - a. 使用以下示例为指南,在您的 XML 文件中为 `STOREREL` 表定义商店实体。

```
<storere1
store_id="@storeent_id_1"
relatedstore_id="@storeent_id_2"
streltype="-4"
sequence="0"
state="1"
/>
```

其中

- `store_id` 是使用相关商店的服务的主商店。
- `relatedstore_id` 是提供主商店所使用的服务的商店。
- `streltype` 是关系的类型。缺省关系类型如下:
  - -1 `com.ibm.commerce.businessPolicy`
  - -3 `com.ibm.commerce.campaigns`
  - -4 `com.ibm.commerce.catalog`
  - -5 `com.ibm.commerce.command`
  - -6 `com.ibm.commerce.hostedStore`
  - -7 `com.ibm.commerce.price`
  - -8 `com.ibm.commerce.referral`
  - -9 `com.ibm.commerce.segmentation`
  - -10 `com.ibm.commerce.URL`
  - -11 `com.ibm.commerce.view`
  - -13 `com.ibm.commerce.inventory`
  - -14 `com.ibm.commerce.storeitem`
  - -15 `com.ibm.commerce.channelStore`
  - -16 `com.ibm.commerce.propertyFiles`
  - -17 `com.ibm.commerce.currency.conversion`
  - -18 `com.ibm.commerce.currency.format`

- -19 com.ibm.commerce.currency.supported
- -20 com.ibm.commerce.currency.countervalue
- -21 com.ibm.commerce.measurement.format
- **sequence** 定义了在为同一关系类型定义多家相关商店时的选择顺序。缺省值为 0。
- **state** 是关系的状态 (0 = 不活动, 1 = 活动)。缺省值为 1。



关于 @ 和 & 用法的更多信息, 请参阅第 417 页的附录 B, 『创建数据』。

---



## 第 15 章 命令、视图和 URL 注册表数据

命令、视图和 URL 注册表是 WebSphere Commerce 命令框架的一部分，命令框架在《WebSphere Commerce 编程指南与教程》的第一章『概述』、第二章『设计模式』和第六章『命令实现』中有更详细的描述。为了理解命令、视图和 URL 注册表是如何符合信息模型的，我们在这里提供一个简要概述。

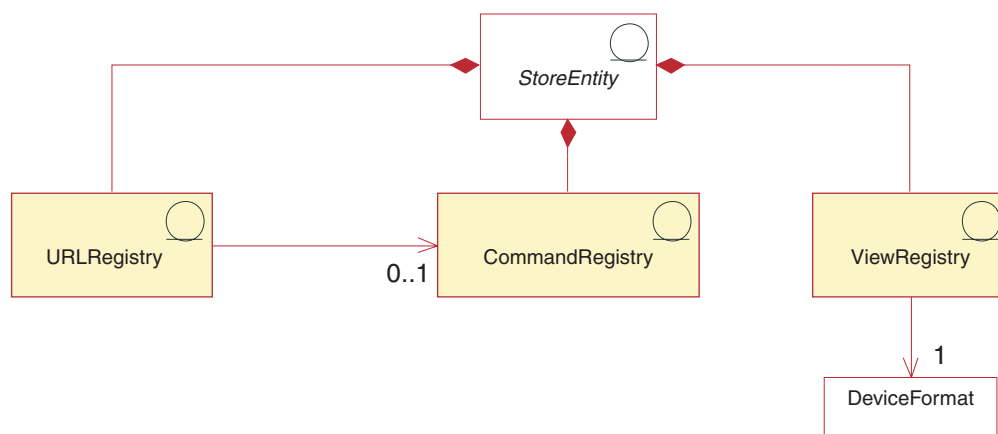


关于 WebSphere Commerce Server 中命令和视图有用资源结构的更多详细信息，请参阅 WebSphere Commerce 联机帮助中的命令和视图数据模型。

### 了解 WebSphere Commerce 中的命令、视图和 URL 注册表

WebSphere Commerce 命令框架决定命令如何执行，然后根据已执行命令返回的视图来返回响应。命令执行和响应是与商店相关的，这意味着对每个商店，同样的命令可以以不同方式实现，同时对每个商店返回不同的响应。

下图例举了 WebSphere Commerce Server 中命令、视图和 URL 注册表的结构。



### URL 注册表

URL 注册表把命令名映射到要执行的命令的实际接口上。每个 URL 注册表条目根据商店的不同而不同，即，每个商店都可为同样的 URL 值定义不同的接口。如果无法找到 URL 注册表的商店版本，则使用为站点（商店 0）定义的 URL 注册表。缺省情况下，为站点定义了所有 URL 注册表。

**Business** 在一家商店中定义和注册的 URL 可以由其它商店使用。为了使一家商店使用另一家商店中定义的 URL，必须在商店之间创建类型为 `com.ibm.commerce.URL` 的商店关系。关于更多信息，请参阅第 129 页的第 14 章，『商店间的关系』。

## 命令注册表

每个命令都可在命令注册表中定义，不管它是控制器还是任务命令。如果命令在命令注册表中定义，则命令执行时该定义会用作命令实现。如果命令没有定义在命令注册表中，则取而代之将使用缺省实现。每个命令接口分配到一个缺省实现，如果命令没有在命令注册表中定义时将使用该缺省实现。

如果命令作为站点级别命令（商店 0）定义在命令注册表中，则将使用站点级别实现，除非命令执行的对象是某个已定义了不同命令实现的商店。

命令注册表允许不同的商店使用同样的命令，但在不更改源命令流的情况下要扩展部分或全部的实现。

**Business** 一家商店中定义和注册的命令可以由其它商店使用。为了使一家商店使用另一家商店中定义的命令，必须在商店之间创建类型为 `com.ibm.commerce.command` 的商店关系。关于更多信息，请参阅第 129 页的第 14 章，『商店间的关系』。

## 视图注册表

在大多数情况下，命令执行后，命令请求程序需要返回一个响应。在确定响应时，命令框架会考虑如下因素：

- 命令执行后在响应属性中找到的视图。
- 将执行命令的商店。
- 发出请求时请求的设备格式。

返回响应的每个视图必须在视图注册表中定义，或者按每个商店定义，或者在缺省情况下按站点定义。每个商店通常都会为传入请求可能有的每个设备格式定义视图。然而，如果商店没有定义某视图，则使用站点的缺省视图。处理请求的适配器在确定调用哪个视图时决定要使用的设备格式和缺省设备格式。由于没有一个类属设备格式，因此视 WebSphere Commerce 可接收的不同请求类型，可能会有为每个设备格式定义的视图。

**Business** 一家商店中定义和注册的视图可以由其它商店使用。为了使一家商店使用另一家商店中定义的视图，必须在商店之间创建类型为 `com.ibm.commerce.view` 的商店关系。关于更多信息，请参阅第 129 页的第 14 章，『商店间的关系』。

## 创建新命令、视图和 URL

创建 WebSphere Commerce Server 实例时，随 WebSphere Commerce 提供的缺省命令、视图和 URL 注册在 WebSphere Commerce Server 数据库相应的表（`CMDREG`、`VIEWREG` 和 `URLREG`）中。这些命令、视图和 URL 可以在驻留于实例中的所有商店内使用。

WebSphere Commerce 还提供显示缺省视图的缺省 JSP 文件。这些 JSP 文件与 `VIEWREG` 表中的视图关联。

如果创建新的命令、视图或 URL，或者定制现有的命令、视图或 URL，您必须在相应数据库表（`CMDREG`、`VIEWREG` 和 `URLREG`）中对其进行注册，才可在商店中使用。如果创建新的 JSP 文件以供在商店中使用，则必须使其与 `VIEWREG` 表中的相应视图关联。

**注:** 如果创建新的 JSP 文件, 但赋予其与关联视图的缺省 JSP 文件相同的名称, 则您不需要在 VIEWREG 表中注册新的 JSP 文件。

**注:** 创建新的视图时, 请确保将访问控制策略与每个新视图相关联。关于更多信息, 请参阅第 280 页的『向商店添加访问控制』。

关于创建或定制命令、视图和 URL 的更多信息, 请参阅《WebSphere Commerce 编程指南与教程》。《WebSphere Commerce 编程指南与教程》还包含关于命令、视图、URL 以及 JSP 文件的注册方法和时间的信息。

---

## 在 WebSphere Commerce 中注册命令、视图和 URL

如果为商店创建或定制多个新的命令、视图、URL 或 JSP 文件, 则您可能希望使用 XML 文件注册它们, 然后可以使用装入程序软件包将此文件装入数据库, 或者作为可使用管理控制台中的发布实用程序发布的商店归档文件的一部分装入数据库。关于装入程序软件包的更多信息, 请参阅第 309 页的第 10 部分, 『发布商店』。

**注:** 创建 XML 文件装入新的或定制的命令之前, 请参阅《WebSphere Commerce 编程指南与教程》以获取关于命令如何工作的更多详细信息。

### 创建 XML 文件以注册命令、视图和 URL

要创建 XML 文件以注册商店新的命令、视图和 JSP 文件, 请执行以下操作:

1. 请复查第 367 页的第 38 章, 『装入 WebSphere Commerce 数据库有用资源组』中的信息。
2. 复查用于注册样本商店的命令、视图、JSP 文件的 XML 文件。每个样本商店包括一个 command.xml 文件, 其中包含注册信息。商店归档文件位于以下目录中:
  - WC\_installdir/samplestores

**注:** 《WebSphere Commerce 样本商店指南》包含关于样本商店中所包含的每个数据有用资源的信息。

要查看商店归档文件的内容, 请使用解压缩程序。command.xml 文件位于 data 目录中。

3. 请复查第 417 页的附录 B, 『创建数据』中的信息。
4. 创建 command.xml 文件, 创建方法可以是复制样本商店归档文件中的一个 command.xml 文件, 也可以是创建新的文件。关于更多信息, 请参阅 wcs.dtd 文件。DTD 文件位于以下目录中:
  - WC\_installdir/schema/xml
5. 控制器命令必须在 URLREG 表和 CMDREG 表中注册。要在 URLREG 表中注册新的或定制的控制命令, 请使用以下示例作为指南在 XML 文件中为每个新的定制控制命令创建一个条目:

```
<urlreg
url="MyProductDisplay"
storeent_id="@storeent_id_1"
interfacename="com.mystore.commerce.catalog.commands.ProductDisplayCmd"
https="0"
description="Product display command for my store"
```

```
authenticated="0"  
internal="0" />
```

其中

- urlreg 是此信息将填充的数据库表（URLREG）的名称。
- url 是 URI 名称
- storeent\_id 是商店实体标识，使用 @ 符号称为内部别名解析。使用内部别名解析时，以别名替换 XML 文档中的主键（标识）。然后在 XML 文件中的其它位置使用此别名来引用该元素。这样就无需了解构建 XML 文件所需的唯一索引。发布期间，标识解析器会以唯一值替换 @ 符号。关于更多信息，请参阅第 417 页的附录 B，『创建数据』。
- interfacename 是控制器命令接口名称
- https 是此 URL 请求所需的安全 HTTP。需要安全 HTTP 时请使用 1，不需要时请使用 0。
- authenticated 是此 URL 请求是否需要用户登录。需要认证时请使用 1，不需要时请使用 0。
- internal 指示命令对于 WebSphere Commerce 是否是内部的。内部的 URL 由 WebSphere Commerce 工具使用。是内部时请使用 1，是外部时请使用 0。您创建的 URL 必须是外部的。

6. 要在 CMDREG 表中注册新的控制器命令或新的任务命令，请使用任务命令的以下示例（来自“多乐五金店”样本商店的 command.xml 文件）作为指南在 XML 文件中为每个新的或定制的控制或任务命令创建一个条目：

```
< cmdreg  
storeent_id="@storeent_id_1"  
interfacename="com.ibm.commerce.payment.commands.DoPaymentCmd"  
classname="com.ibm.commerce.payment.commands.DoPaymentMPFCmdImpl" />
```

其中

- cmdreg 是此信息将填充的数据库表（CMDREG）的名称。
- storeent\_id 是商店实体标识，使用 @ 符号称为内部别名解析。使用内部别名解析时，以别名替换 XML 文档中的主键（标识）。然后在 XML 文件中的其它位置使用此别名来引用该元素。这样就无需了解构建 XML 文件所需的唯一索引。发布期间，标识解析器会以唯一值替换 @ 符号。关于更多信息，请参阅第 417 页的附录 B，『创建数据』。
- interfacename 是命令接口名称
- classname 是命令实现类名称。通常，此名称是在末尾附加 Impl 的接口名称。

7. 要注册新视图，或使新的 JSP 文件与视图关联，请使用以下示例（来自“多乐五金店”样本商店的 command.xml 文件）作为指南在 VIEWREG 表中创建一个条目：

```
< viewreg  
viewname="OrderOptionsView"  
devicefmt_id="-1"  
storeent_id="@storeent_id_1"  
interfacename="com.ibm.commerce.command.ForwardViewCommand"  
classname="com.ibm.commerce.command.HttpForwardViewCommandImpl"
```



```
properties="docname=Shipping.jsp"  
internal="0"  
https="0" />
```

其中

- `viewreg` 是此信息将填充的数据库表 (VIEWREG) 的名称。
- `viewname` 是视图的名称。
- `devicefmt_id` 是将使用此视图的设备类型 (例如浏览器)。
- `storeent_id` 是商店实体标识, 使用 @ 符号称为内部别名解析。使用内部别名解析时, 以别名替换 XML 文档中的主键 (标识)。然后在 XML 文件中的其它位置使用此别名来引用该元素。这样就无需了解构建 XML 文件所需的唯一索引。发布期间, 标识解析器会以唯一值替换 @ 符号。关于更多信息, 请参阅第 417 页的附录 B, 『创建数据』。
- `interfacename` 是视图命令接口名称。缺省选项为 ForwardView、DirectView 和 RedirectView。
- `classname` 是视图实现类名称。通常, 此名称是在末尾附加 Impl 的接口名称。
- `properties` 是设置为命令的输入属性的缺省名称-值对。如果总是显示同一页面, 请设置此属性中的 JSP 文件名, 例如, `docname=Shipping.jsp`。
- `internal` 指示视图对于 WebSphere Commerce 是否是内部的。内部视图由 WebSphere Commerce 工具使用。是内部时请使用 1, 是外部时请使用 0。您创建的视图必须是外部的。
- `https` 是此 URL 请求所需的安全 HTTP。需要安全 HTTP 时请使用 1, 不需要时请使用 0。



关于 @ 和 & 用法的更多信息, 请参阅第 417 页的附录 B, 『创建数据』。

---



---

## 第 16 章 产品目录有用资源

和传统的产品目录一样，在线产品目录也是由提供待售的商品和服务组成的。尽管在线产品目录的大小和结构在商店与商店之间可能极不相同（取决于可购买商品类型和数量），但产品目录需要以下内容：

- 所销售的商品，包括：
  - 价格，它几乎总是包含在在线产品目录中。
  - 产品数据，如商品的描述和图像。
  - 类别，大部分（但是不全部）产品目录都将商品分类以便客户浏览。
- 所销售商品的显示方法。产品目录显示页面概述客户对页面外观的评价，并且在各产品目录页面之间提供一致的外观。如何构建产品目录取决于商品。

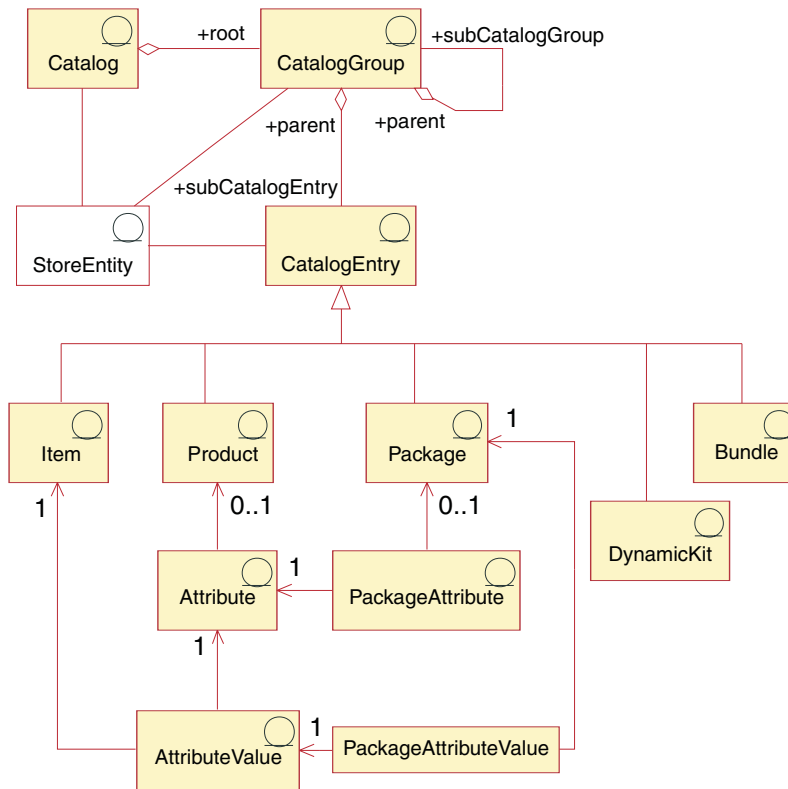
---

### 了解 WebSphere Commerce 中的产品目录

WebSphere Commerce 对商店的在线产品目录有若干要求。WebSphere Commerce 系统中的每个商店都必须具有主产品目录，也简称为产品目录。主产品目录是管理商店商品的中心位置。它是单个目录，其中包含所有产品、商品、关系和商店中所有待售商品的标准定价。

您可以在商店之间共享主产品目录，也可以根据需要定义任意数量的商店。除创建用于产品目录管理的主产品目录以外，您还可以选择创建用于显示的一个或多个销售产品目录。销售产品目录可以包含子集或与主产品目录相同的产品目录条目，但对于客户显示将具有灵活得多的目录结构。主产品目录只能有一个，但您可以按您的希望创建任意多个销售产品目录。但是，由于您需要使用主产品目录管理在线商品，因此我们建议您将主产品目录也用作销售产品目录，从而使维护开销最小化。

如果要为 WebSphere Commerce 商店创建新的主产品目录，或要修改 WebSphere Commerce 样本商店（例如“多乐五金店”）提供的现有主产品目录，则您必须确保产品目录满足这些需求。下图概述了 WebSphere Commerce 中主产品目录的基本结构。



本图和商店数据部分中的其它所有图都是 WebSphere Commerce 信息模型的一部分。关于信息模型的更多信息，请参阅第 97 页的『商店数据信息模型』。关于本图中所使用的约定的更多信息，请参阅第 415 页的附录 A，『UML 图注』。

## 产品目录

产品目录是信息模型中的起点。产品目录包含在线产品目录的所有层次结构和导航信息，是网上商店显示并可购买的产品目录组和产品目录条目的集合。

在 WebSphere Commerce 中，产品目录在数据库中以产品目录实体表示。产品目录实体由唯一的产品目录标识和产品目录的描述（例如产品目录名称）组成。由于每个产品目录都是独立的唯一实体，因此可以很容易与一个或多个商店关联。WebSphere Commerce 系统中的每个商店必须至少与一个产品目录实体相关。

## 产品目录组

产品目录组是产品目录条目的一般分组，是出于划分目的创建的。产品目录组属于产品目录，并可以包含多个产品目录组或产品目录条目。可以使产品目录组与多个产品目录关联。产品目录组也称为类别。

平面产品目录是不将其产品分组为类别的产品目录，而是显示产品列表。尽管可以在 WebSphere Commerce 中创建平面产品目录，但是还是建议您出于结构化和导航的目的创建产品目录组。

创建产品目录组时，必须首先以层次结构或倒排树排列产品目录。树从一般产品目录组（称为根产品目录或顶级产品目录）开始，分支延伸出越来越具体的子类别，直到

无法进一步细分为止。每个仅包含产品的最低级产品目录组为叶。产品目录组是其下类别的父级，是其上类别的子级。例如，男式时装是男式时装类别的分组，而产品目录组长裤和衬衫是产品的分组。

## 产品目录条目

每个产品目录组都包含产品目录条目。在线产品目录中的产品目录条目代表可订购的商品。条目通常具有名称或部件号、描述、一个或多个价格、图像和其它详细信息。产品目录条目可以是产品、商品、捆绑销售商品、打包销售商品、不可改装的成套商品或可改装的成套商品。如果需要的话，您可以创建不适合以下任一现有型号（共六种）的新产品目录条目类型。以下提供了每种类型产品目录条目的更多信息。

### 产品

产品是一种产品目录条目类型。产品充当表现相同属性的商品（或库存标识）组的模板。例如，衬衫是产品目录中的一种产品。将属性和属性值添加到衬衫之后，每个变体都成为商品，如小号黑色衬衫。

### 商品

商品是商品的有形单位，具有特定名称、部件号和价格。例如，小号黑色衬衫是商品，而衬衫是产品。与特定产品相关的所有商品都显示同一组属性，并通过其属性值区分。

**注：**对于 WebSphere 贸易加速器用户，术语“商品”和库存标识视为同义词。使用 WebSphere 贸易加速器中的“产品管理”工具时，可订购商品称为库存标识。在 WebSphere Commerce 数据库模式中，此特定类型的产品目录条目称为商品。

### 捆绑销售商品

捆绑销售商品是产品目录条目的集合，使客户可以一次购买多种商品。例如，计算机的捆绑销售商品可能由中央处理单元、显示器、硬盘驱动器和 CD-ROM 驱动器组成。捆绑销售商品是一组商品，或产品、商品和完全解析的打包销售商品的组合。如果您选择仅包含商品的捆绑销售商品，则此捆绑销售商品分解为独立的可订购库存标识，这些库存标识是单独添加到购物车的。但是，如果您选择包含产品的捆绑销售商品，则需要通过库存标识解析将这些产品解析为商品才能将其添加到购物车。任一情况下一旦分解了捆绑销售商品，且将其组件商品添加到购物车，您就可以修改或删除每个商品。

### 打包销售商品

打包销售商品是产品目录条目的基本集合。例如，计算机打包销售商品可能包含不能单独销售的特定中央处理单元、显示器和硬盘驱动器。与产品类似，打包销售商品具有定义属性，是用于完全解析的打包销售商品的容器。完全解析的打包销售商品可以和库存标识相比。打包销售商品具有自己的价格而且是可以添加到购物车的实际可订购库存标识。浏览期间或将打包销售商品放入购物车之后不能分解或修改打包销售商品。

**注意：**对于 WebSphere 贸易加速器用户，打包销售商品和预置成套商品视作是同义词。使用 WebSphere 贸易加速器中的“产品管理”工具时，打包销售商品称为预置成套商品。在 WebSphere Commerce 数据库模式中，此特定类型的产品目录条目称为打包销售商品。

## 可改装的成套商品

*可改装的成套商品*是一类可由客户动态配置的产品目录条目。此产品配置（或分组）以客户需求为基础并作为单个单元销售。可改装的成套商品的组成部分由外部产品配置器通过一组预定义的规则 and 用户交互作用控制，并在订单登记时提供。将可改装的成套商品添加到订单与添加打包销售商品类似。如同打包销售商品，不能修改可改装的成套商品的单独组件，且整个配置必须作为整体实现。但是，您可以使用外部产品配置器进行重新配置，来更改可改装的成套商品的组件。

## 不可改装的成套商品

*不可改装的成套商品*是作为一个单元进行订购的产品组。关于不可改装的成套商品中所包含产品的信息是预定义的，并在 WebSphere Commerce 内受控。不能修改订单内的单个组成部分，它们必须一起进行处理。如果无法提供不可改装的成套商品中的任何组成部分，那么该成套商品将会延迟交货。

不可改装的成套商品首先创建为打包销售商品，然后由管理员进行配置。

## 产品集

产品集与已发布的产品目录条目相关联。产品集提供将产品目录划分为逻辑子集的机制。此划分操作使您可以向不同用户显示产品目录的不同部分。您可以创建合同并指定合同参与方仅有权购买属于预定义产品集的产品。WebSphere Commerce 提供工具来创建和管理合同，以及针对主产品目录的授权过滤角色。

## 属性

属性是网上商店内产品的属性。有两种类型的属性：

- 定义属性是诸如颜色或大小之类的特征。属性值是属性的特征，例如特定颜色（蓝色或黄色）或大小（中号）。在将属性值指定给商品之前必须对其进行预定义。属性值与其属性隐式相关。属性和属性值每种可能的组合都等于一个新商品。创建属性及其值之后，可以更新诸如名称、描述和类型（文本、整数或小数）之类的信息。定义属性用于库存标识解析，其中属性和属性值的每种可能的组合都定义一种商品。
- 相反地，描述性属性仅提供附加描述。例如，某些衣服只能干洗，不可水洗，那么描述性属性可以指定这种只可干洗的条件。注意，描述性属性不用于库存标识解析，它用来加强产品描述，或者为您的企业特定信息提供简易的定制。

## 属性值

属性值是属性的特性，如特定颜色（蓝色或黄色）或大小（小号、中号或大号）。在将属性值指定给商品之前必须对其进行预定义。每种可能的定义属性值的组合都定义一个商品。

## 打包销售商品属性

*打包销售商品属性*必须从打包销售商品中包含的产品的属性创建而来。仅包含商品的打包销售商品将不具有打包销售商品属性。

## 打包销售商品属性值

*打包销售商品属性值*是对打包销售商品属性指定的值。打包销售商品属性值必须从打包销售商品中包含的产品的属性值创建而来。



关于 WebSphere Commerce 中产品目录有用资源结构的更多详细信息，请参阅 WebSphere Commerce 联机帮助中的产品目录数据模型。

## 在 WebSphere Commerce 中创建产品目录有用资源

要为商店创建产品目录有用资源，您需要通过向若干 WebSphere Commerce 数据库表添加信息创建主产品目录。可以使用由装入程序软件包装入数据库的 XML 文件创建产品目录。如果您正在创建全球化产品目录，则将需要商店所支持的每种语言环境的独立 XML 文件。每个特定于语言环境的 XML 文件添加产品目录、产品目录组和产品目录条目的可翻译信息（例如描述）。

以下是产品目录创建过程的概述：

1. 在 WebSphere Commerce 中，产品目录是使用 XML 文件来创建的。创建产品目录的过程以一个产品目录实体开始，您的数据库等同于一个记录在纸上的产品目录。
2. 通过添加产品目录组来创建产品目录结构和导航，以此确定您的商品的产品目录和布局。
3. 创建库存信息作为产品目录条目的基础。
4. 以产品目录条目的形式添加您的商品，产品目录条目代表产品、库存标识、捆绑销售商品、打包销售商品、不可改装的成套商品和可改装的成套商品。
5. 向产品目录中的产品添加属性和属性值以将不同的库存标识彼此区分开。
6. 您可以为了促销的目的创建打包销售商品和捆绑销售商品以将某些产品目录条目组合在一起。
7. 接下来创建产品目录组和产品目录条目之间的关系。这将确定哪些条目属于产品目录组。
8. 您可以创建产品目录条目的销售策略关联以作为产品推荐措施。
9. 将您的产品目录、产品目录组和产品目录条目关联到您的 WebSphere Commerce 商店。
10. 在最后的步骤中，您需要创建：
  - a. 商品的税款。
  - b. 送货方式。
  - c. 充当库存仓库以及送货和接收中心的供货中心。一个商店可以定义多个供货中心。
  - d. 商品的价格。

### 创建主产品目录

要创建包含多个类别级别的主产品目录，请完成以下任务：

#### 第 1 部分：准备产品目录创建

1. 请复查产品目录信息及其在 WebSphere Commerce 内对应的对象和数据模型。产品目录信息是 WebSphere Commerce Server 的组件，为可订购商品提供在线产品目录导航、划分、分类和关联。
2. 请复查 WebSphere Commerce 装入程序软件包信息。装入程序软件包主要由准备数据并将数据装入 WebSphere Commerce 数据库的实用程序组成。可以使用装入程序

软件包装入大量数据并更新数据库中的数据。关于装入程序软件包的更多信息，请参阅第 309 页的第 10 部分，『发布商店』。

3. 请复查第 417 页的附录 B，『创建数据』中的信息。
4. 通过管理控制台创建组织作为产品目录所有者。关于更多信息，请参阅 WebSphere Commerce 联机帮助主题“创建组织”。
5. 使用“多乐五金店”样本商店中现有的 XML 条目和 catalog.xml 文件作为指南为主产品目录创建新的 XML 文件。如果正在创建全球化产品目录，则为商店支持的每种语言环境创建独立的 catalog.xml 文件。特定于语言环境的文件应当指定所有描述信息，以便比较容易进行翻译。在此示例中，一个 catalog.xml 文件将用于无需翻译的所有信息，另一个 catalog.xml 将用于商店支持的每种语言环境并将包含需要翻译的信息。或者，如果愿意，可以按需要使用“多乐五金店”样本商店现有的 XML 文件并更改信息。“多乐五金”样本商店中的 catalog.xml 文件位于其商店归档文件中。要查看 catalog.xml 文件，请使用 ZIP 程序将该商店归档文件解压缩。catalog.xml 文件位于以下数据目录中：

- WC\_installdir/samplestores

**注：**WebSphere Commerce 联机帮助包含关于样本商店中所包含的每个数据有用资源的信息。

catalog.dtd 文件位于以下目录中：

- WC\_installdir/xml/sar

## 第 2 部分：创建产品目录实体

1. 使用“多乐五金店”样本商店的以下示例作为指南，通过向 CATALOG 和 CATALOGDSC 表添加信息创建产品目录条目。一个产品目录实体代表数据库中的一个产品目录。

```
<catalog
catalog_id="@catalog_id_1"
member_id="@seller_b2b_mbr_id"
  identifier="ToolTech"
description="ToolTech Catalog"
tpclevel="0"
/>
```

其中

- catalog\_id 是内部引用号。
- member\_id 是标识产品目录所有者的内部引用号。
- identifier 是产品目录的外部名称。
- description 是产品目录的描述。

2. 使用“多乐五金店”样本商店的以下示例作为指南，在特定于语言环境的 XML 文件中添加产品目录的描述以便翻译：

```
<catalogdsc
catalog_id="@catalog_id_1"
language_id="@en_US;"
name="Store master catalog"
/>
```

其中

- catalog\_id 是使此语言特定信息与产品目录相关的内部引用号。



- language\_id 是语言的标识。
- name 是产品目录的依赖于语言的名称。

### 第 3 部分: 创建产品目录组

1. 使用“多乐五金店”样本商店的以下示例作为指南, 通过向 CATGROUP 和 CATGRPDESC 表添加信息创建产品目录组。产品目录组(也称为类别)是其它产品目录组或产品的分组。对产品目录中的每个产品目录组完成此任务:

```
<catgroup
catgroup_id="@catgroup_id_1"
member_id="@seller_b2b_mbr_id"
identifier="Woodworking"
markfordelete="0"
/>
```

其中

- catgroup\_id 是产品目录组的内部引用号
  - member\_id 是标识产品目录所有者的内部引用号。
  - identifier 是产品目录的外部名称。
  - markfordelete 指示产品目录组是否已标记为删除。
    - 0 = 否。
    - 1 = 是。
2. 使用“多乐五金店”样本商店的以下示例作为指南, 在特定于语言环境的 XML 文件中添加产品目录组的描述以便翻译。对产品目录中的每个产品目录组完成此任务:

```
<catgrpdesc
language_id="&en_US;"
catgroup_id="@catgroup_id_1"
name="Woodworking"
shortdescription="Woodworking"
longdescription="Woodworking"
published="1"
/>
```

其中

- language\_id 是语言的标识。
- catgroup\_id 是产品目录组的内部引用号。
- name 是产品目录的依赖于语言的名称。
- shortdescription 是产品目录组的简短描述。
- longdescription 是产品目录组的详细描述。
- published 指示是否应对 language\_id 指示的语言显示此产品目录组:
  - 0 = 否。
  - 1 = 是。

**注:** 每次创建产品目录组及其描述时, catgroup\_id 都更改为代表新的产品目录组。例如, catgroup\_id="@catgroup\_id\_2"、catgroup\_id="@catgroup\_id\_3" 和 catgroup\_id="@catgroup\_id\_4" 等等。

3. 创建产品目录组后，请通过向 `CATTGRP` 表添加信息将顶级产品目录组指定给产品目录。此产品目录组是其下紧挨着的产品目录组的父级。对产品目录中的每个顶级产品目录组完成此任务。使用“多乐五金店”样本商店的以下示例作为指南：

```
<cattgrp
catalog_id="@catalog_id_1"
catgroup_id="@catgroup_id_1"
/>
```

其中

- `catalog_id` 是产品目录的引用号。
- `catgroup_id` 是产品目录组的引用号。

**注：**每次将顶级产品目录组指定给产品目录时，`catgroup_id` 修改为代表新的产品目录组关联。例如，`catgroup_id="@catgroup_id_2"`、`catgroup_id="@catgroup_id_3"` 和 `catgroup_id="@catgroup_id_4"`，等等。

4. 一旦对产品目录组确定了父级和子级结构，请通过向 `CATGRPREL` 表添加信息创建产品目录组之间的关系。对产品目录中的每个父级和子级产品目录结构完成此任务。使用“多乐五金店”样本商店的以下示例作为指南：

```
<catgrprel
catgroup_id_parent="@catgroup_id_1"
catgroup_id_child="@catgroup_id_11"
catalog_id="@catalog_id_1"
sequence="0"
/>
```

其中

- `catgroup_id_parent` 是此关系的源产品目录组。
- `catgroup_id_child` 是此关系的目标产品目录组。
- `catalog_id` 是产品目录的引用号。
- `sequence` 是确定产品目录组内容的显示顺序的编号。

**注：**对于每种产品目录组关系，`catgroup_id_child` 和 `sequence` 修改为代表新关系。例如，后续关系将显示为 `catgroup_id_child="@catgroup_id_12"` 与 `sequence="1"`、`catgroup_id_child="@catgroup_id_13"` 与 `sequence="2"`，等等。如果不在产品目录中使用浏览结构，则可以除去 `CATGRPREL` 关系。

## 第 4 部分：创建库存信息

1. 使用“多乐五金店”样本商店的以下示例作为指南，通过向 `BASEITEM`、`BASEITEMDSC`、`ITEMSPC`、`ITEMVERSN`、`VERSIONSPC`、`DISTARRANG` 和 `STOREITEM` 表添加信息创建库存信息。从通过向 `BASEITEM` 表添加信息创建基本商品开始。基本商品表示具有公共名称和描述的一般产品系列。对产品目录中的每组库存商品完成此任务：

```
<baseitem
baseitem_id="@baseitem_id_102"
member_id="@seller_b2b_mbr_id"
markfordelete="0"
partnumber="toolttech_sku_102"
itemtype_id="ITEM"
quantitymeasure="C62"
quantitymultiple="1.0"
/>
```

其中

- `baseitem_id` 是生成的唯一键。
- `member_id` 是基本商品的所有者。
- `markfordelete` 指示基本商品是否已标记为删除：
  - 0 = 否。
  - 1 = 是。
- `partnumber` 唯一标识所有者的基本商品。
- `itemtype_id` 是基本商品的类型：
  - ITEM = 商品、打包销售商品或捆绑销售商品。
  - DNKT = 可改装的成套商品。
  - STKT = 不可改装的成套商品。
- `quantitymeasure` 是数量倍数的计量单位。
- `quantitymultiple` 是以整数单位计量的基本商品数量。它和 `quantitymeasure` 一同指示每个整数单位所表示的金额。

**注：**必须为您在产品目录中创建的每一产品创建一个基本商品。每次创建基本商品，`baseitem_id` 和 `partnumber` 编号都会更改以创建新的基本商品。例如，新的基本商品将包含 `baseitem_id="@baseitem_id_147"` 和 `partnumber="tooltech_sku_147"` 作为条目，而另一基本商品将包含 `baseitem_id="@baseitem_id_192"` 和 `partnumber="tooltech_sku_192"` 作为条目，等等。

2. 使用“多乐五金店”样本商店的以下示例作为指南，将关于指定商品的信息添加到数据库中。指定商品是所有属性都具有值的商品，代表产品目录中的商品、打包销售商品、捆绑销售商品或可改装的成套商品。对产品目录中的每个指定商品完成此任务：

```
<itemspc
itemspc_id="@itemspc_id_106"
baseitem_id="@baseitem_id_102"
markfordelete="0"
partnumber="T0000106"
member_id="@seller_b2b_mbr_id"
discontinued="N"
/>
```

其中

- `itemspc_id` 是生成的唯一键。
- `baseitem_id` 是产品基本商品。
- `markfordelete` 指示指定商品是否已标记为删除：
  - 0 = 否。
  - 1 = 是。
- `partnumber` 唯一标识所有者的指定商品。
- `member_id` 是指定商品的所有者。
- `discontinued` 指示指定商品是否已停止。
  - Y = 已停止，如果有足够库存就可以订购，但是不能延迟交货。
  - N = 正在运行，如果脱销可以延迟交货。

**注:** 必须为您在产品目录中创建的每一商品创建一个指定商品。每次定义指定商品时, `itemspc_id`、`baseitem_id`、`partnumber` 编号都会更改以创建新的指定商品。例如, 新的指定商品将包含 `itemspc_id="@itemspc_id_108"`、`baseitem_id="@baseitem_id_102"` 和 `partnumber="T0000108"` 作为条目, 而另一指定商品将包含 `itemspc_id`、`baseitem_id` 和 `partnumber` 作为条目, 等等。

3. 使用“多乐五金店”样本商店的以下示例作为指南, 将商品版本和基本商品之间关系的信息添加到数据库中。对产品目录中的每个此类关系完成此任务:

```
<itemversn
itemversn_id="@itemversn_id_102"
baseitem_id="@baseitem_id_102"
expirationdate="2010-01-01 00:00:00.000000"
versionname="version"
/>
```

其中

- `itemversn_id` 是生成的引用号, 标识商品版本。
- `baseitem_id` 是基本商品。
- `expirationdate` 是商品版本失效的时间。
- `versionname` 唯一标识基本商品的版本。

**注:** 每次创建商品版本和基本商品之间的关系时, `itemversn_id` 和 `baseitem_id` 编号都会更改以创建新的关系。`baseitem_id` 与现有的基本商品匹配。例如, 新关系将包含 `itemversn_id="@itemversn_id_107"` 和 `baseitem_id="@baseitem_id_107"` 作为条目, 而另一关系将包含 `itemversn_id="@itemversn_id_108"` 和 `baseitem_id="@baseitem_id_108"` 作为条目, 等等。

4. 使用“多乐五金店”样本商店的以下示例作为指南, 将产品版本和指定商品之间关系的以下信息添加到数据库中。对产品目录中的每个此类关系完成此任务:

```
<versionspc
versionspc_id="@versionspc_id_106"
itemspc_id="@itemspc_id_106"
itemversn_id="@itemversn_id_102"
/>
```

其中

- `versionspc_id` 是生成的唯一标识。
- `itemspc_id` 是产品目录条目相关的指定商品。
- `itemversn_id` 标识商品版本。

**注:** 每次创建产品版本和指定商品之间的关系时, `versionspc_id` 和 `itemspc_id` 编号都会更改以创建新的关系。`itemspc_id` 与现有的指定商品匹配。例如, 新关系将包含 `versionspc_id="@versionspc_id_107"` 和 `itemspc_id="@itemspc_id_107"` 作为条目, 而另一关系将包含 `versionspc_id="@versionspc_id_108"` 和 `itemspc_id="@itemspc_id_108"` 作为条目, 等等。

5. 使用“多乐五金店”样本商店的以下示例作为指南, 将分发安排添加到数据库中。分发安排使商店可以销售其自己的库存。对产品目录中的每个分发安排完成此任务:

```

<distarrang
distarrang_id="@distarrang_id_102"
wholesalestore_id="@storeent_id_1"
merchantstore_id="@storeent_id_1"
baseitem_id="@baseitem_id_102"
pickingmethod="F"
startdate="2000-12-25 00:00:00.000000"
enddate="2010-01-01 00:00:00.000000"
/>

```

其中

- distarrang\_id 是分发安排的引用号。
- wholesalestore\_id 是拥有商家商店可销售的库存的批发商店。此批发商店必须与 merchantstore\_id 相同。
- merchantstore\_id 是可以销售批发商店库存的商家商店。此商家商店必须与 wholesalestore\_id 相同。
- baseitem\_id 是分发安排所包括的产品。
- pickingmethod 确定依照此安排从 RECEIPT 表中选取库存的顺序。
  - F = FIFO (先进先出): 最早接收到的库存。
  - L = LIFO (后进先出): 最近接收到的库存。
- startdate 是分发安排生效的时间。
- enddate 是分发安排失效的时间。

**注:** 每次创建分发安排时, distarrang\_id 和 baseitem\_id 编号都会更改以创建新的分发安排。例如, 另一分发安排可能包含值

```

distarrang_id="@distarrang_id_147" 和
baseitem_id="@baseitem_id_147", 而再一个可能包含
distarrang_id="@distarrang_id_192" 和
baseitem_id="@baseitem_id_192", 等等。

```

6. 使用“多乐五金店”样本商店的以下示例作为指南, 将影响特定商店为特定基本商品的指定商品分配库存的属性添加到数据库中。对产品目录中的每个基本商品完成此任务:

```

<storeitem
baseitem_id="@baseitem_id_102"
storeent_id="@storeent_id_1"
trackinventory="Y"
forcebackorder="N"
releaseseparately="N"
returnnotdesired="N"
backorderable="Y"
creditable="Y"
minqtyforsplit="0"
/>

```

其中

- baseitem\_id 是基本商品。
- storeent\_id 是商店或商店组。
- trackinventory 控制是否在 RECEIPT 表中跟踪库存:
  - N = 不跟踪库存, 且 RECEIPT 表中没有条目。
  - Y = 在 RECEIPT 表中跟踪库存。
- forcebackorder 暂挂基本商品的指定商品的分配:

- N = 可以分配库存（正常行为）。
- Y = 即使有足够库存，也不能分配库存。
- `releaseseparately` 控制如何发出基本商品的指定订购商品：
  - N = 订购商品可与其它订购商品一起发出。
  - Y = 订购商品必须分别发货（装在各自的箱子中）。
- `returnnotdesired` 指示不期望退回商品（例如，易腐食品），即使客户愿意或能够退回：
  - N = 根据客户退回商品的意向评估返款请求，但并不期望退货。
  - Y = 视同期望退货，评估返款请求。
- `backorderable` 指示基本商品的指定商品不能延迟交货：
  - N = 商品不可以延迟交货。
  - Y = 商品可以延迟交货。
- `creditable` 指示商家是否将给予此商品的返款而无需佣金：
  - N = 按现状销售。
  - Y = 可返款。
- `minqtyforsplit` 表示如果新订购商品中其余未分配的数量小于特定的最低数量，不会在库存分配期间自动分割订购商品。

**注：**每次为商店商品定义库存分配规则时，`baseitem_id` 编号都会更改以代表新的基本商品。例如，一个新的分配可能包含 `baseitem_id="@baseitem_id_147"`，而另一个可能包含 `baseitem_id="@baseitem_id_192"`，等等。

7. 使用“多乐五金店”样本商店的以下示例作为指南，在特定于语言环境的 XML 文件中添加基本商品的描述以便翻译。对产品目录中的每个基本商品描述完成此任务：

```
<baseitmdsc
baseitem_id="@baseitem_id_102"
language_id="&en_US;"
shortdescription="Circular Saw"
longdescription="Light on weight but not in quality. The Circular Saw
weighs a maximum of 10.9lbs., with a choice of a 12 or 14 amp motor,
and speeds of up to 600 rpms! Low friction 220V aluminum alloy shoe
will ensure the job gets done on time."
/>
```

其中

- `baseitem_id` 是生成的唯一键。
- `language_id` 是此信息的语言。
- `shortdescription` 是基本商品的简短描述。
- `longdescription` 是基本商品的详细描述。

## 第 5 部分：创建产品目录条目

1. 使用“多乐五金店”样本商店的以下示例作为指南，通过向 `CATENTRY` 和 `CATENTDESC` 表添加信息创建产品目录条目。每种类型的产品目录条目（产品、商品、打包销售商品、捆绑销售商品和可改装的成套商品）都代表产品目录中待售商品的订购商品。您需要为每一产品目录条目定义一个基本商品。对产品目录中的每个产品目录条目完成此任务：

```

<catentry
catentry_id="@product_id_102"
baseitem_id="@baseitem_id_102"
member_id="@seller_b2b_mbr_id"
catenttype_id="ProductBean"
partnumber="T0000102"
mfpartnumber="Sprain-Tools-102"
mfname="Sprain Tools"
markfordelete="0"
buyable="1"
/>

```

其中

- catentry\_id 是产品目录条目的内部引用号。
- baseitem\_id 是产品目录条目相关的基本商品。
- member\_id 是标识产品目录条目的引用号。
- catenttype\_id 标识产品目录条目的类型:
  - ItemBean = 标识商品。
  - ProductBean = 标识产品。
  - PackageBean = 标识打包销售商品。
  - BundleBean = 标识捆绑销售商品。
  - DynamicKitBean = 标识可改装的成套商品。
- partnumber 是标识产品目录条目部件号的引用号。
- mfpartnumber 是生产商用于标识产品目录条目的部件号。
- mfname 是产品目录条目生产商的名称。
- markfordelete 指示产品目录条目是否已标记为删除。
  - 0 = 否。
  - 1 = 是。
- buyable 指示是否可以单独购买产品目录条目:
  - 0 = 否。
  - 1 = 是。

**注:** 每次将基本商品添加到产品目录条目中时, catentry\_id 和 baseitem\_id 顺序都会更改以代表新的产品目录条目。catenttype\_id 根据产品目录条目的类型更改。

- 使用“多乐五金店”样本商店的以下示例作为指南, 为每一产品目录条目定义指定商品。对产品目录中的每个产品目录条目完成此任务:

```

<catentry
catentry_id="@catentry_id_106"
itemspc_id="@itemspc_id_106"
member_id="@seller_b2b_mbr_id"
catenttype_id="ItemBean"
partnumber="T0000106"
mfpartnumber="Sprain-Tools-106"
mfname="Sprain Tools"
markfordelete="0"
buyable="1"
/>

```

其中

- catentry\_id 是产品目录条目的内部引用号。
- itemspc\_id 产品目录条目所属的指定商品。
- member\_id 是标识产品目录条目的引用号。
- cattentype\_id 标识产品目录条目的类型:
  - ItemBean = 标识商品。
  - ProductBean = 标识产品。
  - PackageBean = 标识打包销售商品。
  - BundleBean = 标识捆绑销售商品。
  - DynamicKitBean = 标识可改装的成套商品。
- partnumber 是标识产品目录条目部件号的引用号。
- mfpartnumber 是生产商用于标识产品目录条目的部件号。
- mfname 是产品目录条目生产商的名称。
- markfordelete 指示产品目录条目是否已标记为删除。
  - 0 = 否。
  - 1 = 是。
- buyable 指示是否可以单独购买产品目录条目:
  - 0 = 否。
  - 1 = 是。

**注:** 每次将指定商品添加到产品目录条目中时, catentry\_id 和 itemspc\_id 顺序都会更改以代表新的产品目录条目。cattentype\_id 根据产品目录条目的类型更改。在主产品目录结构限制下, 产品目录条目不能属于多个类别。要将产品目录条目放到多个类别中, 您必须使用销售产品目录。

- 使用“多乐五金店”样本商店的以下示例作为指南, 将描述添加到特定于语言环境的 XML 文件中。对产品目录中的每个产品目录条目完成此任务:

```
<catentdesc
catentry_id="@product_id_102"
  language_id="&en_US"
name="Circular"
shortdescription="Circular Saw"
longdescription="Light on weight but not in quality. The Circular Saw
weighs a maximum of 10.9lbs., with a choice of a 12 or 14 amp motor,
and speeds of up to 600 rpms! Low friction 220V aluminum alloy shoe
will ensure the job gets done on time."
thumbnail="images/circular_saw_sm.gif"
fullimage="images/circular_saw.gif"
available="1"
published="1"
/>
```

其中

- catentry\_id 是内部引用号, 指示此特定于语言的信息所相关的产品目录条目。
- language\_id 是语言的标识。
- name 是产品目录条目的依赖于语言的名称。
- shortdescription 是产品目录条目的简短描述。
- longdescription 是产品目录条目的详细描述。
- thumbnail 是缩略图的路径。



- fullimage 是全图像的路径。
- available 指示产品目录条目可用的时间长度。
- published 指示是否应以 language\_id 指示的语言显示此产品目录条目
  - 0 = 显示。
  - 1 = 不显示。

## 第 6 部分: 创建属性和属性值

1. 使用“多乐五金店”样本商店的以下示例作为指南, 通过向特定于语言环境的 XML 文件中的 ATTRIBUTE 和 ATTRVALUE 表添加信息创建产品的属性和属性值以便翻译。产品目录中的每个产品都具有一组特定属性, 如衬衫或长裤的大小和颜色。商品是由属性值定义的。例如, 衬衫是产品, 中号黑衬衫是商品。对产品目录中的每个属性完成此任务:

```
<attribute
attribute_id="@attribute_id_103"
  language_id("&en_US"
attrtype_id="STRING"
name="Amps"
sequence="0"
description="Amps"
catentry_id="@product_id_102"
description2="Amps"
/>
```

其中

- attribute\_id 是属性的内部引用号。
- language\_id 是属性所属的语言。
- attrtype\_id 是相应属性值的类型。
- name 是属性的名称。
- sequence 是确定给定产品的属性显示次序的顺序编号。
- description 是属性的描述。
- catentry\_id 是此属性所属产品的引用号。
- description2 是属性的附加描述。

**注:** 每次将属性添加到由 catentry\_id 定义的产品中时, attribute\_id 顺序都会更改以代表新属性。

2. 使用“多乐五金店”样本商店的以下示例作为指南, 添加属性值。对产品目录中的每个属性值完成此任务:

```
<attrvalue
attrvalue_id="@attrvalue_id_114"
  language_id("&en_US"
attribute_id="@attribute_id_103"
name="12.0amps"
attrtype_id="STRING"
stringValue="12.0amps"
sequence="0"
usage="1"
catentry_id="@catentry_id_106"
/>
```

其中

- attrvalue\_id 是属性值的内部引用号

- language\_id 是此属性值所属的语言
- attribute\_id 是与此值关联的属性的内部引用号
- name 是属性值的名称
- attrtype\_id 是属性值的类型
- stringvalue 是属性值
- sequence 是确定给定属性的属性值显示次序的顺序编号
- usage 是属性的类型:
  - 1 标识用于库存标识解析的定义属性。
  - 0 (或其它值) 标识描述性属性。
- catentry\_id 是此属性值描述的商品标识

**注:** 每次将属性值添加到属性中时, attrvalue\_id 顺序都会更改以代表不同值。attribute\_id 顺序会更改以代表不同属性。sequence 随着每个新的属性值增加。例如, 后续属性值将是 sequence="1"、sequence="2" 和 sequence="3", 等等。

## 第 7 部分: 创建产品和商品之间的关系

1. 为产品目录创建产品和商品之后, 请通过向 CATENTREL 表添加信息定义产品和商品之间的关系。使用“多乐五金店”样本商店的以下示例作为指南。对产品目录中的每个产品和商品关系值完成此任务:

```
<catentrel
catentry_id_parent="@product_id_147"
catreltype_id="PRODUCT_ITEM"
catentry_id_child="@catentry_id_152"
sequence="2"
quantity="1"
/>
```

其中

- catentry\_id\_parent 是此关系中源产品目录条目的引用号, 即产品。
- catreltype\_id 是关系类型: PRODUCT\_ITEM
- catentry\_id\_child 是此关系中目标产品目录条目的引用号, 即商品。
- sequence 是用于确定显示顺序的顺序编号。
- quantity 是可与关系关联的数量。

**注:** 每次添加产品和商品之间的关系时, catentry\_id\_parent 和 catentry\_id\_child 编号都会更改以基于 catreltype\_id 创建不同的关系。对于每种新关系, sequence 编号都不同。例如, 如果您具有 sequence="2", 则下一关系将是 sequence="3", 接着是 sequence="4", 等等。

## 第 8 部分: 创建打包销售商品和捆绑销售商品

1. 一旦创建了产品和商品, 请通过向 CATENTRY、CATENTDESC 和 CATENTREL 表添加信息创建打包销售商品和捆绑销售商品。例如, 通过向 CATENTRY 表添加信息而使用以下代码样本来创建打包销售商品或捆绑销售商品。对产品目录中的每个打包销售商品和捆绑销售商品完成此任务:

```
<catentry
catentry_id="@package_id_102"
member_id="@seller_b2b_mbr_id"
```

```

catenttype_id="PackageBean"
partnumber="sku-@package_id_102"
mfpartnumber="sku-@package_id_102"
mfname="ToolTech"
markfordelete="0"
buyable="1"
/>

```

其中

- catentry\_id 是产品目录条目的引用号。
- member\_id 是标识产品目录条目所有者的引用号。
- catenttype\_id 标识产品目录条目的类型:
  - PackageBean = 标识打包销售商品。
  - BundleBean = 标识捆绑销售商品。
- partnumber 是标识产品目录条目部件号的引用号。
- mfpartnumber 是生产商用于标识产品目录条目的部件号。
- mfname 是产品目录条目生产商的名称。
- markfordelete 指示产品目录条目是否已标记为删除:
  - 0 = 否。
  - 1 = 是。
- buyable 指示是否可以单独购买产品目录条目。
  - 0 = 否。
  - 1 = 是。

**注:** 每次您创建打包销售商品或捆绑销售商品时, catentry\_id、partnumber 和 mfpartnumber 编号都会更改以创建不同的打包销售商品或捆绑销售商品。例如, 要创建新的打包销售商品, 您可以使用 catentry\_id="@package\_id\_103"、partnumber="sku-@package\_id\_103" 和 mfpartnumber="sku-@package\_id\_103", 包括将此条目标识为打包销售商品的 catenttype\_id="PackageBean"。要创建新的捆绑销售商品, 您可以使用 catentry\_id="@package\_id\_110"、partnumber="sku-@package\_id\_110" 和 mfpartnumber="sku-@package\_id\_110", 包括将此条目标识为捆绑销售商品的 catenttype\_id="BundleBean", 等等。

- 例如, 通过向特定于语言环境的 XML 文件中的 CATENTDESC 表添加信息, 从而使用以下代码样本来添加打包销售商品或捆绑销售商品描述以便翻译。对产品目录中的每个打包销售商品和捆绑销售商品的描述完成此任务:

```

<catentdesc
catentry_id="@catentry_id_102"
language_id="-1"
name="computer"
shortdescription="Computer"
longdescription="A combination of a central processing unit, monitor,
hard drive, and color printer. An ideal starter system."
thumbnail="images/package_system_sm.gif"
fullimage="images/package_system.gif"
available="1"
published="1"
/>

```

其中

- catentry\_id 是内部引用号，指示此语言特定信息相关的产品目录条目。
  - language\_id 是语言的标识。
  - name 是产品目录条目的依赖于语言的名称。
  - shortdescription 是产品目录条目的简短描述。
  - longdescription 是产品目录条目的详细描述。
  - thumbnail 是产品目录条目的缩略图路径。
  - fullimage 是产品目录条目的全图像路径。
  - available 指示产品目录条目可用的时间长度。
  - published 指示是否应以 language\_id 指示的语言显示产品目录条目：
    - 0 = 不显示产品目录条目。
    - 1 = 显示产品目录条目。
- 例如，通过向 CATENTREL 表添加信息而使用以下代码样本来创建打包销售商品或捆绑销售商品与其组件之间的关系。对产品目录中的每个打包销售商品或捆绑销售商品组件关系完成此任务：

```
<catentrel
catentry_id_parent="@catentry_id_102"
catreltype_id="PACKAGE_COMPONENT"
catentry_id_child="@catentry_id_97"
sequence="1.0"
quantity="1.0"
/>
```

其中

- catentry\_id\_parent 是此关系中源产品目录条目的引用号，即打包销售商品或捆绑销售商品。
- catreltype\_id 是此关系的类型：
  - PACKAGE\_COMPONENT 表示打包销售商品与其组件之间的关系。
  - BUNDLE\_COMPONENT 表示捆绑销售商品与其组件之间的关系。
- catentry\_id\_child 是此关系中目标产品目录条目的引用号，即组件。
- sequence 是用于确定显示顺序的顺序编号。
- quantity 是可与关系关联的数量。

**注：**每次创建打包销售商品和捆绑销售商品之间的关系时，catentry\_id\_parent 和 catentry\_id\_child 编号都会更改以匹配现有的产品目录条目。对于每种新关系，sequence 编号都不同。例如，如果以 sequence="1.0" 开始，则下一关系将为 sequence="2.0"，接着是 sequence="3.0"，等等。

## 第 9 部分：创建产品目录组和产品目录条目之间的关系

1. 在产品目录中创建产品目录组和产品目录条目之后，请通过向 CATGPENREL 表添加信息定义产品目录组和产品目录条目之间的关系。在主产品目录结构限制下，产品目录条目不能属于多个类别。要将产品目录条目放到多个类别中，您必须使用销售产品目录。使用“多乐五金店”样本商店的以下示例作为指南。对产品目录中的每个产品目录组和产品目录条目关系完成此任务：

```
<catgpenrel
catgroup_id="@catgroup_id_11"
catalog_id="@catalog_id_1"
catentry_id="@product_id_102"
sequence="0"
/>
```

其中

- catgroup\_id 是此关系的源产品目录组。
- catalog\_id 是此关系所在的产品目录。
- catentry\_id 是此关系的目标产品目录条目。
- sequence 是确定产品目录组的内容显示次序的顺序编号。

**注：**每次创建产品目录组和产品目录条目之间的关系时，catgroup\_id 和 catentry\_id 编号都会更改以形成不同产品目录组和产品目录条目的新关系。对于每种新关系，sequence 编号都不同。例如，如果以 sequence="0" 开始，则下一关系将为 sequence="1"，接着是 sequence="2"，等等。

## 第 10 部分：创建销售策略关联

1. 例如，通过向 MASSOCECE 表添加信息而使用以下代码样本来创建产品目录条目之间的销售策略关联。对产品目录中的每个销售策略关联完成此任务：

```
<massoccece
massoccece_id="@relationship_id_100"
massoctype_id="X-SELL"
catentry_id_from="@product_id_1"
catentry_id_to="@product_id_15"
massoc_id="REQUIRES"
quantity="2.0"
rank="1.00000"
/>
```

其中

- massoccece\_id 是此条目的引用号。
- massoctype\_id 是关联类型的标识：
  - X-SELL = 交叉销售。
  - UPSSELL = 优选销售。
  - ACCESSORY = 辅助销售。
  - REPLACEMENT = 代替销售。
- catentry\_id\_from 是产品目录条目，该条目是关联的源。
- catentry\_id\_to 是产品目录条目，该条目是关联的目标。
- massoc\_id 是语义说明符的标识：
  - REQUIRES
  - COMES\_WITH
  - TEMP
  - NONE
- quantity 是与此关联相关的数量。
- rank 是用于显示次序的顺序编号。

注：每次添加销售策略关联时，massoccece\_id 编号都会更改以代表新关系。catentry\_id\_from 和 catentry\_id\_to 编号会更改以创建关联的新商品内容。

## 第 11 部分：将产品目录与商店关联

1. 使用“多乐五金”样本商店中现有的 store-catalog.xml 文件作为指南，通过在数据库中对商店指定产品目录、其产品目录组和产品目录条目，从而使产品目录与商店关联。您还应当对产品目录组和产品目录条目指定显示页面。将此信息添加到 STORECAT、STORECENT、STORECGRP、DISPCGPREL 和 DISPENTREL 表。如果正在创建全球化产品目录，则为商店支持的每种语言环境创建独立的“商店 - 产品目录”关系 XML 文件：

```
<storecat
  catalog_id="@catalog_id_1"
  storeent_id="@storeent_id_1"
  mastercatalog="1"
/>
```

其中

- catalog\_id 是产品目录的引用号。
  - storeent\_id 是数据库中商店实体的引用号。
  - mastercatalog 指定商店的主产品目录。值 1 指示此产品目录指定为主产品目录。
2. 使用“多乐五金店”样本商店的以下示例作为指南，将产品目录条目添加到“商店 - 产品目录”关系。对产品目录中的每个产品目录条目完成此任务：

```
<storecent
  storeent_id="@storeent_id_1"
  catentry_id="@product_id_102"
/>
```

其中

- storeent\_id 是数据库中商店实体的引用号。
- catentry\_id 是产品目录条目的引用号。

注：每次将 catentry\_id 添加到商店实体中时，引用号都会更改以匹配现有产品目录条目。

3. 使用“多乐五金店”样本商店的以下示例作为指南，将产品目录组添加到商店实体中。对产品目录中的每个产品目录组完成此任务：

```
<storecgrp
  storeent_id="@storeent_id_1"
  catgroup_id="@catgroup_id_1"
/>
```

其中

- storeent\_id 是数据库中商店实体的引用号。
- catgroup\_id 是产品目录组的引用号。

注：每次将 catgroup\_id 添加到商店实体中时，引用号都会更改以匹配现有产品目录组。

## 第 12 部分: 将税款与产品目录关联

使税款与特定商店的产品目录条目中的产品和服务关联。您必须通过将此信息添加到 CATENCALCD 表使税款计算代码与产品目录条目关联。关于更多信息, 请参阅第 242 页的『在 WebSphere Commerce 中创建税款有用资源』。

## 第 13 部分: 将送货方式与产品目录关联

要使送货方式与产品目录中的产品和服务关联, 您必须使送货费用计算代码与产品目录条目关联。将此信息添加到 CATENCALCD 表。关于更多信息, 请参阅第 225 页的『在 WebSphere Commerce 中创建送货有用资源』。

## 第 14 部分: 将供货中心与产品目录关联

使产品目录与将产品发送给客户的供货中心关联。供货中心管理商店的产品库存和送货。将此信息添加到 FFMCENTER 表。关于更多信息, 请参阅第 195 页的『在 WebSphere Commerce 中创建供货有用资源』。

## 第 15 部分: 为产品目录条目创建价格

为产品目录条目创建定价。定价代表产品目录条目的价格范围和使用此价格而必须满足的所有条件。要创建正常运作的产品目录, 您需要将出售物信息添加到数据库。将此信息添加到 TRADEPOSCN、TDPSCNCNTR、MGPTRDPSCN、OFFER 和 OFFERPRICE 表。关于更多信息, 请参阅第 172 页的『在 WebSphere Commerce 中创建定价有用资源』。或者可以使用 WebSphere 贸易加速器中的“产品管理”工具创建或更新产品目录条目的定价。

## 第 16 部分: 装入 XML 文件

创建数据之后, 可以使用装入程序软件包或通过发布实用程序将 XML 文件装入到数据库中。关于装入程序软件包的更多信息, 请参阅第 309 页的第 10 部分, 『发布商店』。

**注:** 您也可以使用 WebSphere 贸易加速器的“产品管理”工具为主产品目录创建产品目录有用资源。关于“产品管理”工具的更多详细信息, 请参阅 WebSphere Commerce 联机帮助。

## 显示商店产品目录有用资源

使产品目录、产品目录组和产品目录条目与商店关联后, 通过在数据库中创建这些关系指定 JSP 模板显示产品目录条目和产品目录组。以 XML 文件(可以使用装入程序软件包装入数据库)格式创建这些关系。

“多乐五金”商店样本的 store-catalog.xml 文件位于其商店归档文件中。要查看 store-catalog.xml 文件, 请使用 ZIP 程序将该商店归档文件解压缩。store-catalog.xml 文件位于以下数据目录中:

- WC\_installdir/samplestores

store-catalog.dtd 文件位于以下目录中:

- WC\_installdir/xml/sar

在可以创建“商店 - 产品目录”关系之前, 请确保已创建了商店数据有用资源。请完成以下任务, 其中每项任务都在 store-catalog.xml 文件中创建条目:

1. 要在商店中显示产品目录组(类别), 您必须对产品目录组指定 JSP 模板。可以对产品目录组指定特定显示页模板, 也可以指定缺省模板显示所有产品目录组。使用

“多乐五金店”样本商店的以下示例作为指南，通过向 DISPCGPREL 表添加信息指定产品目录组模板。对希望指定给产品目录组的每个模板完成此任务：

```
<dispcgprel
catgroup_id="@catgroup_id_1"
devicefmt_id="-1"
dispcgprel_id="@dispcgprel_id_1"
mbrgrp_id="0"
pagename="CategoryDisplay.jsp"
storeent_id="@storeent_id_1"
rank="0"/>
```

其中

- `catgroup_id` 是将显示此页面名称的产品目录组的引用号。值 0 指示此页面名称将用于所有产品目录组。
- `devicefmt_id` 是将显示页面的设备类别的引用号。值 -1 指示此模板页面将由 HTTP 浏览器使用。
- `dispcgprel_id` 是此条目的引用号。
- `mbrgrp_id` 是将显示模板页面的成员组的引用号。值 0 指示此模板页面将用于所有成员组。
- `pagename` 是显示模板页面的名称。
- `rank` 是当多个页面满足选择条件时用于断开连接的顺序编号。

**注：**每次对产品目录组指定 JSP 模板时，`catentry_id` 都会更改顺序以匹配现有产品目录条目。

2. 要显示商店中的产品目录条目（产品、商品、打包销售商品、不可改装的成套商品、捆绑销售商品和可改装的成套商品），您必须对产品目录条目指定 JSP 模板。可以指定缺省模板显示所有产品目录条目，或者指定缺省模板显示每种类型的产品目录条目，例如，一个模板用于产品，另一模板用于商品，或特定模板用于特定产品目录条目。使用“多乐五金店”样本商店的以下示例作为指南，通过向 DISPENTREL 表添加信息指定模板。对要指定给产品目录条目的每个模板完成此任务：

```
<dispentrel
auctionstate="0"
catentry_id="0"
catenttype_id="ProductBean"
devicefmt_id="-1"
dispentrel_id="@dispentrel_id_1"
mbrgrp="0"
pagename="ProductDisplay.jsp"
storeent_id="@storeent_id_1"
rank="0"/>
```

其中

- `auctionstate` 指示此模板页面显示正在拍卖的产品目录条目。
  - 0 = 不是拍卖模板。
  - 1 = 拍卖模板。
- `catentry_id` 是将显示此页面名称的产品目录条目的引用号。值 0 指示此页面名称将用于所有产品目录条目。
- `catenttype_id` 是此页面将用于显示的产品目录条目类型。
  - `ProductBean` = 显示产品。



- ItemBean = 显示商品。
- PackageBean = 显示打包销售商品。
- BundleBean = 显示捆绑销售商品。
- DynamicKitBean = 显示可改装的成套商品。
- devicefmt\_id 是将显示页面的设备类别的引用号。值 -1 指示此模板页面将由 HTTP 浏览器使用。
- dispentrel\_id 是产品目录条目的引用号。
- mbrgrp 是将显示此模板页面的成员组的引用号。值 0 指示此模板页面将用于所有成员组。
- pagename 是显示模板页面的名称。
- storeent\_id 是将显示此页面的商店的引用号。
- rank 是当多个页面满足选择条件时用于断开连接的顺序编号。

**注：**每次对产品目录条目指定 JSP 模板时，catentry\_id 都会更改顺序以匹配现有产品目录条目。

---

## 创建销售产品目录

WebSphere Commerce 商店允许两种类型的产品目录：主产品目录和销售产品目录。销售产品目录无需符合施加在主产品目录上的结构限制。销售产品目录用来提供一个灵活的显示结构从而使您能够创建适合商店需求的产品目录。

特别的，销售产品目录无需满足以下施加在主产品目录上的限制：

- 主产品目录必须是恰当的树，这意味着不存在循环且不能使用以下结构：父类别 A 拥有子类别 B。则 B 和 B 的所有子类别都不是 A 的父类别，这一点很重要。
- 产品不能属于多个类别。

以下任务通过修改“时尚潮流”样本商店产品目录来创建销售产品目录。得到的产品目录不能再分类为主产品目录，因为以下步骤将某些产品的分类引入多个类别中。典型的销售产品目录是通过向以下类别关系表添加信息创建的：CATGRPREL（它保存子类别关系）和 CATGPENREL（它保存类别 - 产品关系）。尽管这些示例涉及“时尚潮流”，但是您可对您自己的主产品目录遵循这些基本步骤，进行适当的调整以与产品目录信息、结构和设计相匹配。

## 向另一类别添加产品

此示例显示了如何在保留原始结构的同时，将产品从一个类别复制到另一类别。主页促销类别包含夏日女式睡衣产品，该产品也可归属于女式时装顶级类别的睡衣子类别之下。这些指示信息将向您显示如何将夏日女式睡衣产品及其库存标识复制到睡衣类别。

要通过向另一类别添加产品从而将“时尚潮流”样本商店主产品目录更改为销售产品目录，请执行以下操作：

1. 发布“时尚潮流”商店归档文件以创建“时尚潮流”样本商店。随 WebSphere Commerce 提供了美国英语以及九种本地语言之一的“时尚潮流”。选择一个 FashionFlow\_en\_US\_locale.sar 文件用于发布。
2. 在编辑器中打开 catalog.xml 文件。此文件位于以下 WebSphere Commerce 目录中：

- `WC_installdir/samplestores/FashionFlow/locale/data`

3. 在 `catalog.xml` 文件中定位 `CATGPENREL` 数据部分。对原为**主页促销**类别下产品的**夏日女式睡衣**创建新的产品条目。在 `CATGPENREL` 部分下，添加以下摘要以包含产品：

```
<catgpenrel
catgroup_id="@catgroup_id_18"
catalog_id="@catalog_id_1"
catentry_id="@product_id_2692"
sequence="2"
/>
```

其中

- `catgroup_id` 是由“时尚潮流”样本商店定义的产品目录组内部引用号。在此示例中，`@catgroup_id_18` 是**女式睡衣**类别。
  - `catalog_id` 是“时尚潮流”样本商店定义的产品目录的内部引用号。
  - `catentry_id` 是由“时尚潮流”样本商店定义的产品目录条目内部引用号。在此示例中，`@catentry_id_2692` 是**夏日女式睡衣**产品。
  - `sequence` 是确定由“时尚潮流”样本商店定义的产品目录组的内容显示顺序的编号。在此示例中，**夏日女式睡衣**产品将最后显示。
4. 在添加了**夏日女式睡衣**产品条目之后，如“时尚潮流”样本商店中所定义，在 `CATGPENREL` 部分下添加产品的 `SKU` 条目。当前，**夏日女士睡衣**产品包含十个已定义的库存标识。在 `CATGPENREL` 部分下，添加以下摘要以包含库存标识：

```
<catgpenrel
catgroup_id="@catgroup_id_18"
catalog_id="@catalog_id_1"
catentry_id="@catentry_id_2695"
sequence="2"
/>
```

```
<catgpenrel
catgroup_id="@catgroup_id_18"
catalog_id="@catalog_id_1"
catentry_id="@catentry_id_2696"
sequence="2"
/>
```

```
<catgpenrel
catgroup_id="@catgroup_id_18"
catalog_id="@catalog_id_1"
catentry_id="@catentry_id_2697"
sequence="2"
/>
```

```
<catgpenrel
catgroup_id="@catgroup_id_18"
catalog_id="@catalog_id_1"
catentry_id="@catentry_id_2698"
sequence="2"
/>
```

```
<catgpenrel
catgroup_id="@catgroup_id_18"
catalog_id="@catalog_id_1"
catentry_id="@catentry_id_2699"
sequence="2"
/>
```

```
<catgpenrel
```

```

catgroup_id="@catgroup_id_18"
catalog_id="@catalog_id_1"
catentry_id="@catentry_id_2700"
sequence="2"
/>

<catgpenrel
catgroup_id="@catgroup_id_18"
catalog_id="@catalog_id_1"
catentry_id="@catentry_id_2701"
sequence="2"
/>

<catgpenrel
catgroup_id="@catgroup_id_18"
catalog_id="@catalog_id_1"
catentry_id="@catentry_id_2702"
sequence="2"
/>

<catgpenrel
catgroup_id="@catgroup_id_18"
catalog_id="@catalog_id_1"
catentry_id="@catentry_id_2703"
sequence="2"
/>

<catgpenrel
catgroup_id="@catgroup_id_18"
catalog_id="@catalog_id_1"
catentry_id="@catentry_id_2704"
sequence="2"
/>

```

其中

- `catgroup_id` 是由“时尚潮流”样本商店定义的产品目录组内部引用号。在此示例中，`@catgroup_id_18` 是女式睡衣类别。
- `catalog_id` 是“时尚潮流”样本商店定义的产品目录的内部引用号。
- `catentry_id` 是由“时尚潮流”样本商店定义的产品目录条目内部引用号。在此示例中，`@catentry_id_2695` 到 `@catentry_id_2704` 表示已对夏日女式睡衣产品定义的十个库存标识。
- `sequence` 是确定由“时尚潮流”样本商店定义的产品目录组的内容显示顺序的编号。在此示例中，夏日女式睡衣库存标识将最后显示。

5. 保存 `catalog.xml` 文件。

6. 要查看更改，请执行以下操作之一：从管理控制台发布修改后的“时尚潮流”商店归档文件，或者如第 373 页的『装入数据库有用资源组』中所示利用装入程序软件包装人 `catalog.xml` 文件。

---

## 在 WebSphere Commerce 中管理产品目录有用资源

随着时间的流逝，您将需要更新主产品目录的数据库有用资源信息。维护产品目录是一个进行中的过程，因为您将需要不断地添加和除去商品、创建和关联类别或产品目录组，以及更新诸如描述和价格之类的产品信息。

您可以使用现有的数据库条目和商店的 `catalog.xml` 文件编辑 WebSphere Commerce XML 数据，从而更改您的产品目录有用资源。使用 WebSphere Commerce 样本商店 XML 文件作为参考，该文件位于以下数据目录中：

- `WC_installdir/samplestores`

注：这些示例来源于“时尚潮流”样本商店，标识了为更改产品目录有用资源信息而必须修改哪些 XML 元素。

## 产品目录组

使用 `CATGROUP` 和 `CATGRPDESC` 数据库表在 WebSphere Commerce 产品目录中创建产品目录组。在 `catalog.xml` 文件中，典型的产品目录组看上去与以下抽取内容相似：

```
<catgroup
  catgroup_id="@catgroup_id_1"
  member_id="&MEMBER_ID"
  identifier="Accessories"
  markfordelete="0"
/>
```

`catgroup_id` 是产品目录组的内部引用号。在 WebSphere Commerce 中，每个产品目录组都被指定了一个内部引用号，它在添加产品目录条目时标识该组。`identifier` 是产品目录组的外部名称。在数据库有用资源中这两个元素都是唯一的，并且不可以复制它们。

名称和描述属于特定于语言环境的 `catalog.xml` 文件，对于商店支持的每种语言环境都需要其中一个文件。包含有可翻译信息的典型产品目录组看上去与以下抽取的内容相似：

```
<catgrpdesc
  language_id="&en_US"
  catgroup_id="@catgroup_id_1"
  name="Accessories"
  shortdescription="Accessories"
  longdescription="Accessories"
  published="1"
/>
```

`language_id` 标识产品目录信息所用的语言。该标识必须更改以匹配商店支持的每种语言。`name` 向客户显示，这与 `shortdescription` 和 `longdescription` 元素是相同的，这两个元素可以包含产品目录组的简短描述和详细描述。

当创建新的产品目录组时，请遵循以上信息的结构。

注：

1. 虽然 `identifier` 和 `name` 元素在以上示例中是等价的，但是其内容可能不同。例如，您可能选择将您的产品目录组重命名为 **Complementary Additions**。在这种情况下，您无需更改 `identifier` 中的信息，仅需更改 `name` 中的信息。
2. 当删除产品目录组时，请确保相应的更新了 `catgroup_id` 显示。例如，如果您还希望删除产品目录组下的产品目录条目，则应该除去整个 XML 条目。然而，如果您计划保留产品目录条目，则需要将 `catgroup_id` 更改到正确的组。

## 产品目录条目

产品目录条目是使用 `CATENTRY` 和 `CATENTDESC` 数据库表中信息在 WebSphere Commerce 产品目录中创建的。产品目录条目可以是产品、商品、打包销售商品、捆绑销售商品、不可改装的成套商品或可改装的成套商品。在 `catalog.xml` 文件中，典型的产品目录条目看上去与以下抽取内容相似：

```

<catentry
catentry_id="@product_id_102"
baseitem_id="@baseitem_id_102"
  member_id="&MEMBER_ID"
catenttype_id="ProductBean"
partnumber="product-sku-nf-102"
mfpartnumber="product-sku-nf-102"
mfname="FashionFlow"
markfordelete="0"
buyable="1"
/>

```

catentry\_id 是产品目录条目的内部引用号。baseitem\_id 是产品目录条目为了库存目的而相关的基本商品。partnumber 是标识产品目录条目部件号的引用号。mfpartnumber 是生产商用于标识产品目录条目的部件号。在数据库有用资源中这些元素都是唯一的，并且不可以复制它们。

catenttype\_id 标识产品目录条目的类型：ItemBean、ProductBean、PackageBean、StaticBean、BundleBean 或 DynamicKitBean。

名称和描述属于特定于语言环境的 catalog.xml 文件，对于商店支持的每种语言环境都需要其中一个文件。此文件中还包含了商品图像。包含有可翻译信息的典型产品目录组看上去与以下抽取的内容相似：

```

<catentdesc
catentry_id="@product_id_102"
  language_id="&en_US"
name="Belt"
shortdescription="Classic belt"
longdescription="This classic belt looks great with your favorite jeans,
or takes you to work in style. 1 1/2 inches wide in full-grain leather
with a solid nickel buckle."
thumbnail="images/mens_accessories_belt_sm.gif"
fullimage="images/mens_accessories_belt.gif"
available="1"
published="1"
/>

```

language\_id 标识产品目录信息所用的语言。该标识必须更改以匹配商店支持的每种语言。name 向客户显示，这与 shortdescription 和 longdescription 元素是相同的，这两个元素可以包含产品目录条目的简短描述和详细描述。

当创建新的产品目录条目时，请遵循以上信息的结构。

注：

1. 当删除产品目录条目时，请确保相应更新了出现的每个唯一元素。例如，如果您还希望删除产品目录组下的产品目录条目，则应该除去整个 XML 条目。然而，如果您计划保留产品目录条目，则需要将 catgroup\_id 更改到正确的组。
2. 在创建其它类型的产品目录条目之前，必须先创建产品。

如果不希望手工更改 XML 文件，可以使用“产品管理”工具。

## 产品管理工具

WebSphere 贸易加速器中的“产品管理”工具让您能够使用各种向导和笔记本来管理商店主产品目录中的产品。还可以使用“产品管理”动态表，该表允许您直接更新产品目录条目信息。可更新产品目录内容或创建新的产品目录数据：

- 使用向导或笔记本创建、更新和删除产品和产品详细信息。产品充当库存标识（即最终销售给客户的单个商品）的模板。产品详细信息包括产品代码（唯一地标识了产品）、产品名称和描述、任何销售策略选项（例如向客户显示产品或指示该产品是否是特价商品促销的一部分）、产品图像、税款和送货规格、指定给产品的折扣以及生产商信息。
- 生成、更新和删除可购买的库存标识（或商品）。库存标识表示待售商品的每个可订购商品。与特定产品相关的所有库存标识都显示同一组属性并按其属性值加以区分。对库存标识所作的添加或更改包含了与产品相同的信息，但基于可订购的除外。
- 创建、更新和删除类别（或产品目录组），它们是具有用于组织商店所提供产品或服务的类似属性的一组对象。可通过创建、更改和删除类别及其相关详细信息（例如类别代码、名称和描述，包括父类别和图像），来管理主产品目录的类别层次结构。
- 通过选择父类别或将产品和库存标识从一个类别移动到另一个类别，将产品和库存标识与类别相关联。
- 创建产品的属性和属性值。属性和属性值的每种可能组合都等于一个新库存标识。在将属性值指定给库存标识之前必须对其进行预定义。创建了属性及其值之后，可以创建或更新诸如名称、描述（文本、整数或小数）以及属性和属性值出现顺序之类的信息。
- 创建、更新、删除产品目录定价，并将之与产品相关联。可以用一种或多种货币以及一组条件（例如设置单件或成批数量的价格，为了使用此价格必须满足这些条件），为产品或库存标识定义价格。

可以参阅联机帮助中的“产品管理”部分以获取关于每个任务的详细指示信息。

#### 注:

1. 建议仅对“产品管理”工具作微小更改。对于大量产品目录更新（例如添加或删除季节性商品或准备清货销售），请使用装入程序软件包。
2. 对产品目录数据的任何更改只有禁用了高速缓存或除去了当前高速缓存的 JSP 页面时才能显示在商店中。关于更多信息，请参阅 WebSphere Commerce 联机帮助中的 CacheDelete 命令。CacheDelete 命令启动对动态页面高速缓存的远程清除，并让您能够在不需要直接访问文件系统的情况下管理高速缓存。在使用此命令之前，确保启用了“页面自动失效”。请注意，为使用该命令，您必须作为管理员登录。

## 装入程序软件包

也可通过使用装入程序软件包（先前作为产品目录管理器的一部分）来维护产品目录。装入程序软件包对于将大量现有的产品信息导入数据库是很理想的。在 WebSphere Commerce 中，这是用于创建和管理产品目录信息的主要工具。此软件包主要由用于准备数据和将数据装入 WebSphere Commerce 数据库的命令实用程序组成。装入程序软件包还允许将数据库中的数据抽取为 XML 文档、将 XML 数据转换为备用的 XML 文档，以及在以字符定界的变量格式和 XML 数据格式之间转换数据。

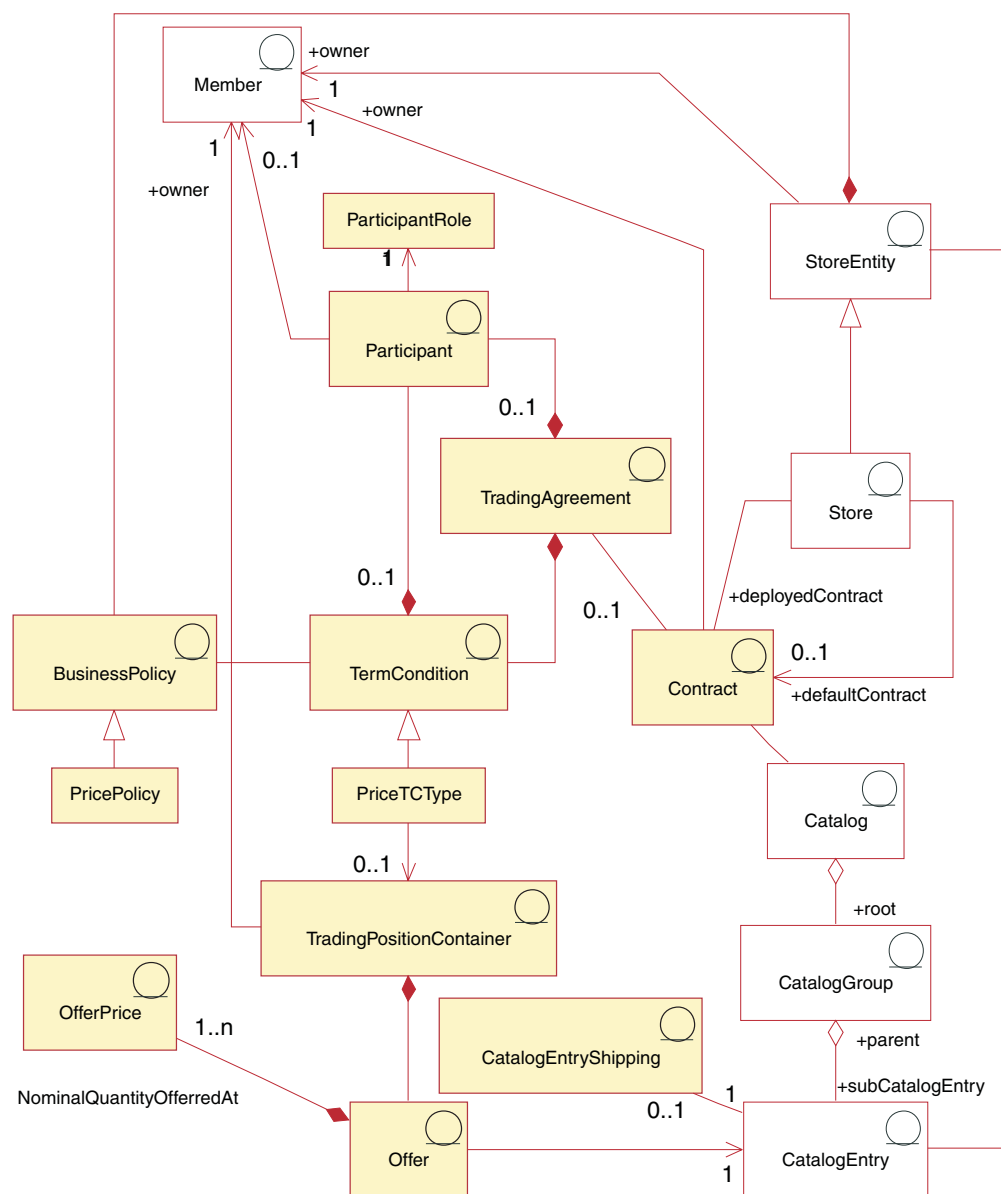
请参阅 WebSphere Commerce 生产联机帮助以获取更多信息。

## 第 17 章 对有用资源进行定价

定价代表产品目录条目的价格以及使用此价格而必须满足的所有条件。要创建正常运作的产品目录，您需要将定价信息添加到数据库中。您可以用 XML 文件（可以使用装入程序软件包装入数据库）格式创建定价信息。或者对于少量定价数据，您可以使用 WebSphere 贸易加速器的“产品管理”工具。

### 了解 WebSphere Commerce 中的定价

下图说明了 WebSphere Commerce Server 中的定价有用资源。





本图和商店数据部分中的其它所有图都是 WebSphere Commerce Server 信息模型的一部分。关于信息模型的更多信息，请参阅第 97 页的『商店数据信息模型』。关于本图中所使用的约定的更多信息，请参阅第 415 页的附录 A，『UML 图注』。

## 报价

报价（或定价）是相同产品或商品针对不同客户或组织的不同价格。报价代表产品目录条目的价格以及客户为支付此价格而必须满足的条件，例如购买数量。例如，商品或服务对儿童、学生、成人和老人通常有不同的价格。在 WebSphere Commerce 中，报价也称为贸易状态，是贸易状态容器的一部分。

## 卖价

卖价是一种价格，商店通过贸易协议或合同以此价格提供产品目录条目。报价可以具有以多种货币定义的一个或多个卖价。

## 贸易状态容器

报价是贸易状态容器（由成员所拥有）的一部分。贸易状态容器包含贸易状态。通过贸易协议或合同以及合同中的条款和条件，可以使其对所有客户都可用，或者仅对特定组中的客户可用。依据合同，贸易状态容器是价格商业对象，可以由多个价格业务策略引用，并且可以由商店组中的某个商店或所有商店共享。贸易状态容器也称为价格列表。

## 条款和条件

条款和条件定义了贸易协议的行为和属性。许多条款和条件都引用业务策略，因为商店操作的若干方面都是由业务策略定义的。

## 定价条款和条件的类型

**Professional Business** 定价条款和条件定义依据合同提供的产品以及客户为产品支付的价格。合同中至少需要以下定价条款中的一项：以下定价条款和条件可在 WebSphere Commerce 中使用：

### 定制价格列表

此条款既指定合同中要进行销售的待售产品及其价格列表是定制的，还指定这些价格也是定制的。商品并不限于商店产品目录中的某一部分，它们可以来自商店产品目录中的任何位置。

### 完整产品调价目录

该条款为商店产品目录中的所有可销售产品提供商店产品目录中定义的基本价格基础上的百分比调价（加价或折扣）。如果未指定调价，则商品以基本价格销售。

### 带有调价的价目表

该条款为价目表中的所有可销售产品提供商店产品目录中定义的基本价格基础上的百分比调价（加价或折扣）。如果未指定调价，则商品以基本价格销售。

### 带有选择性调价的价目表

该条款类似于带有调价的价目表，不同之处在于其调价并不适用于整个价目表。调价是对价格列表的子集进行的。价格列表的子集可以是产品集业务策



略，也可以是定制的产品集。关于产品集类型之间差别的信息，请参阅 WebSphere Commerce 联机帮助中的“合同条款和条件”主题。

### 带有过滤的产品目录

该条款为商店产品目录中的所有可销售产品提供商店产品目录中定义的基本价格基础上的百分比调价（加价或折扣）。该条款还为某类别中的所有可销售产品或是一系列指定产品和商品提供了基本价格基础上的百分比调价（加价或折扣），该基本价格是在该条款所参考的价目表中定义的。该条款还可以声明在合同中哪些类别、产品和商品是待售的而哪些不是的。类别产品集类似于产品集业务策略。商品产品集是定制商品集。

## 贸易协议

**Business** 贸易协议可以是合同、RFQ、商业帐户或拍卖。贸易协议是在卖方和买方之间协商的协议，根据该协议，买方可以依据合同中规定的指定条款和条件以及业务策略购买特定商品。例如，它使客户可以依据定价条款和条件在指定的时间期限内以指定的价格从商店购买产品。在 WebSphere Commerce 中，所有客户必须依据合同在商店中进行购物，一个商店可以部署一个或多个合同，且其中之一可以指定为缺省合同。缺省合同包含与一组商店缺省策略关联的一组条款和条件。贸易协议可以包含零个或多个不同角色的参与方。

## 参与方

参与方可以是贸易协议的一部分，也可以是条款和条件的一部分。参与方是个成员，可以是成员组、组织等等。如果为合同指定了买方角色的参与方，则买方必须是买方参与者的成员以根据合同进行购物。合同中的条款和条件同样也可以包含零个或多个参与方。

## 参与方角色

参与方可以具有以下参与方角色之一：

- 创建者
- 卖方
- 买方
- 供应商
- 核准员
- 帐户持有者
- 买方联系人
- 卖方联系人
- 代理人
- 管理员。

## 合同

合同包含产品的卖价。在 WebSphere Commerce 中，所有客户都必须依据合同进行购物。合同使客户可以依据合同中规定的条款和条件及业务策略，在指定的时间内以指定的价格从商店购买产品。商店拥有零个或多个合同，并拥有至少一个缺省合同。

## 业务策略

**Business** 业务策略是商店或商店组所遵循的规则集合，它们定义业务流程、行业惯例，以及商店或商店组出售物的范围和特征。业务策略的实施涉及以下各项的组合：实现业务策略规则的一个或多个业务策略命令、对规则作用的商业对象的引用以及配置业务策略命令操作的一组属性。

## 价格策略

价格策略包含对价格列表的引用，可以与多个业务策略命令关联，这些命令定义将如何在价格列表上实现业务策略。可以为商店或商店组定义策略。如果为商店组注册策略，则该组中的所有商店都可以使用此策略。

## 产品目录条目送货

产品目录条目送货信息包含关于如何包装产品进行送货的信息。每个产品目录条目可以定义不同类型的送货信息。例如，产品包装后的高度、重量和产品长度。

## 其它定价有用资源

以下有用资源与定价关联：

- 拥有贸易状态容器的成员。一个贸易状态容器仅有一个所有者。
- 商店实体表示 WebSphere Commerce Server 数据库中的商店。
- 产品目录包含将在合同中引用的产品目录条目。产品目录包含在线产品目录的所有层次结构和导航信息，是网上商店中可用于显示和购买的产品目录组和产品目录条目的集合。
- 产品目录组（或类别）是产品目录条目的一般分组，是为导航和产品目录分割而创建的。产品目录组属于产品目录，并可以包含多个产品目录组或产品目录条目。可以使产品目录组与多个产品目录关联。
- 产品目录条目代表在线产品目录中的可订购商品。产品目录条目属于产品目录组。一个报价总与一个产品目录条目关联。



关于 WebSphere Commerce Server 中定价有用资源的结构更多详细信息，请参阅 WebSphere Commerce 联机帮助中的定价对象和数据模型。

---

## 在 WebSphere Commerce 中创建定价有用资源

您有两个选项可用于创建定价有用资源：

- 使用 WebSphere 贸易加速器中的“产品管理”工具创建价格。使用 WebSphere 贸易加速器中的工具最适合为特小型产品目录创建价格。
- 在 XML 文件中创建价格，此文件可以用 WebSphere Commerce 装入程序软件包装入，或可以作为商店归档文件的一部分装入，而商店归档文件可以通过管理控制台发布。此方法最适合创建大量数据。

关于使用 WebSphere 贸易加速器中的“产品管理”工具创建价格的更多信息，请参阅 WebSphere Commerce 联机帮助。关于在 XML 文件中创建价格的更多信息，请参阅第 173 页的『在 XML 文件中创建定价有用资源』。

## 在 XML 文件中创建定价有用资源

请以 XML 文件（这些文件可以使用装入程序软件包装入数据库）格式创建定价有用资源。关于装入程序软件包的更多信息，请参阅第 309 页的第 10 部分，『发布商店』。

1. 复查用于为样本商店创建定价有用资源的 XML 文件。样本商店的所有文件都位于相应的商店归档文件中。

商店归档文件位于以下目录中：

- `WC_installdir/samplestores`

**注：**WebSphere Commerce 联机帮助包含关于样本商店中所包含的每个数据有用资源的信息。

每个样本商店包括两个 `offering.xml` 文件，其中包括定价信息。要查看商店归档文件中的 `offering.xml` 文件，请使用 ZIP 程序将其解压缩。`offering.xml` 文件位于数据目录中。特定于语言的 `offering.xml` 位于数据目录中特定于语言环境的子目录中。

2. 请复查第 417 页的附录 B，『创建数据』中的信息。
3. 通过复制样本商店归档文件中的 `offering.xml` 文件之一或通过创建新文件创建 `offering.xml` 文件。关于更多信息，请参阅对应于 `offering.xml` 的 DTD 文件。DTD 文件位于以下目录中：

- `WC_installdir/xml/sar`

4. 创建一个贸易状态容器。要为商店中的商品提供价格，您必须首先创建贸易状态容器。要创建贸易状态容器，请向 `TRADEPOSCN` 表添加信息。
  - a. 使用以下示例作为指南，在您的 XML 文件中为 `TRADEPOSCN` 表创建贸易状态容器：

```
<tradeposcn
tradeposcn_id="@tradeposcn_id_101"
  member_id="@seller_b2b_mbr_id"
markfordelete="0"
name="ToolTech"
precedence="0"

/>
```

其中

- `tradeposcn_id` 是生成的唯一键
  - `@seller_b2b_mbr_id` 是贸易状态容器的所有者。对于“时尚潮流”样本商店，请用 `@Member_ID`；替换它。
  - `markfordelete` 如下：
    - 0 = `TradingPositionContainer` 可以使用
    - 1 = `TradingPositionContainer` 已经标记为删除（请参阅 `DBClean` 实用程序）并且不应使用
  - `name` 是贸易状态容器的助记名称，对于特定所有者是唯一的。
  - `precedence` 是在特定时间有多个贸易状态容器合格时，使用 `PRECEDENCE` 最高的贸易状态容器。
5. 通过向 `CATGRPTPC` 表添加信息，使主产品目录与贸易状态容器关联。使主产品目录与贸易状态容器关联时，主产品目录中的每个产品目录条目必须有标准价格。关于创建主产品目录的更多信息，请参阅第 161 页的『显示商店产品目录有用资源』。

- a. 使用以下示例作为指南，通过向 `CATGRPTPC` 表添加信息，使主产品目录与贸易状态容器关联：

```
<catgrptpc
catalog_id="@catalog_id_1"
tradeposcn_id="@tradeposcn_id_101"
/>
```

其中

- `catalog_id` 是主产品目录。
- `tradeposcn_id` 是贸易状态容器。

6. 通过向 `OFFER` 和 `OFFERPRICE` 表添加信息，创建产品目录条目的报价和卖价

- a. 使用以下示例作为指南，通过向 `OFFER` 表添加信息，为产品目录条目创建报价。请注意，在可以创建价格之前必须已创建产品目录条目。关于创建产品目录条目的更多信息，请参阅第 161 页的『显示商店产品目录有用资源』。

```
offer
offer_id="@offer_id_138"
startdate="2000-06-19 00:00:00.000000"
catentry_id="@product_id_102"
precedence="0"
published="1"
identifier="1"
  flags="1"
tradeposcn_id="@tradeposcn_id_101"
/>
```

其中

- `offer_id` 是生成的唯一键。
  - `startdate` 是该报价有效的时间范围的开始时间。
  - `catentry_id` 是提供待售的产品目录条目。
  - `precedence` 是在特定时间有多个报价有效时，使用 `PRECEDENCE` 最高的报价。
  - `published` 是
    - 0 = 未发布（临时禁用）
    - 1 = 已发布
    - 2 = 标记为删除（且未发布）
  - `identifier` 是与其指定的产品目录条目和贸易状态容器一起唯一标识此报价的数值。
  - `flags` 是
    - 1 = `shiptoAddressRequired` – 如果为 1，则当 `OrderItem` 引用此 `Offer` 但是没有送货地址时，`OrderPrepare` 将返回错误。
  - `tradeposcn_id` 是贸易状态容器，此报价是该贸易状态容器的组成部分。
- b. 使用以下示例作为指南，通过向 `OFFERPRICE` 表添加信息，为产品目录条目创建卖价。卖价是产品目录条目提供待售的实际价格。请注意，在可以创建价格之前必须已创建产品目录条目。关于创建产品目录条目的更多信息，请参阅第 161 页的『显示商店产品目录有用资源』。

```
<offerprice
offer_id="@offer_id_138"
currency="USD"
price="590.00"
/>
```

其中

- offer\_id 是与此价格关联的报价。
- currency 是提供此价格所用的货币。
- price 是报价所引用的产品额定数量（请参阅 CATENTSHIP.NOMINALQUANTITY）的价格。

**注：**要在商店中显示多种货币，请在 OFFERPRICE 表中为每种货币创建一个单独的 XML 条目。例如，要以加拿大元显示货币，请在新 XML 条目中使用 currency="CAD"。price 值将更改为反映使用加拿大元计算的价格。或者，您可以使用转换，使客户可以基于所选择的货币显示不同的兑换率。关于更多信息，请参阅第 213 页的『使用 XML 文件创建货币有用资源』。

- c. 对产品目录中所有的产品目录条目重复步骤 a 和 b。



关于 @ 和 & 用法的更多信息，请参阅第 417 页的附录 B，『创建数据』。

---



---

## 第 18 章 合同有用资源

在 WebSphere Commerce 中，所有客户都必须依据合同进行购物。合同使客户可以依据特定条件，在指定的时间内以指定的价格从商店购买产品。当浏览商店的产品目录时，客户仅可看见商店内对其授权的合同中所涵盖的产品。

如果您希望与商店没有任何合同的客户（例如临时购物者）可以在商店中购物，或者如果您希望客户可以购买他们的合同中未涵盖的产品，则您的商店将需要缺省合同。

### 重要信息

WebSphere Commerce Professional Edition 和 WebSphere Commerce - Express 仅支持商店缺省合同。

除商店缺省合同之外的合同仅受 WebSphere Commerce Business Edition 支持。

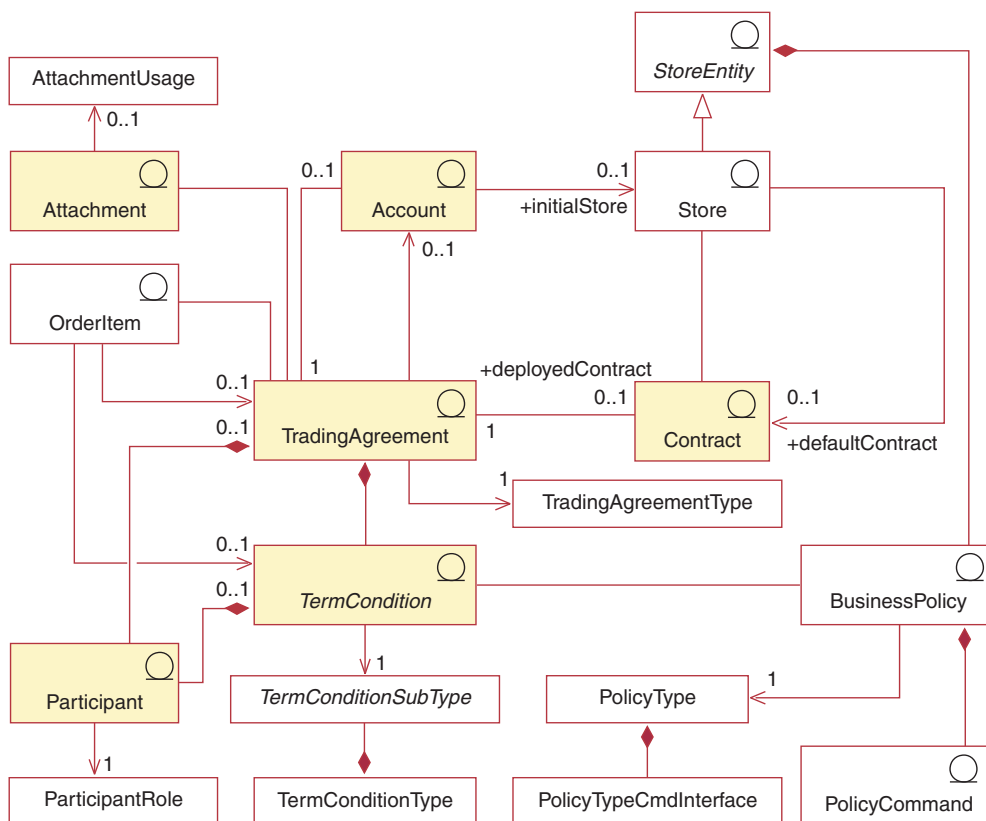
要使所有客户都可以在商店中购物，用 WebSphere Commerce 创建的商店必须包含以下信息：

- 业务策略
- 缺省合同

业务策略由缺省合同引用，由此使所有客户都能在商店中购物。

## 了解 WebSphere Commerce 中的合同

下图说明了 WebSphere Commerce 中的合同结构:



本图和商店数据部分中的其它所有图都是 WebSphere Commerce Server 信息模型的一部分。关于信息模型的更多信息, 请参阅第 97 页的『商店数据信息模型』。关于本图中所使用的约定的更多信息, 请参阅第 415 页的附录 A, 『UML 图注』。

## 帐户 (商业帐户)

Business

商业帐户代表买方组织和卖方组织之间的关系。商业帐户可以用于组织各种贸易协议, 还可以用于指定与买方和卖方之间的关系相关的条款和条件, 例如: 发票定制、购买订单验证或维持买方在卖方下的信用限额。

合同与商业帐户关联, 因为其代表买方和卖方之间的协议。此处的特例是不能与商业帐户关联的商店缺省合同。商业帐户可以有很多合同与之关联。

商业帐户是一种贸易协议类型。关于贸易协议的描述, 请参阅第 179 页的『贸易协议』。

**重要信息:** 商业帐户仅受 WebSphere Commerce Business Edition 支持。



## 合同

有两种类型的与商店关联的活动合同：已部署合同和缺省合同。已部署合同对特定买方组织或个人买方作了授权，并可以在创建商店之后使用 WebSphere 贸易加速器进行创建。一个已部署合同与一个商业帐户相关联。缺省合同定义了商店对与商店没有其它任何合同的买方所采取的缺省行为。仅可使用 XML 创建缺省合同，且对一个商店只能定义一个缺省合同。关于合同的更多信息，请参阅 WebSphere Commerce 生产和开发联机帮助信息。关于创建缺省合同有用资源的信息，请参阅第 184 页的『在 WebSphere Commerce 中创建缺省合同有用资源』。

典型合同由以下元素组成：

### 概要文件

包含合同标识信息的合同概要文件。该信息包括合同的唯一名称、简短描述以及合同有效的时间段。

**参与方** 合同参与方是参与合同的组织。有买方组织、卖方组织以及双方组织的联系人。

### 条款和条件



合同条款和条件是涵盖合同实际实现的规则。合同条款和条件涵盖诸如产品定价、退货与退款、支付、送货以及订单核准等信息。

**附件** 合同附件包含未包含在先前元素中的所有信息，例如提供关于合同的附加信息以及关于合同的任何通用注解的文件附件。WebSphere Commerce 存储合同附件的全局资源标识（URI），而非实际附件。

**引用** 合同可以引用另一合同以共享其条款和条件。例如，合同 A 可以引用合同 B。这样，已受合同 A 约束的买方会受合同 A 的所有条款和条件的约束，以及合同 B 中的所有条款和条件的约束。

## 贸易协议


合同是一种贸易协议类型。WebSphere Commerce 提供了很多贸易机制管理买方与卖方之间的交互。以下贸易机制受不同版本的 WebSphere Commerce 支持：


- 拍卖（Business Edition 与 Professional Edition 均支持）
-  商业帐户
- 合同（请参阅本章先前所讨论的限制条件）
-  报价请求（RFQ）

所有这些贸易机制都具有公共属性。例如，所有的贸易机制都有参与方，并且都有用于管理贸易机制行为的规则。在 WebSphere Commerce 中，管理贸易机制行为的规则称为**条款和条件**。

贸易协议代表贸易机制的一个实例，并记录贸易机制的该实例的属性。WebSphere Commerce 中的每个合同、商业帐户和  RFQ 都由一个贸易协议表示。有一个管理 WebSphere Commerce 中所有拍卖的贸易协议。

贸易协议的组成部分包括：存储在 TRADING 表中的概要文件；存储在 PARTICIPANT 表中的参与方；存储在 TERMCOND 表中的条款和条件；以及作为“全局资源标识”

(URI) 存储在 ATTACHMENT 表中的可选附件。因为一个贸易协议可以有多个附件，所以附件通过 TRDATTACH 表与贸易协议相关。请注意， RFQ 不支持附件。

除了一般贸易协议外，每种贸易协议还在它自己的表中存储特定于该类贸易协议的附加信息：CONTRACT 存储特定于合同的信息； RFQ 存储特定于 RFQ 的信息；而 ACCOUNT 存储特定于商业帐户的信息。

## 参与方

合同参与方在每个合同中承担各种特定角色。参与方可以是买方组织的联系人也可以是卖方组织的联系人。如果合同指定买方参与者为空，那么所有用户（包括临时用户）都会受该合同约束。任何合同都可以指定买方参与者为空。

## 条款和条件

条款和条件定义贸易协议的行为和属性。对于合同，条款和条件定义了如何对买方组织实现合同。它们定义将依据合同进行销售的商品；要销售的商品的价格；如何发送商品；订单的支付方式；如何处理退货；如何核准订单以及从何处发送订单中的商品。

一些条款和条件引用业务策略，因为商店操作的很多方面都是由业务策略定义的。条款和条件为其所引用的业务策略提供参数。向业务策略提供参数使您可以对每个合同的业务策略的行为进行修改。WebSphere Commerce 支持以下条款和条件（引用业务策略的条款和条件以星号 (\*) 指示）。

### 供货中心

该可选条款使您能够指定一组供货中心，在其中必须对遵守合同的订单完成供货。该列表必须是为商店定义的供货中心的子集。供货中心优先级由商店定义，并且不被合同的条款和条件覆盖。

### 订单核准

该条款指定在订单完成前客户组织是否必须核准这些订单。您可以指定可选金额（包括税款和送货费用），这样就可使价值低于一定金额的订单能够完成，而无需由客户组织核准。如果订单总额超过该金额，则必须核准。如果买方所下订单中的商品包含在多个合同中，而且订单中的一个商品有一个指定该条款的合同，那么整个订单就受适用于该商品的订单核准条款的约束。

### 支付方式\*

该可选条款指定受合同约束的订单会接受的支付方式。支付方式可以是一般的支付类型，例如信用卡类型，也可以具体为用于支付的信用卡号。如果合同中指定支付方式条款，则受合同约束的订单将接受商店接受的所有支付方式。

### 定价条款和条件

定价条款和条件定义依据合同可用的产品以及客户为产品支付的价格。在合同中至少需要一个定价条款。以下定价条款和条件可用于 WebSphere Commerce:

#### 定制价格列表

此条款既指定合同中要进行销售的待售产品及其价格列表是定制的，还指定这些价格也是定制的。商品不限于商店产品目录中的某一部分，可以来自商店产品目录中的任何地方。

### **完整产品调价目录**

该条款为商店产品目录中的所有可销售产品提供商店产品目录中定义的基本价格基础上的百分比调价（加价或折扣）。如果未指定调价，则商品以基本价格销售。

### **有调价的价目表\***

该条款为价目表中的所有可销售产品提供商店产品目录中定义的基本价格基础上的百分比调价（加价或折扣）。如果未指定调价，则商品以基本价格销售。

### **具有选择性调价的价目表\***

该条款类似于带有调价的价目表，不同之处在于其调价并不适用于整个价目表。调价是对价格列表的子集进行的。价格列表的子集可以是产品集业务策略，也可以是定制的产品集。关于产品集类型间差别的信息，请参阅 WebSphere Commerce 开发联机帮助。

### **过滤产品目录 \***

该条款为商店产品目录中的所有可销售产品提供商店产品目录中定义的基本价格基础上的百分比调价（加价或折扣）。该条款还为某类别中的所有可销售产品或是一系列指定产品和商品提供了基本价格基础上的百分比调价（加价或折扣），该基本价格是在该条款所参考的价目表中定义的。该条款还可以声明在合同中哪些类别、产品和商品是待售的而哪些不是的。类别产品集类似于产品集业务策略。商品产品集是定制商品集。

## **产品约束条款和条件**

产品约束条款和条件规定按合同要销售哪些产品或不销售哪些产品。产品约束条款是可选的。如果在合同中没有指定产品约束条款和条件，那么合同的价格条款和条件所指定的所有产品都是按合同待售的。以下产品约束条款和条件可用于 WebSphere Commerce:

### **排除定制产品集**

该条款在合同中声明某定制产品集中的商品为非待售商品。

### **包含定制产品集**

该条款在合同中声明某定制产品集中的商品为待售商品。

### **排除产品集\***

该条款在合同中声明某产品集业务策略中的商品为非待售商品。

### **包含产品集\***

该条款在合同中声明某产品集业务策略中的商品为待售商品。

排除条款的优先级在包含条款之上。这意味着如果某产品同时出现在合同的包含条款和排除条款中，那么在该合同中该产品不能出售。关于定制产品集和产品集业务策略间差别的信息，请参阅 WebSphere Commerce 开发联机帮助。

## **退货条款和条件**

退货条款和条件指定按合同如何处理退货。如果没有指定退货条款和条件，那么就无法建立退货过程。如果指定了退货条款和条件，那么它们只应是一个适用于整个合同的条款和条件集。以下退货条款和条件可用于 WebSphere Commerce:

### 退款支付方式\*

该条款指定向客户退款时使用的支付方式。如果指定了退货收费条款，那么至少要同时指定一个退款支付方式条款。如果合同中不允许退货，那么可以不指定该条款。

### 退货收费\*

该条款指定如何自动核准退货以及处理退货是否要从退款中扣除任何费用，例如，重新进货费用。

### 购买权金额

该条款在合同中规定所有订单总价值（包括税款和送货费用）的限制。合同中所有订单的总价值必须小于或等于某指定金额。如果下订单时超出了该限制，那么订单的支付授权会失败。

### 送货条款和条件

送货条款和条件指定如何发送订单中的商品、要把它们发送到哪里以及谁将支付送货费用。以下送货条款和条件可用于 WebSphere Commerce:

### 送货方式\*

该可选条款定义如何送货合同约定下产生的订单中的商品。如果合同中没有指定该条款，那么可以使用商店中可用的任何方式发送订单中的商品。送货方式也称作送货提供者。送货提供者是货物递送者及其送货服务的组合。例如，“XYZ 速递”、“隔夜”是送货提供者。

### 送货地址

该可选条款定义合同中所购产品将发送到何处。通过指定此条款和条件，您能够限制订单中商品的送货目的地。如果没有指定送货地址条款和条件，那么每次按合同下订单时都必须指定送货地址。如果指定了该条款，那么在下订单时买方不能指定新的送货地址，但是必须从送货地址列表中选择一个送货地址。

### 送货收费类型\*

该条款定义谁支付订单中商品的送货费用。以下送货费用类型是受支持的:

- 送货费用由买方支付给卖方。卖方在订单交易成功且送货费用是订单总额的一部分后计算送货费用。
- 送货费用由买方支付给递送者。递送者计算送货费用，并承担向买方收取支付费用的职责。送货费用不在订单交易成功时计算。

### 参照接口 \*

该条款指定商店和远程商店间的关系。它定义远程商店所支持的功能，以及向远程商店发送的消息中要使用的参数。

## 业务策略

业务策略是商店或商店组所遵循的规则集。业务策略定义业务过程、行业惯例，以及商店或商店组出售物的范围和特征。它们是商店或商店组内所有允许的和支持的惯例的中心源和引用模板。

在 WebSphere Commerce 中，业务策略的实施涉及以下各项的组合：实现业务策略规则的一个或多个业务策略命令、对规则作用的商业对象的引用以及配置业务策略命令操作的一组属性。条款和条件可以为其所引用的业务策略提供参数。这使得可以根据引用业务策略的条款和条件来修改业务策略的行为。

**Business** 业务策略是可共享的资源。当您列出可在合同中使用的业务策略时，所列出的业务策略是由创建合同处的商店所拥有的业务策略，以及由任何存在 `com.ibm.commerce.businessPolicy` 商店关系的商店所拥有的业务策略。关于某一站点内的众多商店共享有用资源的更多信息，请参阅第 129 页的第 14 章，『商店间的关系』。

WebSphere Commerce 中提供了以下业务策略类别：

#### 产品目录业务策略

产品目录业务策略定义商店中待售产品目录的范围和特征，包括定义商店产品目录中的产品的价格和分类。

#### 支付业务策略

发票、支付和退款业务策略定义一个商店如何接受支付、如何支付退款并定义商店发票的格式。

#### 退货业务策略

退货业务策略定义是否接受退款、接受退货的时间段以及对退货是否要收取任何重新进货的费用。

#### 送货业务策略

送货业务策略定义商店可以使用的送货提供者，以及与每种类型相关联的费用。

#### 参照接口业务策略

参照接口业务策略定义代理商店和远程商店间的关系。

许多合同条款和条件会引用业务策略。这不仅提供对商店将履行的各种合同的基本部分的一种控制方法，同时还提供制定合同条款和条件的灵活性。关于业务策略的更多信息，请参阅《WebSphere Commerce 编程指南与教程》。

## 附件

附件提供了关于贸易协议的附加信息，该信息未由该贸易协议中的其它元素所涵盖。

示例是一个文件，它提供关于 RFQ 要求和有关 **Business** RFQ 的任何一般注解的附加信息。贸易协议可具有多个附件。附件存储在 WebSphere Commerce 之外，且贸易协议存储了附件的全局资源标识（URI）。URI 的示例包含以下项：

- `http://www.mycompany.com/information/document1.txt`
- `file:///home/joeuser/mydocs/document1`
- `ftp://ftp.mycompany.com/information/attachment.txt`

可以为所有附件指定附件用法，该用法指示附件的用途。附件用法是附件的可选属性。

## 订购商品

订购商品是包含在订单中的产品或商品。一份订单中的不同订购商品可以依据不同的合同贸易协议购买。根据商店设计，买方可以在购物流程开始时或在向其订单添加商品时，选择进行购物所依据的合同贸易协议。依据不同的合同贸易协议购买商品时，应用以下规则：

- 订单中所有商品的合同贸易协议必须至少共享一种支付方式。如果某个商品的合同不共享支付方式，则买方无法将此商品添加到订单。只有订单中所有商品都共享的支付方式才可以用于支付订单。

- 订单中的所有商品必须来自商店缺省合同或属于同一个商业帐户的合同贸易协议。



关于 WebSphere Commerce 中合同有用资源的结构更多详细信息，请参阅 WebSphere Commerce 开发联机帮助中的合同数据模型。

## 在 WebSphere Commerce 中创建缺省合同有用资源

缺省合同定义商店的缺省行为。对于所有合同，可设置可供产品、价格、支付方式、送货方式和其它商店行为。

随 WebSphere Commerce 样本商店提供的商店缺省合同所包含的条款和条件指定了以下内容：

- 客户可按主产品目录中设置的标准价格（没有折扣或加价）购买商店主产品目录中所有可供产品。
- 所有的送货费用都支付给卖方（商店）。
- 客户可在特定天数之内退回购买的产品，而不受到惩罚性收费。
- 客户可使用与用于原始购买的支付方式相同的支付方式接收退款。

并且，商店缺省合同的最普通版本省略了限制客户可使用的支付和送货方式的条款和条件。省略这些条款将让买方能够使用商店支持的任何缺省支付方式对购买进行支付，且能够使用商店中任何可用的送货方式。

缺省合同的属性定义在其条款和条件中。一些条款和条件引用了业务策略。关于业务策略以及条款和条件的更多信息，请参阅 WebSphere Commerce 开发联机帮助。

要创建缺省合同有用资源，请执行以下操作：

1. 复查关于条款和条件、合同、缺省合同以及业务策略的联机信息。
2. 复查 `wcs.bootstrap.xml` 文件中定义的业务策略。有关 `wcs.bootstrap.xml` 文件的信息，请参阅联机信息。
3. 复查用于为样本商店创建缺省合同有用资源的文件。所有的样本商店文件都位于对应的商店归档文件中。每个样本商店都包含 `businesspolicy.xml` 和 `contract.xml`，它们包含了附加业务策略信息和缺省合同信息。商店归档文件在 `WC_installdir/samplestores` 目录中。

**注：**

- a. WebSphere Commerce 联机帮助包含关于样本商店中所包含的每个数据有用资源的信息。
  - b. 要查看商店归档文件中的 `businesspolicy.xml` 和 `contract.xml` 文件，请使用 ZIP 程序将其解压缩。这些文件位于数据目录中。
  - c. 随 WebSphere Commerce Business Edition 提供的“多乐五金店”样本商店的合同有用资源文件包含了除商店缺省合同之外的合同的信息。
4. 请复查第 417 页的附录 B，『创建数据』中的信息。
  5. 通过复制样本商店归档文件中的一个 `businesspolicy.xml` 文件，或者通过创建新文件，来创建 `businesspolicy.xml` 文件。关于创建新文件的指示信息位于第 185 页的『创建业务策略 XML 文件』中。如果希望创建与所讨论策略不同的业务策略，请参阅与 `businesspolicy.xml` 对应的 DTD 文件。DTD 文件在 `WC_installdir/xml/sar` 目录中。

6. 使用装入程序软件包装入 `businesspolicy.xml` 文件。关于装入程序软件包的更多信息，请参阅第 309 页的第 10 部分，『发布商店』。如果正在创建多文化商店，则可能为您的商店支持的每种语言环境创建独立的 XML 文件。特定于语言环境的文件应当指定所有描述信息，以便比较容易进行翻译。
7. 通过复制样本商店归档文件中的一个 `contract.xml` 文件，或者通过创建新文件，来创建 `contract.xml` 文件。创建新文件的指示信息位于第 186 页的『创建缺省合同文件』中。如果希望创建更为复杂的缺省合同，请复查 `B2BTrading.dtd` 或 `Package.xsd` 文件，它们定义了合同文件的结构。`B2BTrading.dtd` 文件位于 `WC_installdir/xml/trading/dtd` 目录中；`Package.xsd` 文件位于 `WC_installdir/xml/trading/xsd` 目录中。
8. 使用 `ContractImportApprovedVersion` 命令发布合同。关于更多信息，请参阅第 377 页的第 39 章，『发布商业帐户和合同』。WebSphere Commerce 开发联机帮助中也提供了关于 `ContractImportApprovedVersion` 命令的信息。

WebSphere Commerce Business Edition 用户可使用 WebSphere 贸易加速器为特定客户定义合同。关于为特定客户制定合同的更多信息，请参阅 WebSphere Commerce 生产联机帮助。

## 创建业务策略 XML 文件

虽然 WebSphere Commerce 提供很多业务策略可供商店缺省合同中的条款和条件引用，但是您仍然必须定义一些业务策略。您必须定义商店缺省合同条款将要引用的所有退货收费、退货核准以及定价业务策略。这些业务策略的命令已提供并且可以在不修改的情况下使用。如果希望创建您自己的业务策略，请参阅《WebSphere Commerce 编程指南与教程》。

要创建商店的业务策略，您必须创建业务策略并使一个或多个命令与此业务策略关联。要创建业务策略，请向 `POLICY` 表添加信息。要使命令与业务策略关联，请向 `POLICYCMD` 表添加信息。

要创建业务策略并使命令与此策略关联，请执行以下操作：

1. 通过向 `POLICY` 表添加信息，在您的业务策略 XML 文件中创建业务策略。使用以下示例作为指南：

```
<policy
policy_id="@policy_id_10"
policyname="MasterCatalogPriceList"
policytype_id="Price"
  storeent_id="@storeent_id_1"
properties="name=&STORE_IDENTIFIER;&orgentity_dn=ORGANIZATION_DN
/>
```

其中

- `policy_id` 是业务策略的唯一的数字标识。
- `policyname` 是此业务策略的唯一名称。
- `policytype_id` 是正在定义的策略的类型。有效的 `Valid policytype_ids` 是：
  - `InvoiceFormat`
  - `Payment`
  - `Price`
  - `ProductSet`

- ReturnApproval
  - ReturnCharge
  - ReturnPayment
  - ShippingCharge
  - ShippingPayment
  - ReferralInterface
- storeent\_id 是商店或商店组。
  - properties 是“名称值”对的列表，将该列表发送到业务策略命令。
2. 通过向 POLICYCMD 表添加信息，使命令与业务策略 XML 文件中的业务策略关联。使用以下示例作为指南：

```
<polycmd
policy_id="@policy_id_10"
businesscmdclass=
"com.ibm.com.commerce.price.commands.RetrievePricesCmdImpl"
/>
```

其中

- policy\_id 是命令与之关联的业务策略的数字标识。
- businesscmdclass 是实现业务策略的 Java 类的名称。

businesscmdclass 属性中的断行仅出于显示目的。



关于 @ 和 & 用法的更多信息，请参阅第 417 页的附录 B，『创建数据』。

## 创建缺省合同文件

要创建缺省合同，您必须定义合同、合同所有者、合同描述、合同参与方以及合同的条款和条件。合同信息存储在四个表中：CONTRACT、PARTICIPANT、TRADING 和 TERMCOND。

使用 STOREDEF 数据库表将缺省合同与商店关联。对于 WebSphere Commerce Business Edition 用户，使用 STORECNTR 数据库表将除缺省合同之外的合同与商店关联。

您可以基于以下两个格式中的一项（XSD 或 DTD），以 XML 创建缺省合同。请参阅下面各部分，以获取关于如何创建每种类型的详细信息。

### 创建 XSD 格式的缺省合同文件

要创建 XSD 格式的缺省合同，请执行以下操作：

1. 在您的 XML 文件中定义缺省合同。缺省合同如下定义在 XML 文件的开头：

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://www.ibm.com/WebSphereCommerce"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.ibm.com/WebSphereCommerce Package.xsd">
<BuyerContract state="Active" contractUsage="Default" comment="">
<ContractUniqueKey majorVersionNumber="1" minorVersionNumber="0"
name="&STORE_IDENTIFIER; Default Contract" origin="Manual">
<ContractOwner>
<OrganizationRef distinguishName="ou=&ORGENTITYNAME;,&ORGANIZATION_DN;" />
</ContractOwner></ContractUniqueKey>
```



请注意，在 Package 和 ContractUniqueKey 元素中的断行只是出于显示的目的。

2. 在您的合同 XML 文件中定义合同参与方。使用以下示例作为指南：

```
<Participant role="Buyer">
</Participant>
<Participant role="Seller">
  <ParticipantMember>
    <OrganizationRef distinguishName="ou=&ORGENTITYNAME;,&ORGANIZATION_DN;" />
  </ParticipantMember>
</Participant>
```

其中 distinguishName 是 LDAP 专有名称格式的作为此合同卖方的用户的名称。例如，uid=johnsmith,ou=People,o=ibm,o=com。在很多情况下，这将与合同所有者相同。

**注：**没有成员指定为买方参与者角色，因为合同对具有买方角色的所有用户都可用。

3. 在您的合同 XML 文件中定义合同描述。使用以下示例作为指南：

```
<ContractDescription title="This is a store default contract." locale="en_US"/>
```

其中

- title 是合同的文本描述。
- locale 是标题所用语言的语言环境。对于 locale，预定义了以下值：
  - en\_US (美国英语)
  - fr\_FR (法语)
  - de\_DE (德语)
  - it\_IT (意大利语)
  - es\_ES (西班牙语)
  - pt\_BR (巴西葡萄牙语)
  - zh\_CN (简体中文)
  - zh\_TW (繁体中文)
  - ko\_KR (韩国语)
  - ja\_JP (日语)

可以通过更新商店的语言有用资源，对 locale 定义其它值。关于语言有用资源的更多信息，请参阅第 209 页的第 22 章，『语言有用资源』。

4. 在您的合同 XML 文件中定义条款和条件。对于不同类型的条款和条件，XML 元素和属性是不同的。使用 Package.xsd 文件学习用于每种条款类型的 XML 元素和属性。定义条款和条件时，通常使用以下属性：

#### **policyName**

条款和条件引用的业务策略的名称。此名称存储在 POLICY.POLICYNAME 中。

#### **策略引用**

条款和条件引用的业务策略的类型。有效值为：

- PricePolicyRef
- ProductSetPolicyRef
- InvoiceFormatPolicyRef

- PaymentPolicyRef
- ReturnApprovalPolicyRef
- ReturnChargePolicyRef
- ReturnPaymentPolicyRef
- ShippingChargePolicyRef
- ShippingModePolicyRef

### storeRef

条款和条件的商店或商店组。

### distinguishName

拥有商店或商店组的用户的名称。名称必须是 LDAP 专有名称格式。例如，uid=wcsadmin,o=Root Organization。

以下样本条款和条件跟在对其所定义内容的描述之后：

- 所有买方都可以按照主产品目录中设置的价格购买商店主产品目录中的所有商品：

```
<PriceTCMasterCatalogWithOptionalAdjustment>
</PriceTCMasterCatalogWithOptionalAdjustment>
```

- 买方向卖方支付所有的送货费用：

```
<ShippingTCShippingCharge>
<ShippingChargePolicyRef policyName="StandardShippingChargeBySeller">
  <StoreRef name="&STORE_IDENTIFIER;">
    <Owner>
      <OrganizationRef distinguishName="ou=&ORGENTITYNAME;,
&ORGANIZATION_DN;" />
    </Owner>
  </StoreRef>
</ShippingChargePolicyRef>
</ShippingTCShippingCharge>
```

- 买方可以退回产品而不支付任何退货收费。必须在 ApprovalByDays 业务策略中定义的天数内退回产品：

```
<ReturnTCReturnCharge>
<ReturnChargePolicyRef policyName="NoCharges">
  <StoreRef name="&STORE_IDENTIFIER;">
    <Owner>
      <OrganizationRef distinguishName="ou=&ORGENTITYNAME;,
&ORGANIZATION_DN;" />
    </Owner>
  </StoreRef>
</ReturnChargePolicyRef>
<ReturnApprovalPolicyRef policyName="ApprovalByDays">
  <StoreRef name="&STORE_IDENTIFIER;">
    <Owner>
      <OrganizationRef distinguishName="ou=&ORGENTITYNAME;,
&ORGANIZATION_DN;" />
    </Owner>
  </StoreRef>
</ReturnApprovalPolicyRef>
</ReturnTCReturnCharge>
```

### WebSphere Commerce Business Edition 用户请注意：

从商店缺省合同中省略这些条款和条件指示了在缺省情况下商店不接受退货。但是其它合同可能通过定义退货条款和条件，允许买方退货。

## WebSphere Commerce Professional Edition 用户请注意:

从商店缺省合同中省略这些条款和条件指示了商店不接受退货。

- 使用与完成订单时买方所使用的同一支付方式来支付退款:

```
<ReturnTCRefundPaymentMethod>
<ReturnPaymentPolicyRef policyName="UseOriginalPayment">
  <StoreRef name="&STORE_IDENTIFIER;">
    <Owner>
      <OrganizationRef distinguishName="ou=&ORGENTITYNAME;,
&ORGANIZATION_DN;" />
    </Owner>
  </StoreRef>
</ReturnPaymentPolicyRef>
</ReturnTCRefundPaymentMethod>
```

5. 如下结束 XML 文件:

```
</BuyerContract>
</Package>
```



关于 @ 和 & 用法的更多信息, 请参阅第 417 页的附录 B, 『创建数据』。

## 创建 DTD 格式的缺省合同文件

要创建缺省合同, 您必须定义合同、合同所有者、合同描述、合同参与方以及合同的条款和条件。合同信息存储在四个表中: CONTRACT、PARTICIPANT、TRADING 和 TERMCOND。

使用 STOREDEF 数据库表将缺省合同与商店关联。对于 WebSphere Commerce Business Edition 用户, 使用 STORECNTR 数据库表将除缺省合同之外的合同与商店关联。

要创建 DTD 格式的缺省合同, 请执行以下操作:

1. 在您的 XML 文件中定义缺省合同。缺省合同如下定义在 XML 文件的开头:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE Trading SYSTEM "B2BTrading.dtd">
<Trading>
<Contract state="Active" origin="Manual"
  name="&STORE_IDENTIFIER; Default Contract" majorVersionNumber="1"
  minorVersionNumber="0" contractUsage="Default">
```

请注意, Contract 元素中的断行仅出于显示目的。

2. 定义合同所有者。使用以下示例作为指南:

```
<ContractOwner>
  <Member>
    <Organization distinguishName="ou=&ORGENTITYNAME;,&ORGANIZATION_DN;" />
  </Member>
</ContractOwner>
```

其中 distinguishName 是 LDAP 专有名称格式的拥有合同的用户的名称。例如, uid=johnsmith,ou=People,o=ibm,o=com。

3. 在您的合同 XML 文件中定义合同描述。使用以下示例作为指南:

```
<ContractDescription title="This is a store default contract." languageId="-1">
</ContractDescription>
```

其中

- title 是合同的文本描述。

- languageId 是标题使用的语言。对于 languageId 预定义了以下值：
  - -1 (英语 - 美国)
  - -2 (法语)
  - -3 (德语)
  - -4 (意大利语)
  - -5 (西班牙语)
  - -6 (巴西葡萄牙语)
  - -7 (简体中文)
  - -8 (繁体中文)
  - -9 (韩国语)
  - -10 (日语)

可以通过更新商店的语言有用资源，对 languageId 定义附加值。关于语言有用资源的更多信息，请参阅第 209 页的第 22 章，『语言有用资源』。

4. 在您的合同 XML 文件中定义合同参与方。使用以下示例作为指南：

```
<Participant role="Buyer">
</Participant>
<Participant role="Seller">
  <Member>
    <Organization distinguishName="ou=&ORGENTITYNAME;,&ORGANIZATION_DN;" />
  </Member>
</Participant>
```

其中 distinguishName 是 LDAP 专有名称格式的作为此合同卖方的用户的名称。例如，uid=erickoeck,ou=People,o=ibm,o=com。在很多情况下，这将与合同所有者相同。

**注：**没有成员指定为买方参与者角色，因为合同对具有买方角色的所有用户都可用。

5. 在您的合同 XML 文件中定义条款和条件。对于不同类型的条款和条件，XML 元素和属性是不同的。使用 B2BTrading.dtd 文件来了解用于每种条款类型的 XML 元素和属性。定义条款和条件时，通常使用以下属性：

#### **policyName**

条款和条件引用的业务策略的名称。此名称存储在 POLICY.POLICYNAME 中。

#### **policyType**

条款和条件引用的业务策略的类型。有效值为：

- Price
- ProductSet
- InvoiceFormat
- Payment
- ReturnApproval
- ReturnCharge
- ReturnPayment
- ShippingCharge
- ShippingMode

## storeIdentity

条款和条件的商店或商店组。

## distinguishName

拥有商店或商店组的用户的名称。名称必须是 LDAP 专有名称格式。例如，uid=wcsadmin,o=Root Organization。

以下样本条款和条件跟在对其所定义内容的描述之后：

- 所有买方都可以按照主产品目录中设置的价格购买商店主产品目录中的所有商品：

```
<TermCondition>
  <PriceTC>
    <PriceTCMasterCatalogWithOptionalAdjustment>
    </PriceTCMasterCatalogWithOptionalAdjustment>
  </PriceTC>
</TermCondition>
```

- 买方向卖方支付所有的送货费用：

```
<TermCondition>
  <ShippingTC>
    <ShippingTCShippingCharge>
      <PolicyReference policyName="StandardShippingChargeBySeller"
        policyType="ShippingCharge" storeIdentity="&STORE_IDENTIFIER;">
        <Member>
          <Organization distinguishName="ou=&ORGENTITYNAME;,
            &ORGANIZATION_DN;" />
        </Member>
      </PolicyReference>
    </ShippingTCShippingCharge>
  </ShippingTC>
</TermCondition>
```

PolicyReference 元素中的断行仅出于显示目的。

- 买方可以退回产品而不支付任何退货收费。必须在 ApprovalByDays 业务策略中定义的天数内退回产品：

```
<TermCondition>
  <ReturnTC>
    <ReturnTCReturnCharge>
      <ReturnChargePolicyReference>
        <PolicyReference policyName="NoCharges"
          policyType="ReturnCharge"
          storeIdentity="&STORE_IDENTIFIER;">
          <Member>
            <Organization distinguishName="ou=&ORGENTITYNAME;,
              &ORGANIZATION_DN;" />
          </Member>
        </PolicyReference>
      </ReturnChargePolicyReference>
      <ReturnApprovalPolicyReference>
        <PolicyReference policyName="ApprovalByDays"
          policyType="ReturnApproval"
          storeIdentity="&STORE_IDENTIFIER;">
          <Member>
            <Organization distinguishName="ou=&ORGENTITYNAME;,
              &ORGANIZATION_DN;" />
          </Member>
        </PolicyReference>
      </ReturnApprovalPolicyReference>
    </ReturnTCReturnCharge>
  </ReturnTC>
</TermCondition>
```

PolicyReference 元素中的断行仅出于显示目的。

**WebSphere Commerce Business Edition 用户请注意:**

从商店缺省合同中省略这些条款和条件指示了在缺省情况下商店不接受退货。但是其它合同可能通过定义退货条款和条件，允许买方退货。

**WebSphere Commerce Professional Edition 用户请注意:**

从商店缺省合同中省略这些条款和条件指示了商店不接受退货。

- 使用与完成订单时买方所使用的同一支付方式来支付退款:

```
<TermCondition>
  <ReturnTC>
    <ReturnTCRefundPaymentMethod>
      <PolicyReference policyName="UseOriginalPayment"
        policyType="ReturnPayment" storeIdentity="&STORE_IDENTIFIER;">
        <Member>
          <Organization distinguishName="ou=&ORGENTITYNAME;,
            &ORGANIZATION_DN;" />
        </Member>
      </PolicyReference>
    </ReturnTCRefundPaymentMethod>
  </ReturnTC>
</TermCondition>
```

请注意，PolicyReference 元素中的断行仅出于显示目的。

6. 如下结束 XML 文件:

```
</Contract>
</Trading>
```



关于 @ 和 & 用法的更多信息，请参阅第 417 页的附录 B，『创建数据』。

---

## 第 19 章 供货有用资源

商店将供货中心同时作为库存仓库以及送货和接收中心来使用。供货中心代表一个地点，产品从该处发送到客户。每个供货中心的库存计数是独立维护的。一个商店可能具有一个或多个供货中心与之关联。供货中心为商店管理产品的库存和送货。供货包含了提货、装货和送货。提货是指在供货中心的一个或多个发货批次中产品的选择，装货是指将这些产品装入送货容器，而送货是指将它们发送给客户。

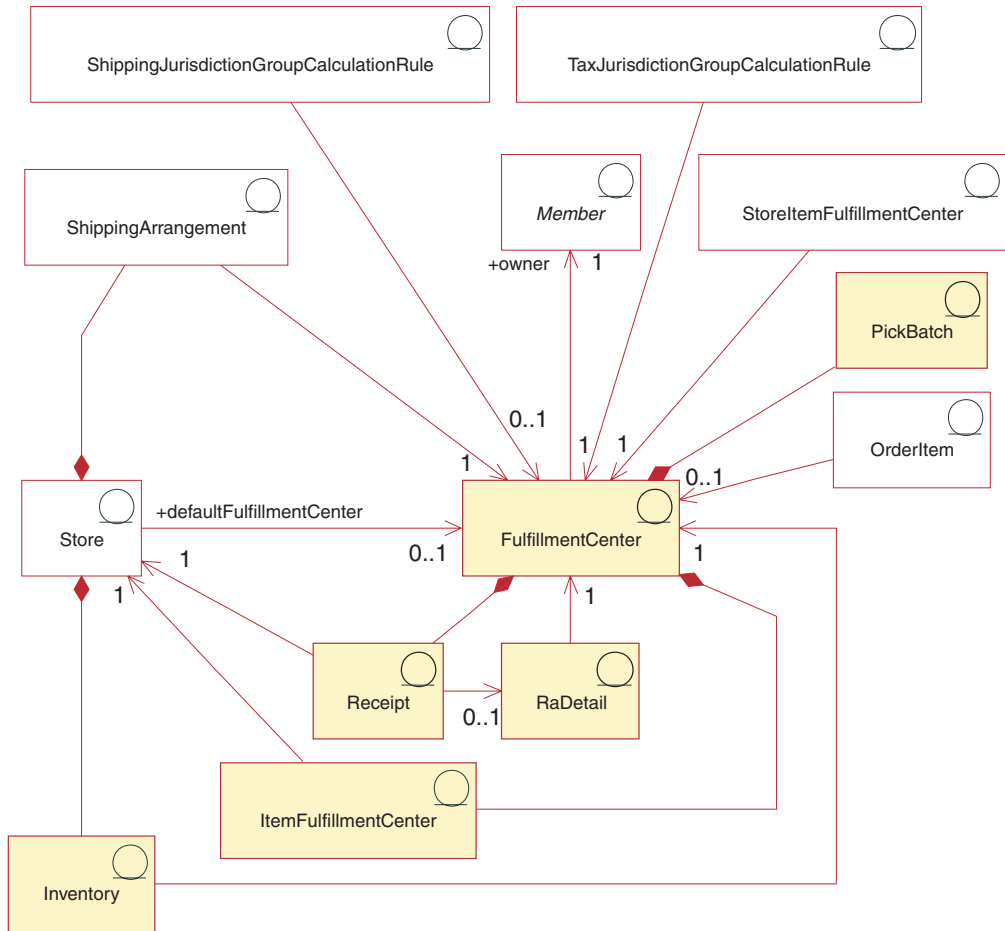
在“产品”向导和“产品”笔记本中配置产品以供货。产品配置提供了选项以跟踪库存、允许延迟交货、强制延迟交货、分别发货以及指定产品不可以退回。

通常，同一时间在供货中心中有很多人员在工作，每个人都有不同的任务要执行。WebSphere 贸易加速器将最常见的任务分成角色，然后将这些角色指定给用户。在 WebSphere 贸易加速器中，如果您对指定了一个或多个与供货相关的角色，则在登录时必须选择一个供货中心。

**注：**关于供货、供货中心和角色的更多信息，请参阅 WebSphere Commerce 联机帮助。

## 了解 WebSphere Commerce 中的供货有用资源

为了了解供货有用资源，有必要了解供货和商店之间的关系。这可以使用信息模型来解释。以下各部分描述供货与商店和其它有用资源之间的关系。



本图和商店数据部分中的其它所有图都是 WebSphere Commerce Server 信息模型的一部分。关于信息模型的更多信息，请参阅第 97 页的『商店数据信息模型』。关于本图中所使用的约定的更多信息，请参阅第 415 页的附录 A，『UML 图注』。

### 供货中心

在前图中，供货中心处于供货过程的中心位置。供货中心具有一个所有者，他在 MEMBER 表中定义。每家商店可以与多个供货中心相关联，因此一个供货中心可以有多个商店与其关联。如图所示，在商店和供货中心之间存在若干交互。关于商店有用资源的更多信息，请参阅第 123 页的『了解 WebSphere Commerce 中的商店有用资源』。

### 接收

供货中心按每天、每周或每月来接收商品的库存。当接收到商品的库存时，在 RECEIPT 表中就创建了一个接收，它记录了有关接收数量的信息以及拥有此库存的商店。处理



订单时，将更新 RECEIPT 表以反映当前可用的库存级别。关于创建接收的信息，请参阅第 262 页的『在 WebSphere Commerce 中创建库存有用资源』。

## RaDetail

*RaDetail* 是关于预期库存记录中的商品的详细信息。此信息可用于估计在供货中心预计何时可以接收到库存，以及用于向客户提供延迟交货商品的预期送货日期。

## 库存

商店拥有与供货中心关联的库存。库存包含可以在供货中心中实际计算的任何东西。库存与一个商店以及一个供货中心关联。还记录有关商店在供货中心中拥有的库存的信息，例如保留数量、延迟交货的数额，以及分配的延迟交货的数额。此信息存储在 ITEMFFMCTR 表中。关于库存和库存有用资源的更多信息，请参阅第 259 页的第 29 章，『库存有用资源』。

## 送货安排

送货安排是一种关系，它使商店能使用供货中心。送货安排指示供货中心可以使用一个送货方式以商店的名义发送产品。每个商店与供货中心之间有一个送货安排，反之亦然。送货安排在 SHPARRANGE 表中设置。关于创建送货安排的信息，请参阅第 233 页的『创建送货供货有用资源』。

## 其它供货有用资源

其它一些与供货中心的关系并不与商店直接有关。一个提货批量与一个供货中心关联。提货批量将各订单发货批次分组在一起以在供货中心作为一个单位进行处理，并且创建提货单和装货单。一旦提取并包装了商品，就可以对订单发货，并确认送货。提货批量信息存储在 PICKBATCH 表中。一个订购商品也与一个供货中心关联。一个商品是产品的一个特定实例，它由属性定义。关于一个订单中每个商品的信息都存储在 ORDERITEMS 表中。关于订单有用资源的更多信息，请参阅第 267 页的第 30 章，『订单有用资源』。

与其它实体相似，供货中心具有管理它的一些活动的规则。每个供货中心都具有针对税款费用和送货费用的规则。这些分别在 TAXJCRULE 和 SHPJCRULE 表中定义。关于税款和送货有用资源的更多信息，请参阅第 223 页的第 26 章，『送货有用资源』和第 239 页的『了解 WebSphere Commerce 中的税款有用资源』。



关于 WebSphere Commerce Server 中供货有用资源的结构更多详细信息，请参阅 WebSphere Commerce 联机帮助中的供货数据模型。

---

## 在 WebSphere Commerce 中创建供货有用资源

在您的商店可以向客户发送商品之前，您必须定义一个或多个供应这些商品的供货中心。以 XML 文件格式创建此信息，可使用装入程序软件包将 XML 文件装入数据库。关于装入程序软件包的更多信息，请参阅第 309 页的第 10 部分，『发布商店』。

创建有用资源之前，您还应熟悉第 367 页的第 38 章，『装入 WebSphere Commerce 数据库有用资源组』中所涵盖的材料。

要使用 XML 文件为您的商店创建供货有用资源，请执行以下操作：

1. 复查用于为样本商店创建供货有用资源的 XML 文件。样本商店的所有文件都位于相应的商店归档文件中。

商店归档文件位于以下目录中:

- `WC_installdir/samplestores`

**注:** WebSphere Commerce 联机帮助包含关于样本商店中所包含的每个数据有用资源的信息。

每个样本商店包含 `fulfillment.xml` 文件, 该文件包含供货信息。要查看商店归档文件中的 `fulfillment.xml` 文件, 请使用 ZIP 程序将其解压缩。`fulfillment.xml` 文件位于 `data` 目录中。

2. 请复查第 417 页的附录 B, 『创建数据』中的信息。
3. 通过复制样本商店归档文件中的一个 `fulfillment.xml` 文件或通过创建一个新文件来创建 `fulfillment.xml` 文件。关于更多信息, 请参阅相应于 `fulfillment.xml` 的 DTD 文件。DTD 文件位于以下目录中:

- `WC_installdir/xml/sar`

4. 定义商店支持的一个或多个供货中心:

- a. 使用以下示例作为指南, 在 XML 文件中定义 FFMCENTER 表中的供货中心:

```
<ffmcenter
ffmcenter_id="@ffmcenter_id_1"
  member_id="@seller_b2b_mbr_id"
  name="ToolTech Home"
  defaultshipoffset="0"
  markfordelete="0"
/>
```

其中

- `ffmcenter_id` 是生成的唯一键
  - `member_id` 是供货中心的所有者
  - `name` 是字符串, 它与所有者一起唯一标识此供货中心。
  - `defaultshipoffset` 是商品从该供货中心处送货所花费时间的估计秒数。可以在 `STORITMFFC` 表中重设此值。
  - `markfordelete` 指示是否应该删除供货中心, 如下: 0 = 不删除。1 = 如果不再使用, 就删除。关于更多详细信息, 请参阅 WebSphere Commerce 联机帮助中有关数据库清理实用程序的信息。
- b. 使用以下示例作为指南, 在 XML 文件中描述 FFMCENTDS 表中的供货中心。如果您正创建多文化的商店, 则应当将此信息包含在特定于语言环境的 XML 文件中。

```
<ffmcentds
ffmcenter_id="@ffmcenter_id_1"
  description="The fulfillment center that supplies products to ToolTech."
  language_id="@en_US"
  displayname="ToolTech Fulfillment"
  staddress_id="@staddress_id_en_US_1"
/>
```

其中

- `ffmcenter_id` 是生成的唯一键
- `description` 是适合于显示给客户的对供货中心的描述。

- language\_id 是显示此信息的语言。(关于对不同语言的支持的更多信息, 请参阅第 287 页的第 34 章, 『全球化』。)
  - displayname 是适合于显示给客户的供货中心的名称。
  - staddress\_id 供货中心的物理位置。
- c. 对商店支持的所有供货中心重复步骤 a 和 b。



关于 @ 和 & 用法的更多信息, 请参阅第 417 页的附录 B, 『创建数据』。

## 创建商店供货有用资源 (非 ATP)

在定义了将为商店供应商品的一个或多个供货中心之后, 必须将供货中心与每个产品关联。即, 必须标识哪个供货中心将供应哪些产品。要创建此关系, 请向 INVENTORY 表添加信息。以 XML 文件格式创建此信息, 可使用装入程序软件包将 XML 文件装入数据库。关于装入程序软件包的更多信息, 请参阅第 309 页的第 10 部分, 『发布商店』。

### 注:

1. 在可以将商店与供货中心关联之前, 您必须先创建商店有用资源。关于创建商店有用资源的更多信息, 请参阅第 124 页的『在 XML 文件中创建商店数据有用资源』。在可以创建商店供货有用资源之前, 您还必须创建产品目录有用资源。关于更多信息, 请参阅第 161 页的『显示商店产品目录有用资源』。
2. 仅在实施非 ATP 实现时才创建商店供货有用资源。包含 ATP 功能的商店不使用 INVENTORY 表。

要使用 XML 文件创建商店供货关系, 请执行以下操作:

1. 复查用于为样本商店创建商店供货有用资源的 XML 文件。样本商店的所有文件都位于相应的商店归档文件中。

商店归档文件位于以下目录中:

- WC\_installdir/samplestores

**注:** WebSphere Commerce 联机帮助包含关于样本商店中所包含的每个数据有用资源的信息。

每个样本商店包含 storefulfill.xml 文件, 该文件包含商店供货信息。要查看商店归档文件中的 storefulfill.xml 文件, 请使用 ZIP 程序将其解压缩。storefulfill.xml 文件位于 data 目录中。

2. 请复查第 417 页的附录 B, 『创建数据』中的信息。
3. 通过复制样本商店归档文件中的一个 storefulfill.xml 文件或通过创建一个新文件来创建 storefulfill.xml 文件。关于更多信息, 请参阅 WC\_installdir/schema/xml 目录中的 wcs.dtd 文件, 或商店归档文件中所包含的 DTD。
4. 使用以下示例作为指南, 通过向 INVENTORY 表添加信息, 在 XML 文件中创建“商店 - 供货中心”关系。

```
<inventory
catentry_id="@catentry_id_1470"
quantity="100"
ffmcenter_id="@ffmcenter_id_1"
```

```
store_id="@storeent_id_1"  
quantitymeasure="C62"  
inventoryflags="0"  
</>
```

其中

- `catentry_id` 是此供货中心将供应的产品目录条目。
- `quantity` 是从该供货中心中可以得到的数额（以 `QUANTITYMEASURE` 指示的单位表示）。
- `ffmcenter_id` 是供应此库存的供货中心。
- `store_id` 是为其供应此库存的商店。
- `quantitymeasure` 是 `QUANTITY` 的计量单位。
- `inventoryflags` 是指示如何使用 `QUANTITY` 的位标志：
  - 1 = `noUpdate`。缺省 `UpdateInventory` 任务命令不更新 `QUANTITY`。
  - 2 = `noCheck`。缺省 `CheckInventory` 和 `UpdateInventory` 任务命令不检查 `QUANTITY`。

5. 对商店中的每个产品目录条目重复步骤 3。



关于 `@` 和 `&` 用法的更多信息，请参阅第 417 页的附录 B，『创建数据』。

---

---

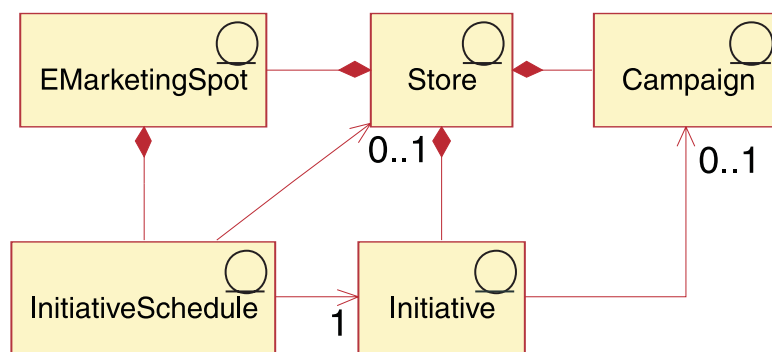
## 第 20 章 竞销有用资源

竞销有助于您组织进行市场营销。竞销通常是市场部经理或销售部经理创建的。它们经常与某一组目标相关联。例如，“回到学校”竞销就可能有在竞销期间增加销售儿童服装的目标。

---

### 了解 WebSphere Commerce 中的竞销

下图说明了 WebSphere Commerce Server 中的竞销有用资源：

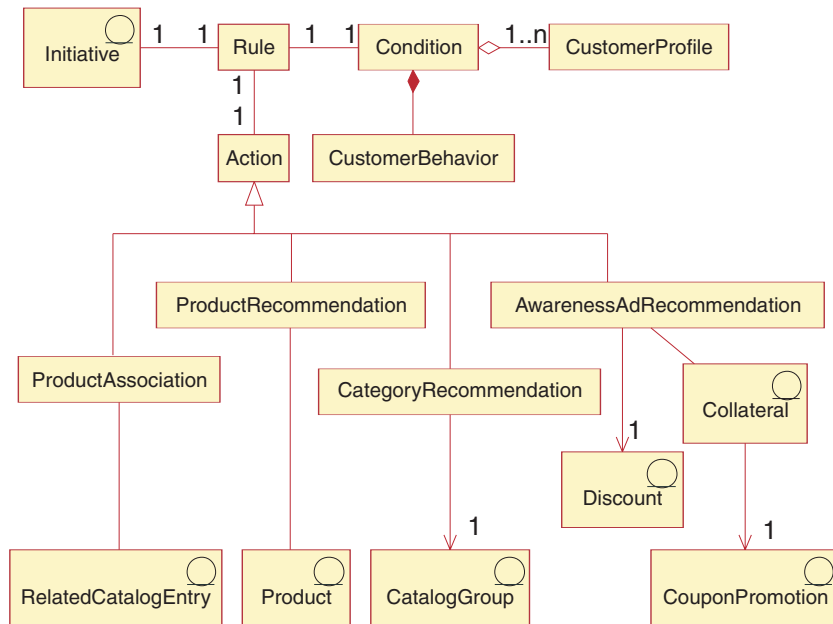


本图和商店数据部分中的其它所有图都是 WebSphere Commerce Server 信息模型的一部分。关于信息模型的更多信息，请参阅第 97 页的『商店数据信息模型』。关于本图中所使用的约定的更多信息，请参阅第 415 页的附录 A，『UML 图注』。

竞销及其相关联的有用资源纳入商店范围。

在 WebSphere Commerce 中，竞销包含任意数量的竞销活动，这些活动将定义条件。当定义的条件评估为真时，竞销活动将为客户生成目标内容。结果是竞销是组织活动的

高级市场营销元素。竞销活动信息模型显示如下：



竞销活动与包含活动集合的竞销相关联。此关系的一个例子是：如果某一办公用品商店开展“回到学校”竞销，则活动将是负责较低级别的活动，例如向以学生身份注册或列出职业的客户以折扣价宣传钢笔或推荐格记录纸。

竞销活动能够显示四种动态内容：

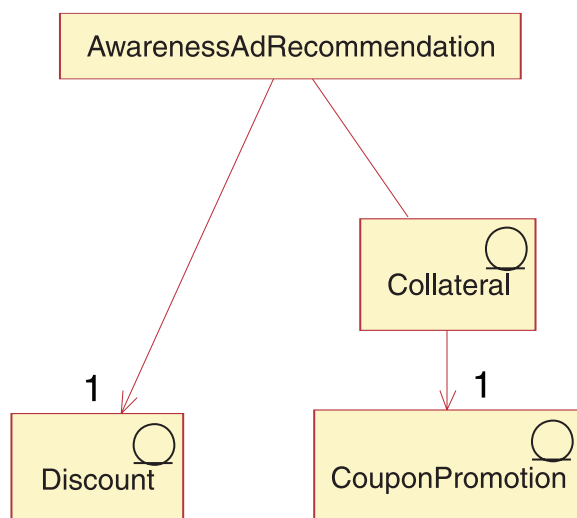
- 推荐销售活动
- 基于协作过滤的推荐
- 推广广告
- 销售关联

推荐销售内容提供基于规则类别和产品推荐，这些推荐针对特定的客户对象，基于客户的概要文件和其它客户的行为。可作为目标的客户行为包括购物车的总值、购物车的内容和客户购买历史记录的内容。

基于协作过滤的推荐也会创建产品推荐，但是它们使用不同的推荐算法，这种算法基于客户整体行为来锁定商品目标，而不是基于预定义的规则。

推广广告针对特定客户对象提供广告内容，以用于推荐销售的相同条件为基础，但是推广广告意在用来让客户加深了解网上商店的活动，突出特价商品并提高品牌知名

度。推广广告遵循以下显示的信息模型:



销售关联基于产品目录中定义的静态关联，创造了追加销售和交叉销售机会。为了创建此类活动，必须定义在关联中选择源产品的方法，这样在调用电子广告位时，可以使用适当的源来返回目标产品。该方法可以基于当前页面的内容、购物车的内容或购买者的购买历史内容来选择源，作为关联的源。

活动可以并入到站点上的任何页面上。站点设计好后，称为电子广告位的特定占位符将置于站点上。当显示给客户时，这些占位符由特定的目标内容替换。通过调度活动指定目标位置可以在理想位置上显示电子广告位。关于向商店添加电子广告位的更多信息，请参阅第 407 页的第 42 章，『向商店添加电子广告位』。

竞销活动包含确定显示时间和显示对象的条件。此条件在活动创建时定义，而且在活动的有效期间可以更改条件以调整活动的可视性和显示的内容。有关客户概要文件的更多信息，请参阅第 273 页的第 32 章，『客户概要文件』。

竞销活动生成关于其使用的统计信息。商家、市场部经理和销售部经理可以使用 WebSphere 贸易加速器查看这些统计信息。统计信息说明了被点击的每一电子广告位的活动点击率。这些统计信息提供了关于活动效果的反馈，以及在不同显示位置的相对成功率。

---

## 在 WebSphere Commerce 中创建竞销有用资源

竞销和竞销活动通常是由市场部经理或销售部经理使用 WebSphere 贸易加速器中的“竞销和竞销活动”向导创建的。关于更多信息，请参阅 WebSphere Commerce 联机帮助。

关于向商店添加电子广告位的更多信息，请参阅第 407 页的第 42 章，『向商店添加电子广告位』。





---

## 第 21 章 Payments 工具

WebSphere Commerce 提供称作 WebSphere Commerce Payments 的可选组件（先前称为 IBM Payment Manager）。

如果想要对商店使用 WebSphere Commerce Payments，您必须在商店归档文件中包含一个支付有用资源文件。在发布商店归档文件前，请确保 Payments 实例已启动，然后当商店归档文件发布后，支付有用资源文件（包含为样本商店归档文件的一部分）在 WebSphere Commerce Payments 中设置以下信息：

- WebSphere Commerce Payments 数据库中的 merchant\_ID。
- 商店中使用的卡匣的类型。
- WebSphere Commerce Payments 数据库中每种货币的帐户，该货币是支付有用资源文件中指定为受商店支持的货币。如果商店不支持支付有用资源文件中指定的货币，那么就不会创建帐户。
- 每个帐户的一个或多个品牌。

要设置支付有用资源文件，并设置商店使用 WebSphere Commerce Payments，请执行以下操作：

- 以 XML 文件（paymentinfo.xml）格式创建支付数据，该数据在使用管理控制台进行商店发布时装入。这为商家配置了 WebSphere Commerce Payments，并配置将发布为商店指定的品牌类型。关于更多信息，请参阅『使用 XML 文件创建支付有用资源』。

**注：**paymentinfo.xml 不填充 WebSphere Commerce Server 数据库中的表。它配置 WebSphere Commerce Payments。仅当您使用脱机信用卡作为支付方式时，paymentinfo.xml 才适用。

在商店归档文件发布后，您可以通过使用在样本商店归档文件中设置的支付信息来订购。如果想要添加新品牌，您必须配置 WebSphere Commerce Payments 以处理每个品牌。

- 如果您使用除 OfflineCard 或 CustomOffline 卡匣之外的 IBM 支付卡匣，请如第 204 页的『定制支付卡匣的环境』中所述修改样本商店 Web 有用资源。
- 使用管理控制台或 WebSphere Commerce Payments 用户界面为您的商店完成 WebSphere Commerce Payments 的设置。如果您使用管理控制台，则菜单项将出现在 Payments 菜单上。如果您使用 WebSphere Commerce Payments 用户界面，则菜单项将出现在导航框架的“管理”下。关于设置任务的更多信息，请参阅 WebSphere Commerce 产品联机帮助中的主题“*给您的商店设置 WebSphere Commerce Payments*”。

如果您想要使用 WebSphere Commerce Payments 之外的支付机制，则使用您的支付机制所要遵循的步骤与以下过程相似。

---

### 使用 XML 文件创建支付有用资源

要使用 XML 文件为您的商店创建支付有用资源，请执行以下操作：

1. 复查用于为样本商店创建支付有用资源的 XML 文件。样本商店的所有文件都位于相应的商店归档文件中。

商店归档文件位于以下目录中:

- `WC_installdir/samplestores`

**注:** WebSphere Commerce 联机帮助包含关于样本商店中所包含的每个数据有用资源的信息。

每个样本商店包括一个 `paymentinfo.xml` 文件, 该文件包含支付信息。要查看商店归档文件中的 `paymentinfo.xml` 文件, 请使用 ZIP 程序将其解压缩。`paymentinfo.xml` 文件位于 `data` 目录中。

2. 通过复制样本商店归档文件中的一个 `paymentinfo.xml` 文件或通过创建一个新文件来创建 `paymentinfo.xml` 文件。关于更多信息, 请参阅对应于 `paymentinfo.xml` 的 DTD 文件。该 DTD 文件位于以下目录: `WC_installdir/xml/sar`。
3. 启用或禁用 WebSphere Commerce Payments。
  - a. 使用以下示例作为指南, 在您的 XML 文件中启用或禁用 WebSphere Commerce Payments 并指定商店接受的支付卡匣、货币和品牌的类型:

```
<paymentinfo>
  <PaymentManager enable="yes"/>
  <Cassette type="OfflineCard">
    <Account currency="USD">
      <Brand type="MasterCard"/>
      <Brand type="VISA"/>
      <Brand type="American Express"/>
    </Account/>
    <Account currency="EUR">
      <Brand type="MasterCard"/>
      <Brand type="VISA"/>
      <Brand type="American Express"/>
    </Account>
  </Cassette>
</paymentinfo>
```

其中:

- `enable` 表示 WebSphere Commerce Payments 是启用还是禁用。当 WebSphere Commerce Payments 禁用时, 您的商店将不能通过 Payments 组件处理支付事务, 尽管 Payments 用户界面仍将起作用。如果您禁用 Payments 组件, 则无需在 `paymentinfo` 元素中指定其它元素。
- `Cassette type` 是受支持的卡匣的类型。
- `Account currency` 是商店支持的货币。如果您使用 `OfflineCard` 卡匣类型, 则帐户货币是必需的。货币必须以符合 ISO 4217 标准的三字母代码来标识。例如, “USD” 用以表示美元。
- `Brand type` 是帐户和货币支持的信用卡类型。

---

## 定制支付卡匣的环境

WebSphere Commerce 提供可以使用 `OfflineCard` 或 `CustomOffline` 卡匣作为支付卡匣来处理支付事务的样本商店。这些卡匣自动配置为可作为样本商店的支付方式使用。样本商店 Web 有用资源需要进行修改以使用任何其它支付卡匣。本节中的指示信息描述如何定制环境, 以便使用随 WebSphere Commerce 提供的其它 IBM 支付卡匣。

对于要使用 IBM 支付卡匣的商店，您必须首先选择要安装的 WebSphere Commerce Payments 组件。对应于您的平台的《WebSphere Commerce 安装指南》中提供了安装说明。WebSphere Commerce 安装程序将安装 Payments 框架和卡匣软件。您接着必须使用 WebSphere Commerce 配置管理器来执行必要的安装后任务，例如创建 Payments 实例和向实例添加卡匣。请参阅《WebSphere Commerce 安装指南》和配置管理器联机帮助，以获得关于配置 Payments 实例的指示信息。

在向 Payments 实例添加支付卡匣后，请检查以下定制步骤，以确保 WebSphere Commerce 样本商店可以通过您选择的支付卡匣处理支付：

1. 修改商店 .jsp 文件来指定支付卡匣。
2. 检查卡匣的卡匣柜概要文件。
3. 检查卡匣支持职员订单（临时订单）放置页面的 .jsp 文件。
4. 配置商家设置。

以下各部分描述了这些步骤。

## 修改商店 .jsp 文件

如果不想对样本商店使用 OfflineCard 或 CustomOffline 卡匣，则必须修改商店的 .jsp 文件。缺省情况下，商店的 .jsp 文件设置为使用 OfflineCard 卡匣；因此，必须修改该文件以使用任何其它卡匣。“时尚潮流”商店还使用 CustomOffline 卡匣。

要获得 .jsp 文件的列表以查看可能的修改，请参阅第 206 页的表 10。

要修改 .jsp 文件，请执行以下这些步骤：


1. 使用样本商店（例如“时尚潮流”）在 WebSphere Commerce 中创建一个商店。
2. 请转至以下目录：

```
WAS_installdir/installedApps/cell_name/  
WC_instance_name.ear/Stores.war/
```

 400 对于 iSeries，该路径是：

```
WAS_userdir/WAS_instance_name/  
installedApps/ cell_name/WC_instance_name.ear/Stores.war/。您创建的商店在 war 目  
录中有其自己的目录。
```

3. 从商店目录中，用文本编辑器打开 OrderSubmitForm.jsp 文件。

 **Business** WebSphere 贸易加速器中的合同工具支持所有的支付卡匣。OrderSubmitForm.jsp 文件必须遵守买方组织和卖方组织之间签订的合同中的支付条款和条件。

4. 在 OrderSubmitForm.jsp 文件中搜索以下文本：

```
if (info[i].getPolicyName().trim().equals("OfflineCard"))
```

根据需要，将支付策略的名称从 OfflineCard 更改为以下项之一：

```
CustomOffline  
BankServACH  
Paymentech  
VisaNet  
VisaNet_PCard
```

关于策略的更多信息，请参阅 WebSphere Commerce 开发联机帮助中的 POLICY 数据库表。

CustomOffline 的策略支持对定制支付事务的处理，如“货到付款”或 COD、“以后再开帐单”或通常在 WebSphere Commerce Payments 之外执行的赠券。

BankServACH 的策略支持使用与自动清算中心网络（ACH）对接的 BankServ 支付网关对在线电子支票支付进行处理。

Paymentech 的策略支持在线授权、信用卡结算以及非基于 PIN 的借记卡支付方式。

VisaNet 的策略支持使用 Vital Processing Services 或 First Horizon Merchant Services（FHMS）金融网络对信用卡交易进行处理。

**注：**如果您在使用有购买卡支持的 Cassette for VisaNet，那么请选择 VisaNet\_PCard 而不是 VisaNet。

关于这些卡匣的更多信息，请参阅卡匣的补充文档。

如果您的商店使用“快速结帐”功能，您还应该更改这些其它文件中的支付策略的名称：

ShoppingArea\CheckoutSection\QuickCheckoutSubsection\QuickCheckoutForm.jsp  
UserArea\AccountSection\QuickCheckoutProfileSubsection\QuickCheckoutProfileForm.jsp

5. （可选）如果您正在使用信用卡支付方式并且需要在用户界面中添加附加字段以便从用户处收集附加信息，请同样复查 StandardCreditCard.jsp 文件中可能的修改。关于路径信息，请参阅表 10。

要使信用卡品牌在以涉及使用信用卡的特定付款方式购买时能够显示出来，则该支付方法的对应 .jsp 文件中务必存在某个选项值。例如，要使信用卡品牌在以 Paymentech 支付方式购买时能够显示出来，则搜索 `<select name="cardBrand">`。在该文本下添加新的一行，并添加以下内容：

```
<option value="Paymentech">Paymentech</option>.
```

表 10. 存储 .jsp 文件供复查

JSP 文件	业务模型 / 样本	更改的目的
/ShoppingArea/CheckoutSection/StandardCheckoutSubsection/OrderSubmitForm.jsp	消费者直销（时尚潮流或 Express），受托管转销商	更改支付策略名 OfflineCard
/ShoppingArea/CheckoutSection/QuickCheckoutSubsection/QuickCheckoutForm.jsp /UserArea/AccountSection/QuickCheckoutProfileSubsection/QuickCheckoutProfileForm.jsp	消费者直销（时尚潮流或 Express）	更改支付策略名 OfflineCard
/ShoppingArea/CheckoutSection/StandardCheckoutSubsection/StandardCreditCard.jsp	消费者直销（时尚潮流或 Express）	使信用卡品牌得以显示
/ShoppingArea/CheckoutSection/StandardCheckoutSubsection/StandardCreditCardDisplay.jsp	B2B 直销（多乐五金），价值链 - 供应	使信用卡品牌得以显示

## 检查卡匣的卡匣柜概要文件

WebSphere Commerce Payments 卡匣柜概要文件应该对任何随 WebSphere Commerce 提供的 IBM 支付卡匣都是可用的。卡匣柜概要文件用于在 Payments 组件中创建订单。

您可能想要编辑卡匣柜概要文件来设置某些参数，例如 APPROVEFLAG 和 DEPOSITFLAG 参数。由于不是所有的卡匣参数都是相似的，因此请参阅卡匣补充文档，以获得关于设置参数的更多信息：

- *WebSphere Commerce Payments CustomOffline Cassette Supplement*
- *WebSphere Commerce Payments OfflineCard Cassette Supplement*
- *WebSphere Commerce Payments Cassette for BankServACH 补充文档*
- *WebSphere Commerce Payments Cassette for Paymentech 补充文档*
- *WebSphere Commerce Payments Cassette for VisaNet 补充文档*

与 IBM 提供的支付卡匣相关联的卡匣柜概要文件包括：

```
WC51_BankServACH.profile
WC51_CustomOffline_BillMe.profile
WC51_CustomOffline_COD.profile
WC51_OfflineCard.profile
WC51_VisaNet.profile
WC51_VisaNet_PCard.profile
WC_Paymentech.profile
```

卡匣概要文件应存储在 WebSphere Commerce 实例概要文件目录中。

要找到存储概要文件的目录，请在 WebSphere Commerce 配置文件中查看您创建的实例。如果您使用了缺省实例名称 “demo”，那么配置文件是：

```
WC_installdir/instances/demo/xml/demo.xml
```

▶ 400 对于 iSeries，该路径会是：

```
WC_userdir/instances/demo/xml/demo.xml
```

然后，查看由配置文件中的 Payment Manager 元素的 ProfilePath 属性所指定的目录。该属性指定概要文件应在的位置。如果您使用了缺省实例名称 “demo”，那么存储概要文件的目录路径会是以下路径：

```
WC_installdir/instances/demo/xml/payment
```

▶ 400 对于 iSeries，该路径是：

```
WC_userdir/instances/demo/xml/payment
```

如果您编辑卡匣的卡匣柜概要文件来设置参数，请确保把概要文件保存到 *WC\_installdir/instances/instance\_name/xml/payment* 目录中，其中 *instance\_name* 是正在使用的实例的名称。

▶ 400 对于 iSeries，该路径是：

```
WC_userdir/instances/instance_name/xml/payment
```

支付业务策略使用的实际卡匣柜概要文件是由支付业务策略的 “属性” 字段中的 profileName 属性值所指定的。关于业务策略的更多信息，请参阅联机帮助中的 POLICY 数据库表。

## 检查卡匣的 .jsp 文件

如果某订单登记员代表客户下了一个临时订单，则将通过 WebSphere 贸易加速器处理支付。卡匣的支付数据通过使用卡匣的 .jsp 文件进行收集。

在 WebSphere Commerce 中卡匣的 .jsp 文件称为“支付属性页”。使用的实际页面是由支付业务策略的“属性”字段中的 attrPageName 属性值指定的。关于更多信息，请参阅 WebSphere Commerce 开发联机帮助中的 POLICY 数据库表。商店流程和 WebSphere 贸易加速器流程都应该使用支付属性页。

卡匣的 .jsp 文件应该已经在以下目录中：

```
WC_installdir/wc.ear/CommerceAccelerator.war/tools/order/buyPages/  
WAS_installdir/installedApps/cell_name/WC_demo.ear/CommerceAccelerator.war/  
tools/order/buypages
```

 对于 iSeries，该路径是：

```
QIBM/userdata/webas5/base/WAS_instance_name /installedApps/  
cell_name/WC_demo.ear/CommerceAccelerator.war/tools/order/buypages
```

如果您想要定制“购买页”信息，请相应地修改 .jsp 文件。

## 配置 WebSphere Commerce Payments 中的商家设置

要为商家配置 IBM 支付卡匣，请遵循在卡匣补充文档中提供的指示信息。可以通过 WebSphere Commerce 管理控制台或 Payments 用户界面

([http://host\\_name:port/webapp/PaymentManager](http://host_name:port/webapp/PaymentManager)) 来修改商家设置。要配置商家设置，您必须在 WebSphere Commerce Payments 中拥有支付管理员或商家管理员权限。

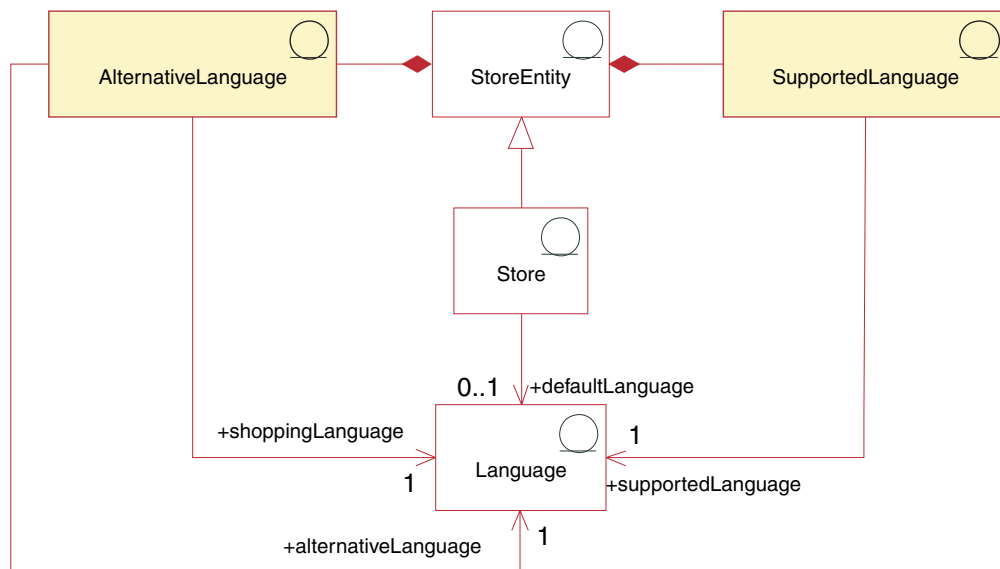
## 第 22 章 语言有用资源

在 WebSphere Commerce 中，您的站点可以定义许多可在 WebSphere Commerce 中使用的语言。在创建实例时，LANGUAGE 表可以有十种受支持的语言，其中包括德语、繁体中文和简体中文、日语、韩国语、意大利语、法语、西班牙语、巴西葡萄牙语和英语。站点可以定义附加的语言，或者现有语言的方言，以便使用适合的方法将信息显示给来自不同文化或地域的客户。

### 了解 WebSphere Commerce 中的语言有用资源

为了了解语言有用资源，有必要了解语言和商店之间的关系。这可以使用下面的信息模型来解释。以下部分描述了语言与商店和其它有用资源之间的关系和关联。

下图描绘了语言有用资源信息模型。



在 WebSphere Commerce 中有四个语言分类。它们是：

- 缺省语言，
- 受支持的语言，
- 可替代的语言，以及
- 购物语言。

这些分类中的每一个在商店中执行不同的角色。所有的语言都存储在 LANGUAGE 表中。

### 缺省语言

缺省语言与每个商店关联。这是商店选择用作其主语言的语言，而且将是向没有明确选择购物语言的客户显示的语言。商店隐式地支持商店的缺省语言；即，商店必须总

是能够以缺省语言显示信息，或者以它的备用语言之一显示信息（如果在 LANGPAIR 表中定义了备用语言）。如果不能以商店支持的语言或备用语言之一显示信息，则信息将以缺省语言显示。

## 支持的语言

STORELANG 表指示每一商店支持的语言。商店必须能够以它的支持语言或备用语言之一（如果在 LANGPAIR 表中定义了备用语言）显示信息。商店也支持其商店组所支持的所有语言。

关于添加受支持语言的更多信息，请参阅第 298 页的『向商店添加语言』。

## 备用语言

如果信息不能以商店的支持语言之一显示，则商店尝试使用备用语言（如果有的话）显示信息。商店可以指定尝试其备用语言的顺序。商店的备用语言包括其商店组的备用语言。当某些信息只能以一种语言显示，但又应当向以不同的、相关的语言购物的客户提供时，备用语言十分有用。例如，并非所有信息都翻译成所有支持的语言时，或者例如支持同一语言的两种十分相近的方言（有时信息完全相同）时，备用语言可能十分有用。



---

关于 WebSphere Commerce Server 中语言有用资源的结构의 更详细信息，请参阅 WebSphere Commerce 联机帮助中的语言数据模型。

---

---

## 在 WebSphere Commerce 中创建语言有用资源

您可以用以下一种方式来定义商店所支持的语言：

- 使用 WebSphere 贸易加速器中的商店工具
- 在将由装入程序软件包或管理控制台中的发布工具装入的 XML 文件中
- 直接使用 SQL 插入编辑数据库
- 使用 SQL 编辑和更新

**注：** 这些工具处理商店归档文件形式的预植入 XML 文件。

关于使用商店工具来定义商店的受支持语言的更多信息，请参阅 WebSphere Commerce 联机帮助。关于在 XML 文件中定义商店支持语言的更多信息，请参阅第 124 页的『在 XML 文件中创建商店数据有用资源』。



## 第 23 章 货币有用资源

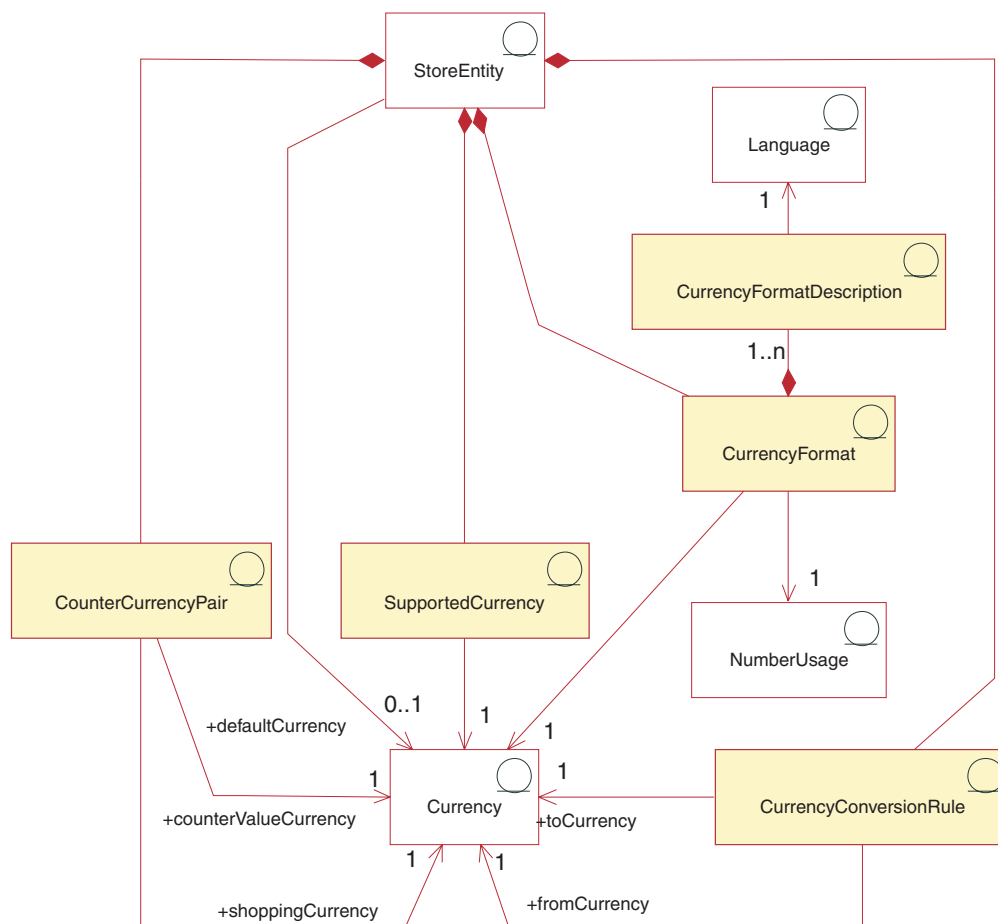
您可以在您的站点中以一种货币显示价格，或者您可以遵循为欧元提供的指示信息来显示多种货币（请参阅第 212 页的『对等货币』）。对于具有多个商店的站点，您可以为商店使用不同的货币，或者您可以对商店组指定货币。根据您正在创建的站点的特点，您可以指定希望使用的货币以及如何显示它们。

在 WebSphere Commerce 中，您可以允许客户选择购物货币。购物货币是客户在特定商店中为产品支付时所用的货币。商店页面上所有的货币金额都以此货币显示。当客户更改了他们的购物货币时，将自动转换并重新计算已添加到购物车的商品的价格以及订单的总额，并以新的购物货币显示它们。

客户可以以多种货币（其中包含欧元）进行购物。欧元于 1999 年 1 月 1 日成为欧盟的合法货币，且现已在金融市场上使用。欧元与所有参与国的货币之间的兑换率是固定的。

### 了解 WebSphere Commerce 中的货币有用资源

下图说明了 WebSphere Commerce Server 中的货币结构：





本图和商店数据部分中的其它所有图都是 WebSphere Commerce Server 信息模型的一部分。关于信息模型的更多信息，请参阅第 97 页的『商店数据信息模型』。关于本图中所使用的约定的更多信息，请参阅第 415 页的附录 A，『UML 图注』。

在上图中，货币处于信息模型的中央。每个商店或商店组都有缺省货币。

## 货币格式

一个商店实体可以具有许多货币格式化规则。如果商店没有特定货币的格式化规则，它将使用其商店组的格式化规则。货币格式在 CURFORMAT 表中设置。

货币格式有用资源可以由其它商店使用，如第 129 页的第 14 章，『商店间的关系』中所述。

## 数值用法

每个已格式化的货币规则都与一个数值用法关联。根据它们关联用法的不同，数值（例如数量或货币金额）可以用不同方式舍入和格式化。商店可以根据使用数值的方式来为显示的数值指定不同的舍入和格式化规则，例如商店可以通过指定单价用法将单价舍入到四位小数，而通过指定缺省用法将其它货币金额舍入到两位小数。数值用法存储在 NUMBRUSG 表中。

## 货币格式描述

货币格式规则可以具有许多货币格式描述。货币格式描述描述了如何以特定货币和特定语言格式化（为了显示目的）货币金额。每个描述与 LANGUAGE 表中的一个语言相关联。关于语言有用资源的更多信息，请参阅第 209 页的第 22 章，『语言有用资源』。关于对全球化的支持的更多信息，请参阅第 287 页的第 34 章，『全球化』。货币格式描述存储在 CURFMTDESC 表中。

## 支持的货币

一个商店实体可以具有许多支持的货币。支持的货币是被承认的可用于支付的货币。

受支持的货币有用资源可以由其它商店使用，如第 129 页的第 14 章，『商店间的关系』中所述。

## 货币转换规则

所有货币都具有规则来管理它们与其它货币之间的相互转换。每个货币转换规则可以用于将价格（以特定的货币存储在数据库中）转换为客户以支持的购物货币支付的金额。

受支持的货币转换规则有用资源可由其它商店使用，如第 129 页的第 14 章，『商店间的关系』中所述。

## 对等货币

对等货币是与支持的货币一起显示的货币金额。不可以用它们进行购买，它们只是用于提供信息。如果客户决定以欧元购物，则他们可以在商店中显示“欧洲货币联盟”

货币金额以及其它货币金额。以购物货币表示的金额将转换为该购物货币的所有对等价格币种。对等货币与一种支持的货币结成配对，例如荷兰盾和欧元。对等货币对存储在 CURCVLIST 表中。

货币对应价值有用资源可以由其它商店所使用，如第 129 页的第 14 章，『商店间的关系』中所述。



关于 WebSphere Commerce Server 中货币有用资源的结构更多详细信息，请参阅 WebSphere Commerce 联机帮助中的货币数据模型。

## 在 WebSphere Commerce 中创建货币有用资源

WebSphere Commerce 中的管理控制台使您能够将受支持的货币添加到您的商店中，并能为您的商店选择缺省的货币。关于您可以使用管理控制台编辑哪些有用资源的更多信息，请参阅 WebSphere Commerce 联机帮助中的主题“更改商店数据库有用资源”。

**注：**管理控制台处理商店归档文件格式的预植入 XML 文件。

您还可以使用 XML 文件将支持的货币和缺省货币添加到您的商店中，XML 文件可以使用装入程序软件包装入到数据库中。此方法还允许您创建其它类型的货币有用资源，包括定义货币兑换率和对等价格货币。

关于处理货币的信息，请参阅 WebSphere Commerce 开发联机帮助。关于以 XML 文件格式创建新的货币有用资源的信息，请参阅『使用 XML 文件创建货币有用资源』。

### 使用 XML 文件创建货币有用资源

以 XML 文件（这些文件可以使用装入程序软件包装入到数据库中）格式为您的商店创建货币有用资源。关于装入程序软件包的更多信息，请参阅第 309 页的第 10 部分，『发布商店』。

在创建有用资源之前，您应熟悉第 367 页的第 38 章，『装入 WebSphere Commerce 数据库有用资源组』中所涵盖的材料。

要使用 XML 文件为您的商店创建货币有用资源，请执行以下操作：

1. 复查用于为样本商店创建货币有用资源的 XML 文件。样本商店的所有文件都位于相应的商店归档文件中。

商店归档文件位于以下目录中：

- `WC_installdir/samplestores`

**注：**WebSphere Commerce 联机帮助包含关于样本商店中所包含的每个数据有用资源的信息。

每个样本商店包含一个 `currency.xml` 文件，它包含货币信息。要查看商店归档中的 `currency.xml` 文件，请使用 ZIP 程序将其解压缩。`currency.xml` 文件位于 `data` 目录中。

2. 请复查第 417 页的附录 B，『创建数据』中的信息。

3. 通过复制样本商店归档中的一个 `currency.xml` 文件，或通过创建一个新的文件来创建 `currency.xml` 文件。关于更多信息，请参阅 `WC_installdir/schema/xml` 目录中的 `wcs.dtd` 文件，或商店归档文件中所包含的 DTD。
4. 定义您的商店支持的货币。

- a. 使用以下示例作为指南，在您 XML 文件中为 CURLIST 表定义商店支持的货币：

```
<curlist currstr="USD" storeent_id="@storeent_id_1" />
```

其中：

- `currstr` 是三个字母的 ISO 4217 货币代码，表示支持的货币。此代码必须出现在 SETCURRE 表中的 SETCCURRE 列。一个商店必须能够接受以所有其支持的货币进行的支付。
  - `storeent_id` 是商店实体。
- b. 为商店支持的每种货币重复此过程。



---

商店的缺省货币在 STOREENT 表中定义。关于更多信息，请参阅第 124 页的『在 XML 文件中创建商店数据有用资源』。

---

5. （可选的）商店显示的货币价格，这取决于您如何设置价格。您可以为商店中使用的每种货币定义价格，或者只为缺省货币定义价格。关于设置价格的更多信息，请参阅第 172 页的『在 WebSphere Commerce 中创建定价有用资源』。

如果您在设置价格时只为缺省货币定义了价格，但又想在商店中以其它支持的货币显示价格，则您必须将兑换率添加到商店中。以此转换率从缺省货币转换为支持的货币。

- a. 确定将进行转换的货币，例如美元（USD），以及将转换成的一种或多种货币，例如日元（JPY）。要确定每种货币的 ISO 货币代码，请参阅国际货币的 ISO 4217 代码。
- b. 使用以下示例作为指南，将转换信息添加到 CURCONVERT 表：

```
<curconvert
  storeent_id="@storeent_id_1"
  fromcurr="USD"
  tocurr="JPY"
  factor="105.10"
  multiplyordivide="M"
  bidirectional="Y"
  updatable="Y"
  curconvert_id="@curconvert_id_1" />
```

其中：

- `storeent_id` 是商店实体。
- `fromcurr` 是您将进行转换的货币。FROMCURRE 货币的金额通常是用于确定与待售产品关联的价格、折扣、送货费用或类似金额的规则或者其它信息的组成部分。
- `tocurr` 是您将转换成的货币。TOCURRE 通常是客户将支付的货币。此货币的金额通常是订购商品的组成部分，例如单价、送货费用和税款金额。
- `factor` 是转换因子。
- `multiplyordivide` 如下：从 FROMCURRE 转换为 TOCURRE：

- M = 乘以 FACTOR
- D = 除以 FACTOR

对于双向规则，允许使用逆操作进行从 TOCURR 到 FROMCURR 的转换。

- `bidirectional` 指示规则是否为双向的。
    - Y = 双向的
    - N = 非双向的
  - `updatable` 是个标志，用户界面将其用于管理货币转换规则。有效值是：
    - N = 转换率是不能取消的 - 永远不能更改
    - Y = 转换率是可以更改的
  - `curconvert_id` 是生成的唯一键。
- c. 对您想用于显示价格的所有货币重复步骤 a 和 b。



即使您已经在您的定价信息中为所有支持的货币定义了价格，您也可能想为商店中支持的货币定义货币兑换率。

6. (可选的) 如果您想同时以购物货币和对等货币显示价格 (例如, 同时以荷兰盾和欧元显示价格), 则您必须将信息添加到 CURCVLIST 表中。
- a. 使用以下示例作为指南, 将转换信息添加到 CURCVLIST 表:

```
<curcvlist
  storeent_id="@storeent_id_1"
  currstr="NLG"
  countervaluecurr="EUR"
  displayseq="1" />
```

其中:

- `storeent_id` 是商店实体。
- `currstr` 是符合 ISO 4217 标准的三个字母的货币代码, 表示货币。此代码必须出现在 SETCURR 表中的 SETCCURR 列。FROMCURR 货币的金额通常是用于确定与待售产品关联的价格、折扣、送货费用或类似金额的规则或者其它信息的组成部分。
- `countervaluecurr` 是三个字母的 ISO 4217 货币代码, 表示对等价格货币。此代码必须出现在 SETCURR 表中的 SETCCURR 列。
- `displayseq` 是指示对等价格货币展示顺序的数字。对等价格货币根据 CURCVLIST 表中的 DISPLAYSEQ 列中所指定的对等价格显示顺序以升序显示。



关于 @ 和 & 用法的更多信息, 请参阅第 417 页的附录 B, 『创建数据』。

## 其它货币任务

关于一般货币和其它货币任务的更多信息, 包括:

- 添加 WebSphere Commerce 当前不支持的新货币
- 更改现有的货币格式

请参阅 WebSphere Commerce 开发联机帮助。



---

## 第 24 章 计量单位有用资源

产品可以按照各种数量单位来出售，库存也可以按照各种数量单位来跟踪，例如，千克、英寸、公升等等。这些单位中，产品可以按最小数量来订购，也可以按特定数量的倍数来订购。

控制器命令使用 UOM（计量单位）来指定数量单位。如果未指定 UOM 参数，则客户指定的数量将乘以 CATENTSHIP 数据库表中的产品目录条目的额定数量。结果称为请求数量。

请求数量将舍入到此产品目录条目的下一个最高的数量倍数。例如，如果倍数为 2 千克，而请求数量为 4.1 千克，则舍入的结果将是 6 千克。当检查库存时将使用舍入的数量，它具有自己的数量单位。如果库存的数量单位不同于产品目录条目的数量单位，则在这两个单位之间必须有一个转换。

当启用可承诺（ATP）库存时（请参阅 STORE 表的 ALLOCATIONGOODFOR 列），库存的数量单位在 BASEITEM 表的 QUANTITYMEASURE 列中定义。否则，它在 INVENTORY 表的 QUANTITYMEASURE 列中定义。

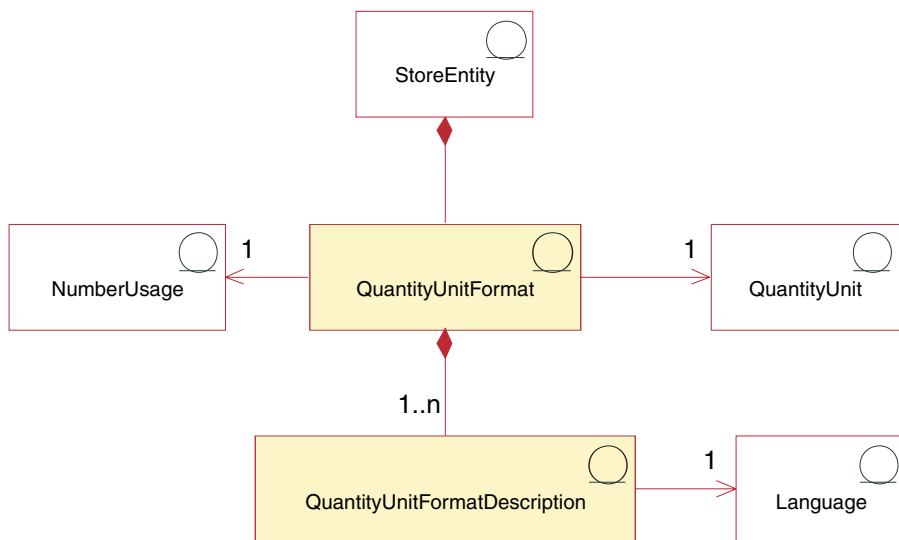
舍入数量除以产品目录条目的额定数量所得的值称为规格化数量。规格化数量存储在订购商品或兴趣商品中，这取决于正在运行的命令。例如，如果舍入数量为 6 千克，额定数量为 2 千克，则规格化数量为 3。

当找到产品目录条目的报价时，请求数量可以影响哪个报价给出最佳价格，所以请求数量确定了将使用的报价。例如，如果舍入数量为 6 千克，而有两个报价，一个指定 2 千克额定数量和 10 千克最小数量的情况下价格为 ¥32，另一个指定 2 千克额定数量和 2 千克最小数量的情况下价格为 ¥36，则仅使用第二个报价。

---

### 了解 WebSphere Commerce 中的计量单位

下图说明了 WebSphere Commerce Server 中的计量单位的结构：



本图和商店数据部分中的其它所有图都是 WebSphere Commerce Server 信息模型的一部分。关于信息模型的更多信息，请参阅第 97 页的『商店数据信息模型』。关于本图中所使用的约定的更多信息，请参阅第 415 页的附录 A，『UML 图注』。

## 数量单位和数量单位格式

数量单位是商店中使用的计量单位，例如，千克、磅、米、英寸、公升等等。数量单位格式是在商店中格式化此数量单位的方式，例如当显示此数量单位时将使用几位小数位。

每个数量单位格式是一个商店实体的组成部分（且只属于此实体），但是每个商店实体可以具有若干个数量单位格式。

每个数量单位和数值的用法都可以有一个数量单位格式；根据商店支持多少种语言，数量单位格式可以具有一个或多个数量单位格式描述。

**Business** 在一家商店中定义的数量单位可以由其它商店使用。为了让一家商店使用在另一家商店中定义的数量单位，必须在商店之间定义类型为 `com.ibm.commerce.measurement.format` 的商店关系。关于更多信息，请参阅第 129 页的第 14 章，『商店间的关系』。

### 数量单位格式描述

数量单位格式描述描述了如何（为了显示的目的）格式化以特定数量单位和特定语言表示的数额。

### 数值用法

数值的用法定义了在使用应用程序中使用数值的方式。例如，通过使用 WebSphere Commerce 代码中的数值用法代码，您可以选择您喜欢的方式来格式化或舍入该数值（货币或数量）。这些代码（在 `NUMBRUSG` 表中定义）使数值根据在 `CURFORMAT`、`CURFMTDESC`、`QTYFORMAT` 和 `QTYFMTDESC` 表中为该类型的数值用法指定的规则来进行格式化。这使商店能够以不同的方式来格式化数值以满足不同场合的需要。





---

关于 WebSphere Commerce Server 中计量单位有用资源的结构更多详细信息，请参阅 WebSphere Commerce 联机帮助中的数量单位数据模型。

---

---

## 在 WebSphere Commerce 中创建计量单位

当创建实例时，在 WebSphere Commerce Server 数据库中预填充了计量单位。关于更多信息，请参阅第 109 页的第 11 章，『站点有用资源』。

您还可以在 WebSphere Commerce 中定义在商店中使用的新的计量单位，或者删除您不想在商店中使用的计量单位。

要定义在商店中使用的新的计量单位，请向以下数据库表添加信息：

- QTYUNIT
- QTUNITDSC
- QTYFORMAT
- QTYFMTDESC
- QTYUNITMAP
- QTYCONVERT



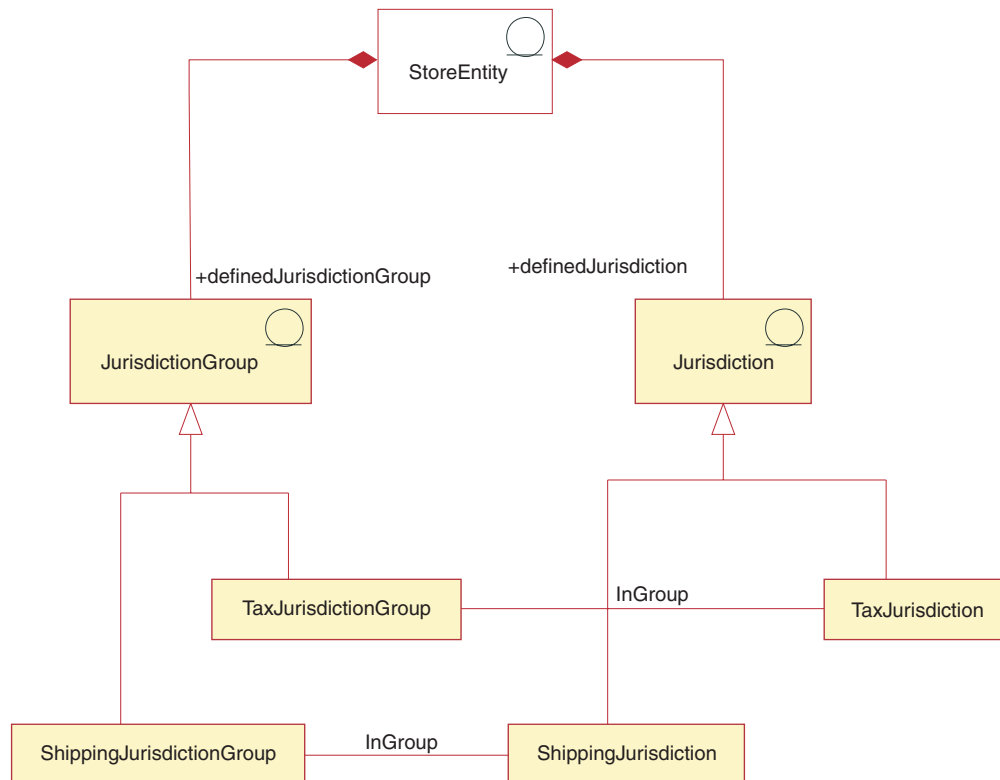
## 第 25 章 地区有用资源

地区是地理意义上的区域，它代表您对其销售商品的一个国家或地区、省 / 直辖市或领土，或邮政编码范围。可以将地区分组在一起形成地区组。

地区组用于订单上的送货费用和税款费用的计算。即，地区组可以用于限定所使用的送货费用和税款计算规则。只有商品将发送到与计算规则关联的地区组中的某个地区内的地址时，这些符合条件的计算规则才适用于订单上的这些商品。所以，根据订单中不同商品的送货地址，送货费用和税款金额的计算会有不同。

### 了解 WebSphere Commerce 中的地区有用资源

下图说明了地区和地区组如何整合到 WebSphere Commerce Server 中。



本图和商店数据部分中的其它所有图都是 WebSphere Commerce Server 信息模型的一部分。关于信息模型的更多信息，请参阅第 97 页的『商店数据信息模型』。关于本图中所使用的约定的更多信息，请参阅第 415 页的附录 A，『UML 图注』。

在 WebSphere Commerce 中，地区或地区组是商店的组成部分，而对于为其创建该地区或地区组的商店或商店组而言，该地区或地区组是此商店或商店组专用的。例如，如果您的商店创建了三个地区，然后删除了该商店，则同时也会删除地区。这些地区对于任何其它现有的商店或可能在将来创建的任何商店而言，都是不可使用的。

不过，如果您为商店组创建了地区，在删除该组中的商店时并不会删除地区。地区将可用于在该商店组中新创建的商店。

WebSphere Commerce 支持两种类型的地区：送货地区和税收地区。可以将送货地区分组在一起形成送货地区组，送货地区组限定送货费用计算规则。类似地，可以将税收地区分组在一起形成税收地区组，税收地区组限定税款计算规则。



关于 WebSphere Commerce Server 中地区有用资源的结构更多详细信息，请参阅 WebSphere Commerce 联机帮助中的地区数据模型。

---

---

## 在 WebSphere Commerce 中创建地区有用资源

为了应用税款和送货费用，您必须为您的商店创建地区有用资源。关于创建地区的更多信息，请参阅第 242 页的『在 WebSphere Commerce 中创建税款有用资源』或第 225 页的『在 WebSphere Commerce 中创建送货有用资源』。

一旦为您的商店创建了地区，您就可以使用 WebSphere 贸易加速器上商店工具中的“税款和送货笔记本”来编辑地区或创建新的地区。

**注：**对于每个创建的地区，会自动创建一个地区组。地区是为商店创建的，而不是为商店组创建的。

## 第 26 章 送货有用资源

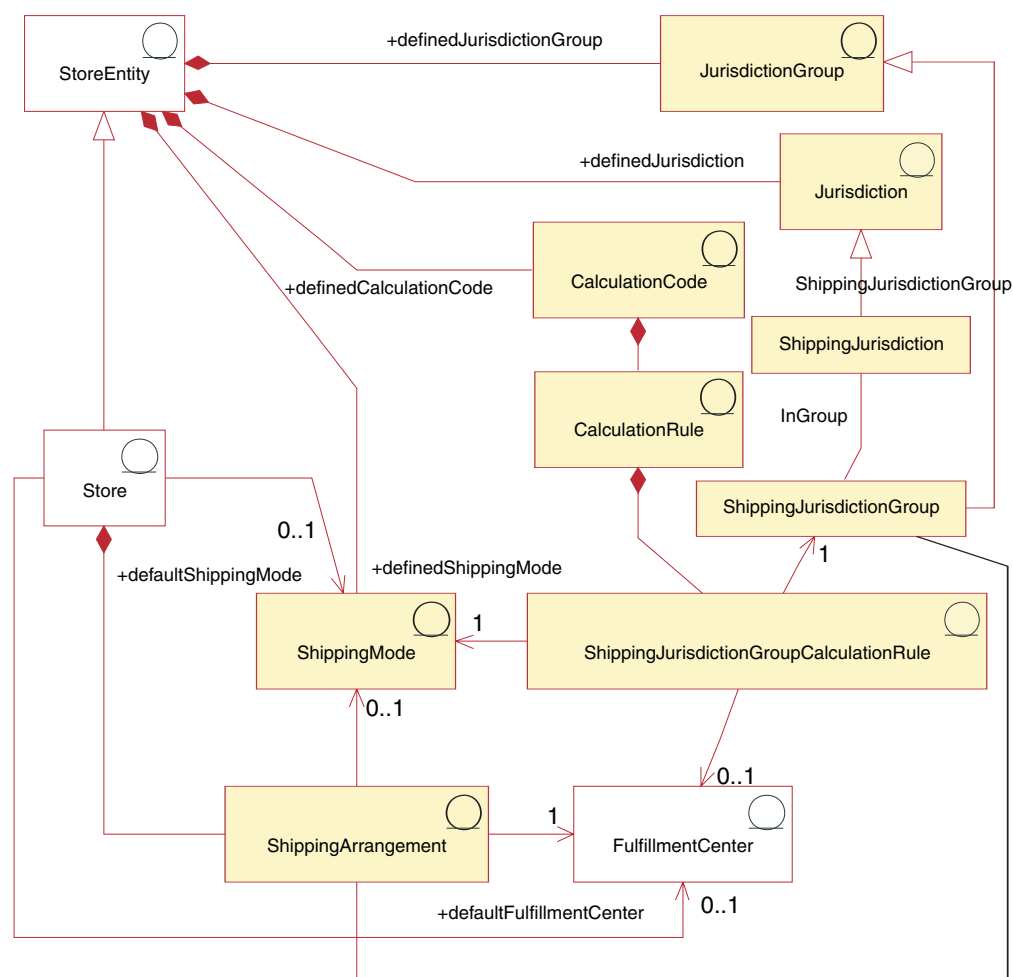
送货指的是商店如何处理物理递送给客户的商品。大部分情况下，商品从供货中心发送，供货中心是负责储存商店商品的独立中介。

为了提供送货服务，并对这些服务收费，用 WebSphere Commerce 创建的商店应当包含以下内容：

- 至少一种送货方式
- 至少一个送货费用计算代码
- 地区和地区组

### 了解 WebSphere Commerce 中的送货有用资源

下图说明了 WebSphere Commerce Server 中的送货结构。





---

本图和商店数据部分中的其它所有图都是 WebSphere Commerce Server 信息模型的一部分。关于信息模型的更多信息，请参阅第 97 页的『商店数据信息模型』。关于本图中所使用的约定的更多信息，请参阅第 415 页的附录 A，『UML 图注』。

---

## 送货方式

送货方式是发送商品的方式。更具体地说，送货方式是货物递送者（它是提供从供货中心到客户的送货服务的公司）和该货物递送者所提供的送货服务的组合。例如，“ABC 送货公司，隔夜服务”以及“ABC 送货公司，快递”都是送货方式。

送货方式属于商店实体。如果删除了商店实体，则在该商店实体内定义送货方式也将被删除。商店并不一定要有缺省的送货方式，但建议还是设置一个缺省送货方式。

### 送货安排

送货安排是商店和供货中心之间的一种安排，它指示供货中心将使用指定的送货方式发送特定商店的商品。可以对某个送货安排设置某些限制，包括送货安排的有效期限和送货地区。

如果送货安排与某个送货方式关联，则该送货安排仅适用于该送货方式。否则，送货安排适用于所有可用的送货方式。送货安排是商店的组成部分，如果删除了商店，则也将删除它。

## 计算代码

计算代码用于计算送货费用，即，送货计算代码指示如何计算订购商品的送货费用。要计算订购商品的送货费用，您必须将送货费用计算代码指定给一个产品目录条目或一组产品目录条目。

计算代码是商店实体的组成部分。一个计算代码只能与一个商店实体相关联，但是一个商店实体可以有几个计算代码。如果删除了商店实体，则与该商店实体关联的计算代码也将被删除。



---

关于计算代码使用的更多信息，请参阅《WebSphere Commerce 计算框架指南》。

---

### 计算规则

每一计算代码都有一组计算规则。根据送货方式、供货中心以及送货地区的不同，订购商品的送货费用可能会发生变化。ShippingJurisdictionGroupCalculationRules 是将送货费用计算规则与地区、供货中心和送货方式相关联的关系对象，用于确定每一订购商品应当使用的计算规则。

如果删除了计算规则或者 ShippingJurisdictionGroupCalculationRules 引用的任何其它对象，则也将删除 ShippingJurisdictionGroupCalculation 规则。



---

关于计算代码使用的更多信息，请参阅《WebSphere Commerce 计算框架指南》。

---

## 地区和地区组

地区是地理意义上的区域，它代表您对其销售商品的一个国家或地区、省 / 直辖市或领土，或邮政编码范围。将地区分组在一起就形成地区组。

WebSphere Commerce 支持两种类型的地区：送货地区和税收地区。这些地区中每一个都是相应组的组成部分，例如，送货地区在送货地区组中，而税收地区在税收地区组中。

地区组与计算规则关联。计算规则使用地区组作为计算的一部分以确定送货费用金额。

地区和地区组是商店实体的组成部分。如果删除了商店实体，则与该商店实体关联的地区和地区组也将被删除。

一个送货地址可以对应到若干送货地区。例如，美国纽约的送货地址适用于以下送货地区：“美国纽约”、“美国”和“世界”。当送货地址适用于多个送货地区时，将适用若干送货费用计算规则。在此情况下，关联的 ShippingJurisdictionGroupCalculationRules 的优先级用于确定使用什么规则。



---

关于 WebSphere Commerce Server 中送货有用资源结构的更多详细信息，请参阅 WebSphere Commerce 联机帮助中的送货数据模型。

---

## 在 WebSphere Commerce 中创建送货有用资源

WebSphere 贸易加速器中的送货工具使您能够创建与编辑某些送货有用资源（例如送货方式和地区），但不是所有的送货有用资源。下表详细描述了可以由送货工具编辑的数据库表：

- CALCODE
- CALCODEDSC
- CALRULE
- SHPICRULE
- CRULESCALE
- CALSCALE
- CALSCALEDS
- CALRANGE
- CALRLOOKUP
- SHIPMODE
- SHPMODEDSC
- SHPARRANGE
- SHPARJURGP
- JURST
- JURSTGROUP
- JURSTGPREL
- CATENCALCD

- CATGPCALCD

您还能以 XML 文件（这些文件可以使用装入程序软件包装入到数据库中）格式创建送货有用资源。这样，有以下两种选择可用于创建送货有用资源：

- 创建新的送货有用资源或编辑随 WebSphere Commerce 提供的样本商店之一的现有送货有用资源
- 以 XML 文件的格式创建新的送货有用资源。

关于使用 WebSphere 贸易加速器 创建或编辑送货有用资源的信息，请参阅 WebSphere Commerce 产品联机帮助。关于以 XML 文件格式创建新的送货有用资源的信息，请参阅『使用 XML 文件创建送货有用资源』。

## 使用 XML 文件创建送货有用资源

请以 XML 文件（这些文件可以使用装入程序软件包装入到数据库中）格式创建送货有用资源。关于装入程序软件包的更多信息，请参阅第 309 页的第 10 部分，『发布商店』。如果正在创建全球化商店，则可能要为商店支持的每种语言环境创建各自的 XML 文件。特定于语言环境的文件应当指定所有描述信息，以便比较容易进行翻译。关于创建全球化商店的更多信息，请参阅第 287 页的第 34 章，『全球化』。

样本商店（这些任务中的很多示例从其中获取）对所有不需要翻译的信息使用一个 shipping.xml 文件，对需要翻译的信息针对商店支持的每种语言环境使用另一个 shipping.xml 文件。特定于语言环境的文件包含所有描述信息，因此可以容易地进行翻译。

要使用 XML 文件为您的商店创建送货有用资源，请执行以下操作：

1. 复查《WebSphere Commerce 计算框架指南》。WebSphere Commerce 计算框架将计算与客户选择购买的产品或服务相关联的货币金额（例如，送货费用）。
2. 请复查第 367 页的第 38 章，『装入 WebSphere Commerce 数据库有用资源组』中的信息。
3. 复查用于为样本商店创建送货有用资源的 XML 文件。样本商店中的所有文件都位于相应的商店归档文件中。每个样本商店包含两个或更多 shipping.xml 文件，这些文件包含送货信息。商店归档文件位于以下目录中：

- WC\_installdir/samplestores

**注：**《WebSphere Commerce 样本商店指南》包含关于样本商店中所包含的每个数据有用资源的信息。

要查看商店归档文件中的 shipping.xml 文件，请使用 ZIP 程序将其解压缩。shipping.xml 文件位于 data 目录中。特定于语言的 shipping.xml 位于 data 目录中特定于语言环境的子目录中。

4. 请复查第 417 页的附录 B，『创建数据』中的信息。
  5. 通过复制样本商店归档文件中的一个 shipping.xml 文件或通过创建一个新文件来创建 shipping.xml 文件。关于更多信息，请参阅 wcs.dtd 文件。DTD 文件在以下目录中：
- WC\_installdir/schema/xml
6. 定义地区和地区组，您正在将商品和服务发送到该地区和地区组。所有地区都必须属于一个地区组。



- a. 使用以下示例作为指南，在您的 XML 文件中为 JURSTGROUP 表定义一个地区组：

```
< jurstgroup
  jurstgroup_id="@jurstgroup_id_1"
  description="Jurisdiction Group1 for Shipping"
  subclass="1"
  storeent_id="@storeent_id_1"
  code="World"/>
```

其中

- jurstgroup\_id 是生成的唯一键
  - description 是地区组的简短描述，它适合于在管理地区组的用户界面中显示。
  - subclass 是如下地区组子类：
    - 1 = ShippingJurisdictionGroup
    - 2 = TaxJurisdictionGroup
  - storeent\_id 是与此地区组关联的商店实体。
  - code 与其商店实体和子类一起唯一标识该地区组。
- b. 使用以下示例作为指南，在您的 XML 文件中为 JURST 表定义地区。

```
< jurst
  jurst_id="@jurst_id_1"
  storeent_id="@storeent_id_1"
  code="World"
  subclass="1"/>
```

其中

- jurst\_id 是生成的唯一键
  - storeent\_id 是与此地区组关联的商店实体。
  - code 与其商店实体和子类一起唯一标识该地区组。
  - subclass 是如下地区子类：
    - 1 = ShippingJurisdiction
    - 2 = TaxJurisdiction
- c. 使用以下示例作为指南，通过向 JURSTGRPRL 表添加信息，将您在步骤 b 中创建的地区与在步骤 a 中创建的地区组相关联。

```
< jurstgrprrl
  jurst_id="@jurst_id_1"
  jurstgroup_id="@jurstgroup_id_1"
  subclass="1"/>
```

其中

- jurst\_id 是地区
- jurstgroup\_id 是地区组
- subclass 是地区和地区组的子类。它们应当匹配：
  - 1 = ShippingJurisdiction[Group]

- 2 = TaxJurisdiction[Group]

d. 对您商店支持的所有地区和地区组重复步骤 a 到 c。

7. 定义您的商店将使用的送货方式。

a. 使用以下示例作为指南，在您的 XML 文件中为 SHIPMODE 表定义一个送货方式：

```
<shipmode
  shipmode_id="@shipmode_id_1"
  field1
  storeent_id="@storeent_id_1"
  code="Ground 1 week"
  carrier="XYZ Carrier" />
```

其中：

- shipmode\_id 是生成的唯一键。
- field1 是可供定制的字段。
- storeent\_id 是与此送货方式关联的商店实体。
- code 是商家指定的代码，对于商店实体而言它是唯一的。
- carrier 是递送者的名称或标识。

b. 使用以下示例作为指南，将有关送货方式的信息添加到 SHPMODEDSC 表。如果您正创建多文化的商店，则应当将此信息包含在特定于语言环境的 XML 文件中：

```
< shpmodedsc
  description="International mail"
  field1="USD$5.00 per order plus USD$1.00 for each item"
  field2="5 business days"
  shipmode_id="@shipmode_id_1"
  language_id="&en_US;" />
```

其中：

- description 是 ShippingMode 的简短描述，适合于对客户显示以供选择。
- field1 和 field2 是可供定制的字段。
- shipmode\_id 是生成的唯一键。
- language\_id 是使用的语言。

c. 对您的商店中所有的送货方式重复步骤 a 和 b。

8. 定义您的商店要使用的计算代码。

a. 使用以下示例作为指南，在您的 XML 文件中为 CALCODE 表定义计算代码。

```
< calcode
  calcode_id="@calcode_id_1"
  code="shipping Code 1- per/order"
  calusage_id="-2"
  storeent_id="@storeent_id_1"
  groupby="0"
```

```
published="1"
sequence="+0.00E+000"
calmethod_id="-23"
calmethod_id_app="-24"
calmethod_id_qfy="-22"
flags="0" />
```

其中:

- `calcode_id` 是生成的唯一键。
- `code` 是在给定特定的 `CalculationUsage` 和 `StoreEntity` 时, 唯一标识此 `CalculationCode` 的字符串。
- `calusage_id` 指示使用此 `CalculationCode` 的计算的类型。例如, `CalculationCode` 可能用于计算以下一种货币金额:
  - 折扣 (-1)
  - 送货费用 (-2)
  - 销售税 (-3)
  - 送货税 (-4)
  - 赠券 (-5)
- `storeent_id` 是与此计算代码关联的商店实体。
- `groupby` 是位标志, 它向 `CalculationCodeCombineMethod` 指示在执行计算时应该如何将 `OrderItem` 分组。0 = 不分组。将所有可应用的 `OrderItem` 放置在一个组中。关于更多信息, 请参阅 `WebSphere Commerce` 联机帮助中的 `CALCODE` 表: [详细信息](#)。
- `published` 指定是否发布了计算代码:
  - 0 = 未发布 (临时禁用)
  - 1 = 已发布
  - 2 = 标记为删除 (且未发布)
- `sequence` 按照从低到高的顺序计算并应用 `CalculationCode`。如果两个计算代码具有相同的序号, 则将首先计算具有较低 `calcode_id` 的计算代码。
- `calmethod_id` 是 `CalculationCodeCalculateMethod`, 它定义了如何计算此 `CalculationCode` 的货币金额。`calmethod_id="-23"` (送货的 `CalculationCodeCalculateMethod`) 是随 `WebSphere Commerce` 提供的唯一送货计算方法。
- `calmethod_id_app` 是 `CalculationCodeApplyMethod`, 它存储关联的 `OrderItem` 的已计算金额。`calmethod_id_app="-24"` (送货的 `CalculationCodeApplyMethod`) 是随 `WebSphere Commerce` 提供的唯一送货应用方法。
- `calmethod_id_qfy` 是 `CalculationCodeQualifyMethod`, 它定义了哪些 `OrderItem` 与此 `CalculationCode` 关联。`calmethod_id_qfy="-22"` (送货的 `CalculationCodeQualifyMethod`) 是随 `WebSphere Commerce` 提供的唯一送货限定方法。
- `flags` 指定是否应该调用此 `CalculationCode` 的 `CalculationCodeQualifyMethod`.
  - 0 = 没有限制。不调用此方法
  - 1 = 限制。将调用此方法

- b. 使用以下示例作为指南，在您的 XML 文件中为 CALCODEDSC 表添加计算代码描述信息。如果您正创建全球化商店，则应当将此信息包含在特定于语言环境的 XML 文件中。

```
<calcodedsc
  calcode_id="@calcode_id_3"
  description="5.00USD per order"
  language_id="@en_US"
  longdescription="This shipping calculation code charges
5.00USD per order."
/>
```

其中

- calcode\_id 是此信息应用的计算代码。
  - description 是此计算代码的简短描述。
  - language\_id 是此信息所应用的语言。
  - longdescription 是此计算代码的详细描述。
- c. 对商店中使用的每个计算代码重复步骤 a 和 b。

9. 定义您的商店的计算规则。

- a. 使用以下示例作为指南，在您的 XML 文件中为 CALRULE 表设置计算规则：

```
< calrule
  calrule_id="@calrule_id_1"
  calcode_id="@calcode_id_1"
  startdate="1900-01-01 00:00:00.000000"
  enddate="2100-01-01 00:00:00.000000"
  sequence="+1.000000000000000E+000"
  combination="2"
  calmethod_id="-27"
  calmethod_id_qfy="-26"
  flags="1"
  identifier="1" />
```

- 其中
- calrule\_id 是生成的唯一标识。
  - calcode\_id 是计算代码，此计算规则是其组成部分。
  - startdate 是此计算规则生效的时间。
  - enddate 是此计算规则失效的时间。
  - sequence 是处理此计算规则的顺序。按照从低值到高值的顺序处理相同计算代码的计算规则。
  - combination 指定特殊处理的位标志，缺省的 CalculationRuleCombineMethod 实现将执行这些特殊处理。关于更多信息，请参阅 WebSphere Commerce 联机帮助中的 CALRULE 表。
  - calmethod\_id 是 CalculationRuleCalculateMethod，它计算一系列 OrderItem 的货币结果。
  - calmethod\_id\_qfy 是 CalculationRuleQualifyMethod，它确定一系列 OrderItem 中的哪些 OrderItem 应该发送到 CalculationRuleCalculateMethod。

- flags 由 CalculationRuleCombineMethod 使用，以确定此计算规则如何与其它计算规则组合。关于更多信息，请参阅 CALRULE 表。
- identifier 与其计算代码结合标识了此计算规则。

关于更多信息，请参阅 WebSphere Commerce 联机帮助中的 CALRULE 表。

- 对您的商店中使用的每个计算规则重复步骤 a。注意，每个计算代码可以具有几个计算规则。例如，calcode\_id="@calcode\_id\_1" 可以与若干个 calrule\_id 关联。

#### 10. 定义您的商店的计算标度。

计算标度是将应用于计算的一组范围。例如，对于送货费用，您可以具有一组重量范围，每个范围对应一个特定的费用。即，重量在 0 到 5 千克之间的产品送货费用可能为 ¥80.00。而重量在 5 到 10 千克之间的产品送货费用可能为 ¥120.00。这些范围创建了一个标度。

- 使用以下示例作为指南，在您的 XML 文件中为 CALRULE 表设置计算标度：

```
<calscale
  calscale_id="@calscale_id_1"
  code="Scale Code 1 per order USD"
  storeent_id="@storeent_id_1"
  calusage_id="-2"
  setccurr="USD"
  calmethod_id="-28"/>
```

其中

- calscale\_id 是生成的唯一标识。
- code 是在给定特定的计算用法和商店实体时，唯一标识此计算标度的字符串。
- storeent\_id 是商店实体，此计算标度是其组成部分。
- calusage\_id 指示使用此 CalculationScale 的计算的类型。例如，CalculationScale 可能用于计算以下一种货币金额：
  - 折扣 (-1)
  - 送货费用 (-2)
  - 销售税 (-3)
  - 送货税 (-4)
  - 赠券 (-5)
- setccurr 如果指定的话，它指示此计算标度的计算范围对象的范围开始值的货币。CalculationScaleLookupMethod 应该返回以此货币表示的“查找数值”。
- calmethod\_id 是 CalculationScaleLookupMethod: 给定的一组订购商品时，该 CalculationScaleLookupMethod 确定查找值、基本货币值、结果乘数以及一组数学权值，它们可以由计算标度用来计算货币金额。要确定使用何种 CalculationScaleLookupMethod，请执行以下操作：
  - 请参阅 WebSphere Commerce 联机帮助中的 CALMETHOD 表。请参阅 SUBCLASS 列的描述。单击“CALMETHOD 表: 详细信息”的链接。此表列出了可用的计算方法类型。MonetaryCalculationScaleLookupMethod 方法是 9。

- 打开引导程序文件 `wcs.bootstrap_xx_XX.xml`，其中 `xx_XX` 是语言环境的代码。引导程序文件位于以下目录中：
  - `WC_installdir/schema`
- 定位列有可用计算方法（CALMETHOD）的部分。
- 使用税款的 `calusage_ID` 值（-3 代表销售税，-4 代表送货税）定位计算方法。
- 定位子类为 7 的计算方法；有若干个。请选择满足您需要的一种。

关于更多信息，请参阅 WebSphere Commerce 联机帮助中的 CALSCALE 表。

- b. 对您的商店中使用的每个计算标度重复步骤 a。例如，对于送货费用，“时尚潮流”会创建按照订单计费的费用以及按照商品计费的费用。

#### 11. 定义计算标度的计算范围。

- a. 使用以下示例作为指南，在您的 XML 文件中为 CALRANGE 表设置计算范围。

```
<calrange
calrange_id="@calrange_id_1"
calscale_id="@calscale_id_1"
calmethod_id="-33"
rangestart="0.00000"
cumulative="0"/>
```

其中

- `calrange_id` 是生成的唯一标识。
- `calcode_id` 是计算标度，此计算范围是其组成部分。
- `calmethod_id` 是 CalculationRangeMethod，它确定来自 CalculationRangeLookupResult 的货币金额。例如，FixedAmountCalculationRangeCmd、PerUnitAmountCalculationRangeCmd 或 PercentageCalculationRangeCmd。要确定 CalculationRangeMethod，请执行以下操作：
  - 请参阅 WebSphere Commerce 联机帮助中的 CALMETHOD 表。请参阅 SUBCLASS 列的描述。单击“CALMETHOD 表：详细信息”的链接。此表列出了可用的计算方法类型。CalculationRangeMethod 是 10。
  - 打开引导程序文件 `wcs.bootstrap_xx_XX.xml`，其中 `xx_XX` 是语言环境的代码。引导程序文件位于以下目录中：
    - `WC_installdir/schema`
  - 定位列有可用计算方法（CALMETHOD）的部分。
  - 使用税款的 `calusage_ID` 值（-3 代表销售税，-4 代表送货税）定位计算方法。
  - 定位子类为 9 的计算方法；有若干个。请选择满足您需要的一种。
- `cumulative` 是有效值：
  - 0 = 仅使用与最高 RANGESTART 值匹配的 CalculationRange。
  - 1 = 使用所有匹配的 CalculationRange。将已计算的货币金额相加得到最后的结果。

关于更多信息，请参阅 WebSphere Commerce 联机帮助中的 CALRANGE 表。

- b. 对与商店中使用的计算标度关联的每个计算范围重复步骤 a。
12. 定义计算标度的计算查找值。计算查找值是与计算标度关联的值。例如，计算标度包含以下重量范围以及关联的送货价格：
- 0 - 5 千克的费用是 ¥80.00
  - 5 - 10 千克的费用是 ¥120.00

查找值是 ¥80.00 和 ¥120.00。

- a. 使用以下示例作为指南，在 XML 文件中为 CALRLOOKUP 表设置计算查找值。如果您正创建多文化的商店，则应当将此信息包含在特定于语言环境的 XML 文件中（商店支持的每种语言环境都有一个特定于该语言环境的 XML 文件）。例如，如果商店为美国和日本的客户送货，您应该在一个 XML 文件中添加美元查找值，在另一个 XML 文件中添加日元查找值。

```
<calrlookup  
calrlookup_id="@calrlookup_id_1"  
setccurr="USD"  
calrange_id="@calrange_id_1"  
value="5.00"/>
```

其中

- calrlookup\_id 是生成的唯一标识。
- calrange\_id 是计算范围，此计算范围查找结果是其组成部分。
- value 是计算范围查找结果的值，它由计算范围的计算范围方法使用以确定货币结果。

关于更多信息，请参阅 WebSphere Commerce 联机帮助中的 CALLOOKUP 表。

- b. 对与商店中使用的计算标度关联的每个查找值重复步骤 a。
13. 将计算规则与计算标度关联。
- a. 使用以下示例作为指南，在您的 XML 文件中为 CRULESCALE 表将计算标度与计算规则关联。

```
< crulescale  
calrule_id="@calrule_id_1"  
calscale_id="@calscale_id_1" />
```

其中

- calrule\_id 是计算规则。
- calscale\_id 是计算标度。

- b. 对每个计算标度和规则的关联重复步骤 a。



关于 @ 和 & 用法的更多信息，请参阅第 417 页的附录 B，『创建数据』。

---

## 创建送货供货有用资源

为使您的送货有用资源在商店中正确工作，您必须将送货地区组与计算规则关联，并将供货中心与商店中使用的送货方式关联。

您必须在可以将送货有用资源关联到供货中心之前创建供货中心有用资源。关于创建供货有用资源的更多信息，请参阅第 195 页的『在 WebSphere Commerce 中创建供货有用资源』。

在创建供货有用资源之后，请通过向 SHPJCRULE 和 SHPARRANGE 表添加信息来将送货有用资源与供货有用资源关联。请执行以下操作：

1. 复查 《WebSphere Commerce 计算框架指南》。WebSphere Commerce 计算框架计算与客户选择购买的产品或服务相关联的货币金额（例如，送货费用）。
2. 请复查第 367 页的第 38 章，『装入 WebSphere Commerce 数据库有用资源组』中的信息。
3. 复查用于为样本商店创建送货供货有用资源的 XML 文件。样本商店的所有文件都位于相应的商店归档文件中。每个样本商店包含一个 shipfulfill.xml 文件，它包含送货供货信息。要查看商店归档文件中的 shipfulfill.xml 文件，请使用 ZIP 程序将其解压缩。shipfulfill.xml 文件位于数据目录中。

商店归档文件位于以下目录中：

- WC\_installdir/samplestores

**注：** WebSphere Commerce 联机帮助包含关于样本商店中所包含的每个数据有用资源的信息。

4. 请复查第 417 页的附录 B，『创建数据』中的信息。
5. 通过复制样本商店归档文件中的一个 shipfulfill.xml 文件或通过创建一个新文件来创建 shipfulfill.xml 文件。关于更多信息，请参阅 wcs.dtd 文件。DTD 文件位于以下目录中：
  - WC\_installdir/schema/xml
6. 通过向 SHPJCRULE 表添加信息，将计算规则与送货地区组关联。使用以下示例作为指南。如果您正创建多文化商店，则还需要为商店支持的每种语言环境创建一个 XML 文件。

```
<shpjcrule
calrule_id="@calrule_id_1"
ffmcenter_id="@ffmcenter_id_1"
jurstgroup_id="@jurstgroup_id_1"
precedence="0"
shipmode_id="@shipmode_id_1"
shpjcrule_id="@shpjcrule_id_1"
```

其中

- calrule\_id 是使用的计算规则。
  - ffmcenter\_id 是供货中心。如果这为 NULL，则此关联将应用到所有的供货中心上。
  - jurstgroup\_id 是送货地区组。如果这为 NULL，则此关联将应用到所有的送货地区组上。
  - precedence 是对于同一个供货中心和送货方式，送货地址落在多个指定的送货地区组中的情况。只有具有最高的 SHPJCRULE.PRECEDENCE 值的计算规则才适用。
  - shipmode\_id 是送货方式。
  - shpjcrule\_id 是生成的唯一标识。
7. 对商店中每个地区组、供货中心和规则的关联重复步骤 3。



8. 通过向 SHPARRANGE 表添加信息，将送货方式和供货中心与商店关联。使用以下示例作为指南：

```
<shparrange
  shparrange_id="@shparrange_id_2"
  store_id="@storeent_id_1"
  ffmcenter_id="@ffmcenter_id_1"
  shipmode_id= "@shipmode_id_2"
  startdate="1970-06-22 23:00:00.000000"
  enddate= "2008-06-22 23:00:00.000000"
  precedence= "0"
  flags="0"
/>
```

其中

- shparrange\_id 是生成的唯一标识。
  - store\_id 是商店。
  - ffmcenter\_id 是供货中心。
  - shipmode\_id 是送货方式。NULL 指示不论何种送货方式都可以使用此送货安排。
  - startdate 是此送货安排开始生效的时间。
  - enddate 是此送货安排失效的时间。
  - precedence 是当特定时间（对于同一个商店和送货方式）有多个送货安排有效时，使用 PRECEDENCE 最高的一个安排。
  - flags 包含位标志：
    - 1 = 限制 - 该送货安排仅适用于某些订购商品，这些订购商品的送货地址和（通过 SHPARJURGP 表）与此送货安排关联的送货地区组之一相匹配。
9. 对商店中使用的所有送货方式重复步骤 5。



关于 @ 和 & 用法的更多信息，请参阅第 417 页的附录 B，『创建数据』。

## 创建商店 - 产品目录 - 送货有用资源

为了将送货方式与商店关联，您必须对商店包含的每个合同，将计算代码与商店中的产品目录条目关联。

您必须在可以创建“商店 - 产品目录 - 送货”有用资源之前，创建您的商店和产品目录有用资源。关于创建商店有用资源的更多信息，请参阅第 124 页的『在 XML 文件中创建商店数据有用资源』。关于创建产品目录有用资源的更多信息，请参阅第 161 页的『显示商店产品目录有用资源』。

要创建“商店 - 产品目录 - 送货”有用资源，请执行以下操作：

1. 复查《WebSphere Commerce 计算框架指南》。WebSphere Commerce 计算框架计算与客户选择购买的产品或服务相关联的货币金额（例如，送货费用）。
2. 请复查第 367 页的第 38 章，『装入 WebSphere Commerce 数据库有用资源组』中的信息。
3. 复查用于为样本商店创建送货供货有用资源的 XML 文件。样本商店的所有文件都位于相应的商店归档文件中。

商店归档文件位于以下目录中：

- *WC\_installdir/samplestores*

**注:** WebSphere Commerce 联机帮助包含关于样本商店中所包含的每个数据有用资源的信息。

每个样本商店包含一个 *store-catalog-shipping.xml* 文件，它包含送货供货信息。要查看商店归档文件中的 *store-catalog-shipping.xml* 文件，请使用 ZIP 程序将其解压缩。*store-catalog-shipping.xml* 文件位于 *data* 目录中。

4. 请复查第 417 页的附录 B，『创建数据』中的信息。
5. 通过复制样本商店归档文件中的一个 *store-catalog-shipping.xml* 文件或通过创建一个新文件来创建 *store-catalog-shipping.xml* 文件。关于更多信息，请参阅 *wcs.dtd* 文件。DTD 文件位于以下目录中：

- *WC\_installdir/schema/xml*

6. 通过向 *CATENCALCD* 表添加信息，创建“商店 - 产品目录 - 送货”关系。使用以下示例作为指南：

```
<catencalcd
  calcode_id="@calcode_id_1"
  catencalcd_id="@catencalcd_id_1"
  store_id="@storeent_id_1"
/>
```

其中

- *calcode\_id* 是计算代码。
- *catencalcd\_id* 是生成的唯一标识。
- *store\_id* 是商店。



---

关于 @ 和 & 用法的更多信息，请参阅第 417 页的附录 B，『创建数据』。

---

## 创建缺省送货方式

为了设置商店的缺省送货方式，您必须向 *STOREDEF* 表添加信息。要向 *STOREDEF* 表添加信息，请执行以下操作：

1. 请复查第 367 页的第 38 章，『装入 WebSphere Commerce 数据库有用资源组』中的信息。
2. 复查用于为样本商店创建商店缺省有用资源的 XML 文件。样本商店的所有文件都位于相应的商店归档文件中。

商店归档文件位于以下目录中：

- *WC\_installdir/samplestores*

**注:** WebSphere Commerce 联机帮助包含关于样本商店中所包含的每个数据有用资源的信息。

每个样本商店包含一个 *store-defaults.xml* 文件，它包含缺省送货信息。要查看商店归档文件中的 *store-defaults.xml* 文件，请使用 ZIP 程序将其解压缩。*store-defaults.xml* 文件位于 *data* 目录中。

3. 请复查第 417 页的附录 B，『创建数据』中的信息。

4. 通过复制样本商店归档文件中的一个 `store-defaults.xml` 文件或通过创建一个新文件来创建 `store-defaults.xml` 文件。关于更多信息，请参阅 `wcs.dtd` 文件。DTD 文件位于以下目录中：

- `WC_installdir/schema/xml`

5. 使用以下示例作为指南，在 XML 文件中，通过向 `STOREDEF` 表添加信息，指定商店的缺省送货方式：

```
<storedef
store_id="@storeent_id_1"
  shipmode_id="@shipmode_id_1"
/>
```

其中

- `store_id` 是商店。
- `shipmode_id` 是商店的缺省送货方式。



关于 `@` 和 `&` 用法的更多信息，请参阅第 417 页的附录 B，『创建数据』。

---



## 第 27 章 税款有用资源

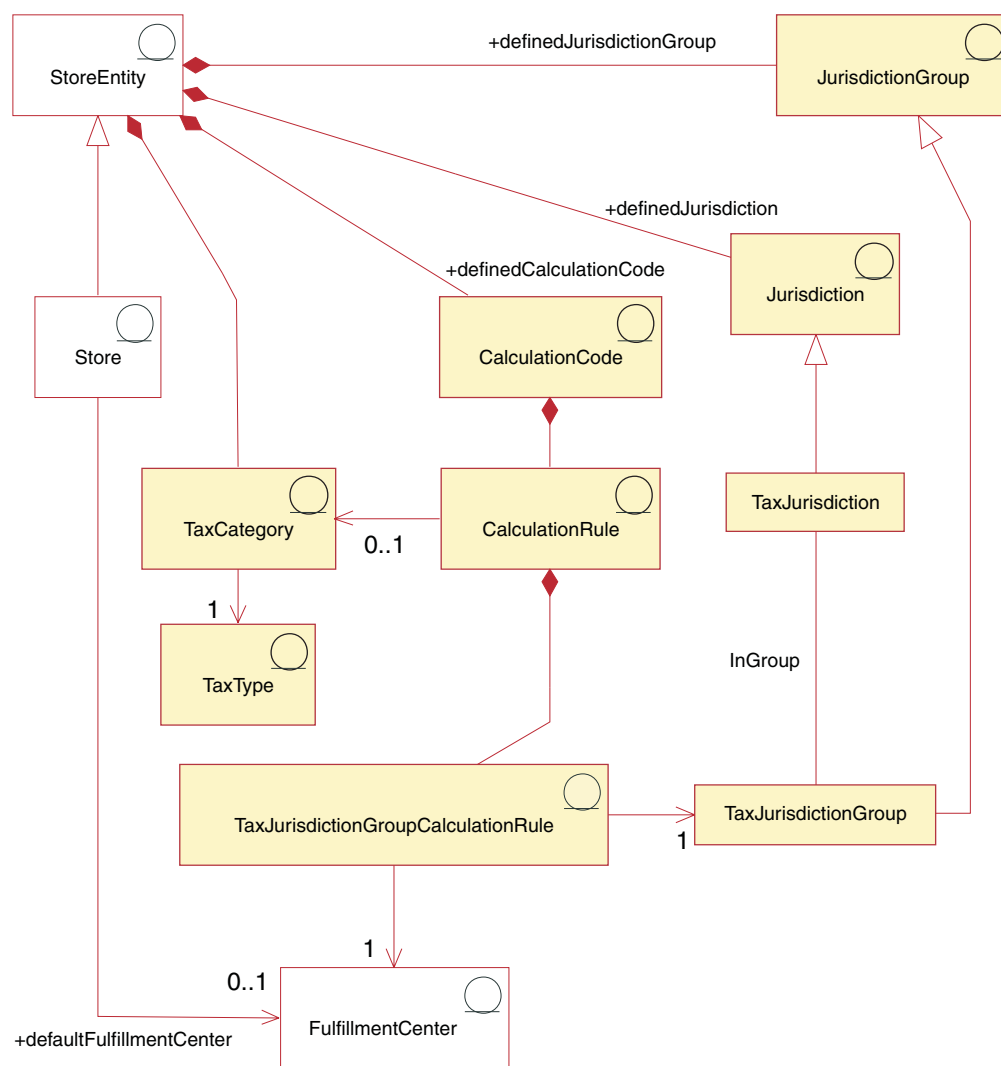
为了对您的商店提供的商品和服务收取税款，用 WebSphere Commerce 创建的商店必须包含以下内容：

- 税类别
- 计算代码
- 地区和地区组

税类别、计算代码以及地区和地区组的组合创建了此商店的税款费用。

### 了解 WebSphere Commerce 中的税款有用资源

下图说明了 WebSphere Commerce Server 中税务的结构。





本图和商店数据部分中的其它所有图都是 WebSphere Commerce Server 信息模型的一部分。关于信息模型的更多信息，请参阅第 97 页的『商店数据信息模型』。关于本图中所使用的约定的更多信息，请参阅第 415 页的附录 A，『UML 图注』。

## 税类别

税类别对应于商店可能需要收取的不同类型的税款，例如国税和地税。

税类别是一个商店实体的组成部分，尽管商店实体可以具有若干个税类别。如果删除了商店实体，则与该商店实体关联的税类别也将被删除。

### 税款类型

商店通常收取两种类型的税款：销售税或使用税，以及送货税。每个税类别具有一个税款类型。尽管每个税类别可能只属于一种税款类型（例如税类别“国税”是销售税类型），但是可能有若干个不同的税类别属于同一个税款类型（例如，税款类型“销售税”适用于类别“国税”和“地税”）。

## 计算代码

计算代码用于计算税款费用，即，税款计算代码指示如何计算订购商品的税款。要计算订购商品的税款，您必须将销售税款和送货税款计算代码指定给一个产品目录条目或一组产品目录条目。对于一个特定的产品目录条目或产品目录条目组，仅可以应用每个税款类型的一种税款计算代码。通常情况下，对净价征收销售税或使用税，对送货费用征收送货税。

计算代码是商店实体的组成部分。一个计算代码只能与一个商店实体相关联，但是一个商店实体可以有几个计算代码。如果删除了商店实体，则与该商店实体关联的计算代码也将被删除。



关于使用计算代码的更多信息，请参阅《IBM WebSphere Commerce 计算框架指南》。

### 计算规则

每个计算代码至少有一个计算规则，计算规则定义每个税类别的计算并指定进行计算的条件。每个税款计算规则都与一个税类别、一个地区组和一个供货中心关联，它们共同定义了计算规则的使用条件。例如，根据订单中指定的送货地址和供货中心的不同，可能选择不同的规则来计算特定税类别的金额。

每个计算规则只属于一个计算代码。

一个特定的税款计算代码可以具有若干个计算规则，每个与此商店关联的税类别、税收地区组和供货中心的组合都有一个计算规则。每个销售税款和送货税款计算规则都可以和多个 TaxJurisdictionGroupCalculationRules (TaxRules) 关联。如下表所示，计算规则 10001 对地区组 1234 和 1235 都适用。

TAXJCRULE_ID	CALRULE_ID	FFMCENTER_ID	JURSTGROUP_ID	PRECEDENCE
10001	10001	NULL	1234	0
10002	10001	NULL	1235	0

每个 TaxRule 定义了应当应用计算代码的条件。例如，您可以为商店送货的每一地区组定义一个计算规则。在以下示例中，计算规则 10001 对地区组 1234 和 1235 都适用。

在以下示例中，当税收地区为阿尔伯特时，税款计算代码对地方销售税类别使用计算规则 A；当税收地区为英属哥伦比亚时，则使用计算规则 C。

税收地区	国家销售税	地方销售税
阿尔伯特，加拿大	计算规则 B，它给出 Y%	计算规则 A，它给出 X%
英属哥伦比亚，加拿大	计算规则 B，它给出 Y%	计算规则 C，它给出 Z%

当一个送货地址与多个税收地区组相匹配时，则将使用其关联的 TAXJCRULE.PRECEDENCE 列值最高的计算规则。

TaxJurisdictionGroupCalculationRules (TaxRule) 与计算规则的关联确定了何时适用此计算规则。当满足 TaxRules 给定的任一条件时，将适用销售税或送货税计算规则。在以下示例中，当您发送到地区组 1001 时，或者从供货中心 1001 送货时，适用计算规则 10001。

CALRULE_ID	FFMCENTER_ID	JURSTGROUP_ID
10001	NULL	1001
10001	1001	1001

每个 TaxJurisdictionGroupCalculationRule 都至多与一个地区组关联。计算规则自身并不直接与地区组关联。



关于使用计算规则的更多信息，请参阅《IBM WebSphere Commerce 计算框架指南》。

## 地区和地区组

地区是地理意义上的区域，它代表您对其销售商品的一个国家或地区、省 / 直辖市或领土，或邮政编码范围。将地区分组在一起就形成地区组。

WebSphere Commerce 支持两种类型的地区：送货地区和税收地区。这些地区中每一个都是相应组的组成部分，例如，送货地区在送货地区组中，而税收地区在税收地区组中。

地区和地区组确定使用何种计算规则来计算税款费用。

地区和地区组是商店实体的组成部分。每个地区和地区组都是一个商店实体的组成部分，然而一个商店实体可以具有若干个地区或地区组。如果删除了商店实体，则与该商店实体关联的地区和地区组也将被删除。



关于 WebSphere Commerce Server 中税款有用资源的结构的信息，请参阅 WebSphere Commerce 联机帮助中的税款数据模型。

---

## 在 WebSphere Commerce 中创建税款有用资源

WebSphere 贸易加速器中的税款工具使您能够创建和编辑某些税款有用资源（例如税款类别和地区），但不是所有的税款有用资源。

下表详细描述了可以由税款工具编辑的数据库表：

- CALCODE
- CALCODEDSC
- CALRULE
- TAXJCRULE
- CRULESCALE
- CALSCALE
- CALSCALEDSC
- CALRANGE
- CALRLOOKUP
- TAXCGRY
- TAXCGRYDS
- JURST
- JURSTGROUP
- JURSTGPREL
- CATENCALCD
- CATGPCALCD

您还能以 XML 文件（这些文件可以使用装入程序软件包装入到数据库中）格式创建税款有用资源。这样，有以下两种选择可用于创建送货有用资源：

- 创建新的税款有用资源或编辑随 WebSphere Commerce 提供的样本商店之一的现有税款有用资源。
- 以 XML 文件的格式创建新的税款有用资源。

关于在现有商店归档文件中编辑税款有用资源的信息，或一般税款信息，请参阅 WebSphere Commerce 联机帮助。关于以 XML 文件格式创建新的税款有用资源的信息，请参阅『使用 XML 文件创建税款有用资源』。

### 使用 XML 文件创建税款有用资源

请以 XML 文件（这些文件可以使用装入程序软件包装入到数据库中）格式创建税款有用资源。关于装入程序软件包的更多信息，请参阅第 309 页的第 10 部分，『发布商店』。如果正在创建全球化商店，则可能要为商店支持的每种语言环境创建各自的 XML 文件。特定于语言环境的文件应当指定所有描述信息，以便比较容易进行翻译。

样本商店（这些任务中的很多示例从其中获取）对所有不需要翻译的信息使用一个 tax.xml 文件，对需要翻译的信息针对商店支持的每种语言环境使用另一个 tax.xml 文件。特定于语言环境的文件包含所有描述信息。

要使用 XML 文件为您的商店创建税款有用资源，请执行以下操作：



1. 请复查第 367 页的第 38 章,『装入 WebSphere Commerce 数据库有用资源组』中的信息。复查《IBM WebSphere Commerce 计算框架指南》。WebSphere Commerce 计算框架计算与客户选择购买的产品或服务相关联的货币金额(例如,税款)。

2. 复查用于为样本商店创建税款有用资源的 XML 文件。样本商店的所有文件都位于相应的商店归档文件中。

商店归档文件位于以下目录中:

- `WC_installdir/samplestores`

3. 请复查第 417 页的附录 B,『创建数据』中的信息。

4. 通过复制样本商店归档文件中的一个 `tax.xml` 文件或通过创建一个新文件来创建 `tax.xml` 文件。关于更多信息,请参阅相应于 `tax.xml` 的 DTD 文件。DTD 文件在以下目录中:

- `WC_installdir/schema`

5. 定义您正将商品和服务发送到的地区和地区组。根据适用的税类别计算规则,将税收地区指定到地区组。

a. 使用以下示例作为指南,在您的 XML 文件中为 JURSTGROUP 表定义一个地区组:

```
< jurstgroup
  jurstgroup_id="@jurstgroup_id_2"
  description="Tax Jurstiction Group 1"
  subclass="2"
  storeent_id="@storeent_id_1"
  code="World"/>
```

其中

- `jurstgroup_id` 是生成的唯一键
- `description` 是地区组的简短描述,它适合于在管理地区组的用户界面中显示。
- `subclass` 是如下地区组子类:
  - 1 = `ShippingJurisdictionGroup`
  - 2 = `TaxJurisdictionGroup`
- `storeent_id` 是与此地区组关联的商店实体。
- `code` 与其商店实体和子类一起唯一标识该地区组。

b. 使用以下示例作为指南,在您的 XML 文件中为 JURST 表定义地区。

```
< jurst
  jurst_id="@jurst_id_2"
  storeent_id="@storeent_id_1"
  code="World"
  subclass="2"/>
```

其中

- `jurst_id` 是生成的唯一键
- `storeent_id` 是与此地区组关联的商店实体。
- `code` 与其商店实体和子类一起唯一标识该地区组。
- `subclass` 是如下地区子类:

- 1 = ShippingJurisdiction
  - 2 = TaxJurisdiction
- c. 使用以下示例作为指南，通过向 JURSTGRPREL 表添加信息，将您在步骤 b 中创建的地区与在步骤 a 中创建的地区组相关联。

```
<jurstgprel
  jurst_id="@jurst_id_2"
  jurstgroup_id="@jurstgroup_id_1"
  subclass="2"/>
```

其中

- jurst\_id 是地区
  - jurstgroup\_id 是地区组
  - subclass 是地区和地区组的子类。它们应当匹配：
    - 1 = ShippingJurisdiction[Group]
    - 2 = TaxJurisdiction[Group]
- d. 对您商店支持的所有地区和地区组重复步骤 a 到 c。
6. 定义您的商店将使用的税类别。
- a. 使用以下示例作为指南，在您的 XML 文件中为 TAXCGRY 表定义一个税类别：

```
<taxcgry
  taxcgry_id="@taxcgry_id_1"
  taxtype_id="-3"
  storeent_id="@storeent_id_1"
  name="Sales Tax"
  displayseq="0"
  displayusage="0"/>
```

其中：

- taxcgry\_id 是生成的唯一键。
  - taxtype\_id="-3" 是此税类别的税款类型。WebSphere Commerce 支持两种税款类型：
    - 销售税或使用税 (-3)
    - 送货税 (-4)
  - storeent\_id 是与此税类别关联的商店实体。
  - name 是税类别的名称。此名称与商店实体一起唯一标识此税类别。
  - displayseq 指定税款金额显示的顺序（从低到高），例如在订单中。
  - displayusage 指定此税类别与 PriceDataBean 有关，如下：
    - 0 = 不计算
    - 1 = 已计算

PriceDataBean 可以用于获取应当与产品价格一起显示的税款金额。
- b. 对商店中使用的每个税类别重复步骤 a。
- c. 使用以下示例作为指南，在您的 XML 文件中为 TAXCGRYDS 表添加税类别描述信息。如果您正创建多文化的商店，则应当将此信息包含在特定于语言环境的 XML 文件中。

```
<taxcgryds
  taxcgry_id="@taxcgry_id_1"
  description="Sales Tax"
  language_id="&en_US"/>
```

其中

- taxcgry\_id 是税类别。
- description 是适合显示给客户的税类别的简短描述。
- language\_id 是显示此信息的语言。

d. 对商店中使用的每个税类别重复步骤 c。

7. 定义您的商店要使用的计算代码。

a. 使用以下示例作为指南，在您的 XML 文件中为 CALCODE 表定义计算代码。

```
<calcode
  calcode_id="@calcode_id_3"
  code="Tax Code 1"
  calusage_id="-3"
  storeent_id="@storeent_id_1"
  groupby="0"
  published="1"
  sequence="0"
  calmethod_id="-43"
  calmethod_id_app="-44"
  calmethod_id_qfy="-42"
  displaylevel="0"
  flags="0"
  precedence="0"
/>
```

其中:

- calcode\_id 是生成的唯一键。
- code 是在给定特定的计算用法和商店实体时，唯一标识此计算代码的字符串。
- calusage\_id 指示使用此计算代码的计算类型。例如，计算代码可能用于计算以下一种货币金额：
  - 折扣 (-1)
  - 送货费用 (-2)
  - 销售税 (-3)
  - 送货税 (-4)
  - 赠券 (-5)
- storeent\_id 是与此计算代码关联的商店实体。
- groupby 是位标志，它向计算代码组合方法指示在执行计算时应该如何将 OrderItem 分组。零指定不进行分组（所有适用的订购商品都在单个的组中）。关于更多信息，请参阅 WebSphere Commerce 联机帮助中的 CALCODE 表：*详细信息*。
- published 指定是否发布了计算代码：
  - 0 = 未发布（临时禁用）
  - 1 = 已发布
  - 2 = 标记为删除（并且不发布）

- `sequence` 是计算此计算代码的顺序。按照从低到高的顺序来计算和应用计算代码。如果两个计算代码具有相同的序号，则将首先计算具有较低 `calcode_id` 的计算代码。
  - `calmethod_id` 计算代码计算方法，它定义了如何计算此计算代码的税款金额。为了确定使用何种计算代码计算方法，请执行以下操作：
    - 请参阅 WebSphere Commerce 联机帮助中的 CALMETHOD 表。请参阅 SUBCLASS 列的描述。单击“CALMETHOD 表：详细信息”的链接。此表列出了可用的 CALMETHOD 类型。计算代码计算方法类型是 3。
    - 打开引导程序文件 `wcs.bootstrap_xx_XX.xml`，其中 `xx_XX` 是语言环境的代码。引导程序文件位于以下目录中：
      - `WC_installdir/schema/xml`
    - 定位列有可用计算方法（CALMETHOD）的部分。
    - 使用税款的 `calusage_ID` 值（-3 代表销售税，-4 代表送货税）定位计算方法。
    - 定位子类为 3 的计算方法。此计算方法是 -43。
  - `calmethod_id_app` 是 `CalculationCodeApplyMethod`，它存储关联的 `OrderItem` 的已计算金额。使用 `calmethod_id` 中所述的方法来确定使用何种计算代码应用方法。
    - `calmethod_id_app="-44"` 是销售税的 `CalculationCodeApplyMethod`。
  - `calmethod_id_qfy` 是 `CalculationCodeQualifyMethod`，它定义了哪些订购商品与此计算代码关联。使用 `calmethod_id` 中所述的方法来确定使用何种计算代码限定方法。
    - `calmethod_id_qfy="-42"` 是销售税的 `CalculationCodeQualifyMethod`。
  - `display level` 确定由该计算代码计算的金额是否应该与以下每项一起显示：
    - 0 = 订购商品
    - 1 = 订单
    - 2 = 产品
    - 3 = 商品
    - 4 = 合同
  - `flags` 指定是否应该调用此计算代码的 `CalculationCodeQualifyMethod`。
    - 0 = 没有限制。不调用此方法
    - 1 = 限制。将调用此方法
- b. 使用以下示例作为指南，在您的 XML 文件中为 CALCODEDSC 表添加计算代码描述信息。如果您正创建多文化的商店，则应当将此信息包含在特定于语言环境的 XML 文件中。

```
<calcodedsc
  calcode_id="@calcode_id_3"
  description="Vitamins
  language_id="@en_US"
  longdescription="In Ontario vitamins are taxed federally, but
  not provincially."
/>
```

其中

- `calcode_id` 是此信息应用的计算代码。

- description 是此计算代码的简短描述。
  - language\_id 是此信息所应用的语言。
  - longdescription 是此计算代码的详细描述。
- c. 对商店中使用的每个计算代码重复步骤 a 和 b。
8. 定义您的商店的计算规则。
- a. 使用以下示例作为指南，在您的 XML 文件中为 CALRULE 表设置计算规则：

```
<calrule
  calrule_id="@calrule_id_10"
  calcode_id="@calcode_id_3"
  startdate="1900-01-01 00:00:00.000000"
  taxcgry_id="@taxcgry_id_1"
  enddate="2100-01-01 00:00:00.000000"
  flags="1"
  identifier="1"
  combination="2"
  calmethod_id="-47"
  calmethod_id_qfy="-46"
/>
```

其中

- calrule\_id 是生成的唯一标识。
- calcode\_id 是计算代码，此计算规则是其组成部分。
- startdate 是此计算规则生效的时间。
- taxcgry\_id 是此计算规则对其生效的税类别。
- enddate 是此计算规则失效的时间。
- combination 由 CalculationRuleCombineMethod 使用，以确定此计算规则如何与其它计算规则组合。关于更多信息，请参阅 CALRULE 表。
- identifier 与其计算代码结合标识了此计算规则。
- flags 指定位标志以指示将由缺省 CalculationRuleCombineMethod 实现执行的特殊处理。关于更多信息，请参阅 WebSphere Commerce 联机帮助中的 CALRULE 表。
- calmethod\_id 是 CalculationRuleCalculateMethod，它计算一系列订购商品的货币结果。要确定使用何种计算规则计算方法，请执行以下操作：
  - 请参阅 WebSphere Commerce 联机帮助中的 CALMETHOD 表。请参阅 SUBCLASS 列的描述。单击“CALMETHOD 表：详细信息”的链接。此表列出了可用的 CALMETHOD 类型。计算规则计算方法是 7。
  - 打开引导程序文件 wcs.bootstrap\_xx\_XX.xml，其中 xx\_XX 是语言环境的代码。引导程序文件位于以下目录中：
    - WC\_installdir/schema/xml
  - 定位列有可用计算方法（CALMETHOD）的部分。
  - 使用税款的 calusage\_ID 值（-3 代表销售税，-4 代表送货税）定位计算方法。
  - 定位子类为 7 的计算方法。此计算方法是 -47。
- calmethod\_id\_qfy 是 CalculationRuleQualifyMethod，它确定一系列 OrderItem 中的哪些 OrderItem 应该发送到 CalculationRuleCalculateMethod。使用 calmethod\_id 中所述的方法来确定使用何种计算规则限定方法。

- b. 对您的商店中使用的每个计算规则重复步骤 a。注意，每个计算代码可以具有几个计算规则，一个规则用于一个适用的税类别。例如，`calcode_id="@calcode_id_1"` 可以与若干个 `calrule_id` 关联。
9. 定义您的商店的计算标度。

计算标度是将应用于计算的一组范围。这些范围创建了一个标度。

- a. 使用以下示例作为指南，在您的 XML 文件中为 CALRULE 表设置计算标度：

```
<calscale
  calscale_id="@calscale_id_19"
  code="Sales Tax 1"
  storeent_id="@storeent_id_1"
  calusage_id="-3"
  setcurr="USD"
  calmethod_id="-53"
/>
```

其中

- `calscale_id` 是生成的唯一标识。
- `code` 是在给定特定的计算用法和商店实体时，唯一标识此计算标度的字符串。
- `storeent_id` 是商店实体，此计算标度是其组成部分。
- `calusage_id` 指示使用此 `CalculationScale` 的计算的类型。例如，`CalculationScale` 可能用于计算以下一种货币金额：
  - 折扣 (-1)
  - 送货费用 (-2)
  - 销售税 (-3)
  - 送货税 (-4)
  - 赠券 (-5)
- `setcurr` 如果指定的话，它指示此计算标度的计算范围对象的范围开始值的货币。`CalculationScaleLookupMethod` 将以此货币返回一个“查找号码”。在此情况下号码是未指定的；`CalculationScaleLookupMethod` 将以订单中的货币返回一个查找号码。货币是不需要指定的，除非标度范围开始值不是零。
- `calmethod_id` 是 `CalculationScaleLookupMethod`，给定一组订购商品，它就确定了一个查找数值、基本货币值、结果乘法器以及一组数学重量，它们可以由计算标度使用来计算货币金额。要确定使用何种 `CalculationScaleLookupMethod`，请执行以下操作：
  - 请参阅 `WebSphere Commerce` 联机帮助中的 `CALMETHOD` 表。请参阅 `SUBCLASS` 列的描述。单击“`CALMETHOD` 表：详细信息”的链接。此表列出了可用的 `CALMETHOD` 类型。`MonetaryCalculationScaleLookupMethod` 方法是 9。
  - 打开引导程序文件 `wcs.bootstrap_xx_XX.xml`，其中 `xx_XX` 是语言环境的代码。引导程序文件位于以下目录中：
    - `WC_installdir/schema/xml`
  - 定位列有可用计算方法（`CALMETHOD`）的部分。
  - 使用税款的 `calusage_ID` 值（-3 代表销售税，-4 代表送货税）定位计算方法。
  - 定位子类为 9 的计算方法。有若干个计算方法的子类是 9。从中选取一个适合您需要的方法。

关于更多信息，请参阅 WebSphere Commerce 联机帮助中的 CALSCALE 表。

- b. 对您的商店中使用的每个计算标度重复步骤 a。
- c. 使用以下示例作为指南，在您的 XML 文件中为 CALSCALDS 表添加计算标度描述信息。如果您正创建多文化的商店，则应当将此信息包含在特定于语言环境的 XML 文件中。

```
<calscalds
  calscale_id="@calscale_id_19"
  description="Sales Tax 5% "
  language_id="&en_US"
/>
```

其中

- calscale\_id 是此描述适用的计算标度。
- description 是计算标度的简短描述，适合显示给客户以解释计算是怎样进行的。例如，“每千克 ¥.80，最少收费 ¥40.00。”或“5 个或 5 个以上 9 折。”
- language\_id 是显示此信息的语言。

- d. 对您的商店中使用的每个计算标度重复步骤 c。

#### 10. 定义计算标度的计算范围。

- a. 使用以下示例作为指南，在您的 XML 文件中为 CALRANGE 表设置计算范围。

```
<calrange
  calrange_id="@calrange_id_37"
  calscale_id="@calscale_id_19"
  calmethod_id="-59"
  rangestart="0.00000"
  cumulative="0"
/>
```

其中

- calrange\_id 是生成的唯一标识。
- calcode\_id 是计算标度，此计算范围是其组成部分。
- calmethod\_id 是 CalculationRangeMethod，它确定来自 CalculationRangeLookupResult 的货币金额。例如，FixedAmountCalculationRangeCmd、PerUnitAmountCalculationRangeCmd 或 PercentageCalculationRangeCmd。要确定 CalculationRangeMethod，请执行以下操作：
  - 请参阅 WebSphere Commerce 联机帮助中的 CALMETHOD 表。请参阅 SUBCLASS 列的描述。单击“CALMETHOD 表：详细信息”的链接。此表列出了可用的 CALMETHOD 类型。CalculationRangeMethod 是 10。
  - 打开引导程序文件 wcs.bootstrap\_xx\_XX.xml，其中 xx\_XX 是语言环境的代码。引导程序文件位于以下目录中：
    - WC\_installdir/schema/xml
  - 定位列有可用计算方法（CALMETHOD）的部分。
  - 使用税款的 calusage\_ID 值（-3 代表销售税，-4 代表送货税）定位计算方法。
  - 定位子类为 10 的计算方法。有若干种计算方法的子类为 10。请选择满足您需要的一种。

- rangestart 表示如果查找号码大于或等于 RANGESTART，或者 RANGESTART 是 NULL，则此行与查找号码匹配。
- cumulative 如下：
  - 0 = 仅使用与最高 RANGESTART 值匹配的 CalculationRange。
  - 1 = 使用所有匹配的 CalculationRange。将已计算的货币金额相加得到最后的结果。

关于更多信息，请参阅 WebSphere Commerce 联机帮助中的 CALRANGE 表。

- 对与商店中使用的计算标度关联的每个计算范围重复步骤 a。由于所有金额都以相同的税率征税，所以在以上例子中只有一个范围。
11. 定义计算标度的计算查找值。计算查找值是计算标度关联的值。例如，计算标度包含以下范围以及关联的税率（对旅馆提供的餐饮征收的安大略地方销售税）：
- \$0.00 - \$3.99 以 0.00% 税率征税
  - \$4.00 及以上以 8.00% 税率征税
- 查找值是 0.00 和 8.00。

- 使用以下示例作为指南，在您的 XML 文件中为 CALRLOOKUP 表设置计算查找。

```
<calrlookup
  calrlookup_id="@calrlookup_id_37"
  calrange_id="@calrange_id_37"
  value="5.00"
/>
```

其中

- calrlookup\_id 是生成的唯一标识。
- calrange\_id 是计算范围，此计算范围查找结果是其组成部分。
- value 是计算范围查找结果的值，它由计算范围的计算范围方法使用以确定货币结果。在此例中，税率是 5.00%。

关于更多信息，请参阅 WebSphere Commerce 联机帮助中的 CALLOOKUP 表。

- 对与商店中使用的计算标度关联的每个查找值重复步骤 a 和 b。在此例中，由于 CALRLOOKUP.SETCCURR 为 NULL，所以只有一个 CALRLOOKUP 值；并且由于所有金额的税率是一样的，所以只有一个 CALRANGE。
12. 将计算规则与计算标度关联。

- 使用以下示例作为指南，在您的 XML 文件中为 CRULESCALE 表将计算标度与计算规则关联。

```
<crulescale
  calrule_id="@calrule_id_10"
  calscale_id="@calscale_id_19"
/>
```

其中

- calrule\_id 是计算规则。
  - calscale\_id 是计算标度。
- 对每个计算标度和规则的关联重复步骤 a。在以上示例中，每个计算规则只有一个计算标度。



**注：**如果税率依购买金额而发生变化，则您将需要创建范围起始值非零的标度。那样，对于您未建立转换率（请参阅 CURCONVERT 表）的每种支持的货币（将 CALSCALE.SETCCURR 设置为相应的货币）创建计算标度，并且将它们全部与该特定税类别的计算规则关联。例如，对 \$4.00 以下的餐饮不征收安大略地方税。如果您的商店支持以美元销售餐饮，则需要建立从美元到加元的转换，或者以适当的范围起始值（可能是 \$6.00 USD）创建单独的税款计算标度，并将其与相同的税款计算规则关联。根据订单的货币，将只使用相应的计算标度。



关于 @ 和 & 用法的更多信息，请参阅第 417 页的附录 B，『创建数据』。

## 创建税款供货有用资源

为使您的税款有用资源在商店中正确工作，您必须将商店中的税款地区组关联到商店使用的供货中心，然后将计算规则同时关联到税款地区组和供货中心。

您必须先创建供货中心有用资源，才能将税款有用资源关联到供货中心。关于创建供货有用资源的更多信息，请参阅第 195 页的『在 WebSphere Commerce 中创建供货有用资源』。

在您创建供货有用资源之后，请通过向 TAXJCRULE 表添加信息来将税款有用资源与其相关联。请执行以下操作：

1. 复查《IBM WebSphere Commerce 计算框架指南》。WebSphere Commerce 计算框架计算与客户选择购买的产品或服务相关联的货币金额（例如，税款）。
2. 复查用于为样本商店创建税款供货有用资源的 XML 文件。样本商店的所有文件都位于相应的商店归档文件中。

商店归档文件位于以下目录中：

- WC\_installdir/samplestores

每个样本商店包括一个 taxfulfill.xml 文件，其中包括税款信息。要查看商店归档文件中的 taxfulfill.xml 文件，请使用 ZIP 程序将其解压缩。taxfulfill.xml 文件位于 data 目录中。

3. 请复查第 417 页的附录 B，『创建数据』中的信息。
4. 通过复制样本商店归档文件中的一个 taxfulfill.xml 文件或通过创建一个新文件来创建 taxfulfill.xml 文件。关于更多信息，请参阅相应于 taxfulfill.xml 的 DTD 文件。DTD 文件位于以下目录中：

- WC\_installdir/xml/sar

5. 使用以下示例作为指南，在您的 XML 文件中为 TAXJCRULE 表添加信息：

```
<taxjcrule
taxjcrule_id="@taxjcrule_id_1"
calrule_id="@calrule_id_10"
ffmcenter_id="@ffmcenter_id_1"
jurstgroup_id="@jurstgroup_id_2"
precedence="0"
/>
```

其中

- taxjcrule\_id 是生成的唯一标识。

- `calrule_id` 是使用的计算规则。
- `ffmcenter_id` 是供货中心。如果这为 `NULL`，则此关联将应用到所有的供货中心上。
- `jurstgroup_id` 是税收地区组。如果这为 `NULL`，则此关联将应用到所有的税收地区组上。
- `precedence` 当对于同一个供货中心，送货地址落在多个指定的税收地区组中时，则只有 `TAXJCRULE.PRECEDENCE` 值最高的计算规则适用。

6. 对商店中每个地区组、供货中心和规则的关联重复步骤 3。



关于 `@` 和 `&` 用法的更多信息，请参阅第 417 页的附录 B，『创建数据』。

## 创建商店 - 产品目录 - 税款有用资源

为了将税款与商店中的商品和服务关联，您必须对商店包含的每个合同将计算代码与商店中的产品目录条目关联。

您必须在可以创建“商店 - 产品目录 - 税款”有用资源之前，创建您的商店和产品目录有用资源。关于创建商店有用资源的更多信息，请参阅第 124 页的『在 XML 文件中创建商店数据有用资源』。关于创建产品目录有用资源的更多信息，请参阅第 161 页的『显示商店产品目录有用资源』。

要创建“商店 - 产品目录 - 税款”有用资源，请执行以下操作：

1. 复查《*IBM WebSphere Commerce 计算框架指南*》。WebSphere Commerce 计算框架计算与客户选择购买的产品或服务相关联的货币金额（例如，送货费用）。
2. 复查用于为样本商店创建“商店 - 产品目录 - 税款”有用资源的 XML 文件。样本商店的所有文件都位于相应的商店归档文件中。

商店归档文件位于以下目录中：

- `WC_installdir/samplestores`

3. 请复查第 417 页的附录 B，『创建数据』中的信息。
4. 通过复制样本商店归档文件中的一个 `store-catalog-tax.xml` 文件或通过创建一个新文件来创建 `store-catalog-tax.xml` 文件。关于更多信息，请参阅相应于 `store-catalog-tax.xml` 的 DTD 文件。DTD 文件位于以下目录中：

- `WC_installdir/xml/sar`

5. 通过向 `CATENCALCD` 表添加信息，创建“商店 - 产品目录 - 税款”关系。使用以下示例作为指南：

```
<catencalcd
  calcode_id="@calcode_id_3"
  catencalcd_id="@catencalcd_id_3"
  store_id="@storeent_id_1"
/>
```

其中

- `calcode_id` 是计算代码。
- `catencalcd_id` 是生成的唯一标识。
- `store_id` 是商店。



关于 @ 和 & 用法的更多信息，请参阅第 417 页的附录 B，『创建数据』。

---

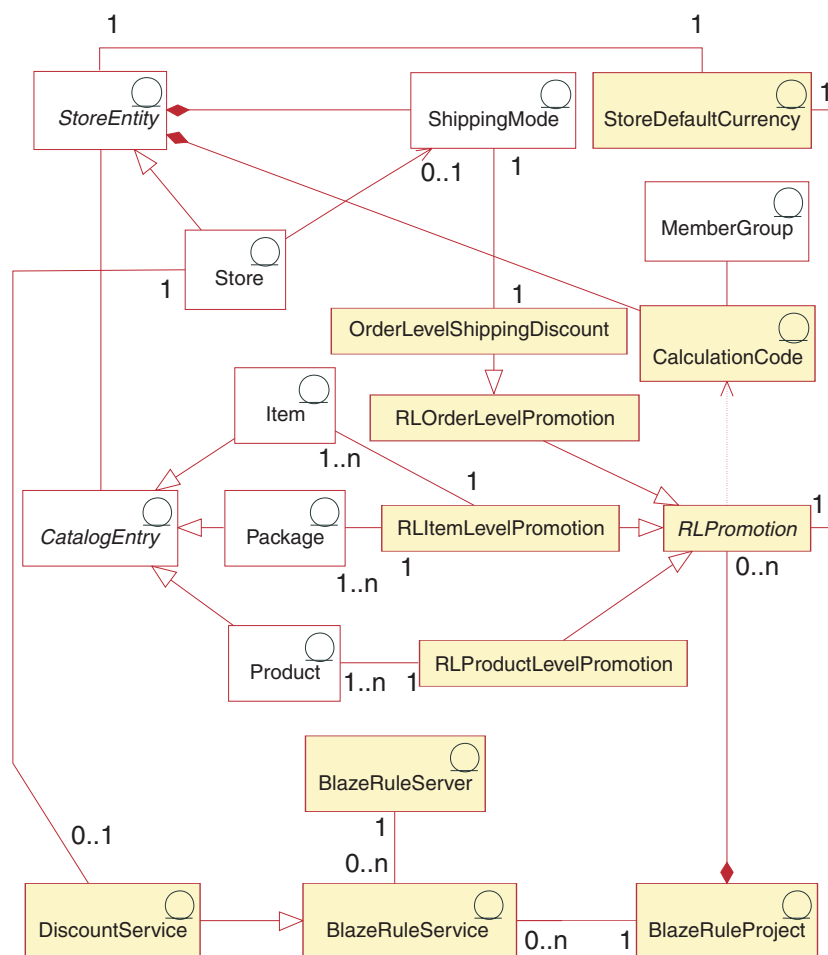


## 第 28 章 折扣有用资源

折扣让您能够在价格上吸引客户，从而驱动销售或促销产品。WebSphere Commerce 具有两种可用的折扣实现：基于规则的折扣以及基于模式的折扣。每种实现都可以提供百分比折扣（如打九折）或固定金额的折扣（如降价 ¥ 120）。折扣可以适用于特定的产品或适用于整个购买。例如，您可以向老年客户提供 20% 的折扣；如果库存中有许多红色棒球帽，则可以在一定时期内对这种帽子提供 25% 的折扣。基于规则的折扣超出这些折扣类型以提供送货折扣，并为符合条件的购物提供免费礼物。

### 了解 WebSphere Commerce 中基于规则的折扣

下图说明了 WebSphere Commerce Server 中基于规则的折扣的结构。



本图和商店数据部分中的其它所有图都是 WebSphere Commerce Server 信息模型的一部分。关于信息模型的更多信息，请参阅第 97 页的『商店数据信息模型』。关于本图中所使用的约定的更多信息，请参阅第 415 页的附录 A，『UML 图注』。

## 存储缺省货币

这是定义在 STOREENT 表中的商店的缺省货币。基于规则的折扣以此货币定义，但需要时可能使用商店所支持的任何货币进行评估。关于货币用法的更多信息，请参阅第 211 页的第 23 章，『货币有用资源』。

## 计算代码

折扣在计算框架中由折扣计算代码表示。折扣计算代码指示订购商品是如何通过相应的计算规则计算折扣的。

计算代码属于商店实体。可以在商店实体内定义多个计算代码。如果删除了商店实体，则在该商店实体内定义的计算代码也将被删除。

每个折扣计算代码可以具有一个开始日期和一个结束日期，它们定义了时间周期，在此时间周期内折扣是有效的。折扣计算代码也可以与一个或多个成员组关联，它定义可选择的成员组。

## RLPromotion

这是基于规则的折扣的父对象。当 RLPromotion 是对象类型名称时，您应该了解它对应于基于规则的折扣。每种基于规则的折扣都有名称、在不同情况中显示的各种描述、优先级、目标段（预定义客户概要文件）以及可同时管理日期和时间的执行调度表。

优先级属性应进一步阐明。存在优先级属性是为了解决存在可同时应用的多个折扣时出现的冲突。适用的折扣根据其各自的优先级值所定义的顺序以降序应用。也就是说，先应用具有最高优先级值的折扣。

下列所有子对象将基于规则的折扣类型进一步分类，并在需要折扣类型的地方给出特定值。这些对象中的每个对象还包含适当的逻辑，以便操纵定义折扣的域 XML 文件。

### RLProduct 级促销

这些对象代表产品级基于规则的折扣。该类是从 RLPromotion 类派生的。该类需要附加的属性，SKU，该属性标识了目标产品。

### RLItem 级促销

这些对象代表商品级基于规则的折扣。该类是从 RLPromotion 类派生的。该类需要附加的属性，catEntryID，该属性标识目标产品。由于预置的成套商品可通过其自身的 catEntryID 和价格分别订购，因此这些商品级促销还可用来对这些成套商品应用基于规则的折扣。基于规则的折扣不可应用于捆绑销售商品和可改装的成套商品。

### RLOrder 级促销

这些对象代表订单级基于规则的折扣。这是由 RLPromotion 派生的。该类需要附加的属性，inCombineWithProductLevelDiscount，该属性确定是否可以在产品级促销的同时应用订单级的折扣。

**订单级的送货折扣：** 该类是从 RLOrder 级促销类派生的。该类需要附加的属性，该属性定义要使用的送货方式以及折扣率。

## Blaze 规则项目

Blaze 规则项目包含当前为商店定义的所有折扣，该规则项目由域 XML 文件生成。规则项目驻留在文件系统中，用于植入折扣服务。

## Blaze 规则服务

这是 WebSphere Commerce 用来与 Blaze 规则服务器进行通信的接口。

## 折扣服务

从 Blaze 规则服务扩展而来，并包含其中将发生评估和计算的规则项目实例。该服务将折扣上下文（一种包含有关适用折扣的信息的对象）传递给 WebSphere Commerce 计算框架。

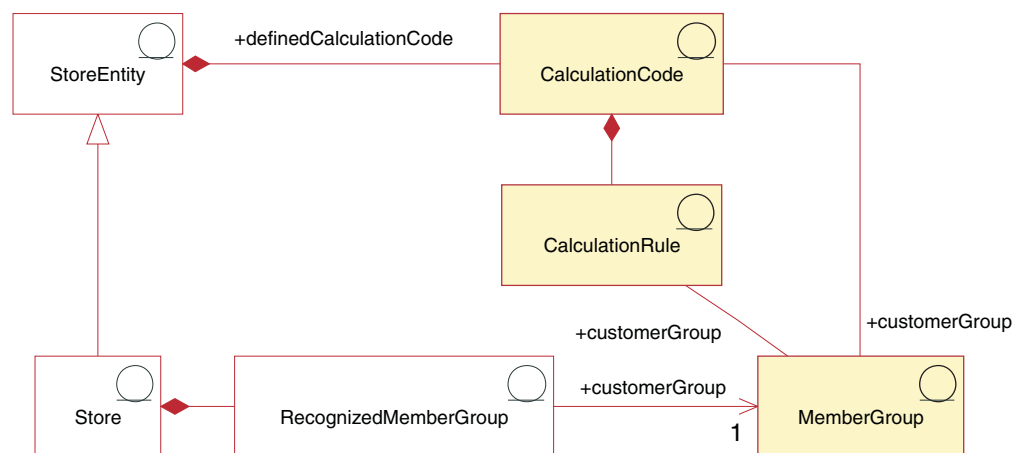
## Blaze Rule Server

每当请求从订单处理命令到达时将评估规则项目的 Blaze 软件。

---

## 了解 WebSphere Commerce 中基于模式的折扣

下图说明了 WebSphere Commerce Server 中基于模式的折扣的结构。



## 计算代码

折扣是使用折扣计算代码表示和计算的。折扣计算代码指示了如何为订购商品计算折扣。

计算代码属于商店实体。可以在商店实体内定义多个计算代码。如果删除了商店实体，则在该商店实体内定义的计算代码也将被删除。

每个折扣计算代码可以具有一个开始日期和一个结束日期，它们定义了时间周期，在此时间周期内折扣是有效的。折扣计算代码也可以与一个或多个成员组关联，它定义可选择的成员组。

折扣计算代码可以附加在一个或多个产品目录条目以及产品目录组上。将计算代码附加到产品目录组上的效果与将其直接附加到该产品目录组中所有产品目录条目上的效果相同。但是，如果产品目录组 A 包含产品目录组 B，则附加在产品目录组 A 上的计算代码不会附加在产品目录组 B 中的产品和商品上。

产品目录条目或产品目录组可以具有多个与它们关联的折扣。当对一个订单应用多个折扣计算代码时，将按照其计算代码顺序属性的升序执行折扣计算。

**注：** 定义折扣顺序以在折扣上实现折扣。

按照以下一种方式将订购商品分组以供计算：

- 每个贸易协议
- 每个产品
- 每个报价
- 每个送货地址

关于更多信息，请参阅 [WebSphere Commerce 联机帮助](#)。



关于使用计算代码的更多信息，请参阅《*IBM WebSphere Commerce 计算框架指南*》。

---

### 计算规则

每个计算代码具有一组计算规则，这些规则定义了执行计算的条件。每个折扣计算规则与一个或多个成员组关联（折扣对该成员组有效）。成员组一次可以选择多个折扣。

**注：** 如果可选择的成员组是在计算代码级别定义的，它不必在计算规则级别再次定义。



关于使用计算规则的更多信息，请参阅《*IBM WebSphere Commerce 计算框架指南*》。

---

---

## 在 WebSphere Commerce 中创建折扣有用资源

在用 WebSphere Commerce 创建的商店中创建折扣的主要方法是使用 WebSphere 贸易加速器中的“折扣”向导。关于使用 WebSphere 贸易加速器创建折扣的更多信息，请参阅 [WebSphere Commerce 联机帮助](#)。

折扣还可以通过使用一个 XML 文件创建，然后由装入程序软件包装入。然而，虽然以这种方法创建的折扣以及在从先前版本迁移过程中导入的折扣将具有正常功能，但是在 WebSphere 贸易加速器中可能会无法正常显示。



---

## 第 29 章 库存有用资源

库存包含可以在供货中心中实际计算的任何东西。对于可以实现的库存类型有特定的定义，例如商品、产品、库存标识、捆绑销售商品以及打包销售商品；但是我们都当作库存。在“产品”向导和“产品”笔记本中配置产品以供货。这包含若干选项，有跟踪库存、允许延迟交货、强制延迟交货、分别发货以及指定产品不可以退回。WebSphere 贸易加速器区分可接收的两种主要类型的库存：

- 预期库存，具有关联的预期库存记录
- 特别库存，或者未记录为预期库存的库存

预期库存是从供应商处接收的，并且通常用购买订单来支付。WebSphere 贸易加速器跟踪带有预期库存记录的预期库存，并允许您记录外部标识，通常是来自外部系统的购买订单号。在这种方式下，您可以轻易的跟踪已订购的库存以及已到货和未到货的库存。预期库存详细信息是有关预期库存记录中产品的具体信息，例如供货中心（它预期此产品）、预期接收日期、期望的数量以及注释。

一旦已根据预期库存记录接收其库存，则不能将其删除；一旦已接收该预期库存中任何库存，则不能更改或删除预期库存记录的详细信息。

当对供货中心可提供的库存下订单时，订单系统将库存分配给这些订单。将库存分配给订单后将使库存不再对订单系统可用。如果取消订单，库存又变为可用。如果为不可用的库存下订单，则可以创建延迟交货订单。如果有可以用于实现延迟交货订单的预期库存，则将预期库存分配给延迟交货订单，并可以向客户提供预期的送货日期。

特别库存接收在库存到达供货中心而没有相应的预期库存记录时创建。这可能是因为意外的库存到达，或者可能是因为商家或销售商选择了不使用预期库存记录来记录库存接收。

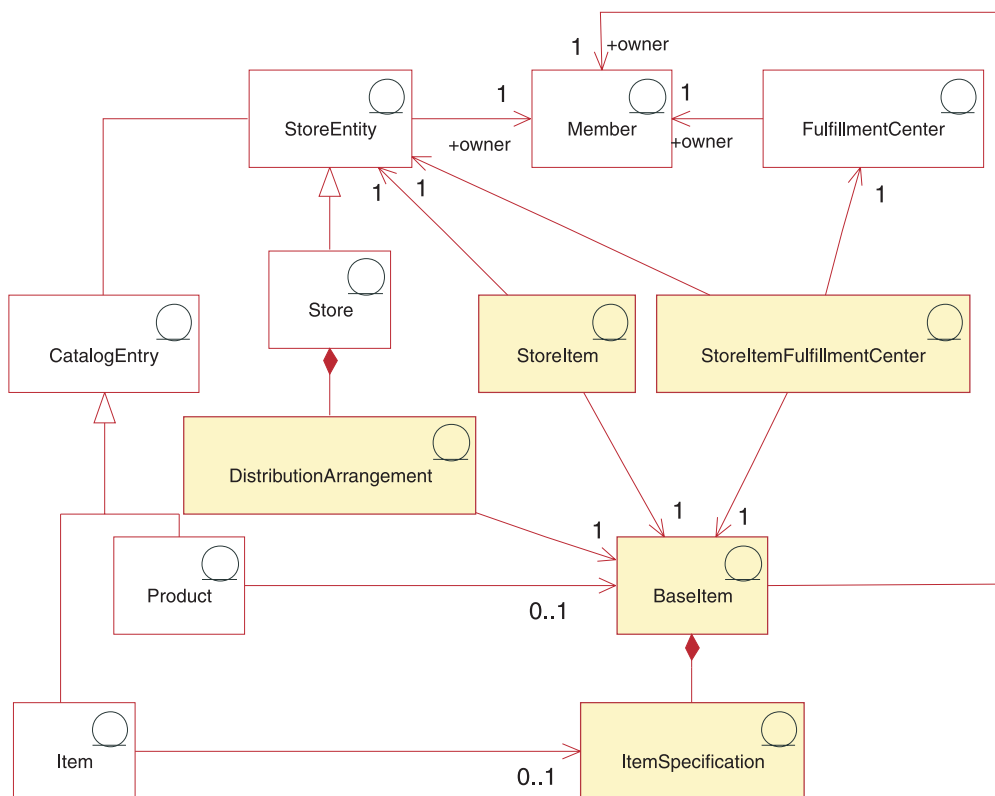
**注：**不论库存接收是预期的还是特别的，产品都必须在 WebSphere Commerce 系统中存在以便接收。

---

### 了解 WebSphere Commerce 中的库存有用资源

要了解库存有用资源，有必要了解库存和商店之间的关系。这可以使用信息模型来解释。以下部分描述了库存与商店和其它有用资源之间的关系和关联。下面这些图描述了可承诺（ATP）库存和非 ATP 库存的关系和关联。商店可以使用 ATP 或非 ATP 库存方法。如果 STORE 数据库表的 ALLOCATIONGOODFOR 列包含大于 0 的值，则可认为某个商店为实现 ATP 而启用。每张图与它的关联描述如下。关于 ATP 的更多信息，请参阅联机帮助。

## ATP 库存



本图和商店数据部分中的其它所有图都是 WebSphere Commerce Server 信息模型的一部分。关于信息模型的更多信息，请参阅第 97 页的『商店数据信息模型』。关于本图中所使用的约定的更多信息，请参阅第 415 页的附录 A，『UML 图注』。

### 基本商品

基本商品是库存图的中心，它代表具有公共名称和描述的商品的一般系列。基本商品专用于供货，并且不是特定于任何商店的。为了供货的目的，每个产品目录条目（它代表产品目录中的一个产品）都具有一个相应的基本商品。基本商品在 BASEITEM 表中定义。

### 商品规格

商品规格是一个基本商品，它带有为其所有属性定义的值。为了供货的目的，每个产品目录条目（它代表产品目录中的一个商品）都具有一个相应的商品规格。

### 产品目录条目

产品和商品都是目录条目。产品目录条目与商店实体关联，这意味着产品目录条目（例如产品和商品）可以在商店中找到。

### 分发安排

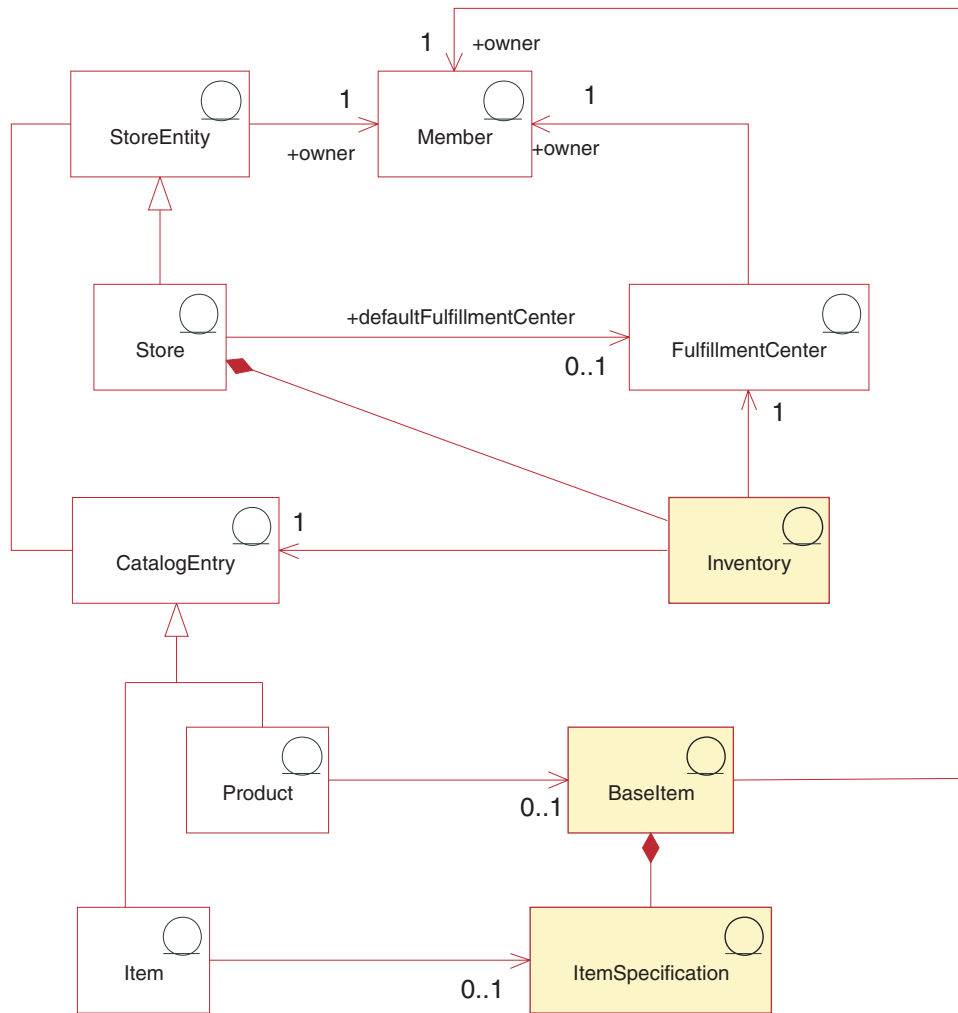
分发安排与一个基本商品关联，以使商店能够销售它自己的库存。分发安排存储在 DISTARRANG 表中。

## 商店商品

商店商品代表一些属性，这些属性影响了特定商店或商店组为特定基本商品的指定商品分配库存的方式，包括是否允许延迟交货以及是否允许跟踪库存。STORITMFFC 表定义了从订购商品已针对供货发货开始算起直至订购商品发送到客户为止，其间估计经过的时间（以秒为单位）。仅当商店希望定义一个值覆盖商店商品的 FFMCENTER 缺省送货偏移量时，才填充此表。

商店商品有用资源可由其它商店使用，如第 129 页的第 14 章，『商店间的关系』中所述。

## 非 ATP 库存



本图和商店数据部分中的其它所有图都是 WebSphere Commerce Server 信息模型的一部分。关于信息模型的更多信息，请参阅第 97 页的『商店数据信息模型』。关于本图中所使用的约定的更多信息，请参阅第 415 页的附录 A，『UML 图注』。

基本商品也是非 ATP 库存图的中心。基本商品与产品、商品和产品目录条目之间的关系与常规库存图中的关系相同。基本商品仍然由一个成员所有，并且一旦该成员定义

了基本商品，则该基本商品就可以在商店中销售。然而，在这种情况下，将没有分发安排、商店商品关联或者商店商品供货中心。

## 供货中心

库存与一个供货中心和一个商店关联。商店可以指定一个缺省的供货中心。与基本商品相同，供货中心归成员所有。关于供货有用资源的更多信息，请参阅第 193 页的第 19 章，『供货有用资源』。



关于 WebSphere Commerce Server 中库存有用资源的结构更多详细信息，请参阅 WebSphere Commerce 联机帮助中的库存对象和数据模型。

## 在 WebSphere Commerce 中创建库存有用资源

因为库存是可操作的数据，所以当客户从您的商店中购买产品或退回产品时，库存每天都会更改。因此，当您销售产品时以及供货中心从供应商处接收新的库存时，您的库存级别会上下浮动。WebSphere 贸易加速器允许您完成以下与库存有关的任务：

- 记录预期库存
- 从供应商处接收预期库存和特别库存
- 调整库存
- 维护退货记录
- 维护退货原因
- 从客户处接收退回的库存
- 管理退回库存的处理

关于使用 WebSphere 贸易加速器管理库存的更多信息，请参阅 WebSphere Commerce 联机帮助。

## 管理库存调整代码

库存调整代码由 WebSphere Commerce 提供，使用户可以指定要对库存作特定调整的原因。下表列出了最初与 WebSphere Commerce 一起提供的库存调整代码。您也可以通过输入针对 INVADJCODE 和 INVADJDESC 数据库表的 SELECT \* 数据库查询来查看这些代码。作为商店开发者，您可以按需要添加、更改或删除这些代码，以满足库存环境的需要。在下表中描述的代码应看作是可以使用的代码类型的示例。可以通过添加您自己的调整代码或修改预装入的代码来获得定制程度更高的环境。

表 11. 预装入的库存调整代码

调整代码	描述	解释
RTND	已返回	使用该代码可指定某商品*已退回库存。例如，客户可能会因为商品的颜色或大小问题而退回订单中的商品。该代码已预装入供所有样本商店使用。
EXPD	已失效	使用该代码可指定库存中的某商品已失效。例如，处方药或储存期限很短的产品（例如奶制品）会失效。该代码已预装入供所有样本商店使用。

表 11. 预装入的库存调整代码 (续)

调整代码	描述	解释
DMGD	已损坏	使用该代码可指定某商品已损坏（例如，可能有凹痕、抓痕、已损坏或有缺陷）。该代码已预装入供所有样本商店使用。
LOST	丢失	使用该代码可指定某商品已在供货中心或存储位置丢失或被偷。该代码已预装入供所有样本商店使用。
MSCT	计数错误	使用该代码可指定先前对商品的计数是错误的。该代码已预装入供所有样本商店使用。
PCNT	物理计数	使用该代码可指定商品实际的物理计数。该代码对消费者直销和 B2B 直销样本商店是预装入的，但对主管商店未预装入。
SPLG	变质	使用该代码可指定某商品已变质。该代码对消费者直销和 B2B 直销样本商店是预装入的，但对主管商店未预装入。
DISC	废弃	使用该代码可指示某商品已经或将要从库存中废弃。该代码对消费者直销和 B2B 直销样本商店是预装入的，但对主管商店未预装入。
*这里所使用的词“商品”指的是普通含义的商品。库存产品可以是商品、产品、最小库存单位（SKU）、捆绑销售商品和打包销售商品。		

在本表中，显示在“调整代码”列中的值代表在 INVADJCODE 数据库表的 ADJUSTCODE 列中发现的值。显示在“描述”列中的值代表在 INVADJDESC 数据库表中的 DESCRIPTION 列中发现的值。

要添加、更改和删除库存调整代码，请使用装入（massloader）命令。第 336 页的『装入命令』中提供了关于装入命令的详细信息。

关于如何在样本商店中组织数据的更多信息，请参阅第 367 页的第 38 章，『装入 WebSphere Commerce 数据库有用资源组』。

在使用装入命令前，请确保 wcs.dtd 文件已包含在路径（WC\_installdir\schemaxml）中。massloader.cmd 文件也必须在路径（WC\_installdir\bin）中。

## 添加库存调整代码

要添加您的商店的库存调整代码，请执行以下操作：

1. 创建一个与以下示例相似的 XML 文件（您自己选择名字），并把它放在装入程序可以找得到的位置。在 XML 文件中，为 invadjcode 和 invadjdesc 元素指定值，从而把新库存调整代码添加到两个数据库表 INVADJ 和 INVADJDESC 中。对任何给定代码而言，在两个表中 invadjcode\_id 应该有相同的值。

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE import SYSTEM "wcs.dtd">
<import>
<invadjcode invadjcode_id="404" adjustcode="BRKN" storeent_id="-1"
  markfordelete="0"
<invadjdesc invadjcode_id="404" description="BROKEN" language_id="-1" />
</import>
```

其中：

- `invadjcode_id` 是您指定的调整代码标识。标识是内部使用的，它不会在 WebSphere Commerce 用户界面上显示给用户。
- `adjustcode` 是唯一标识代码的 4 个字符的代码，它适合在用户界面中显示。
- `storeent_id` 是商店实体或商店组的标识。要对所有商店添加此库存调整代码，请输入值“-1”。否则，提供一个特定的商店实体或商店组标识。
- `description` 是库存调整代码的文本描述，它适合在用户界面中显示。
- `language_id` 是对在商店中购物的客户显示信息时使用的缺省语言。关于语言支持的更多信息，请参阅第 287 页的第 34 章，『全球化』。

如果需要添加多个库存调整代码，您可以在 XML 文件中指定多个代码。

2. 对 XML 文件运行装入命令以将数据装入到两个目标数据库。

Windows

```
massload.cmd -dbname dbname -dbuser dbuser -dbpwd dbpwd -infile
xml_file_name -method sqlimport
```

AIX

Solaris

Linux

首先成为 `wasuser` ( WebSphere Application Server 用户标识) :

```
su - wasuser
```

然后，发出此命令：

```
./massload.sh -dbname dbname -dbuser dbuser -dbpwd dbpwd -infile
xml_file_name -method sqlimport
```

400

启动一个 QShell 会话 (STRQSH)。然后，从 `WC_installdir/bin` 目录运行以下命令：

```
massload.sh -dbname dbname
-dbuser dbuser -dbpwd dbpwd -infile
xml_file_name -method sqlimport
```

3. 请确认库存调整代码是否已添加，方式是运行数据库查询来查看新代码值是否已添加到这两个表中。

## 更改库存调整代码

要更改库存调整代码的描述，请执行以下操作：

1. 执行与添加新代码相似的步骤，但是把 XML 文件中的 `adjustcode` 和描述元素更改为特定的 `invadjcode_id` 的值。例如：

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE import SYSTEM "wcs.dtd">
<import>
<invadjcode invadjcode_id="404" adjustcode="DEFE" storeent_id="-1"
  markfordelete="0"
<invadjdesc invadjcode_id="404" description="DEFECTIVE" language_id="-1" />
</import>
```

在该示例中，先前的调整代码和描述 (BRKM、BROKEN) 已更改为显示的新值。如果需要，您还可以更改 `invadjcode_id` 的值。

要查找 `invadjcode_id`，您可以发出 `massextract.cmd`（抽取命令）。关于抽取命令的更多信息可以在 `WebSphere Commerce 开发联机帮助` 中获得。`massextract.cmd` 从数据库中抽取数据，并把它放到 XML 文件中。然后，您可以浏览文件以查找 `invadjcode_id` 值。或者，您也可以对 `INVADJCODE` 和 `INVADJDESCA` 数据库表运行 `SELECT *` 查询。

2. 如同对添加新调整代码一样，运行 `massload.cmd`:

Windows

```
massload.cmd -dbname dbname -dbuser dbuser -dbpwd dbpwd -infile  
xml_file_name -method sqlimport
```

AIX

Solaris

Linux

以 `wasuser` 身份，运行以下命令:

```
./massload.sh -dbname dbname -dbuser dbuser -dbpwd dbpwd -infile  
xml_file_name -method sqlimport
```

400

启动一个 QShell 会话 (STRQSH)。然后，从 `WC_installdir/bin` 目录运行以下命令:

```
massload.sh -dbname dbname  
-dbuser dbuser -dbpwd dbpwd -infile  
xml_file_name -method sqlimport
```

3. 请确认库存调整代码已在数据库表中更改了。

## 删除库存调整代码

要从 `WebSphere Commerce` 数据库表中删除库存调整代码，请执行以下操作:

1. 创建一个包含您想要删除的代码的 XML 文件。请参阅第 263 页的『添加库存调整代码』中作为示例的样本 XML 文件。
2. 运行 `massload.cmd`，但是请确保如下所示指定了删除方法

Windows

```
massload.cmd -dbname dbname -dbuser dbuser -dbpwd dbpwd -infile  
xml_file_name -method delete
```

AIX

Solaris

Linux

以 `wasuser` 身份，运行以下命令:

```
./massload.sh -dbname dbname -dbuser dbuser -dbpwd dbpwd -infile  
xml_file_name -method delete
```

400

启动一个 QShell 会话 (STRQSH)。然后，从 `WC_installdir/bin` 目录运行以下命令:

```
massload.sh -dbname dbname  
-dbuser dbuser -dbpwd dbpwd -infile  
xml_file_name -method delete
```

3. 请确认库存调整代码已从数据库表中删除。



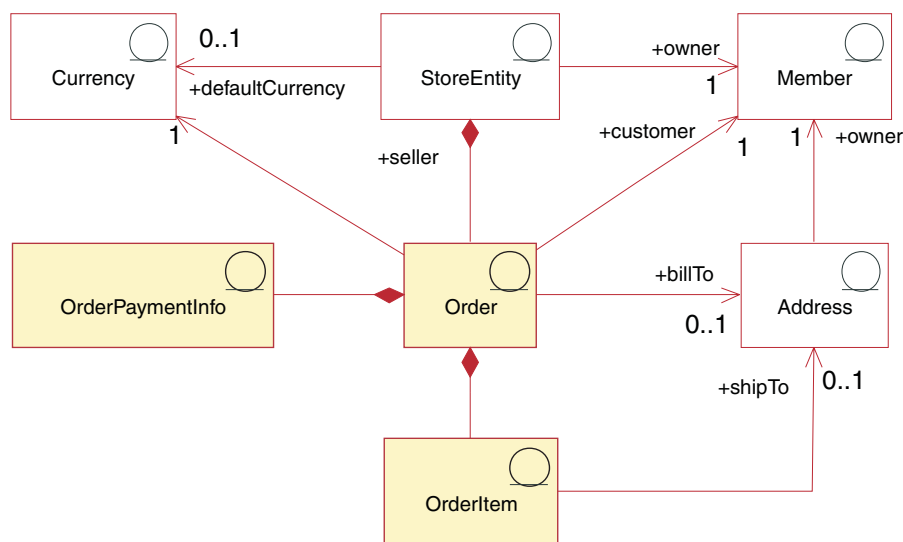


## 第 30 章 订单有用资源

WebSphere Commerce 系统中的订单有用资源提供了购物车、订单管理以及订单处理功能。订单处理功能包含快速订购或购买、固定订购、多个未决订单、重新订购以及分割订单和延迟交货订单。相关的服务（例如定价、税务、支付、库存和供货）也是订单有用资源的组成部分。

### 了解 WebSphere Commerce 中的订单有用资源

下图说明了 WebSphere Commerce Server 中的订单有用资源。每个有用资源的描述在此图之后。



本图和商店数据部分中的其它所有图都是 WebSphere Commerce Server 信息模型的一部分。关于信息模型的更多信息，请参阅第 97 页的『商店数据信息模型』。

### 订单和订购商品

在 WebSphere Commerce 系统中，对于客户或购物者而言，订单是选定产品的列表（例如订单可以包含两本书和一张 CD），而该列表上的每个产品是一个订购商品（例如，每本书和每张 CD 都是同一个订单的订购商品）。当客户在商店中下了一个订单时，此客户必须提供开票地址，商店会把发票发送到此地址。每个订单有一个单一货币标识与之关联。从商店角度而言，订单是订购商品的列表。它是商店数据的组成部分。

#### 货币

商店可以用一种货币来显示价格，或者也可以使用多种货币。每个商店还必须定义一个缺省货币。您还可以允许客户选择购物货币。如果购物货币与商店的缺省货币相同，则它在 STOREENT 表中已受到支持。如果购物货币不是商店的缺省货币，则您必须将此货币添加到 CURLIST 表。客户使用购物货币在您的商店中下订单。

## 支付信息

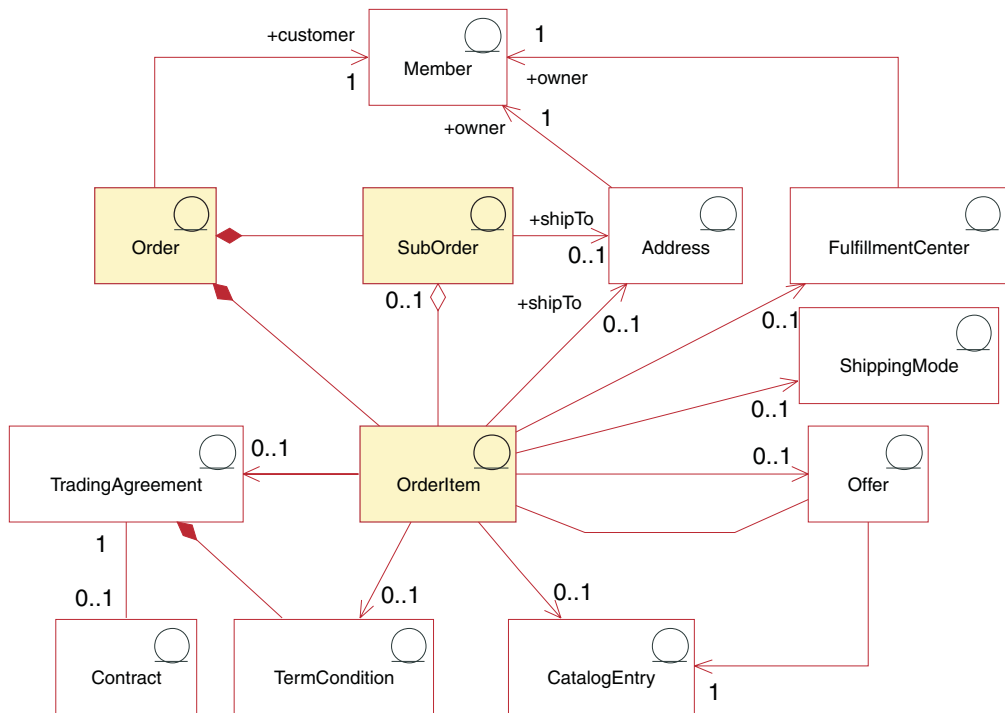
一旦客户选择了首选购物货币，则所有的支付将使用该货币来处理。根据商店的支付支持和策略，客户可以使用联机支付（这种情况下，客户通过因特网在商店的站点上提供支付信息）或脱机支付（这种情况下，客户不通过因特网通道提供支付信息，例如可通过电话或传真）来支付购买的商品。不论是采用联机支付方式还是脱机支付方式，客户必须在下订单时提供支付信息，包含以下任何信息：

- 支付方式：客户支付订单的方式。根据在 **WebSphere Commerce Payments** 中为商店配置支付的卡匣，您可以设置商店，使之接受脱机支付，或对不需要客户使用在线电子钱包的在线支付使用其它支付协议，或使用定制支付方式。
- 对于信用卡支付，有关信用卡的信息：客户用于支付订单的信用卡的品牌、号码和失效日期。如果商店支持联机支付，则通常需要信用卡信息。
- 购买订单号码：购买订单的号码，客户在商店中订购时可能已经提供了此号码。正如商店和客户之间的合同中的条款所规定的，购买订单号码将客户识别成一个已授权可从商店订购商品的人。

## 订购商品

订购商品是构成订单的独立的产品或商品。一个订单必须具有至少一个订购商品。每个订购商品代表客户选择购买的一件物品。此外，每个订购商品都引用了贸易协议（通常是合同）、送货方式、供货中心以及报价。折扣、送货费用和总税款与每个订购商品存储在一起。

下图说明了 **WebSphere Commerce** 订购商品有用资源。每个有用资源的描述在此图之后。





---

本图和商店数据部分中的其它所有图都是 WebSphere Commerce Server 信息模型的一部分。关于信息模型的更多信息，请参阅第 97 页的『商店数据信息模型』。关于本图中所使用的约定的更多信息，请参阅第 415 页的附录 A，『UML 图注』。

---

## 子订单


订购商品被分组形成子订单。子订单是订单中被交付到指定地址的那一部分。例如，客户可以为购物车中的不同产品指定不同的送货地址。每个送货地址以及与其相关的产品即构成一个子订单。子订单中的订购商品具有相同的送货地址，而且可以用于显示其订购商品金额的小计。

OrderItem 对象的数量属性是一个无单位的数值，它可以与 CatalogEntryShippingInformation 对象（此对象与 CatalogEntry 对象关联）的额定数量属性相乘得到由 OrderItem 代表的实际数量。CatalogEntryShippingInformation 对象指定了表示数量的计量单位。

尽管订单通常与单个商店关联，但是订单概要文件是一种特殊类型的订单，它可以与一个商店或一个商店组关联。在对象模型中，订单概要文件由状态为“Q”的订单来代表。订单概要文件存放有关客户的缺省信息，例如支付信息、送货地址、送货方式和开票地址。

## 其它订购商品有用资源

一个订购商品可以与以下对象中的一个关联，也可以不与以下对象中的任何一个关联。

- 客户的送货地址（此客户下的订单中包含订购商品）。客户必须在订购过程中指定一个送货地址，以便商店的供货中心可以使用此地址来相应地交付订购商品。
- 供货中心，它发送和接收客户订单所需的订购商品，并且存储订购商品的库存。
- 订购商品的送货方式，它是货物递送者（它是提供从供货中心到客户的送货服务的公司）和该货物递送者所提供的送货服务的组合。例如，“ABC 送货公司，隔夜服务”以及“ABC 送货公司，快递”都是送货方式。
- 与此订购商品关联的报价。通过将不同的报价包含在不同的价格列表（或“贸易状态容器”）中，商店可以对不同的客户显示同一个产品或库存标识的不同价格。例如，旅行社可以在四个不同的价格列表中提供机票：成人定价、老人定价、儿童定价和学生定价。
- 订购商品的产品目录条目；即，每个订购商品从产品目录中订购一个商品。
- 贸易协议，它定义了条款和条件，在该条款和条件下才可以订购此商品。这通常是一个合同，但也可能是一个  报价请求（RFQ），它代表一次协商，直至提交订单以供处理为止。



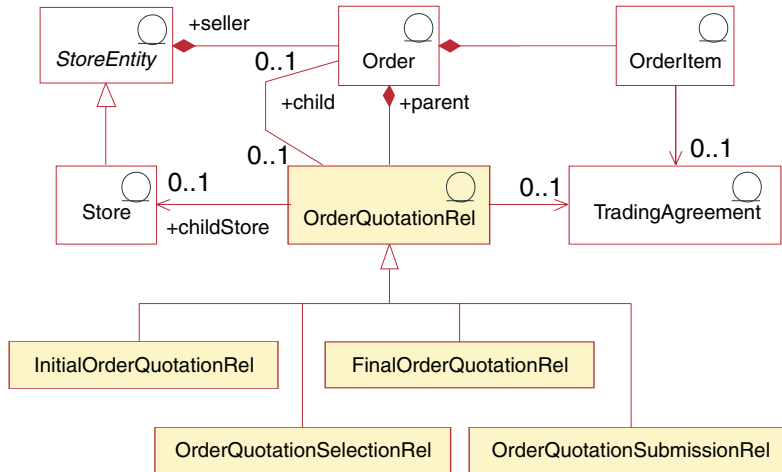
---

关于 WebSphere Commerce Server 中订单有用资源的结构更多详细信息，请参阅 WebSphere Commerce 联机帮助中的订单对象和数据模型。

---

## 订单报价关系

下图描述订单报价关系。



关于 WebSphere Commerce Server 中订单有用资源的结构更多详细信息，请参阅 WebSphere Commerce 联机帮助中的订单对象和数据模型。

订单报价关系（OrderQuotationRel）是父级购物车订单与子级报价相关订单之间的关系。在价值链（需求）业务模型中（销售是通过渠道进行的），渠道区域购物者可以使用购物车从多家商店获取产品或服务的报价，在生成的报价中选择数量，然后向提供报价的商店提交订单。订单报价关系指示请求了报价的商店和合同（贸易协议）。

父级购物车对于每个报价商店和合同对，可拥有以下类型的子级报价相关订单：初始报价、当前报价选择、最终报价以及一份或多份提交的订单。

InitialOrderQuotationRel 的子级订单代表针对父级购物车订单中商品所接收到的初始报价。初始报价可以包括在父级订单中没有明确提到的替代产品和相关产品。

OrderQuotationSelectionRel 的子级订单代表当前从初始报价或最终报价中选择的商品及其数量。OrderQuotationSelectionRel 子级订单可以提交到其商店进行处理。

FinalOrderQuotationRel 的子级订单代表针对选择订单中商品所接收到的最终报价。发送对最终报价的请求之后，最终报价仅提供选择订单中那些商品的价格和数量。

OrderQuotationSubmissionRel 的子级订单代表已提交的订单。选择订单提交后，其 OrderQuotationSelectionRel 对象就更改为 OrderQuotationSubmissionRel 对象。

## 在 WebSphere Commerce 中创建订单有用资源

客户可以从商店下订单，或（使用 WebSphere 贸易加速器）请求商店的客户服务代表协助完成该任务。要代表消费者直销客户创建订单，请参阅 WebSphere Commerce 联机帮助中的主题“为注册客户创建订单”和“为非注册客户创建订单”。要代表 B2B 直销客户创建订单，请参阅帮助主题“为商业用户创建订单”。

## 第 31 章 供应商有用资源

供应商代表在供货中心接收的、或预期要在供货中心接收的商品的源。根据商店模型，供应商由买方、产品经理、卖方（商家）或其他一些授权角色定义。通过 WebSphere 贸易加速器，您可以查看所有供应商的列表、创建新的供应商、更改现有供应商和删除供应商。

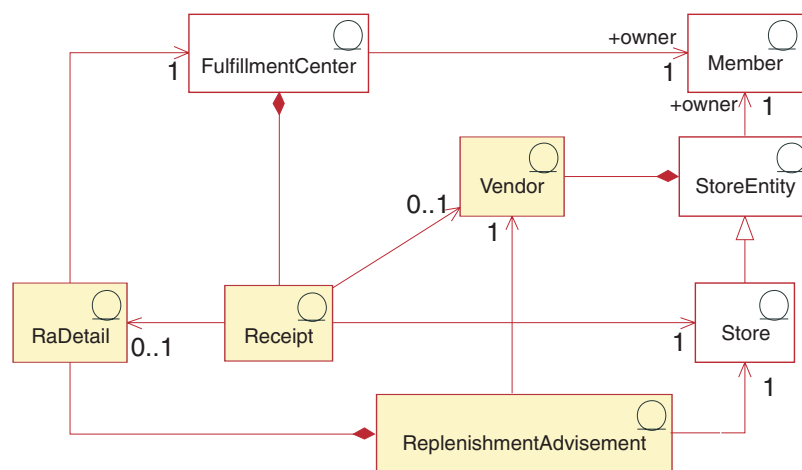
供应商记录包括关于供应商的信息，例如姓名、地址和联系信息。必须首先创建供应商，然后商店才能创建预期库存记录。

预期库存记录按供应商、外部标识（通常为购买订单号）和“预期库存”页面上的订单日期显示。

关于管理供应商记录的更多信息，请参阅 WebSphere Commerce 产品联机帮助中的供应商信息主题。

### 了解 WebSphere Commerce 中的供应商有用资源

下面这一节描述供应商与商店和其它有用资源之间的关系。



本图和商店数据部分中的其它所有图都是 WebSphere Commerce Server 信息模型的一部分。关于信息模型的更多信息，请参阅第 97 页的『商店数据信息模型』。关于本图中所使用的约定的更多信息，请参阅第 415 页的附录 A，『UML 图注』。

供应商代表商店接收的商品的源，可被认为是商店的供应商。

补给意见与预期库存记录同义。

收据是指库存收据。通常，收据由预期库存记录生成。

RaDetail 代表关于预期库存记录上的商品的详细信息，例如库存的预期日期、供货中心标识和订购量。

关于库存收据的更多信息，请参阅第 259 页的第 29 章，『库存有用资源』。

---

## 创建供应商有用资源

可以使用 WebSphere 贸易加速器创建供应商有用资源。有关指示信息请参阅 WebSphere Commerce 产品联机帮助中的主题“创建供应商”。

---

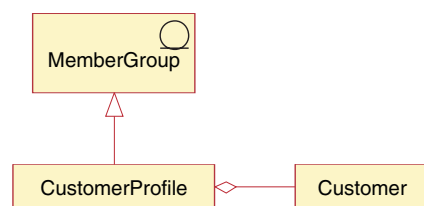
## 第 32 章 客户概要文件

客户概要文件通过分组市场营销消息的目标来帮助您组织市场营销工作。客户概要文件通常是由任意一个商家使用 WebSphere 贸易加速器创建的。

---

### 了解 WebSphere Commerce 中的客户概要文件

下图说明了 WebSphere Commerce Server 中的客户概要文件有用资源:



本图和商店数据部分中的其它所有图都是 WebSphere Commerce Server 信息模型的一部分。关于信息模型的更多信息，请参阅第 97 页的『商店数据信息模型』。关于本图中所使用的约定的更多信息，请参阅第 415 页的附录 A，『UML 图注』。

客户概要文件包含注册信息、人口统计信息、地址信息、客户文化、购买历史和其它杂项属性，这些属性定义了客户或帐户的动态组。客户概要文件可以充当广告、促销、推荐销售、折扣以及电子邮件活动的目标。您必须在开展竞销之前创建客户概要文件。概要文件看作是动态的，因为这是根据客户的个人数据和购买历史来生成的，而个人数据和购买历史都是可能更改的。例如，您可能根据客户的注册状态来创建概要文件。如果您创建了客户需要注册之后才具备资格的概要文件，则未注册的客户将被排除。如果同一个客户在稍后的日期注册，则他们将成为目标概要文件的成员，并且继续作为成员直到删除该概要文件为止。

客户概要文件还支持静态条件。您可以明确地包含或排除特定的客户或帐户，这些条件覆盖所有已定义的动态条件。例如，通过这种方法，您可以在概要文件中包含在其它情况下不匹配的客户，或从概要文件中排除在其它情况下匹配的帐户。换句话说，如果在同一个客户概要文件中同时包含了静态和动态的条件，则将先评估动态条件，然后评估静态条件。

而且，您可以根据 WebSphere Commerce Analyzer 的数据挖掘结果明确地包含或排除客户或帐户。WebSphere Commerce 集成了 WebSphere Commerce Analyzer 的高级分析方法，从而使商家能够轻松地根据 WebSphere Commerce Analyzer 所生成的段创建显式的客户概要文件。





---

## 第 7 部分 向商店添加访问控制



---

## 第 33 章 商店中的访问控制

WebSphere Commerce 允许您通过访问控制来确定特定用户（无论是客户还是管理员）可以执行的任务。本章着重描述了如何在商店中添加访问控制，以便限制您的客户可以看到的页面以及他们在商店中可以执行的任务。

关于 WebSphere Commerce 中的访问控制的更多信息，请参阅第 31 页的第 4 章，『WebSphere Commerce 中的访问控制』。关于访问控制模型的更多详细信息，请参阅《WebSphere Commerce 安全性指南》。

---

### 了解 WebSphere Commerce 中的访问控制

#### 商店中的访问控制

WebSphere Commerce 中创建的所有商店都受到由拥有商店的组织所预订的或从该组织祖先所继承的缺省访问控制策略的控制。

缺省情况下，根组织预订管理和运营策略组。如果您以 WebSphere Commerce 中提供的样本商店之一为基础创建商店，则将创建由拥有该商店的组织所拥有并预订的特定于商店的访问控制策略和策略组。除了特定于商店的策略组，该拥有商店的组织还可以根据商店的性质来预订管理和运营、公共购物和 B2C 或 B2B 策略组。有关预订策略组的更多信息，请参阅《WebSphere Commerce 安全性指南》。

当您创建自己的商店时，无论它是否基于样本，您都可能希望创建新的访问控制策略或修改现有策略，这些策略将只适用于该组织所有的商店。例如，如果您创建新的视图用来显示商店页面，则必须对这些视图指定访问控制策略。

访问控制数据是在高级访问控制策略文件中定义的。这些文件定义了可由任何策略使用的可能的操作、操作组、资源、资源组和关系。它们也定义了特定于某特定组织的策略和策略组预订。与 WebSphere Commerce 一起提供的样本商店包含这些高级访问控制策略文件。以下部分说明了样本商店如何使用这些访问控制策略文件来定义由组织预订的访问控制策略组。

#### 样本商店中的访问控制

所有样本商店都包含高级访问控制策略文件，这些文件定义专门为商店创建的访问控制策略和策略组。拥有商店的组织拥有这些访问控制数据。

样本商店的高级访问控制策略文件如下：

- 消费者直销
  - FashionFlowAccessControl.xml
-  B2B 直销
  - ToolTechAccessControl.xml
-  需求链
  - CommercePlazaAccessControl.xml（对于渠道中心）

- ResellerStoreFrontAssetStoreAccessControl.xml
- **Business** 供应链
  - SupplierHubAccessControl.xml
  - SupplierAssetStoreAccessControl.xml
- **Business** 主管
  - CommerceHostingHubAccessControl.xml
  - HostedStoreFrontAssetStoreAccessControl.xml
  - StoreDirectoryAccessControl.xml

这些文件位于以下目录中:

- *WC\_installdir /samples/stores/businessmodel*

**注:** **Express** Express Store 高级访问控制文件位于 Express Store 商店归档文件 (ExpressStore.sar) 中。高级访问控制文件名为 AccessControl.xml。

**了解样本商店访问控制策略文件:** 要了解访问控制是如何在商店级别添加的, 请熟悉高级样本商店访问控制策略文件。以下示例取自 **Business** ToolTechAccessControl.xml 文件。

**定义操作:** **Business** ToolTechAccessControl.xml 文件的第一部分定义商店中的新操作, 引导访问控制策略没有涵盖到这些操作。在此情况下, 操作即为商店中使用的所有视图。为了在商店中使用可直接从 URL 调用的视图来显示页面, 或使用可以由其它命令通过重定向来启动的视图 (相对于通过转发到视图来启动) 来显示页面, 必须将其定义为一项操作。请看以下示例:

```
<!-- [Start of Action definitions] -->
<!-- [this is the dictionary of possible actions --->
<Action Name="GenericApplicationError"
  CommandName="GenericApplicationError">
  </Action>

<Action Name="GenericSystemError"
  CommandName="GenericSystemError">
  </Action>

<Action Name="OrderOptionsView"
  CommandName="OrderOptionsView">
  </Action>

<!--[End of Action definitions] -->
```

其中

- Action Name 是 XML 文件中用来引用此项操作的标号。在这些示例中, 标号与视图名称相同。
- CommandName 是存储在 VIEWREG 表 VIEWNAME 列中的视图的名称。CommandName 将存储在 ACACTION 表的 Action 列中。

**定义操作组:** 第二部分定义操作组。操作组是对文件第一部分中定义的操作的分组。在“多乐五金”示例中, 所有新用户视图都分组到 ToolTechAllUserViews 组中 (该组将

在允许所有用户访问那些视图的策略中使用)，或分组到 ToolTechRegisteredCustomerViews 中（它将在仅允许注册用户访问那些视图的策略中使用）。

**注：**您也可以的操作组中添加在 WebSphere Commerce 中其它地方定义的操作。如果在 WebSphere Commerce 中的其它地方定义，则这些操作需要在第 278 页的『定义操作』中讨论的“操作”列表中重新定义。

```
<!-- [Start of Action Group definitions] -->
<!-- Dictionary of grouped actions usable in policies -->
<!-- cross-component view-related action groups -->
<ActionGroup Name="ToolTechAllUsersViews"
  OwnerID="RootOrganization">

  <ActionGroupAction Name="UserRegistrationForm"/>
  <ActionGroupAction Name="UserRegistrationErrorView"/>
  <ActionGroupAction Name="GenericApplicationError"/>
  <ActionGroupAction Name="GenericSystemError"/>
  <ActionGroupAction Name="LogonForm"/>
  </ActionGroup>

<!-- [End of Action Group definitions] -->
```

其中

- ActionGroup Name 是操作组的名称。操作组名称在 ACACTGRP 表中定义。
- OwnerID 是操作组的所有者。根组织通常为操作组的所有者。如果使用任何其它组织，则使用该组织的 orgentity 标识。
- ActionGroupAction Name 是属于此组的操作的名称。ActionGroupAction Name 必须与第 278 页的『定义操作』的“Action Name”元素中定义的名称匹配。这种“操作对操作组”关系存储在 ACACTACTGP 表中。

定义策略：下一部分定义商店中使用的新策略。

```
<!-- [Start of Policy definitions] -->

<!-- AllUsers for ToolTech can execute ToolTechAllUsersViews -->

<Policy Name="AllUsersForToolTechExecuteToolTechAllUsersViews"
  OwnerID="&seller_b2b_mbr_id;"
  UserGroup="AllUsers"
  UserGroupOwner="RootOrganization"
  ActionGroupName="ToolTechAllUsersViews"
  ResourceGroupName="ViewCommandResourceGroup"
  PolicyType="groupableStandard">
  </Policy>

<!-- RegisteredApprovedUsers for ToolTech can execute
ToolTechRegisteredApprovedUsersViews -->

  <Policy Name="RegisteredCustomersForOrgForTool
TechExecuteToolTechRegisteredCustomerViews"
  OwnerID="&seller_b2b_mbr_id;"
  UserGroup="RegisteredCustomersForOrg"
  UserGroupOwner="RootOrganization"
  ActionGroupName="ToolTechRegisteredCustomerViews"
  ResourceGroupName="ViewCommandResourceGroup"
  PolicyType="groupableTemplate">
  </Policy>

<!-- [End of of Policy definitions] -->
```

其中

- Policy Name 是定义的策略的名称。该策略是在 ACPOLICY 表中定义的。
- OwnerId 是策略的所有者。在此情况下，策略的所有者是拥有商店的组织。
- UserGroup 是策略所适用的用户组（访问组）。
- UserGroupOwner 是访问组的所有者。在此示例中，访问组的所有者与策略所有者不同。如果策略所有者和 UserGroupOwner 相同，则可以省略此元素。
- ActionGroupName 是策略所适用的操作组。
- ResourceGroupName 是策略所适用的资源组。在此示例中，资源组 ViewCommandResourceGroup 已经在引导程序数据中定义。因此，不需要在此高级 xml 文件中重新定义它。
- PolicyType 是策略的类型。策略可以是 groupableStandard 也可以是 groupableTemplate 类型。关于策略类型的更多信息，请参阅《WebSphere Commerce 安全性指南》。

定义策略组： 最后一部分定义商店的特定策略组。

```
<PolicyGroup Name="ToolTechPolicyGroup" OwnerID="&seller_b2b_mbr_id;">
  <PolicyGroupPolicy Name="AllUsersForToolTechExecuteToolTechAllUsersViews"/>
  <PolicyGroupPolicy Name="RegisteredCustomersForOrgForToolTechExecuteTool
TechRegisteredCustomerViews"/>
<PolicyGroupSubscription OrganizationID="&seller_b2b_orgentity_id;" />
</PolicyGroup
```

其中

- PolicyGroup Name 是策略组的名称。它定义在 ACPOLGRP 表中。
- OwnerID 是策略组的所有者。在此示例中，策略组的所有者是拥有该商店的组织。
- PolicyGroupPolicy Name 是属于该组的策略的名称。这种“策略对策略组”关系存储在 ACPOLGPPOL 表中。
- PolicyGroupSubscription OrganizationID 是预订此策略组的组织。该预订存储在 ACPLGPSUBS 表中。

有关将 XML 文件用于访问控制的更多信息，请参阅《WebSphere Commerce 安全性指南》中的『使用 XML 定制访问控制策略』一章。

---

## 向商店添加访问控制

从商店开发角度来看，需要的最常见类型的访问控制是用于为商店创建的新视图和命令的访问控制。但是，您可能希望对商店添加其它类型的访问控制。关于对视图、命令和其它功能的访问控制的更多信息，请参阅《WebSphere Commerce 安全性指南》。请确保复查《WebSphere Commerce 安全性指南》，然后继续执行本指南中概括的以后步骤。

如果您在对基于样本商店的商店添加新的访问控制功能，请编辑现有高级访问控制策略 XML 文件。如果您在对未基于样本商店的商店添加访问控制，则需要创建新的高级访问控制策略 XML 文件。关于这两个方案的详细指示信息，请参阅『创建或编辑商店中的访问控制』。

## 创建或编辑商店中的访问控制

访问控制有用资源与商店中的其它有用资源不同，因为对于访问控制，您创建高级访问控制 XML 文件，然后转换并装入它们。

要创建或编辑访问控制有用资源，请执行以下操作：

1. 复查用于为样本商店创建商店有用资源的高级 XML 文件：  
*samplestorenameAccessPolicies.xml*。这些文件位于以下目录中：

- *WC\_installdir/samples/stores/businessmodel*

有关高级访问控制 xml 文件的语法，请参阅 *samplestorenameAccessPolicies.xml* 所引用的 DTD 文件。DTD 文件位于以下目录中：

- *WC\_installdir /xml/policies/dtd*

如果正在更改基于样本的商店，请编辑这些文件。如果您正在添加访问控制的商店不是基于该样本，则通过复制其中一个样本商店文件来为您的商店创建新的高级访问控制文件。复制文件之后，将其重命名并进行适当的编辑。关于向您的商店添加访问控制的更多信息，请参阅《*WebSphere Commerce 安全性指南*》。

2. 将适当的 *samplestorenameAccessPolicies.xml* 文件复制到以下目录：

- *WC\_installdir/xml/policies/xml*

3. 用实际值来替换以下图表中标识的占位符。要找到这些值，请执行以下步骤：

- a. 通过使用以下查询，找出专有名称（DN）的值。在查询中使用下表中提供的 DN 值：

- `select orgentity_id from orgentity where dn= <the DN of the organization>`

示例：在 *CommercePlazaAccessControl.xml* 文件中，当 DN 值为 `ou=Reseller Hub Organization,o=Demand Chain Management Organization,o=Root Organization` 时，以使用以上查询找到的值替换 `&channel_mbr_id;` 和 `&channel_orgentity_id;`。

表 12.






文件名	以实际值替换以下占位符	DN 值
 Business CommercePlazaAccessControl.xml	<code>&amp;channel_mbr_id;</code> <code>&amp;channel_orgentity_id;</code>	<code>ou=Reseller Hub Organization,o=Demand Chain Management Organization,o=Root Organization</code>
	<code>&amp;proxy_orgentity_id;</code>	<code>ou=Distributor Proxy Organization,o=Demand Chain Management Organization,o=Root Organization</code>
 Business ResellerStoreFrontAssetStoreAccessControl.xml	<code>&amp;profile_mbr_id;</code>	<code>ou=Asset Store Organization,o=Demand Chain Management Organization,o=Root Organization</code>
 Business SupplierAssetStoreAccessControl.xml	<code>&amp;profile_orgentity_id</code> <code>&amp;;profile_mbr_id;</code>	<code>ou=Asset Store Organization,o=Supply Chain Management Organization,o=Root Organization</code>

表 12. (续)

 SupplierHubAccessControl.xml	&channel_orgentity_id; &channel_orgentity_id;	ou=Supplier Hub Organization,o=Supply Chain Management Organization,o=Root Organization
FashionFlowAccessControl.xml	&seller_b2c_mbr_id; &seller_b2c_orgentity_id;	"ou=B2C,o=Seller Organization,o=Root Organization"
 ToolTechAccessControl.xml	&seller_b2b_mbr_id; seller_b2b_orgentity_id;	ou=B2B,o=Seller Organization,o=Root Organization
CommerceHostingHubl AccessControl.xml	&channel_mbr_id; &channel_orgentity_id;	ou=Hosting Hub Organization,o=Hosting Organization,o=Root Organization
HostedStoreFrontAsset StoreAccessControl.xml	&profile_mbr_id;	ou=Asset Store Organization, o=Hosting Organization,o=Root Organization
StoreDirectoryAccess Control.xml	&public_mbr_id; &public_orgentity_id;	ou=Store Directory Organization,o=Hosting Organization,o=Root Organization

4. 在文件中添加适当的访问控制信息。关于更多信息，请参阅第 277 页的『样本商店中的访问控制』和《WebSphere Commerce 安全性指南》。
5. 运行 `acpload` 命令来转换文件 `samplestorenameAccessPolicies.xml`，并将其装入到数据库中。

**注：**数据库用户标识必须具有以下许可权以便执行 `acpload` 命令：

- 对 `WC_installdir/xml/policies`、`WC_installdir/logs` 的目录、子目录和文件的读 / 写 / 执行权限。
  - 对 `WC_installdir/bin` 目录及其文件的读 / 执行权限。
- a. 在命令提示符下，转至以下目录：
    - `WC_installdir/bin`
  - b. 运行 `acpload` 命令文件。
    -  **语法：** `acpload.cmd databasename database user database user password Policies XML file [schema name]` **示例：** `acpload mall dbuser dbusrpwd ChannelHubAccessControl.xml`
    -     **语法：** `acpload.sh databasename database user database user password Policies XML file [schema name]` **示例：** `acpload.sh mall dbuser dbusrpwd ChannelHubAccessControl.xml`
  - c. 检查以下日志文件以确保访问控制数据已成功装入：
    - `WC_installdir/logs/acpload.log`
    -  `WC_userdir/instances/acpload.log`





关于 @ 和 & 用法的更多信息，请参阅第 417 页的附录 B，『创建数据』。

---



---

## 第 8 部分 使您的商店全球化



---

## 第 34 章 全球化

---

### 支持全球化

WebSphere Commerce 翻译成以下 10 种语言:

- 美国英语
- 法语
- 德语
- 意大利语
- 西班牙语
- 巴西葡萄牙语
- 简体中文
- 繁体中文
- 韩国语
- 日语

这包括软件、软件文档、用户界面和样本。您可以添加其它语言的支持。您可以翻译站点的许多特征，例如页面上的产品描述、消息和文本。这适用于商店上的页面，并适用于基于浏览器的 WebSphere Commerce 工具（例如 WebSphere 贸易加速器和管理控制台）上的页面。

WebSphere Commerce 包括几个功能部件，它们使您可以定制创建站点，使之满足国际化或多文化客户群的需要。通过使用 Java 技术以及灵活的数据库模式，您可以根据地点或客户喜好改动站点的一些文化特征，其中包括:

**语言** 一个商店可以通过一种以上语言进行查看。例如，您可能希望允许客户选择他们查看站点时想用的语言，或者您可能希望能根据商店的地点自动选择一种缺省语言。

**货币** 一个商店可以一种以上货币来显示价格。在 WebSphere Commerce 中，特定货币的选择与特定语言的选择无关。

#### 文化格式

数据可以各种可定制的格式显示。来自不同文化的客户对于某些信息应如何显示，可能会有不同的期望。例如，根据客户的语言、国家或地区或区域，十进制数可以使用逗号来表示，也可以使用句点来表示。

**地址** 地址可以各种格式输入、存储和显示，以符合不同的国际标准。

**税务** 可以针对不同地区来定义税务规则。这包括销售税和其它营业税的规则。

**送货** 可以针对不同地区来定义送货规则和承运人。

#### 支付方式

可以针对不同地区来定义支付方式。

**价格** 可以按一种货币设置价格，兑换系数可从此货币转换到其它货币，或者可对每种货币均设置价格。

## 联机产品目录内容

产品和类别描述、属性和图像可能随地点而变化。WebSphere Commerce 允许商家在语言表中定义每种格式后，通过选择显示格式来管理联机产品目录内容。您还可以维护一个主产品目录，该产品目录可供许多商店共享。

WebSphere Commerce 数据库已设计成允许您通过使用 Unicode UTF-8 编码或 UCS-2（在发送到 Web 浏览器时它们可转换成大多数的国际编码格式）来灵活地创建和维护可全球识别的数据。关于针对许多国家或地区进行全球化的站点工作样本，请参阅任一家样本商店。

管理控制台和 WebSphere 贸易加速器支持全球化使用。这些工具以 WebSphere Commerce 支持的 10 种本地语言中的任意语言显示。它们使用的程序设计模块允许在不影响页面的一般运作或外观的情况下添加其它语言。

您可以将这些工具翻译成其它语言，就如同翻译商店或其它 Web 站点一样。要翻译这些工具，请修改以下目录下的相应属性文件：

```
WAS_userdir/installedApps/cell_name/WC_instance_name.ear/properties  
/com/ibm/commerce/tools/
```

## 样本商店

样本商店和站点为创建商店奠定了基础。所有样本商店均演示如何能创建和维护全球化的站点。

样本商店允许客户选择他们查看站点时希望使用的语言。在“时尚潮流”和 Commerce Plaza 中，您从列表中选择希望的显示格式。该列表显示于贯穿站点的左侧框架上的下拉列表中。客户可以浏览整个站点，以他们选择的语言进行查看。

**Business** 在“多乐五金”中，您通过转至个人信息页面来选择首选语言，从而切换语言。

**Business** 在“商业托管中心”和“商业供应商中心”中，登录页面上提供一个下拉列表，您可以从中选择语言。

样本商店对所有商店和语言程序设计模块使用单个模板。每种受支持的语言都有自己的属性文件，该文件包含该语言的已翻译文本和消息。所有商店归档文件都包含：`publishNLS.properties`、`publishNLS_en_US.properties`。这些只在发布向导中使用。

对于“时尚潮流”以及基于“时尚潮流”的主管商店：

- `infashiontext_locale.properties`
- `infashiontext_dynamic_locale.properties`
- `infashiontext_dynamic_labels_locale.properties`（仅用于“更改页面”GUI）
- `AuctionSample_locale.properties`

**Business** 对于“多乐五金”，该文件是：

- `tooltechttext_locale.properties`

**Business** 对于 Commerce Plaza，该文件是：

- `pcdmarket_locale.properties`

**Business** 对于“商业供应商中心”，该文件是：

- `tooltechttext_locale.properties`

**Business** 对于“商业托管中心”，该文件是：

- `b2cHostingChannel_locale.properties`

**Business** 对于“主管中的商店目录”样本，该文件是：

- `b2cHostingPublic_locale.properties`

在属性文件中，许多页面元素已翻译：

**文本** 页面的文本内容。

**标签** 格式字段标签。

**消息** 错误、状态和确认消息。

**备用文本**

用于图像、Java applet 和其它嵌入对象。

**确定字符集**

文本在浏览器中显示时所用的字符集是使用 `ENCODESTATEMENT` 属性在属性文件中定义的。例如，`infashiontext_en_US.properties` 文件包含以下语句：`ENCODESTATEMENT = text/html; charset=ISO_8859-1`。由于编码是在属性文件而不是 JSP 模板中指定的，所以对于每种语言，字符集可能会有所不同。生成的 JSP 页面的字符编码是通过在 JSP 模板中使用以下语句来设置的：

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

运行时，每个请求的 JavaServer Page 都包含文件 `EnvironmentSetup.jsp`。在这个文件中，检索命令上下文，从而使用语言环境来检索 `infashiontext Properties java` 对象，该对象从特定于语言环境的相应目录中的 `infashiontext_locale.properties` 文件获取值。然后，通过使用 `ResourceBundle` 对象的 `getString()` 方法，模板有权按需要访问每个属性。

**注：**这用于仅使用商店创建向导所创建的价值链主管商店。当卖方创建主管商店时，这就不是常规商店发布了，并且商店缺省语言之外的其它商店语言有用资源将不安装到主管商店中。因此，如果卖方将某受支持的语言添加到商店，则该语言的商店有用资源不可用。如果要将某受支持的语言添加到主管商店，则请确保已翻译的有用资源（属性文件）可用于商店，否则商店页面将无法正确运行。

**显示已翻译的图像**

运行时检索语言环境和语言，以确定正确的文件夹，从中寻找图像文件。模板可能会查找文件 `FashionFlow/language_Locale/images/go_button.gif`，其中 `language_Locale` 来自命令上下文的显示格式替换。例如，生成的页面将显示图像：`FashionFlow/en_US/images/go_button.gif` 或 `FashionFlow/jp_JA/images/go_button.gif`。

**显示产品目录内容**

产品目录包含多种翻译，每个受支持的语言环境对应一种。运行时，通过数据 bean 发送命令上下文，以确定从数据库中检索和在页面上显示哪种翻译。

## 资源束和属性文件

资源束和属性文件使您能够维护 Java 对象的集合，这些 Java 对象是特定于您的 JavaServer Pages 语言环境的。当页面需要特定于语言环境的资源时（例如格式字段标签，图形用户界面消息或下拉菜单的值），页面可以从适合于选定语言环境的资源束或属性文件将其装入，从而允许客户以他们自己的语言查看页面。这样，您可以创建在很大程度上独立于客户语言环境的 JSP 模板，分离出资源束或属性文件中所有特定于语言环境的信息。

尽管资源束和属性文件执行类似的功能，处理它们的方式上还是存在一些区别的。下表显示资源束和属性文件之间一些较重要的区别：

表 13. ListResource 束和属性文件


属性文件	ListResource 束
文本文件	编译格式
性能有轻微下降	性能有轻微上升
如果更改了属性文件，则 WebSphere Application Server 必须重新启动才能查看更改。	如果更改了某个资源束，则必须重新编译，且 WebSphere Application Server 必须重新启动才能查看更改。
与语言和语言环境相关。对于每种语言环境，应存在一个文件。您需要对非 ISO-8859-1 字符运行 native2ascii。	与语言和语言环境相关。对于每种语言环境，应存在一个文件。您需要对非 ISO-8859-1 字符运行 native2ascii。

关于如何在为许多国家或地区全球化的站点中使用属性文件的示例，请参考样本商店。关于这些主题的更多信息，请访问 Sun Microsystems Java Web 站点。

## 数据存储和传送

单个商店可以多种语言显示页面，即使这些语言使用不同的字符集。要实现这一点，请将数据以可应用于多种语言的通用格式存储在 WebSphere Commerce 数据库中。由于不是所有的 Web 浏览器均支持相同字符集，所以当 JavaServer Page 请求数据时，数据会转换到合适的字符集。

下面描述数据如何从数据库传送到浏览器：

1. 文本数据使用 Unicode UTF-8 编码或  UCS-2 格式存储在 WebSphere Commerce 数据库中。
2. JDBC 驱动程序从数据库装入数据，将其从 UTF-8 转换到 Java 的本机 16 位 Unicode 编码。
3. JSP 文件使用 Java 16 位编码输出数据。
4. WebSphere Application Server 将 JSP 文件输出从 16 位 Unicode 转换为目标编码。这种编码可以在 JSP 文件或属性文件中指定。例如，要为某日语页面指定 Shift-JIS 编码，可以执行以下操作：
  - JSP 文件 `<%@ page contentType="text/html; charset=Shift-JIS"%>`。
  - 属性文件 ENCODESTATEMENT = text/html; charset=Shift-JIS 生成的 JSP 文件的字符编码是使用以下 JSP 模板中的语句设置的：  
`<%response.setContentType(fashionflowtext.getString ("ENCODESTATEMENT")); %>`

由于不是所有浏览器均可处理每种编码方案，因此建议您指定较通用的编码方案，例如 UTF-8 和 Shift-JIS。



5. 转换过的数据发送回浏览器。
6. 浏览器基于在页眉处指定的编码来解释 HTTP 应答。

下面描述数据如何从浏览器传送到数据库:

1. 数据输入到浏览器。可以使用某种输入方法输入多语言数据。
2. WebSphere Commerce 使用 `setCharacterEncoding()` 方法将来自浏览器的数据转换成 Java 16 位编码。LANGUAGE 表中的每个 LANGUAGE\_ID 都使用 ENCODING 列映射到一个编码值。此值用于解释来自浏览器的数据。
3. 数据将发送到数据库, 在那里从 Java 16 位转换成 UTF-8 编码, UTF-8 编码是数据在数据库中存储的方式。

---

## 显示格式

显示格式允许商家基于全球化、多语言的客户进行销售。每个显示格式可由三个因子进行标识: 语言、区域和您可定义的变体。您可以设计您的站点, 以向有任何不同因子的组显示不同的内容。例如, 您可以使用语言和语言环境使美国英语和加拿大英语具有各自独立的格式。这些显示格式可能具有相同的文本, 但会有不同的货币和计量单位。您可以添加加拿大法语的显示格式。该显示格式可以显示与加拿大英语相同的货币和计量单位, 但文本将以加拿大法语显示。您甚至可以使用第三个因子, 使得针对某个文化中的特定对象具有独立的格式, 例如青少年、科学家或技术专业人士, 调整您的站点以适合他们的需要。

客户可以选择他们查看站点时想要使用的格式, 或者您可以为他们设置缺省值。有关格式的信息是在您希望更改语言时通过 URL 参数传递的。例如, 如果以 `langId=-2` 传递, 则会话将把当前语言设置为法语。`langID` 存储在会话中。当客户请求某个页面时, 显示格式确定要检索的 Web 有用资源和产品目录信息。

## 创建新的显示格式

要创建新的显示格式, 请执行以下操作:

1. 运行以下命令: `select * from language` 该命令返回正由当前提供的显示格式使用的语言标识。选择下一个可用的 ID\_VALUE。
2. 运行以下命令: `insert into language (LANGUAGE_ID, ENCODING, LOCALENAME, LANGUAGE, COUNTRY) values (ID_Value, ENCODING_VALUE, 'x', 'y', 'z')` 其中:

### **ID\_VALUE**

您在步骤 2 中选择的值。

### **LANGUAGE\_ID (必需)**

独特标识显示格式的标识。

### **ENCODING\_VALUE (必需)**

浏览器将用于显示此语言的页面的字符编码值。这应该与属性文件中使用的编码值相同。ENCODESTATEMENT = `text/html;`

`charset=[ENCODING_VALUE]`。受 Sun JDK 支持的编码值的列表可在 Sun Java 站点 [www.java.sun.com](http://www.java.sun.com) 处找到。

### **LOCALENAME (必需)**

用来代表具有独特的格式化语言和习俗的政治、地理或文化区域的 java 语

言环境。LOCALENAME 是两个字母的 ISO 639 语言代码后跟两个字母的 ISO 3166 国家或地区代码，以下划线分隔。

#### **LANGUAGE (可选)**

语言名称。

#### **COUNTRY (可选)**

显示格式的国家或地区。

#### **VARIANT (可选)**

VARIANT 列是额外的列，允许您描述特定文化中的子组，例如青少年、技术人员以及任何其它类别。

例如，要为在美国所讲的意大利语添加显示格式，可以执行以下语句：`insert into language (LANGUAGE_ID, ENCODING, LOCALENAME, LANGUAGE, COUNTRY) values ('333', 'ISO8859-1', 'it_US', 'Italian', 'United States')`。您可能还想要指定备用语言，那么请参阅『[创建新显示格式的示例](#)』。

3. 将一个条目添加到 LANGUAGEDS 表。有关示例，请参阅『[创建新显示格式的示例](#)』。
4. 将一个条目添加到 LANGPAIR 表。有关示例，请参阅『[创建新显示格式的示例](#)』。
5. 向商店添加语言。关于如何向您的商店添加语言的详细信息，请参阅第 298 页的『[向商店添加语言](#)』。

### **创建新显示格式的示例**

以下示例显示可如何创建显示格式，以泰语查看“时尚潮流”样本商店页面。

1. 将 `infashiontext_locale.properties` 文件翻译成泰语。
2. 确保属性文件中的编码语句引用目标浏览器将支持的字符集。对于泰语，编码语句如下：`ENCODINGSTATEMENT = text/html; charset=MS874`
3. 将文件另存为 `infashiontext_th_TH.properties`
4. 打开 DB2 命令窗口。
5. 运行以下命令：`select * from language` 该命令返回正由当前提供的显示格式使用的语言标识。选择下一个可用的 ID\_VALUE。在这个示例中，泰语的 ID\_VALUE 为 3。
6. 运行以下命令：`insert into language (LANGUAGE_ID, ENCODING, LOCALENAME, LANGUAGE, COUNTRY, MIMECHARSET) values (ID_Value, ENCODING_VALUE, 'w', 'x', 'y','z')` 使用以下值：`insert into language (LANGUAGE_ID, ENCODING, LOCALENAME, LANGUAGE, COUNTRY) values ('3', 'MS874', 'th_TH', 'Thai', 'Thailand', 'MS874')` 其中：

#### **ID\_VALUE**

您在步骤 2 中选择的值。

#### **LANGUAGE\_ID (必需)**

独特标识显示格式的标识。

#### **ENCODING\_VALUE (必需)**

浏览器将用于显示此语言的页面的字符编码值。受 Sun JDK 支持的编码值的列表可在 Sun Java 站点 [www.java.sun.com](http://www.java.sun.com) 处找到。编码值必须受 Sun JDK 支持。

### LOCALENAME (必需)

用来代表具有独特的格式化语言和习俗的政治、地理或文化区域的 java 语言环境。LOCALENAME 是两个字母的 ISO 639 语言代码后跟两个字母的 ISO 3166 国家或地区代码，以下划线分隔。关于 ISO 语言代码，请参考以下国际标准组织 Web 站点：[www.iso.ch](http://www.iso.ch)。

### LANGUAGE (可选)

语言名称。

### COUNTRY (可选)

显示格式的国家或地区。

### MIMECHARSET (可选)

用于 MIME 消息传递的字符集。

### VARIANT (可选)

VARIANT 列是额外的列，允许您描述特定文化中的子组，例如青少年、技术人员以及任何其它类别。

7. 将一个条目添加到 LANGUAGEDS 表:

```
languages language_id=-1, description=your_language_name_in_English,  
language_id_desc=-11.
```

例如: "languages language\_id=-1, description=French, language\_id\_desc=-11, languages language\_id=-11, description=Your\_language\_name\_in\_your\_own\_language, language\_id\_desc=-11, desc=-11.

例如: "languages language\_id=-11, description=Francais, language\_id\_desc=-11.

8. 如果以泰语请求的数据不存在，则请为所使用的泰语创建可替代的语言。如果数据库中的所有数据并不都翻译成新语言，这会很有用。要创建可替代的语言，请运行以下命令: `insert into langpair(LANGUAGE_ID, LANGUAGE_ID_ALT, SEQUENCE, STOREENT_ID) values (ID_Value, ID_Value_ALT, 'x', 'y')` 使用以下值: `insert into langpair(LANGUAGE_ID, LANGUAGE_ID_ALT, SEQUENCE, STOREENT_ID) values ('3','-1', '1' '12345')` 其中 LANGUAGE\_ID 是所请求的语言。LANGUAGE\_ID\_ALT 是可替代的语言。SEQUENCE 如果请求的语言如 STORELANG 表中指定的那样受支持，但不提供该语言的信息，则以升序的 SEQUENCE 来尝试每种可替代的语言。商店可以覆盖为其商店组指定的 SEQUENCE。STOREENT\_ID 此关系所属的商店实体。商店的可替代语言关系包括其商店组的可替代语言关系。上面的 insert 语句将指定英语 (语言标识 = -1) 作为第一可替代语言，在未找到泰语数据的情况下为标识为 "12345" 的商店尝试该语言。
9. 将所有属性文件从本机转换成 ascii: 将 `infashiontext_th_TH.properties` 复制到临时目录 (例如: `/tmp`)。运行以下命令: `JDK_dir/bin/native2ascii -encoding TIS620 /tmp/infashiontext_th_TH.properties /tmp/infashiontext_th_TH_new.properties`。将此 `/tmp/infashiontext_th_TH_new.properties` 复制到此目录:  
`/WAS_userdir/installedApps/cell_name/WC_instance.ear/Stores.war`  
`/WEB-INF/classes/storeDir/infashiontext_th_TH.properties`。其中 `JDK_dir` 是 JDK 的路径。

## 多语言数据条目

如果浏览器支持 Unicode，您可以同时在一个浏览器中显示多种语言。Netscape Navigator 和 Internet Explorer V4 或更高版本均支持 Unicode 显示。然而，您的操作系统可能不具有您以某些语言输入文本所需要的字符。要做到这一点，您可能需要使用输入方法。输入方法是一个软件组件，它将按键操作转换成不能直接输入的文本输入。对于

所拥有的字符多于键盘上可装配字符的那些语言，例如日语、中文、韩国语、泰语和印地语，通常使用输入方法来输入文本。常见的输入方法工具是 Microsoft® Global Input Method Editor，它可以从 Microsoft Web 站点获取。Global IME 既适用于客户在购物页面上输入数据，也适用于管理员在 WebSphere 贸易加速器和管理控制台中输入数据。

如果选择不使用输入方法，您也能以不同语言输入数据，条件是您安装了数台机器，每台机器都使用不同的操作系统语言和正确配置的浏览器。浏览器将自动支持安装该浏览器的机器的本机语言。例如，要输入日语和德语数据，您可以安装两台机器，一台使用德语操作系统，另一台使用日语操作系统，每台机器均具有能够显示来自该机器上操作系统的数据的浏览器。关于此问题的更多信息，请查阅您的操作系统或 Web 浏览器的文档。

**Unicode:** WebSphere Commerce 文本数据是使用 Unicode 字符集进行编码的。Unicode 可以显示几种主要语言中使用的字符，包括欧洲语、中东语和亚洲语。在 WebSphere Commerce 中，Unicode UTF-8 标准用于在相同的数据库实例中以多种语言形式存储数据。尽管客户不需要使用支持 Unicode 的浏览器来查看由 WebSphere Commerce 驱动的站点，但如果管理员想要在同一机器上以多种语言查看他们的站点，管理员可能需要一个这样的浏览器。如果您想要以非英语语言来查看站点，则需要启用了 Unicode 的浏览器。关于 Unicode 的更多信息，请访问 Unicode Web 站点。

---

## 创建全球化商店

要创建全球化商店，请执行以下操作

1. 创建一家商店。
2. 管理模板。
3. 向商店添加语言。
4. 创建全球化产品目录。
5. 管理全球化有用资源。
6. 翻译属性文件。

## 创建商店

您可以通过发布样本商店归档文件之一和编辑最后得到的商店来创建商店，或者可以分别创建商店前台、业务逻辑或数据有用资源。

- 商店前台：商店的外部部分或向客户显示的部分被称为商店前台。商店前台由 Web 有用资源（如 HTML 页面、JSP 文件、样式表、图像、图形和其它多媒体文件类型）组成。关于更多信息，请参阅第 73 页的第 4 部分，『开发商店前台』。
- 业务逻辑：商店处理客户请求的部分（包括命令和定制代码）称为业务逻辑。关于创建业务逻辑或定制代码的更多详细信息，请参阅《WebSphere Commerce 编程指南与教程》。
- 商店数据：组成商店的数据有用资源。为了正常运作，商店必须准备好数据以支持所有客户活动。例如，为让客户能够进行购买，商店必须包含待售商品的产品目录、处理订单的过程、实现请求的库存以及送货过程。您的商店还必须具有处理和收取支付款项的方法。创建商店数据中涉及到的概念和任务在第 107 页的第 6 部分，『开发商店数据』中进行讨论。

关于如何发布样本商店归档文件的更多信息，请参阅 WebSphere Commerce 产品联机帮助中的主题“发布商店归档文件”。

## 管理全球化站点的模板

要管理全球化站点中的静态页面和动态模板，存储文件的目录结构必须允许简单快捷地标识文件及其所属的语言环境。

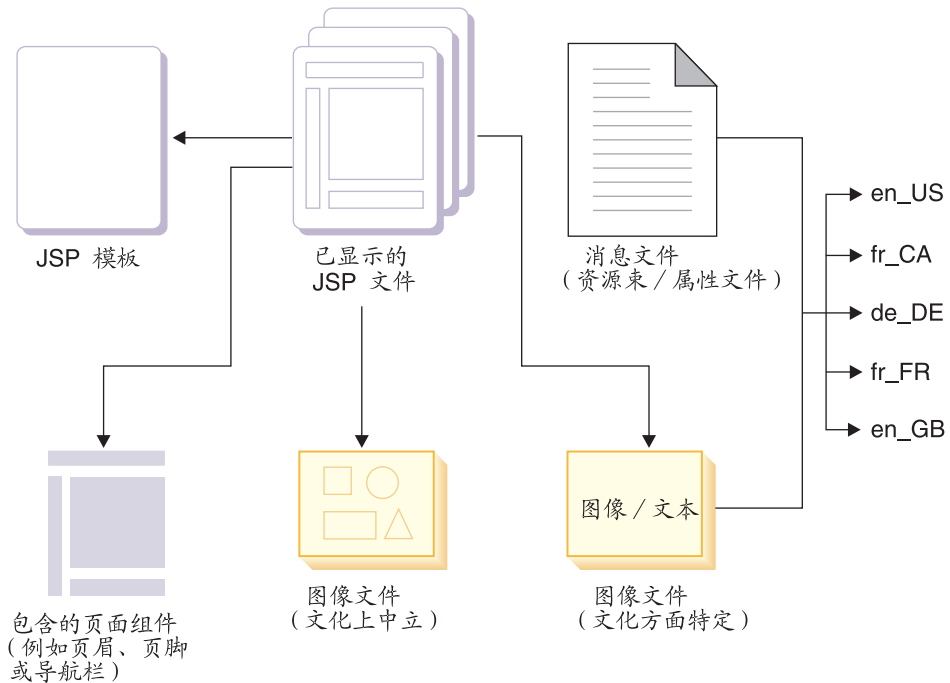
文件目录路径是根据 WebSphere Commerce 实例、商店概要文件内包含的 Business 商店路径以及已注册的文件路径来构造的。创建全球化站点时，您创建多家商店，每家商店代表该站点的一个受支持的送货地区，而且每家商店具有受支持语言的列表。由于模板文件会影响站点的外观，所以模板文件存储在特定于语言环境的目录下，这样就能使用语言环境值，以类似于选择资源束的方法来选择它们。当系统选择一个模板以用于特定语言格式时，会使用语言环境来确定语言格式，该语言格式将用来确定检索文件的目录。

对于全球化环境中的模板存储器，有三个模型：

表 14. 全球化环境中的模板存储器

	所有商店和语言使用一个模板	每种语言使用一个模板	每家商店使用一个模板
<b>定制</b>	对于大多数商店，提供每家商店和每种商店语言格式之间足够级别的定制。	允许每家商店和每种语言格式之间最大级别的定制。	每家商店之间某一级别的定制。
<b>页面外观</b>	页面外观相似。	页面差别可能很大。	页面具有相同的常规布局。
<b>维护</b>	由于仅有一个模板需要更改，所以考虑在站点范围内简易地更改页面设计。对于大多数全球化站点，此模型提供最佳级别的可维护性和可伸缩性。	必须管理每个模板的多个副本。必须对每个模板进行影响所有商店或所有语言格式的更改。	必须跨多个模板对 JSP 文件外观做站点范围的更改。
<b>何时使用</b>	在每家商店的外观和每种语言非常相似时使用。	在各语言之间的页面外观和内容差别非常大时使用。在这种情况下，可以跨语言共享的部分并不多，为每种语言开发独立的页面比较方便。	在商店外观差别显著，但不考虑语言时则保持比较相同的情况下使用。
<b>何时不使用</b>	如果站点在商店和语言之间的差别看起来非常大，请不要使用。	如果页面在商店之间和语言格式之间非常类似，请不要使用。	如果商店外观非常类似，请不要使用。
<b>属性文件</b>	必需。每种受支持的语言还拥有自己的属性文件，在生成页面时会包含该属性文件。	不是必需的。每个商店和语言环境组合都有自己的 JavaServer Pages 模板。	必需。考虑到各种语言格式之间的模板共享。
<b>购物流程</b>	购物流程在语言和商店方面保持相同。	购物流程在语言方面会有显著更改。	购物流程在语言和商店方面保持相同。

## 所有商店和语言程序设计模块使用一个模板



在“所有商店和语言程序设计模块使用一个模板”中，每个页面均由单个 JavaServer Pages 模板组成，包含基本页面布局和在文化上中立的数据和图像。运行时，此模板基于客户选择的显示格式，与文化方面敏感的组件相结合。无论支持多少种文化，只需对页面设计进行一次更改。由于文化方面敏感的功能与页面的其它功能是分开的，所以添加或除去语言或文化非常简单。

下表显示可以如何组织文件。请注意，*webapp* 是指站点或应用程序的根目录。在该目录内，您可能拥有一个公共目录，并且针对每种显示格式或受支持的语言 - 语言环境组合都有一个目录。精确的结构则由您决定。运行时，模板使用来自命令上下文的语言和语言环境信息，并使用它来确定适当的文件夹，从中检索属性文件、图像文件和任何其它文化方面的特定内容。例如，如果命令上下文指示 *en\_US* 显示格式，它将使用 *site\_root/en\_US/sensitivetext.properties* 文件，并将从 *site\_root/en\_US/images/* 目录检索图像。

表 15. 所有商店和语言程序设计模块使用一个模板

模板	/webapp/common/web/template/template.jsp 对所有显示格式使用同一个模板。
所包含的页面组件	/webapp/common/web/template/header.jsp /webapp/common/web/template/footer.jsp。公共页面组件在此目录中。
文化上中立的图像文件	/webapp/common/web/images/image.gif 图像位于公共目录中，并用于所有显示格式。

表 15. 所有商店和语言程序设计模块使用一个模板 (续)

<p>属性文件 (选择以下两种方法中的一种来存储属性文件)</p>	<p>/webapp/language_LocaleA/web/sensitivetext.properties /webapp/language_LocaleB/web/sensitivetext.properties 每种显示格式都有独立的属性文件。不同显示格式的属性文件具有相同的基本名, 在文件扩展名前面的名称中追加语言环境后缀 <i>xx_XX</i>。它们位于同一个目录中。目录名是基于 <i>Language_Locale</i> 组合的, 就象它在 LANGUAGE 表的 LOCALNAME 列中出现一样。关于此方法的示例, 请参阅样本商店。或者您可以使用这种方法: /webapp/properties/sensitivetext_Language_LocaleA.properties /webapp/properties/sensitivetext_Language_LocaleB.properties 属性文件存储在单个目录中, 但具有特定于语言环境的文件名。</p>
<p>文化方面特定的图像文件</p>	<p>/webapp/language_LocaleA/web/images/image.gif /webapp/language_LocaleB/web/images/image.gif 对于每种显示格式, 均存储独立的已翻译图像。这些文件具有相同的名称, 但位于不同的目录中, 与它们所适用的显示格式的名称相对应。 <i>Language_Locale</i> 组合代表显示格式, 就象它在 LANGUAGE 表的 LOCALENAME 列中出现一样。</p>

### 所有商店和语言目录结构使用一个模板

在每家商店的外观和每种语言非常相似的情况下, 这个模型是合适的。您只需维护一组 JSP 模板, 但必须管理一系列属性文件。这种方法就是要维护的模板管理模型, 由于只有一个模板需要更改, 所以它考虑在站点范围内更改页面设计。

例如, 如果您有两个商店地点, 分别显示美国英语和加拿大法语, 则可以组织 JSP 模板如下: /webapp/common/web/template/abc.jsp

此 JSP 模板的属性文件的路径将存储如下: /webapp/common/web/properties/en\_US/abc.properties/webapp/common/web/properties/fr\_CA/abc.properties。

在此情况下, 当注册 JSP 文件时, 仅文件类型需要包含在文件注册表中。使用此方法, 对于所有商店和所有语言环境, 仅需注册一组 JSP 文件。这里, 属性文件必须分开存储, 因为它们包含文化方面敏感的信息, 而模板本身由于是完全中立的, 故存储在公共目录中。关于此模板管理战略的工作示例, 请参考样本商店。

### 每种语言目录结构使用一个模板

要使用此模型, 必须在每家商店的模板目录中为每种受该商店支持的语言创建独立的目录。在这些与语言相关的目录中, 必须都存储不同的模板。此模型中任何属性文件都不是必需的, 因为每个商店和语言环境组合都有自己的 JavaServer Pages 模板。

例如, 如果您有两个商店地点, 分别显示美国英语和加拿大法语, 您可以组织 JSP 模板如下:

/webapp/StoreA/web/template/en\_US/abc.jsp

/webapp/StoreA/web/template/fr\_CA/abc.jsp

/webapp/StoreB/web/template/en\_US/abc.jsp

/webapp/StoreB/web/template/fr\_CA/abc.jsp 在此情况下, 当注册 JSP 模板时, 必须在文件注册表中包含语言环境和文件类型。每家商店和每个语言环境必须拥有一组完整的已注册模板。

## 每种商店目录结构使用一个模板

此模型的 JSP 模板可在一家商店内共享，但仅供单家商店专用。此模型中要求有 JSP 包含文件，以顾及各种语言格式的模板共享。

例如，如果您有两个商店地点，分别显示美国英语和加拿大法语，您可以组织 JSP 模板如下：

```
/webapp/StoreA/web/template/abc.jsp /webapp/StoreB/web/template/abc.jsp
```

此模板管理模型中属性文件的路径将类似这个示例：

```
/webapp/StoreA/web/properties/en_US/abc.properties
```

```
/webapp/StoreA/web/properties/fr_CA/abc.properties
```

```
/webapp/StoreB/web/properties/en_US/abc.properties
```

```
/webapp/StoreB/web/properties/fr_CA/abc.properties
```

当注册此模型的 JSP 模板时，在文件注册表中仅需包含文件类型。每家商店必须注册自己的一组完整的 JSP 文件。

## 向商店添加语言

要向现有商店添加对新语言的支持，请执行以下操作：

1. 确保该语言对于您的站点是可用的。关于受支持的 10 种本地语言的列表，请参考多语言支持。如果语言可用，转至步骤 3；如果不可用，转至下一步。
2. 创建该语言的新显示格式。使用“商店概要文件”笔记本将该语言添加到受商店支持的语言列表。
3. 将本地语言文件（例如，对于“时尚潮流”，将为 `infashiontext_locale.properties`）复制到此位置：

```
AppServer/installedApps/host/WC_demo1.ear/Stores.war/WEB-INF/classes/storeDir.
```

有许多其它 XML 文件需要使用装入程序来翻译和植入，如

```
tax.xml、store.xml、fulfillment.xml、catalog.xml、businesspolicy.xml、contract.xml、accesscontrol.xml、shipping.xml。
```

## 创建全球化联机产品目录

要创建适合于全球化站点的灵活联机产品目录，应包含关于每个产品的众多详细信息，每批详细信息都对应您想要支持的一种语言或文化。考虑到各种文化之间除语言之外常常存在其它区别，例如某些类型的数据的表示方法。例如，在某些文化中，十进制数由逗号表示，而在其它文化中则使用句点表示。

1. 对于商店支持的每种语言，均必须创建一个产品目录。选择以下产品目录创建方法中的一种：
  - 使用装入程序软件包或您选择的产品目录工具创建产品目录。
  - 以 XML 文件创建产品目录数据，并使用装入程序软件包将其装入到数据库，或者使用管理控制台以商店归档文件格式进行发布。关于更多信息，请参阅『创建产品目录』。
  - 将现有的产品目录转换成适用于装入程序的 XML 文件格式，然后将信息装入到数据库。关于更多信息，请参阅『装入程序软件包』。
  - 以样本商店为基础创建产品目录，然后使用产品管理工具更改信息（这仅对少量数据起作用）。
2. 对于您创建的每个产品目录，考虑如何表示以下类型的信息。



## 联机产品目录产品

对于每个产品目录条目，有多个语言描述。

## 产品描述

描述的语言和习惯用语可以是各种各样的，对不同的客户组突出显示不同的特征。

**价格** 价格可以是各种各样的，以反映关税和其它送货费用，并可以不同的货币表示。

## 文化格式

可以制定日期、姓名、计量单位和其它数据的格式，以适合文化需要。

## 产品图像

您可能想要将不同的产品图像显示给不同的客户。

## 管理全球化有用资源

要管理 Web 有用资源，建议您使用一个全球化程序设计模块，该模块对所有商店和语言使用一个 JSP 模板，包括每个页面的基本设计，以及任何文化上中立的信息。剩余的文化方面敏感的文本通过使用资源束或属性文件，在运行时添加到页面上。

确定要翻译哪些 Web 有用资源。这个列表可能包括：标题、图像、applet、文本、消息和页面中显示的其它文化方面敏感的内容。对这些组件中的一些组件创建多个版本，每个版本都对应站点支持的一种语言或文化。关于可如何在全球化商店中管理有用资源的示例，请参考样本商店。

## 翻译属性文件

要翻译属性文件，请执行以下操作：

1. 使用任一文本编辑器打开属性文件。
2. 翻译属性文件中的文本，并注意以下几点：
  - 不要翻译关键字。关键字是等号左边的内容。源：lastName.Label=Last Name  
翻译：lastName.Label=Nom de famille
  - 对于选项属性，仅翻译分号（;）右边的值。  
源：title.Options=MR;Mr.|MRS;Mrs.|MS;Ms.  
翻译：title.Options=MR;M.|MRS;Mme.|MS;Mlle. 源：publishPhone.Options=Y;Yes|N;No  
翻译：publishPhone.Options=Y;Oui|N;Non
  - 是否翻译注释（即以井字符 # 开头的行）是可选的。
3. 将属性文件另存为文本。如果正在使用程序设计模块，其中：
  - a. 属性文件具有相同的名称，但存储在特定于语言环境的目录中
    - 将文件保存到正确的目录
  - b. 属性文件存储在相同的目录中，但有语言环境追加到名称后面
    - 向文件名追加相应的语言环境。扩展名必须为 .properties
4. 如果属性文件包含非 Latin 1 和非 Unicode 字符，则使用 native2ascii 转换器将数据从非 ascii 格式转换成 Unicode ascii 表示。此过程将使属性文件平台中包含的数据独立。native2ascii 转换器位于以下目录中：

`WC_installdir\jdk\bin`

▶ 400 `WC_installdir/Java400/jdk13/bin`

关于 `native2ascii` 转换器的附加信息，请参考以下站点：[www.java.sun.com](http://www.java.sun.com)

---

## 第 9 部分 封装商店



---

## 第 35 章 封装商店

如果您希望使用您的商店作为样本提供给其他人，则请使用管理控制台中的发布实用程序发布它，或在另一服务器或平台上部署它，您可以以商店归档文件的格式将其打包。

通常，商店归档文件由以下文件组成：

- **Web 有用资源：**创建商店页面的文件，如 HTML 文件、JSP 文件、图像、图形和包含文件。
- **属性资源束：**包含商店页面的文本。如果您的商店支持多种语言，则商店归档文件将包含多个资源束，每种受支持的语言对应一个资源束，再加上一个缺省的资源束（它不包含语言环境）。例如，`AddressText_en_US.properties` 和 `AddressText.properties`。
- **商店数据有用资源：**要装入到数据库中的数据。商店数据有用资源包括竞销、产品目录条目、货币、供货信息、定价、送货、存储和税务信息等数据。关于商店数据有用资源的更详细的列表，请参阅第 107 页的第 6 部分，『开发商店数据』。

随 WebSphere Commerce 提供的样本商店归档文件中的商店数据有用资源是对装入程序软件包有效的、格式良好的 XML 文件的一部分。商店数据有用资源最好是可移植的，并且不应包含特定于某数据库特定实例的已生成主键。相反，它们使用商店发布时由标识解析器解析的内部别名。这些约定的使用允许移植样本商店归档文件。关于更多信息，请参阅第 301 页的第 9 部分，『封装商店』。

关于装入程序软件包的更多信息，请参阅第 323 页的第 37 章，『装入商店数据的概述』。

**注：**商店数据有用资源还包含合同信息，这些信息提供创建合同所必需的信息。合同信息不是通过装入程序软件包装入的；它向创建合同的命令提供输入。

- **支付有用资源：**WebSphere Commerce 支付的配置信息。支付信息不是通过装入程序软件包装入的；它向配置 WebSphere Commerce Payments 的命令提供输入。
- **描述符：**描述关于它如何发布的商店归档文件和信息的 XML 文件。这些文件包括：`store-refs.xml`、`ibm-wc-load.xml`、`unpack.xml` 和 `ForeignKeys.dtd`。

样本商店归档文件中的文件可以按以下结构分组：

- *Store directory*
  - JSP 文件、HTML：按功能区分组成各个子目录。例如：ShoppingArea、AuctionArea、CustomerServiceArea。然后，这些区域中的每一个会分组成各个部分。例如，ShoppingArea 可以再分成 CatalogSection、CheckoutSection、DiscountSection 和 ShopcartSection。如有必要，这些部分中的每一个还可以分成多个部分。
  - 图像：样本商店归档文件中的图像可以按语言环境分组。
- SAR-INF
  - 包含特定于发布该商店归档文件的信息，其中包括以下文件：
    - `store-refs.xml`：定义对该商店归档文件使用的发布参数。

- 属性文件：用于描述此商店归档文件的发布参数的文本。该样本商店归档文件还包含该属性文件的特定于语言环境的版本。
- `unpack.xml`：确定商店归档文件中哪些有用资源将被解包、如何将它们解包以及解包到哪里。

**注：**关于 `store-refs.xml` 和 `unpack.xml` 文件的更多信息，请参阅第 311 页的第 36 章，『发布完整的商店』。

- WEB-INF

- 属性文件：商店的属性文件在 WEB-INF 中的以下目录结构中：
  - Classes
    - *Store directory*
- 商店数据有用资源：商店的数据有用资源（格式为 XML）在 WEB-INF 中的以下目录结构中：
  - 商店

**注：**组合商店归档文件包含其它业务模型目录。

- *Store directory*

- 数据：如果商店包含多种语言，则按语言环境将特定于语言环境的 XML 文件分组。数据目录还包含发布商店归档文件所需的以下文件：
  - `ForeignKeys.dtd`：存储发布参数值。这些参数是由商店数据有用资源 XML 引用的实体名称 - 值对。
  - `ibm-wc-load.xml`：控制数据的装入。
  - `store-data-assets.xml`：包含所有要装入的商店数据的 XML 文件。

**注：**

1. 关于 `ForeignKeys.dtd` 和 `ibm-wc-load.xml` 文件的更多信息，请参阅第 311 页的第 36 章，『发布完整的商店』。
2. 格式为 XML 文件的支付配置信息也包含在此数据目录中。

- 用于更改商店流程的文件：如果可以使用 WebSphere 贸易加速器中的商店工具更改商店的流程，则必要的文件按以下目录结构存放：
  - xml
    - tools
      - stores
        - *Store directory*
          - devtools
            - flow

---

## 创建商店归档文件

商店归档文件的打包结构是灵活的。这里给出的指示信息反映了随 WebSphere Commerce 提供的样本的结构，然而，您可以改变这种结构以满足您的需要。您的商店归档文件必须包括含有 `store-refs.xml` 文件和 `unpack.xml` 文件的 SAR-INF 目录。这两个文件中定义的路径必须与您的商店归档文件的结构一致。

要将商店封装为商店归档文件，请执行以下操作：

1. 复查与 WebSphere Commerce 一同提供的样本商店归档文件的结构和内容。  
商店归档文件位于以下目录中：
  - *WC\_installdir/samplestores/businessmodel*要查看商店归档文件，请使用解压缩程序。
2. 在 WebSphere Commerce Server 上为商店创建一个临时目录。例如，*mystore*。
3. 创建以下子目录：
  - *Store directory* (商店的名称)
  - SAR-INF
  - WEB-INF
4. 在 *Store directory* 目录中，请执行以下操作：
  - 按功能区创建 JSP 文件的子目录。有关示例，请参阅样本商店归档文件。将 JSP 文件和所有必要的 HTML 文件复制到这些子目录中。
  - 创建图像文件的子目录。如果商店支持多种语言，使用语言环境名称为特定语言的信息创建子目录。例如，*en\_US*。将您的图像文件复制到这些子目录中。
5. 在 SAR-INF 目录中，请执行以下操作：
  - a. 为商店归档文件创建 *store-refs.xml* 文件。使用样本商店归档文件中的现有的 *store-refs.xml* 文件作为示例，为您的商店创建 *store-refs.xml* 文件。关于 XML 规范的更多信息，请参阅以下目录中的描述符 *store-refs.dtd*：
    - *WC\_installdir/xml/sar*

注：关于 *store-refs.xml* 文件的更多信息，请参阅第 311 页的第 36 章，『发布完整的商店』。
  - b. (可选) 如果您希望商店归档文件具有用户从管理控制台发布时可以选择的参数，则创建描述这些参数的属性文件。将此文件保存到命名为属性的子目录中。
  - c. 使用样本商店归档文件中的现有的 *unpack.xml* 文件作为示例，为您的商店创建 *unpack.xml* 文件。*unpack.xml* 文件确定如何解包商店归档文件。关于 XML 规范的更多信息，请参阅以下目录中的描述符 *unpack.dtd*：
    - *WC\_installdir/xml/sar*

注：关于 *unpack.xml* 文件的更多信息，请参阅第 311 页的第 36 章，『发布完整的商店』。
6. 在 WEB-INF 目录中，请执行以下操作：
  - a. 为商店的属性文件创建以下子目录结构：
    - *classes*
      - *Store directory*
  - b. 将属性文件复制到 *Store directory*。
  - c. 为商店的数据有用资源创建以下子目录结构：
    - *stores*
      - *Store directory*
        - 数据：如果商店支持多种语言，则使用语言环境名称为特定于语言的信息创建子目录。例如，*en\_US*。
  - d. 将数据有用资源复制到数据目录和相应的子目录。

- e. 使用样本商店归档文件中现有的 `ibm-wc-load.xml` 文件作为示例，为您的商店创建 `ibm-wc-load.xml` 文件。`ibm-wc-load.xml` 文件确定如何装入商店数据。关于 XML 规范的更多信息，请参阅以下目录中的描述符 `ibm-wc-load`:

- `WC_installdir/xml/sar`

**注：**关于 `ibm-wc-load.xml` 文件的更多信息，请参阅第 311 页的第 36 章，『发布完整的商店』。

- f. （可选）如果您希望商店归档文件具有用户从管理控制台发布时可以选择的参数，则您必须创建存储这些参数的值的 `ForeignKeys.dtd` 文件。使用样本商店归档文件中现有的 `ForeignKeys.dtd` 文件作为示例，为您的商店创建 `ForeignKeys.dtd` 文件。

**注：**关于 `ForeignKeys.dtd` 文件的更多信息，请参阅第 311 页的第 36 章，『发布完整的商店』。

7. 创建由 `Store directory`、`SAR-INF` 目录以及 `WEB-INF` 目录组成的 ZIP 文件。将此 ZIP 文件命名为 `storearchivename.sar`。
8. 如果您希望使用管理控制台发布商店归档文件，请参阅第 321 页的『使商店归档文件可用于管理控制台』。

---

## 创建样本商店归档文件

在将商店打包为商店归档文件后，您可以在管理控制台中作为样本商店选择使用它。要将商店归档文件用作样本商店归档文件，请执行以下操作：

1. 将商店归档文件保存到以下目录：
  - `WC_installdir/samplestores`
2. （可选的）创建预览页面。要在管理控制台中显示商店页面的预览，您必须创建预览页面。请执行以下操作：
  - a. （可选）在管理控制台的发布实用程序中，选择要发布的商店归档文件，然后单击**预览**。显示的页面称为预览页面。这些页面是 HTML 文件，它们显示预定义的样本购物流程，并作为样本商店的预览。
  - b. 确定希望在预览页面中显示的购物流程。
  - c. （可选的）在发布了的商店中创建一些样本数据。例如，向购物车中添加商品，并创建少许送货地址和开票地址。将正在从此商店创建预览页面，数据使页面看起来更加真实。
  - d. 使用 Internet Explorer 浏览商店。通过选择“文件”->“另存为”保存每个页面的 HTML。您还应当保存样式表（.css）和图像。将文件保存到以下目录：
    - `stylesheet.css`
      - `WC_installdir/wc.ear/SiteAdministration.war/tools/devtools/preview/locale/businessmodel/storedir/`
    - HTML
      - `WC_installdir/wc.ear/SiteAdministration.war/tools/devtools/preview/locale/businessmodel/storedir`
    - 与语言环境无关的图像



- *WC\_installdir/wc.ear/SiteAdministration.war/  
tools/devtools/preview/images/  
businessmodel/storedir*
    - 与语言环境有关的图像
      - *WC\_installdir/wc.ear/SiteAdministration.war/  
tools/devtools/preview/locale/  
businessmodel/storedir/images*
  - e. 由于图像和 `css` 文件的位置已经更改，您必须更改 `HTML` 页面中至图像和 `css` 文件的引用。更改完引用之后，请确保在浏览器中打开 `HTML` 页面时能够查看图像。
  - f. 将 `HTML` 页面中的链接从命令更改为引用 `HTML` 文件的链接。
3. 将商店归档文件添加到 `sarregistry.xml` 文件从而使其在管理控制台的发布实用程序中显示。关于指示信息，请参阅第 321 页的『使商店归档文件可用于管理控制台』。



---

## 第 10 部分 发布商店

为了创建正常工作的商店，必须将商店前台 Web 有用资源发布到 WebSphere Commerce Server，且必须将商店数据发布到 WebSphere Commerce 数据库。

本部分中的各章讨论 WebSphere Commerce 提供的发布选项：

- 第 311 页的第 36 章，『发布完整的商店』 - 该章讨论在商店处于商店归档文件格式的情况下，使用管理控制台或命令行发布来发布整个商店（商店前台和商店数据有用资源）。
- 第 323 页的第 37 章，『装入商店数据的概述』 - 该章讨论使用装入程序软件包将商店数据有用资源发布到数据库。
- 第 367 页的第 38 章，『装入 WebSphere Commerce 数据库有用资源组』 - 该章讨论使用装入程序软件包将商店数据有用资源组或所有商店数据发布到数据库。
- 第 377 页的第 39 章，『发布商业帐户和合同』 - 该章讨论发布帐户、合同和产品集有用资源。
- 第 379 页的第 40 章，『发布商店前台有用资源和商店配置文件』 - 该章讨论发布商店前台有用资源和商店配置文件。



---

## 第 36 章 发布完整的商店

为了创建正常工作的商店，必须将商店前台 Web 有用资源发布到 WebSphere Commerce Server，且必须将商店数据发布到 WebSphere Commerce 数据库。本章讨论在商店处于商店归档文件格式的情况下，使用管理控制台中的发布实用程序或使用命令行发布来发布整个商店（商店前台和商店数据有用资源）。

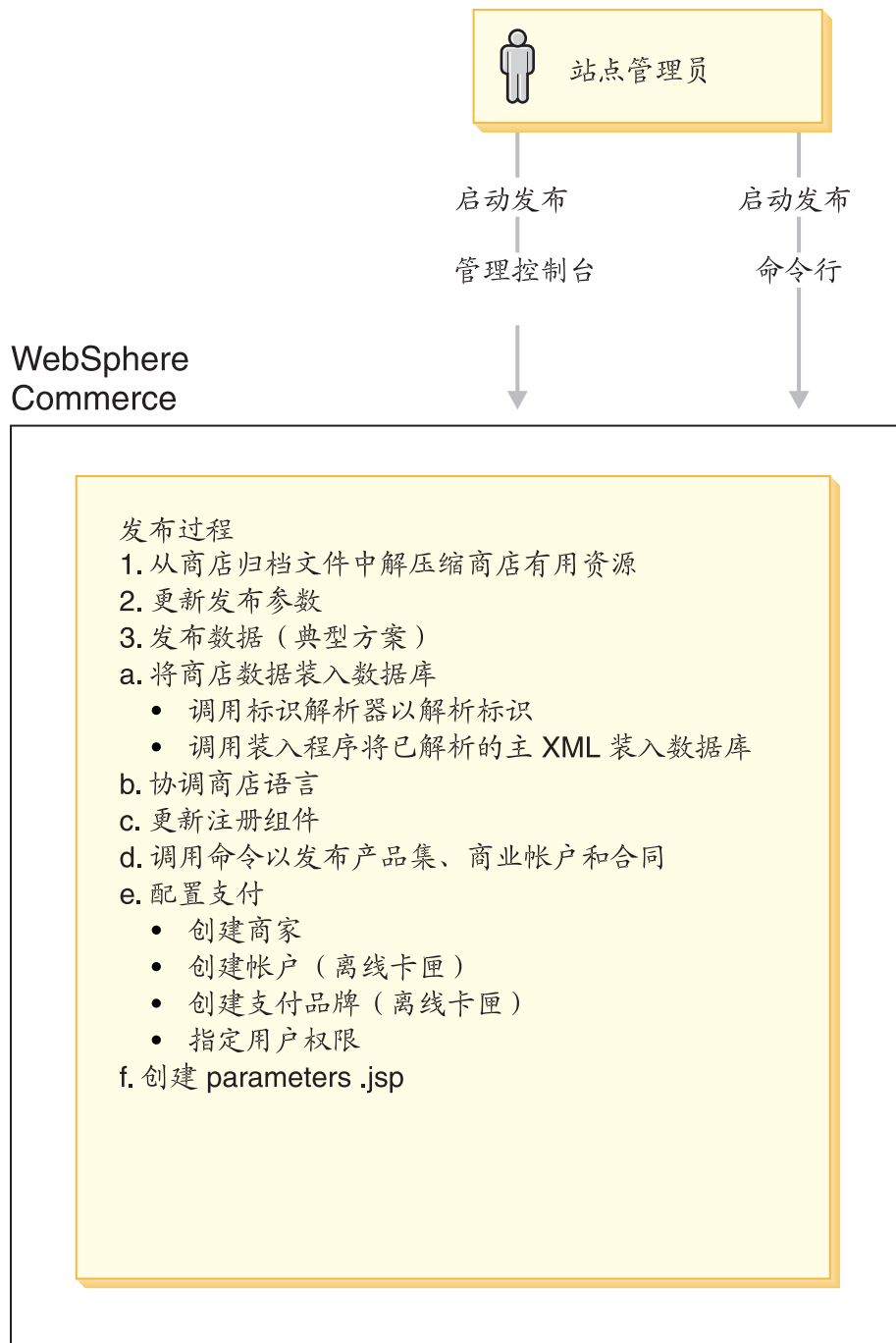
**注：** 如果不希望将商店封装为商店归档文件，可单个地发布有用资源。关于更多信息，请参阅第 323 页的第 37 章，『装入商店数据的概述』、第 367 页的第 38 章，『装入 WebSphere Commerce 数据库有用资源组』、第 377 页的第 39 章，『发布商业帐户和合同』和第 379 页的第 40 章，『发布商店前台有用资源和商店配置文件』。

---

### 了解 WebSphere Commerce 中的发布

管理控制台或命令行提供的发布选项让您能够立即可以发布一个全部完整的商店（商店前台和商店数据有用资源）。为使用此选项，必须以商店归档文件的格式封装商店有用资源。关于将商店封装为商店归档文件的更多信息，请参阅第 301 页的第 9 部分，『封装商店』。

下图概述了发布过程中的步骤。



## 启动发布

为了发布商店，您必须具有站点管理员权限。站点管理员可使用以下方法之一启动发布过程：

- 管理控制台
- 命令行

两种发布方法都要求您指定您希望发布的商店归档文件。

**注：**为了使用管理控制台发布商店归档文件，商店归档文件必须在正确的位置或已注册。关于更多信息，请参阅第 321 页的『使商店归档文件可用于管理控制台』。您可以接着更改选定参数的值（如果有的话），这些参数包括商店标识（唯一标识商店的名称）、商店目录（将向之发布 JSP 文件和图像的唯一位置）以及组织（将向之发布商店归档文件的组织）。

## 管理控制台中的发布参数

管理控制台的发布实用程序中的发布参数由 `store-refs.xml` 文件在每个商店归档文件中定义。

从 `ConsumerDirectStore.sar` 文件查看以下的 `store-refs.xml` 文件示例。

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE store-refs SYSTEM "store-refs.dtd">
<store-refs
  target-dtd="WEB-INF/stores/FashionFlow/data/ForeignKeys.dtd"
  deploy-descriptor="WEB-INF/stores/FashionFlow/data/ibm-wc-load.xml"
  resource-bundle="/SAR-INF/properties/publishNLS">
  ref id="storeDir" entity="STORE_DIR" >
    <input type="text"/>
  </ref>
  <ref id="storeIdent" entity="STORE_IDENTIFIER" >
    input type="text"/>
  </ref>
  <ref id = "parentOrg" entity="ORGANIZATION_DN">
    <input type="member" />
  </ref>
</store-refs>
```

该文件为 `ConsumerDirectStore.sar` 文件定义了三个发布参数：

- `<ref id="storeDir" entity="STORE_DIR" >`  
  `<input type="text"/>`

该实体创建了发布参数商店目录。

- `<ref id="storeIdent" entity="STORE_IDENTIFIER" >`  
  `< input type="text"/>`

该实体创建了发布参数商店标识。

- `<ref id = "parentOrg" entity="ORGANIZATION_DN">`  
  `<input type="member" />`

该实体创建了发布参数组织。

其中

- `ref id` 用作属性文件中由 `store-refs` 资源束属性指定的键。它用于获取显示在发布参数页面中的可转换参数名以及描述。
- `entity` 是 `target-dtd` 中由该参数编辑的 `ENTITY` 的名称。
- `input type` 控制参数如何在屏幕上显示。如果输入类型是文本，则参数将显示在可编辑的字段中。如果输入类型是成员，则所有的现有组织将显示在下拉列表中。不能编辑只读参数。

用户为这些参数输入的值存储在 `target-dtd` 文件中所标识的文件中。`target dtd` 在以下代码中定义：

```
target-dtd="WEB-INF/stores/ConsumerDirect/data/ForeignKeys.dtd
```

该文件还是商店归档文件的一部分并且可随商店数据有用资源一起解包。已解包的文件中将更新对应于每个参数的实体值。商店归档文件中的 DTD 将不更新。这些参数值将存储在文件（在此案例中是 ForeignKeys.dtd）中，直到发布实例化为止。

最后，如果商店可以用多种语言发布（就像样本商店一样），则可以在语言环境特定文件中找到发布参数及其伴随的描述。每个发布参数的字段标签和描述在 store-refs.xml 的资源束属性中所定义的属性文件中。发布期间，发布将查找管理控制台中使用的语言的特定语言环境。stores-ref.xml 文件还定义以下这些文件：

- resource-bundle="/SAR-INF/properties/publishNLS"

**注：**发布参数仅可通过管理控制台使用。如果您通过命令行发布商店归档文件，则您无法指定参数值。将使用商店归档文件中包含的缺省值。

部署描述符指定文件（ibm-wc-load.xml）的位置，该文件控制发布过程的发布数据部分。例如，

```
deploy-descriptor="WEB-INF/stores/FashionFlow/data/ibm-wc-load.xml"
```

选择了参数之后，应单击**完成**来启动发布。在使用管理控制台或命令行启动了发布过程后，您无需再执行任何操作。列在前图和本章中的所有其它步骤都是由 WebSphere Commerce 系统完成的。

关于如何使用管理控制台或命令行发布商店归档文件的更详细的信息，请参阅 WebSphere Commerce 产品联机帮助中的主题“发布商店归档文件”。

## 从商店归档文件中解包有用资源

在管理控制台的发布向导中单击**完成**或从命令行运行发布之后，WebSphere Commerce 会从商店归档文件解包有用资源到 WebSphere Commerce Server。解包有用资源由 unpack.xml 文件控制，该文件在商店归档文件的 SAR-INF 目录中。

以下的 unpack.xml 文件示例来自 ConsumerDirect.sar 文件：

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE ibm-wc-unpack SYSTEM "unpack.dtd">
<ibm-wc-unpack>
  <unpack>
    <include file="*" />
    <exclude file="SAR-INF/*" />
    <exclude file="*.zip" />
    <exclude file="*.war" />
    <exclude file="FashionFlow/devtools/flow/ui/*.properties" />
    <exclude file="xml/*" />
    <rename-store-dir target-name="FashionFlow">
      <store-dir path="WEB-INF/stores/FashionFlow" />
      <store-dir path="FashionFlow" />
      <store-dir path="WEB-INF/classes/FashionFlow" />
      <store-dir path="WEB-INF/xml/tools/stores/FashionFlow" />
    </rename-store-dir>
  </unpack>
  <unpack dest="{wc:ToolsStoresPropertiesPath}">
    <include file="FashionFlow/devtools/flow/ui/*.properties" />
    <rename-store-dir target-name="FashionFlow">
      <store-dir path="FashionFlow" />
    </rename-store-dir>
  </unpack>
</ibm-wc-unpack>
```



```

<unpack dest="{wc:instanceDir}">
  <include file="xml/member/MemberRegistrationAttributes.xml"/>
</unpack>
</ibm-wc-unpack>

```

unpack.xml 文件确定要解包哪些文件（使用 include 和 exclude 元素）以及这些文件将解包到哪些目录（使用 unpack dest 实体），并且还重命名目录（使用 rename-store-dir 实体）。

缺省情况下，解包操作将解包它所在商店归档文件中的所有文件。然而，如果指定的话，解包操作也可仅解包商店归档文件中的某些文件。同样在缺省情况下，解包操作会将这些文件解包到通过组合实例 XML 中 DevTools 元素的 StoreDocRoot 和 StoreWebPath 路径所获得的路径。该路径指向商店 Web 模块的文档根路径。然而，如果像在上述示例中指定的那样，<unpack dest="{wc:ToolsStoresPropertiesPath}">，解包操作会将这些文件解包到另一个位置。请注意：解包操作接受变量。在此案例中，ToolsStoresPropertiesPath 变量 "{wc:ToolsStoresPropertiesPath}" 是 instance.xml 中 devtools 元素的一个属性。

## 更新发布参数

在有用资源从商店归档文件解包之后，WebSphere Commerce 将把 DTD 文件（在样本商店中是 ForeignKeys.dtd）更新为在发布向导中创建或选定的发布参数值。例如，如果原始文件已包含 <!ENTITY STORE\_IDENTIFIER "FashionFlow">，则已更新的文件会包含 <!ENTITY STORE\_IDENTIFIER "MyFashion">。

**注：**发布参数仅可通过管理控制台使用。如果您通过命令行发布商店归档文件，您就不能选择发布参数

## 发布数据

在解包了文件并更新了发布参数后，会为发布过程创建调度作业。用于发布商店归档文件的调度作业编号在管理控制台的发布实用程序中显示。

当调度程序运行发布作业时，WebSphere Commerce 通常完成以下操作：

- 将商店数据从商店归档文件的 XML 装入数据库
- 使商店数据一致
- 更新注册表组件
- 调用命令发布商业帐户和合同
- 配置支付
- 创建 parameters.jsp 文件

发布作业由包含在每个商店归档文件中的 ibm-wc-load.xml 文件控制。该文件由 stores-refs.xml 文件中的 deploy-descriptor 属性指定。

### ibm-wc-load.xml

ibm-wc-load.xml 文件确定在发布作业中要完成哪些任务以及这些任务的顺序。下面是 ibm-wc-load.xml 文件的一个示例：

```

<data-deploy base-dir="." default-target="all">
  <asset id="master" location="store-data-assets.xml"/>
  <asset id="resolved.master" location="store-data-assets.resolved.xml"/>
  <asset id="foreignKeys" location="ForeignKeys.dtd" type="dtd"/>
  <asset id="pmconfigfile" location="paymentinfo.xml"/>

```

```

    <deploy-task-cmd name="configPM" class="com.ibm.commerce.tools.devtools.
publish.tasks.payment.ConfigurePaymentTaskCmd"/>
    <target id="all">
      <task name="idresolve">
        param name="infile" value="\${asset:master}" />
        param name="outfile" value="\${asset:resolved.master}" />
      </task>
      <task name="massload">
        param name="infile" value="\${asset:resolved.master}" />
        param name="maxerror" value="1" />
        param name="nopriamry" value="error" />
      </task>
      <task name="configPM">
        param name="paymentConfigFilename" value="\${asset:pmconfigfile}" />
        param name="storeIdentifier" value="
\${asset:foreignKeys#STORE_IDENTIFIER}" />
        param name="organizationDN" value="
\${asset:foreignKeys#ORGANIZATION_DN}" />
      </task>
    </target>
  </data-deploy>

```

其中

- `base-dir` 是将要发布的信息的目录。"." 指示信息与 `ibm-wc-load.xml` 文件处在同一目录中。
- `default-target` 是要执行的目标的标识。发布期间仅将执行一个目标。
- `asset id` 是分配给将要发布的有用资源的标识。由于发布过程中对有用资源可能执行多个任务，因此会为这些有用资源分配多个标识。
- `location` 是分配有标识的有用资源的位置，该位置相对于基本目录（`base-dir`）。
- `deploy-task-cmd name` 是指定给发布过程中使用的任务命令的短名称。
- `deploy-task-cmd class` 是发布过程中使用的任务命令的全名。
- `target id` 是指定给作为组执行的一系列任务的名称。可以定义多个目标，但在发布过程中只执行缺省目标所引用的一个目标。
- `task name` 是要完成的任务名称。请注意：在此样本中，使用任务的短名称。您可以向 `ibm-wc-load.xml` 文件添加新的任务，但任何新任务必须发出以下命令：  
`com.ibm.commerce.tools.devtools.publish.tasks.DeployTaskCmd`。
- `param name` 是任务的参数名称。输入参数作为名称 - 值对字符串传递给任务。
- `value` 是参数的值。请注意：值可以是变量。这些变量可以在任务运行时解析。

## 将商店数据从商店归档文件的 XML 文件装入到数据库中

将商店数据从商店归档文件中的 XML 文件装入到数据库中时，WebSphere Commerce 执行以下操作：

**调用标识解析器解析标识：** 标识解析器（一个装入程序软件包实用程序）在商店归档文件 XML 文件中生成 XML 元素的唯一标识。例如，标识解析器以唯一值替换样本商店 XML 文件中使用的 @ 别名。关于样本商店中使用的内部别名解析的示例，请参阅第 417 页的附录 B，『创建数据』。

**注：** 当您重新发布时，标识解析器还可以为已发布的商店解析标识。例如，如果您已发布了商店归档文件一次，而且您需要重新发布商店归档文件或其一部分，则标识解析器从数据库中检索唯一标识并在重新发布的过程中使用它们。

关于标识解析器和装入程序软件包其它组件的更多信息，请参阅第 323 页的第 37 章，『装入商店数据的概述』。

当发布调用标识解析器时，必须指定要使用的标识解析器方法。标识解析器具有若干方法，这些方法可用于处理标识解析器输入；明确地讲，是如同标识存在于原始数据中一样地处理数据（更新方法），还是如同标识不存在于原始数据中一样地处理数据（装入方法）。当一些标识存在而另一些不存在时，则使用混合方法。您可以在 WebSphere Commerce 配置文件 *instance\_name.xml* 中指定发布将使用的方法。缺省情况下，发布使用混合方法。关于标识解析器方法的更多信息，请参阅第 327 页的『标识解析命令』。

发布还必须指定使用标识解析器要用到的定制程序文件。缺省的定制程序文件如下：DBConnectionCustomizer 或 OracleConnectionCustomizer。




OracleConnectionCustomizer 定制程序文件位于以下目录中：

- *WAS\_installdir/installedApps/cell\_name/*  
*WC\_instance\_name.ear/properties*



DBConnectionCustomizer 文件位于以下 ZIP 文件中：

- *WAS\_installdir/installedApps/cell\_name/*  
*WC\_instance\_name.ear/properties*
-  *WAS\_userdir/WAS\_instance\_name/installedApps/*  
*cell\_name/WC\_instance\_name.ear/lib/loader/IdResGen.zip*

注： 如果希望指定您自己的定制程序文件，则必须更改 *instance\_name.xml* 文件的 DevTools 部分中的以下属性的值：

- `IDResolverCustomizerFile="myIDResolverCustomizerFile"`

*store-data-asset.xml*: 每个样本商店归档文件包含一个 *store-data-asset.xml* 文件。*store-data-asset.xml* 文件为商店归档文件中将包含在发布过程中的所有数据有用资源文件包含占位符。

下面是 ConsumerDirect.sar 的 *store-data-asset.xml* 文件的一部分，其中例举了占位符：

```
<?xml version="1.0"?>
<!DOCTYPE import SYSTEM "store-data-assets.dtd">

<import>
  &modelorg.xml;
  &modelorgrole.xml;
  &storeorg.xml;
  &storeorgrole.xml;
  &fulfillment.xml;
  &store.xml;
  &en_US_store.xml;
  &en_US_fulfillment.xml;
  &catalog.xml;
  &en_US_catalog.xml;
  &tax.xml;
```

发布期间，所有在 `store-data-asset.xml` 文件中用占位符标识的数据有用资源都合并到 `store-data-asset.xml` 文件中，从而创建一个大的数据文件。

标识解析器使用 `store-data-asset.xml` 和相应的 DTD 文件 `store-data-asset.dtd` 来解析这些标识。解析了标识之后，标识解析器创建以下文件：

`store-data-asset.resolved.xml` 文件，该文件包含这些唯一的标识。如果在标识解析过程中发生错误，则装入程序软件包会将一个条目添加到 `messages.txt` 文件中。关于更多信息，请参阅第 320 页的『发布日志文件』。

**调用装入程序软件包将经解析的主 XML 文件装入数据库：** 装入程序软件包将经解析的 `store-data-asset.resolved.xml` 装入数据库。如果在装入过程中发生错误，则装入程序软件包会将一个条目添加到 `messages.txt` 文件中。

关于装入程序软件包的更多信息，请参阅第 323 页的第 37 章，『装入商店数据的概述』。

当管理控制台或命令行发布调用装入程序软件包时，必须指定要使用的装入程序方法。管理控制台可以使用以下的任何装入程序方法：

- SQL 导入
- 导入
- 装入

**注：** 缺省情况下，管理控制台使用 SQL 导入方法。

您可以使用 DevTools 元素中 `LoaderMode` 属性在 WebSphere Commerce 配置文件 `instance_name.xml` 中指定管理控制台或命令行发布将调用的方法。

- SQL 导入：此方法使用 Java 数据库连接性 (JDBC) 插入和更新数据，提供了最灵活的操作方法，但是也是将大量数据导入少量表的最慢方法。它允许列级别的更新。建议您使用 SQL 导入。

**注：** SQL 导入方法是使用最安全的方法，因为当数据无效时它不会使数据库毁坏。在您可以使用 SQL 导入装入之前，记录必须符合数据库模式约束。其它的装入程序方法更快，因为它们成批装入数据库而不进行过多的检查。因此在使用其它方法之前，您必须确保数据的正确性。

- 导入：此方法使用 DB2 本机导入功能并允许单元格级别的更新，具有中等的速度和灵活性。此方法不可用于 Oracle。
- 装入：此方法使用 RDBMS (DB2 Load 或 SQLLoad) 的本机工具，且是将大量数据装入到少量表中的最快方法。

关于装入命令中的方法的更多信息，请参阅第 336 页的『装入命令』。

管理控制台和命令行发布还必须指定使用装入程序时要用到的定制程序文件。如果未在 WebSphere Commerce 配置文件 `instance_name.xml` 中指定定制程序文件，则发布代码将使用缺省定制程序文件：`MassLoadCustomizer`。

**注：** 如果希望指定您自己的定制程序文件，则必须更改 `instance_name.xml` 文件的 DevTools 部分中的以下属性的值：

- `LoaderCustomizerFile="myLoaderCustomizerFile"`

缺省情况下，WebSphere Commerce 配置文件 `instance_name.xml` 不为此属性指定值。

## 协调商店语言

样本商店归档文件包含受 WebSphere Commerce 支持的所有语言的数据。结果是，当装入程序软件包装入商店数据时，将为商店装入所有语言信息。然而，商店可以仅支持商店驻留所在的实例所支持的语言。协调商店语言任务确保只有实例中支持的语言才可在商店中启用。如果您希望对实例添加其它语言的支持，则可使用配置管理器来实现。有关更多信息，请参阅 WebSphere Commerce 产品联机帮助中的主题“配置管理器”。

## 更新注册表组件

发布过程还更新注册表组件。发布通过调用

`com.ibm.commerce.scheduler.commands.RefreshRegistryCmd` 更新 WebSphere Commerce 中所有的注册表。关于注册表的更多信息，请参阅 WebSphere Commerce 产品和开发联机帮助。

## 调用命令发布商业帐户和合同

一些商店数据库有用资源（合同和商业帐户）不能由装入程序软件包装入，因此发布还调用对应的命令将这些有用资源发布到 WebSphere Commerce Server。这些命令如下：

- `AccountImport` - 从商店归档文件中的 `businessaccount.xml` 文件中创建商业帐户。
- `ContractImportApprovedVersion` - 从商店归档文件中的 `contract.xml` 文件导入合同。
- `ProductSetPublish` - 在创建商业帐户和合同之前，使数据库中的产品集数据与产品目录同步。管理控制台和命令行发布调用 `ProductSetPublish` 命令，该命令接着调用 `AccountImport` 和 `ContractImportApprovedVersion` 命令。

关于发布商业帐户和合同的更多信息，请参阅第 377 页的第 39 章，『发布商业帐户和合同』。

## 配置支付

发布过程还包含配置支付的步骤。WebSphere Commerce 支持 WebSphere Commerce Payments。如果您计划使用 WebSphere Commerce Payments 作为处理支付的方法，则应当如第 203 页的第 21 章，『Payments 工具』中所述创建支付 XML 文件。如果支付 XML 文件包含在正在发布的商店归档文件中，则 WebSphere Commerce 将在发布期间完成以下支付配置：

- 创建商家。
- 创建帐户（仅用于脱机卡匣）。
- 创建 `paymentinfo.xml` 中指定的品牌（仅用于脱机卡匣）。
- 指定用户权限。


**错误处理：** 如果在发布过程的配置支付阶段发生错误，则您可以在发布日志中查看错误消息（请参阅第 320 页的『发布日志文件』）。

## 创建 `parameters.jsp` 文件

发布过程创建文件 `parameters.jsp`。该文件包含 `storeId` 参数。样本商店中的 `index.jsp` 文件使用该参数启动商店。

`parameters.jsp` 位于以下目录中：

- `WAS_installdir/installedApps/cell_name/WC_instance_name.ear/Stores.war/storedir/include`




-  `WAS_userdir/WAS_instance_name/installedApps/cell_name/WC_instance_name.ear/Stores.war/storedir/include`

## 错误处理



如果在发布过程的发布有用资源阶段发生错误，则您可在发布日志（请参阅『发布日志文件』）或通过管理控制台的“发布摘要”页面来查看错误消息。

## 发布日志文件


发布过程的发布有用资源阶段遇到的所有错误写入以下日志和跟踪文件中：

- `activity.log`: WebSphere Application Server 日志。WebSphere Commerce 中所有的错误消息都将写入 `activity.log`。在发布失败时首先检查此日志。`activity.log` 在以下目录中：
  - `WAS_installdir/logs`
  -  `WAS_userdir/logs`
  -  `WCDE_installdir/workspace_name/.metadata/plugins/com.ibm.etools.server.core/tmp0/logs`
- `SystemOut.log` 和 `SystemErr.log`: `SystemOut.log` 在以下目录中。它包含任何在商店发布期间写入标准输出和标准错误的信息。`SystemOut.log` 和 `SystemErr.log` 在以下目录中：
  - `WAS_installdir/logs/instance_name`
  -  `WAS_userdir/WAS_instance_name/logs/WC_instance_name`

**注：**`SystemOut.log` 和 `SystemErr.log` 在开发环境中不可用。写入标准输出或标准错误的信息将显示在 WebSphere Studio 控制台中并可在以下位置捕获：




- `workspaceDir\metadata\plugins\com.ibm.etools.server.core\{tmp0}\logs\server1\trace.log`
- `messages.txt`: 包含发布过程的装入程序软件包部分的错误消息。在发布失败时首先检查此日志。在这些错误消息中提及的行或列号指的是 `store-data-asset.resolved.xml` 文件中的行或列。`messages.txt` 在以下目录中：
  - `WC_installdir/instances/instance_name/logs`
  -  `WC_userdir/instances/instance_name/logs`
  -  `WCDE_installdir/Commerce/logs`

在这些错误消息中提及的行或列号也可以是指 `store-data-asset.xml` 文件中的行或列。



- `trace.txt`: 包含发布过程的装入程序软件包部分的跟踪信息。该文件还包含有关发布过程的标识解析器部分的消息。缺省情况下 `trace.txt` 是关闭的。`trace.txt` 在以下目录中：
  - `WC_installdir/instances/instance_name/logs`
  -  `WC_userdir/instances/instance_name/logs`

**注：**缺省情况下，跟踪会打开，并生成循环日志文件。

-  `WCDE_installdir/Commerce/instances/instance_name /logs`

- `trace.log`: WebSphere Application Server 跟踪日志的一部分。包含发布过程的跟踪信息（如果已启用 `WC_DEVTOOLS` 跟踪组件）。关于启用日志记录的更多信息，请参阅《WebSphere Commerce 管理指南》的『配置』一章。`trace.log` 在以下目录中：
  - `WAS_installdir/logs/WC_instance_name`
  -  `WAS_userdirlogs/WC_instance_name`
  -  `workspaceDir\metadata\plugins\com.ibm.etools.server.core\tmp0\logs\server1`
-  `RESWCSID.txt`: 包含发布过程的标识解析器部分的消息。这些消息中提及的行或列号指的是输入文件（例如 `store-data-assets.xml` 文件）中的行或列。`RESWCSID.txt` 在以下目录中：
  - `WC_userdir/instances/instance_name/logs`

要配置 `trace.txt` 和 `messages.txt` 日志文件（即，调整日志级别或其它选项），请编辑以下文件：

- `WC_installdir/instances/instance_name/xml/loader/WCALoggerConfig.xml`
-  `WC_userdir/instances/instance_name/xml/loader/WCALoggerConfig.xml`
-  `WCDE_installdir/Commerce/instances/instance_name/xml/loader/WCALoggerConfig.xml`

---


## 使商店归档文件可用于管理控制台

为了从管理控制台发布商店归档文件，必须通过下列一种方法使商店归档文件对管理控制台可用：

- 在 `SARRegistry.xml` 文件中注册商店归档文件
- 将商店归档文件复制到适用的商店归档文件目录

### 在 `SARRegistry.xml` 文件中注册商店归档文件

为了从管理控制台发布商店归档文件并预览商店，商店归档文件必须在 `SARRegistry.xml` 文件中注册。`SARRegistry.xml` 文件在以下目录中：

- `WC_installdir/xml/tools/devtools`
-  `WC_userdir/xml/tools/devtools`

要注册商店归档文件，请包含商店归档文件的路径以及 `SARRegistry.xml` 文件中所有预览文件的路径。以下示例说明了在 `SARRegistry.xml` 文件中注册消费者直销样本商店归档文件的条目。

```
<!-- Consumer Direct -->
<SampleSAR fileName="ConsumerDirect.sar" relativePath="ConsumerDirect">
  <view name="ConsumerDirect" />
  <view name="default" />
  <html locale="de_DE" featureFile="" sampleSite="de_DE/B2C/FashionFlow/index.html"/>
  <html locale="en_US" featureFile="" sampleSite="en_US/B2C/FashionFlow/index.html"/>
  <html locale="es_ES" featureFile="" sampleSite="es_ES/B2C/FashionFlow/index.html"/>
```

```
<html locale="fr_FR" featureFile="" sampleSite="fr_FR/B2C/FashionFlow/index.html"/>
<html locale="it_IT" featureFile="" sampleSite="it_IT/B2C/FashionFlow/index.html"/>
<html locale="ja_JP" featureFile="" sampleSite="ja_JP/B2C/FashionFlow/index.html"/>
<html locale="ko_KR" featureFile="" sampleSite="ko_KR/B2C/FashionFlow/index.html"/>
<html locale="pt_BR" featureFile="" sampleSite="pt_BR/B2C/FashionFlow/index.html"/>
<html locale="zh_CN" featureFile="" sampleSite="zh_CN/B2C/FashionFlow/index.html"/>
<html locale="zh_TW" featureFile="" sampleSite="zh_TW/B2C/FashionFlow/index.html"/>
</SampleSAR>
```

其中

- `fileName` 是商店归档文件的名称。
- `relativePath` 是 WebSphere Commerce 配置文件 `instance_name.xml` 中 DevTools 元素的 `sampleSarPath` 属性的相对目录路径。
- `view name` 是从“发布”用户界面显示商店归档文件的视图（显示商店归档文件的方式）的名称。请注意：商店归档文件可以以多个视图显示。例如，可以同时以“消费者直销”视图与“缺省”视图显示。
- `html locale` 是 HTML 预览文件的语言环境。
- `featureFile` 不赞成使用特征文件，并且将不再使用。

## 将商店归档文件复制到适用的商店归档文件目录

如果希望从管理控制台发布商店归档文件，但又计划不包含预览页面，或在管理控制台中指定商店显示方式的“发布”视图，则可以仅将商店归档文件复制到以下目录：

- `WC_installdir/instances/instance_name/sar`

商店归档文件将显示在“缺省”视图中。



---

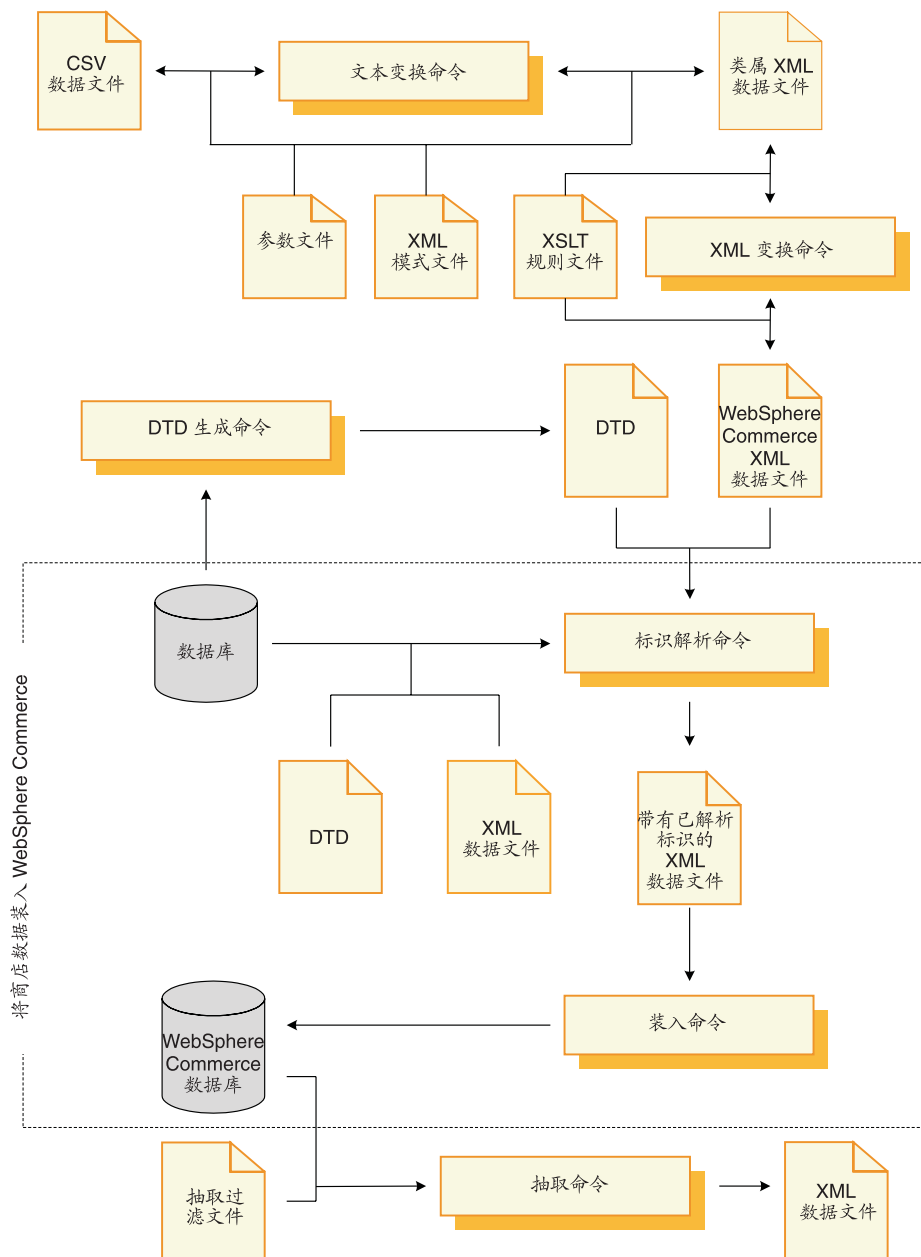
## 第 37 章 装入商店数据的概述

创建商店数据之后，您可选择使用 WebSphere Commerce 管理控制台将它封装为商店归档文件并发布它，或者可使用 WebSphere Commerce 装入程序软件包将它直接装入 WebSphere Commerce Server 数据库。请参阅第 367 页的第 38 章，『装入 WebSphere Commerce 数据库有用资源组』和第 373 页的『装入数据库有用资源组』以获取关于 WebSphere Commerce 数据库有用资源组的装入过程的信息。

装入程序软件包提供了六个命令行实用程序和两个相关的管理工具，您可用来准备要装入的数据并将数据装入商店。这些命令和工具使用“可扩展标记语言（XML）”数据文件管理信息。

## 了解 WebSphere Commerce 中的数据装入

下图显示了可使用装入程序软件包命令执行的数据准备、装入和抽取过程。



请注意虚线指示最常用于将商店数据装入 WebSphere Commerce Server 数据库的两个过程：解析标识和装入数据。这些过程是本章的讨论重点。

有关准备数据以供装入到 WebSphere Commerce Server 数据库的更多信息，请参阅第 95 页的第 5 部分，『商店数据概述』。

以下两个装入程序软件包命令行实用程序常用于将数据装入 WebSphere Commerce Server 数据库:

- **标识解析命令**

要使用装入程序软件包将 XML 数据装入 WebSphere Commerce Server 数据库, XML 元素必须直接映射到目标 WebSphere Commerce Server 数据库的模式。其具有的属性相应于数据库模式中的唯一键或主键的所有 XML 元素都必须具有唯一标识; 数据库模式所有的不可为 null 的列都必须用非 null 值定义相应的属性。标识解析器可以为获得授权的 XML 元素的唯一键或主键属性生成唯一标识。

**注:** 正如此文档中所示, 标识是数据库表(它为每行给予一个唯一标识)的单个列中的一个值。如果您使用标识解析器来生成标识, 则它从 KEYS 或 SUBKEYS 表中获得一个基础值, 并顺序递增该值以解析数据库表中每行的标识。

关于此命令的信息, 请参阅第 327 页的『标识解析命令』、第 356 页的『使用装入程序软件包命令和脚本』和第 357 页的『解析标识的示例』。

- **装入命令**

装入程序使用有效而组织良好的 XML 文件作为输入, 将数据装入数据库。XML 文档的元素映射到数据库中的表名; 元素属性映射到列。

**注:** 关于有效性和格式良好性约束的描述, 请参阅万维网联盟(W3C)XML 原则。

关于此命令的信息, 请参阅第 336 页的『装入命令』、第 356 页的『使用装入程序软件包命令和脚本』和第 364 页的『装入数据的示例』。

这些命令是本章主要的重点。

也可使用以下装入程序软件包命令行实用程序管理数据:

- **DTD 生成命令**

DTD 生成程序生成文档类型定义(DTD), 后者描述要向其装入 XML 数据的目标数据库的表和列。DTD 生成程序还可为数据库生成 XML 模式。

DTD 生成程序可基于 WebSphere Commerce 数据库模式创建 DTD。如果使用随样本商店归档文件提供的 DTD 且未修改数据库模式, 则通常无需使用 DTD 生成程序生成 DTD。

关于更多信息, 请参阅第 345 页的『DTD 生成命令』。

- **抽取命令**

抽取程序对照数据库使用查询, 以将选定的数据子集从数据库抽取到 XML 文档。

您可以使用此命令从数据库中抽取数据并转换为 XML 格式。

关于更多信息, 请参阅第 348 页的『抽取命令』。

- **文本转换命令**

文本转换程序在字符定界的变量格式与 XML 数据格式之间转换数据。

如果您的数据无法直接从数据库中抽取为 XML 格式，则（比如说）您可以将数据保存为字符限定变量格式，然后使用此命令将其转换成 XML 格式。

关于更多信息，请参阅第 351 页的『文本转换命令』。

- **XML 转换命令**

XML 转换程序将 XML 文档中的数据转换为备用的 XML 格式。它使用“可扩展样式表语言（XSL）”定义转换的映射规则。

您可以使用此命令将您的 XML 数据转换成一个格式，该格式直接映射到目标 WebSphere Commerce 数据库的模式（您希望将数据装入到此数据库）。

关于更多信息，请参阅第 352 页的『XML 转换命令』。

这些命令不是本章主要的重点。关于这些命令的详细信息，请参阅 WebSphere Commerce 产品和开发联机帮助。

WebSphere Commerce Loader 软件包还包含以下工具来帮助管理其数据管理功能：

- **文本转换工具**

文本转换工具有助于使用“文本转换”命令来处理字符定界的变量格式和 XML 数据格式之间的数据转换。

- **XSL 编辑器**

XSL 编辑器提供了用于编辑 XSL 文件（可由 XML 转换程序使用）的可视界面。

使用 XSL 编辑器，在对 XML 格式之间转换数据定义映射规则时，建立从源 DTD 中的元素至目标 DTD 中的元素的关联。

这些工具不是本章主要的重点。关于这些工具的详细信息，请参阅 WebSphere Commerce 产品和开发联机帮助的最新版本。

## 用于装入商店数据的装入程序软件包命令

### 标识解析命令

此命令为在可将 XML 数据元素装入数据库之前需要标识的 XML 数据元素生成唯一标识。如果源 XML 数据已提供了必要的唯一标识，则无需运行标识解析器。

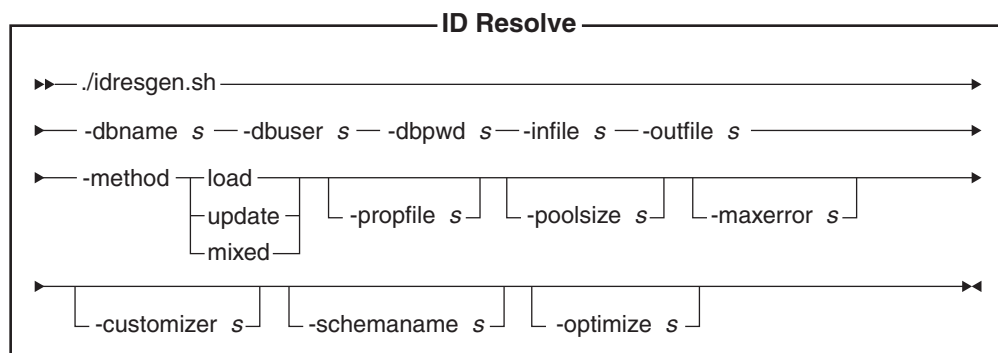
WebSphere Commerce 数据库模式定义其表中的主键和外键，这些键用于表示表之间的各种关系。对于此原因，WebSphere Commerce XML 元素必须包含带有唯一标识的相应属性。在 WebSphere Commerce Server 数据库中，必须解析其标识的表是定义在 KEYS 和 SUBKEYS 表中的那些表。这些表在 WebSphere Commerce 中被称为主表。有关 KEYS 和 SUBKEYS 表的更多信息，请参阅 WebSphere Commerce 联机帮助。

**注：** 如果必须解析在 KEYS 或 SUBKEYS 表中未定义的表的标识，在运行标识解析器前将该表添加至 SUBKEYS 表。

因为 WebSphere Commerce XML 元素和属性意在可跨数据库和跨数据库实例进行移植，因此通常使用内部别名表示其标识。在可将数据装入任何 WebSphere Commerce Server 数据库之前，必须将这些别名解析为有效的数字标识。关于更多信息，请参阅第 417 页的附录 B，『创建数据』。

**注：**

1. 上图主要用作命令参数的参考。
2. 指定为此命令的参数的文件名可以放置在相对或绝对路径之后。



**注：** 指定为此命令的参数的文件名可以放置在相对或绝对路径之后。

参数值:

**-dbname**

▶ AIX ▶ Linux ▶ Solaris 目标数据库的名称。

▶ 400 这是显示在关系数据库目录（WRKRDBDIRE）中的数据库名称。

**-dbuser**

▶ AIX ▶ Linux ▶ Solaris 连接到数据库的用户名称。 ▶ 400 它通常与实例用户名相同。

**-dbpwd**

连接至数据库的用户的密码

**-infile** 包含表记录的输入 XML 文档的名称。

**-outfile**

要产生的输出 XML 文件的名称；此文件可用作对装入程序的输入

**-method**

要用于处理输入文件的方法。该命令可以象记录不存在于数据库中那样处理输入文件（装入），或者如同输入对象的标识已存在那样处理输入文件（更新）。当某些记录不存在于数据库中而某些存在时，则使用混合方法。缺省方法是 load。

**-propfile**

包含 name=value 对格式的 Java 属性的文本文件。此属性文件设置标识解析器解析标识的方式。它用于描述应当将主条目的哪些列用作对需要主行标识的表的查找。此文件定义用于外键标识查找的列名称和用于主表（例如 CATEGORY 和 PRODUCT）查询的选择谓词。可以省略此文件中具有已定义唯一索引（不包含标识）的表的条目。此参数是可选的。IdResolveKeys.properties 是缺省文件。此属性文件可以如以下示例之一所示来指定：

▶ AIX ▶ Linux ▶ Solaris

```
-propfile WC_installdir/my_directory/file_name.properties  
-propfile WC_installdir/my_directory/file_name
```

▶ 400

```
-propfile WC_userdir/my_directory/file_name.properties  
-propfile WC_userdir/my_directory/file_name
```

如果在当前目录中存在此文件，则可以如以下示例所示指定相同的文件：

```
-propfile file_name.properties
```

如果在类路径系统环境变量中指定的目录中存在此文件，则可以如以下示例所示指定相同的文件：

```
-propfile file_name
```

其中，*my\_directory* 是用户定义的目录，而 *file\_name* 是您将要使用的属性文件的名称

关于创建和指定新属性文件以供标识解析器使用的更多信息，请参阅 WebSphere Commerce 产品和开发联机帮助。

### **-poolsize**

要保留的标识数目。此参数是可选的。缺省值是 50。

### **-maxerror**

将导致标识解析器终止的错误数。此参数是可选的。缺省值是 1。

### **-customizer**

要使用的定制程序属性文件名。此参数是可选的。定制程序属性文件设置标识解析器起作用的方式。缺省文件为:

 DB2ConnectionCustomizer.properties

 ISeries\_RESWCSID\_Customizer.properties

如果您已配置实例来使用工具箱驱动程序, 则使用为工具箱驱动程序提供的 Toolbox\_RESWCSID\_Customizer 定制程序文件。还必须指定 `-dbname` 参数的主机名。以下是调用 `idresgen.sh` 脚本的示例:

```
./idresgen.sh -dbname MY.HOSTNAME.CA -dbuser instance -dbpwd mypass  
-infile /path/infile.xml -outfile /path/outfile.xml -method sqlimport  
-customizer Toolbox_RESWCSID_Customizer
```

定制程序属性文件可以如以下示例之一所示来指定:



```
-customizer WC_installdir/my_directory/file_name.properties
```

```
-customizer WC_installdir/my_directory/file_name
```



```
-customizer WC_userdir/my_directory/file_name.properties
```

```
-customizer WC_userdir/my_directory/file_name
```

其中, `my_directory` 是用户定义的目录, 而 `file_name` 是您想要使用的属性文件的名称。

如果在当前目录中存在此文件, 则可以如以下示例所示指定相同的文件:

```
-customizer file_name.properties
```

如果在类路径系统环境变量中指定的目录中存在此文件, 则可以如以下示例所示指定相同的文件:

```
-customizer file_name
```

关于创建和指定新定制程序属性文件的更多信息, 请参阅 WebSphere Commerce 产品和开发联机帮助。

### **-schemaname**

目标数据库模式的名称。此参数是可选的。

如果在运行命令时未指定此参数, 则此命令在指定了 `SchemaName` 的值的定制程序属性文件中查找 `name=value` 对。如果此对出现在属性文件中, 则命令使用指定的值。如果此参数的命令行或属性文件规范都不存在, 则命令缺省为数据库

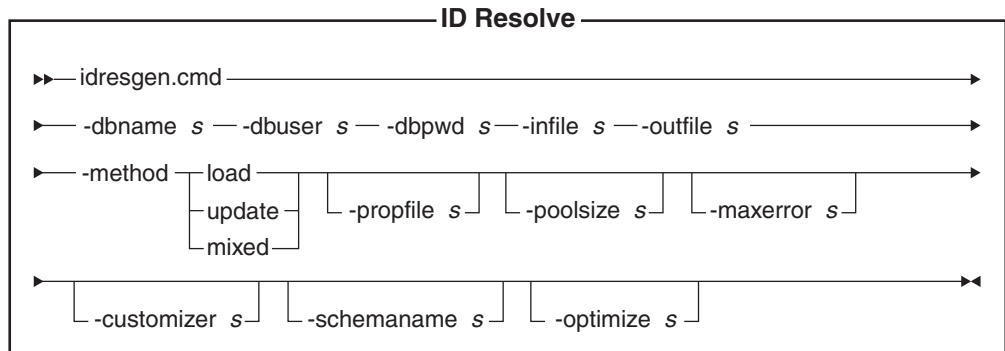
中 `KEYS` 表的模式名称。  如果 `-schemaname` 参数的命令行或属性文件规范都不存在, 则命令缺省为 `-dbuser` 参数的值。

## -optimize

-optimize no

标识解析器将先跳过重复记录检查，然后将已解析的记录写到输出文件中。该选项使用户能关闭标识解析器中的优化功能。

Windows





参数值:

**-dbname**

目标数据库的名称

**-dbuser**

连接至数据库的用户的名称

**-dbpwd**

连接至数据库的用户的密码

**-infile** 包含表记录的输入 XML 文档的名称。

**-outfile**

要产生的输出 XML 文件的名称; 此文件可用作对装入程序的输入

**-method**

要用于处理输入文件的方法。该命令可以象记录不存在于数据库中那样处理输入文件（装入），或者如同输入对象的标识已存在那样处理输入文件（更新）。当某些记录不存在于数据库中而某些存在时，则使用混合方法。缺省方法是 load。

**-profile**

包含 name=value 对格式的 Java 属性的文本文件。此属性文件设置标识解析器解析标识的方式。它用于描述应当将主条目的哪些列用作对需要主行标识的表的查找。此文件定义用于外键标识查找的列名称和用于主表（例如 CATEGORY 和 PRODUCT）查询的选择谓词。可以省略此文件中具有已定义唯一索引（不包含标识）的表的条目。此参数是可选的。IdResolveKeys.properties 是缺省文件。此属性文件可以如以下示例之一所示来指定:

```
-profile WC_installdir\my_directory\ file_name.properties
```

```
-profile WC_installdir\my_directory\file_name
```

如果在当前目录中存在此文件，则可以如以下示例所示指定相同的文件:

```
-profile file_name.properties
```

如果在类路径系统环境变量中指定的目录中存在此文件，则可以如以下示例所示指定相同的文件:

```
-profile file_name
```

其中，my\_directory 是用户定义的目录，而 file\_name 是您想要使用的属性文件的名称。

关于创建和指定新属性文件以供标识解析器使用的更多信息，请参阅 WebSphere Commerce 产品和开发联机帮助。

**-poolsize**

要保留的标识数目。此参数是可选的。缺省值是 50。

**-maxerror**

将导致标识解析器终止的错误数。此参数是可选的。缺省值是 1。

**-customizer**

要使用的定制程序属性文件名。此参数是可选的。定制程序属性文件设置标识解析器起作用的方式。DB2ConnectionCustomizer.properties 是缺省文件。定制程序属性文件可以如以下示例之一所示来指定:

```
-customizer WC_installdir\my_directory\file_name.properties  
-customizer WC_installdir\my_directory\file_name
```

如果在当前目录中存在此文件，则可以如以下示例所示指定相同的文件：

```
-customizer file_name.properties
```

如果在类路径系统环境变量中指定的目录中存在此文件，则可以如以下示例所示指定相同的文件：

```
-customizer file_name
```

其中，*my\_directory* 是用户定义的目录，而 *file\_name* 是您将要使用的属性文件的名称。关于创建和指定新定制程序属性文件的更多信息，请参阅 WebSphere Commerce 产品和开发联机帮助。

#### **-schemaname**

目标数据库模式的名称。此参数是可选的。

如果在运行命令时未指定此参数，则此命令在指定了 *SchemaName* 的值的定制程序属性文件中查找 *name=value* 对。如果此对出现在属性文件中，则命令使用指定的值。如果此参数的命令行或属性文件规范都不存在，则命令缺省为数据库中 KEYS 表的模式名称。

#### **-optimize**

```
-optimize no
```

标识解析器将先跳过重复记录检查，然后将已解析的记录写到输出文件中。该选项使用户能关闭标识解析器中的优化功能。

#### **解析技术：**

根据是否使用属性文件，标识解析器使用两个或三个以下技术的组合解析标识。

- **内部别名解析**

使用内部别名标识解析时，在源 XML 文档中以别名替换唯一键（标识）。然后在 XML 文件的其它位置使用此别名来引用该元素。

必须在整个 XML 文件中一致地使用内部别名。例如，如果通讯录标识 ADDRBOOK\_ID 的别名是 @addrbook\_1，则文件中对该标识的所有外键引用都必须使用 @addrbook\_1。

请注意别名对于特定 XML 文件是瞬时的。它们不会被保存；并且如果不再引进该别名的话它无法在独立的 XML 文件中使用。而在管理控制台中的发布期间，发布将并置 XML 文件以使解析能够跨所有数据发生。

- **唯一索引解析**

标识解析器还可分析数据库模式以确定是否存在满足其需求的唯一索引。标识解析器仅当属性文件中不存在被分析的表的条目时或者在没有属性文件时查找唯一索引。如果这些条件为真，则执行唯一索引检查。唯一索引在其存在且未包含表的主键时视为有效。

## • 属性文件规范

标识解析器让您能够使用备用的 Java 属性文件来描述应当将主条目的哪些列用作对需要主行标识的表的查找。

随 WebSphere Commerce 提供的样本商店归档文件在它们的 XML 文件中使用内部别名。这使商店归档文件可以跨数据库移植。尽管唯一索引和属性文件规范技术也允许跨数据库的移植，但是用户可以随时更改唯一列，这在稍后这些技术用于标识解析时会造成问题。例如，如果用户更改了唯一列，则必须在属性文件定义中更改列名。然而，利用内部别名技术，数据库中的更改不会必然导致 XML 或属性文件中的更改。从 WebSphere Commerce 管理控制台或使用装入程序软件包进行发布期间，标识解析器会以唯一值替换别名。一旦装入数据，别名对用户是透明的。关于更多信息，请参阅第 417 页的附录 B，『创建数据』。

标识解析器使用以下过程：

- 如果输入 XML 数据具有来自主表的已具有硬编码标识（例如“12345”）的元素，则标识解析器不对该元素创建新标识。
- 如果输入 XML 数据具有来自主表的不具有标识的元素，则标识解析器查看数据库以确定是否已存在此元素的行。

在数据库中查找元素要求使用元素中的其它列构成唯一键。可以在属性文件中指定这些其它列，或者，可让标识解析器确定要使用哪些列。

- 如果正在使用属性文件且在属性文件中存在正被分析的表的条目，则标识解析器使用在属性文件中指定的列构成唯一键。
- 如果未在使用属性文件或属性文件中不存在正被分析的表的条目，则标识解析器使用唯一索引解析。

唯一索引解析使用表中任何已指定的唯一索引作为定位标识的方法。例如，MEMBER\_ID 加上 IDENTIFIER 是 CATALOG 表的唯一索引，因此可用作 CATALOGDSC 表的外键 CATALOG\_ID 的解析点。

如果存在着具有相同唯一键的行，则元素被认为已存在于数据库中；否则，它被视为一项新数据。

- 如果元素作为数据库中的行已存在，则将检索并保存其标识以便以后可以使用。否则，将由标识解析器使用 KEYS 或 SUBKEYS 表中提供的值生成新标识。
- 如果在 XML 文档中指定了元素的内部别名（例如“store\_id\_1”），则将该别名与标识关联，以便以后可以使用同一内部别名查找标识。
- 后面的 XML 文档元素（它们需要引用主表中的元素）使用内部别名（如果主表元素具有一个内部别名，例如“store\_id\_1”），或使用查找列的值（如果主表元素没有内部别名，例如“WC2001@100”）。对于上述两种情况的任一种，将使用指定的值查找实际标识，并用该标识替换值。
- 当产生了输出 XML 文档时，所有主表元素自身具有实际标识，且引用这些主表元素的所有元素使用实际标识引用它们（而不是使用上面提及的内部别名或查找列值）。这是完全解析的 XML 文档。

### 标识解析命令的方法：

标识解析命令让您选择 load、update 或 mixed 方法以处理输入文件。

### 装入方法：

标识解析器的装入方法用于为装入到数据库中的所有新记录生成新标识。

**注：** 如果为标识解析器指定装入方法，则输入文件中的记录不应已存在于数据库中。如果对标识解析器使用装入方法而源 XML 文件中的记录已存在于目标数据库中，则装入程序在您装入数据时将生成错误。标识解析器在标识解析期间将对 XML 文件中的记录指定新的主键；但是当您将数据装入数据库时，将生成错误。装入程序在处理重复记录时将不会停止；但是将报告错误且不会将重复记录装入数据库。

以下示例用于为对于数据库来说是新元素的数据元素生成标识：

- ▶ AIX ▶ Linux ▶ 400 ▶ Solaris  

```
./idresgen.sh -dbname db -dbuser user -dbpwd pwd -infile input.xml  
-outfile output.xml -method load -customizer customizer -schemaname wcsadmin
```
- ▶ Windows  

```
idresgen -dbname db -dbuser user -dbpwd pwd -infile input.xml  
-outfile output.xml -method load -customizer customizer -schemaname wcsadmin
```

**注：** 有关相应的标识解析命令或脚本的位置，请参阅第 356 页的『使用装入程序软件包命令和脚本』。

**更新方法：**

如果为标识解析器指定更新方法，则输入文件中的记录应当已存在于数据库中。标识解析器如页面 333 中所述在数据库中定位标识。如果记录不存在于数据库中，则标识解析器无法解析此记录的标识，并指示已发生错误。

以下示例用于定位已存在于数据库中的数据元素的标识:

- ▶ AIX ▶ Linux ▶ 400 ▶ Solaris

```
./idresgen.sh -dbname db -dbuser user -dbpwd pwd -infile input.xml  
-outfile output.xml -method update -customizer customizer -schemaname wcsadmin
```

- ▶ Windows

```
idresgen -dbname db -dbuser user -dbpwd pwd -infile input.xml  
-outfile output.xml -method update -customizer customizer -schemaname wcsadmin
```

**注:** 有关相应的标识解析命令或脚本的位置, 请参阅第 356 页的『使用装入程序软件包命令和脚本』。

**混合方法:**

如果输入数据文件包含已存在于数据库中的记录以及一些新的记录, 则必须使用混合方法运行标识解析器。使用此方法, 标识解析器仅当记录未存在于数据库中时为记录创建新标识。否则, 从数据库获取现有标识。以下示例用于为新数据生成标识以及定位已存在于数据库中的数据元素的标识:

- ▶ AIX ▶ Linux ▶ 400 ▶ Solaris

```
./idresgen.sh -dbname db -dbuser user -dbpwd pwd -infile input.xml  
-outfile output.xml -method mixed -customizer customizer -schemaname wcsadmin
```

- ▶ Windows

```
idresgen -dbname db -dbuser user -dbpwd pwd -infile input.xml  
-outfile output.xml -method mixed -customizer customizer -schemaname wcsadmin
```

**注:**

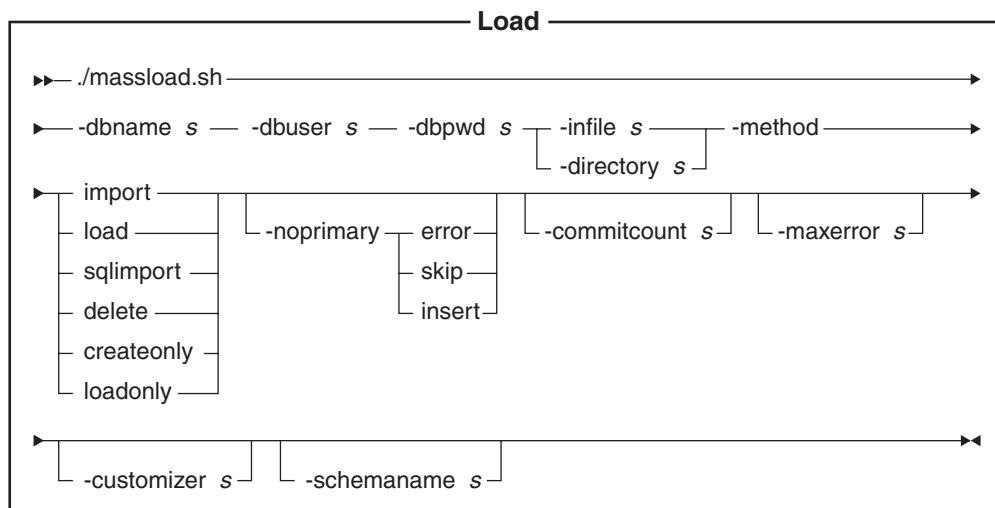
1. 有关相应的标识解析命令或脚本的位置, 请参阅第 356 页的『使用装入程序软件包命令和脚本』。
2. 混合方法是建议用于管理控制台的方法。

关于设置和定制用于运行此命令的文件的详细信息, 请参阅 WebSphere Commerce 产品和开发联机帮助。

## 装入命令

此命令将 XML 输入文件装入目标数据库。

AIX Linux 400 Solaris



注:

1. 上图主要用作命令参数的参考。
2. 指定为此命令的参数的文件名可以放置在相对或绝对路径之后。

参数值:

**-dbname**

AIX Linux Solaris 目标数据库名称

400 这是显示在关系数据库目录（WRKRDBDIRE）中的数据库名称。

**-dbuser**

AIX Linux Solaris 连接到数据库的用户名称。

400 它通常与实例用户名相同。

**-dbpwd**

连接至数据库的用户的密码

**-infile** 输入 XML 文件的名称

**-directory**

如果您如下面所述为 **-method** 参数使用除 **loadonly** 之外的任何选项，那么请使用 **-infile** 参数。如果您使用 **loadonly** 方法，那么必须用 **-directory** 参数来替换 **-infile** 参数，否则会导致错误。对于使用 **loadonly** 方法时 **-directory** 参数的值，指定 **MassLoadOutputFiles** 目录的全限定路径，该目录是使用下面所述的 **createonly** 方法创建的。

**-method**

装入程序在使用输入数据修改数据库时要使用的操作方式

**load** load 方法使用来自数据库供应商的本机装入程序。您可以对本地和远程 Oracle 数据库都使用装入方法；但装入方法只能用于本地 DB2 数据库。

▶ 400 装入方法不支持位数据或 DBCLOB 字段。

### import

此导入方法使用 import 或 update 选项（如果数据库供应商提供的话）。如果未提供 import 或 update 选项，将使用 JDBC 的 SQL 语句更新数据库。缺省为 import。

▶ 400 导入方法只能在本地数据库上使用。

### sqlimport

SQL 导入 (sqlimport) 方法可以与本地和远程数据库一同使用。

### delete

delete 方法是从数据库删除数据

### createonly

要改进实例创建期间的性能，请使用 createonly 方法。使用 createonly 方法创建大量装入数据 (MLD) 文件，而无需向数据库装入数据。使用此方法创建的文件 (.mld 和 .cmd 文件) 将放在名为 MassLoadOutputFiles 的目录中。▶ AIX ▶ Linux ▶ Solaris 该目录是作为您运行装入命令的目录（您的工作目录）下的子目录而创建的。

这样，您的工作目录必须是可写入的目录。▶ 400 该目录是作为临时目录的子目录而创建的，位于实例的根目录中。该目录的缺省位置是 WC\_userdir/instances/instance\_name/temp/MassLoadOutputFiles。

这里是使用 createonly 方法运行装入命令的示例：

```
./massload.sh -dbname mall -dbuser db2admin -dbpwd db2admin  
-infile WC_installdir/data/example.xml  
-method createonly
```

▶ 400

```
./massload.sh -dbname mall -dbuser db2admin -dbpwd db2admin  
-infile WC_userdir/data/example.xml  
-method createonly
```

您可以稍后使用本机数据库装入实用程序来装入 MLD 文件，该文件是通过使用如下所述的 loadonly 方法运行装入命令而在 WebSphere Commerce 数据库中创建的。

**注：**程序会获得关于本机数据库装入实用程序的信息，该实用程序是数据库产品从定制程序属性文件中使用的。

### loadonly

使用 loadonly 方法可装入使用前面所述的 createonly 方法创建的 MLD 文件。在使用 loadonly 方法时，您还必须使用 -directory 参数，否则会导致错误。

**注：**如果您正在使用除 loadonly 之外的任何方法，那么 -directory 参数会替换您指定的 -infile 参数。

对于 `-directory` 参数的值，您必须指定 `MassLoadOutputFiles` 目录的全限定路径，该目录是使用 `createonly` 方法创建的。

这里是使用 `loadonly` 方法（以及必需的 `-directory` 参数）运行装入命令的示例：

```
./massload.sh -dbname mall -dbuser db2admin -dbpwd db2admin
-method loadonly -directory WC_installdir/bin/MassLoadOutputFiles
-schemaname wcsadmin
```

在使用此方法运行装入命令时，请始终通过使用 `-schemaname` 参数指定目标数据库模式的名称。否则，程序会使用在最初创建 `MassLoadOutputFiles` 目录及其文件时获得的数据库模式名。在使用 `loadonly` 方法时，错误和其它消息将保存在以 `.log` 为扩展名的文件中。这些日志文件会写到为 `-directory` 参数指定的 `MassLoadOutputFiles` 目录。

`loadonly` 方法只用于创建实例。如果在其它时间使用它，结果会很不理想。

### **-noprimary**

当输入文件中记录的主键丢失时，装入程序必须执行的操作

- `error` 选项指示它应当将丢失主键报告为错误并终止。
- `skip` 选项跳过输入文件中没有主键的任何记录。
- 插入选项尝试插入或删除数据。

此参数是可选的。缺省操作是 `error`。

### **-commitcount**

在使用操作的 `SQL import` 方法时，数据库提交发生前处理的记录数目。此参数是可选的。缺省数目是 1。

### **-maxerror**

操作的 `SQL import` 方法中的错误数目，在这些错误发生后装入程序将终止。此参数是可选的。缺省值是 1。

### **-customizer**

要使用的定制程序属性文件名。此参数是可选的。定制程序属性文件设置装入程序运作的方式。缺省文件为：

```
▶ AIX ▶ Linux ▶ Solaris MassLoadCustomizer.properties
```

```
▶ 400 ISeries_LODWCSDTA_Customizer.properties
```

如果已配置实例来使用工具箱驱动程序，则使用为工具箱驱动程序提供的 `Toolbox_LODWCSDTA_Customizer` 定制程序文件。还必须指定 `-dbname` 参数的主机名。以下是调用 `massload.sh` 脚本的示例：

```
./massload.sh -dbname MY.HOSTNAME.CA -dbuser instance -dbpwd mypass
-method sqlimport -customizer Toolbox_LODWCSDTA_Customizer
-infile /path/file.xml
```

定制程序属性文件可以如以下示例所示来指定：

```
▶ AIX ▶ Linux ▶ Solaris
```

```
-customizer WC_installdir/my_directory/file_name.properties
```



-customizer *WC\_installdir/my\_directory/file\_name*

▶ 400

-customizer *WC\_userdir/my\_directory/file\_name.properties*

-customizer *WC\_userdir/my\_directory/file\_name*

如果在当前目录中存在此文件，则可以如以下示例所示指定相同的文件：

-customizer *file\_name.properties*

如果在类路径系统环境变量中指定的目录中存在此文件，则可以如以下示例所示指定相同的文件：

-customizer *file\_name*

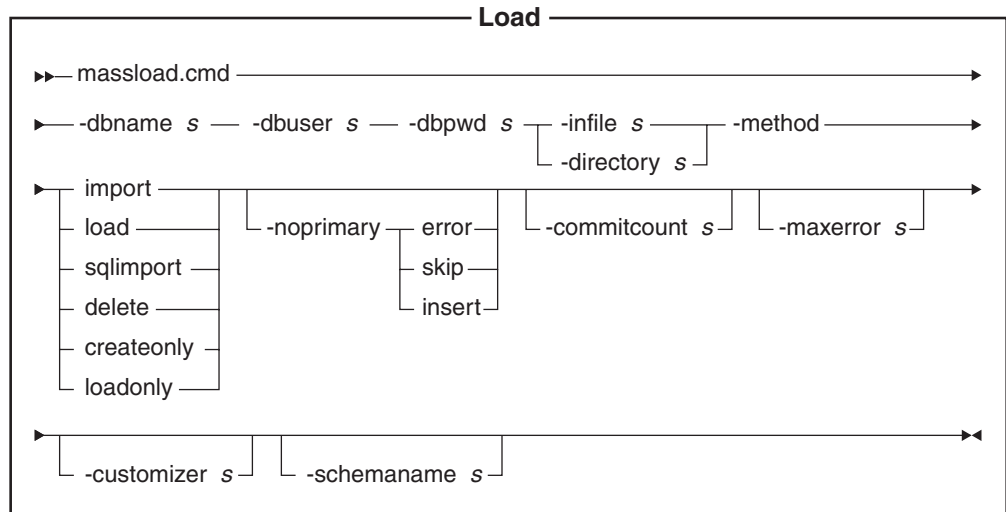
其中，*my\_directory* 是用户定义的目录，而 *file\_name* 是您将要使用的属性文件的名称。有关创建和指定新定制程序属性文件的更多信息，请参阅 **WebSphere Commerce** 产品和开发联机帮助。

### -schemaname

目标数据库模式的名称。此参数是可选的。

如果在运行命令时未指定此参数，则此命令在指定了 **SchemaName** 的值的定制程序属性文件中查找 **name=value** 对。如果此对出现在属性文件中，则命令使用指定的值。如果此参数的命令行或属性文件规范都不存在，则命令缺省为数据库中 **KEYS** 表的模式名称。▶ 400 如果 **-schemaname** 参数的命令行或属性文件规范都不存在，则命令缺省为 **-dbuser** 参数的值。

▶ Windows



注：指定为此命令的参数的文件名可以放置在相对或绝对路径之后。

参数值：

### -dbname

目标数据库的名称

**-dbuser**

连接至数据库的用户的名称

**-dbpwd**

连接至数据库的用户的密码

**-infile** 输入 XML 文件的名称

**-directory**

如果您如下面所述为 **-method** 参数使用除 **loadonly** 之外的任何选项，那么请使用 **-infile** 参数。如果您使用 **loadonly** 方法，那么必须用 **-directory** 参数来替换 **-infile** 参数，否则会导致错误。对于使用 **loadonly** 方法时 **-directory** 参数的值，指定 **MassLoadOutputFiles** 目录的全限定路径，该目录是使用下面所述的 **createonly** 方法创建的。

**-method**

装入程序在将数据插入到数据库时要使用的操作方式

**load** **load** 方法使用来自数据库供应商的本机装入程序。您可以对本地和远程 Oracle 数据库都使用装入方法；但装入方法只能用于本地 DB2 数据库。尽管导入方法可以用于将数据装入本地或远程数据库，但是它通常用于将数据装入到远程 DB2 数据库。

**import**

此导入方法使用 **import** 或 **update** 选项（如果数据库供应商提供的话）。如果未提供 **import** 或 **update** 选项，将用使用 JDBC 的 SQL 语句更新数据库。缺省为 **import**。

**sqlimport**

SQL 导入（**sqlimport**）方法可以与本地和远程数据库一同使用。

**delete**

**delete** 方法从数据库删除数据。

**createonly**

要改进实例创建期间的性能，请使用 **createonly** 方法。使用 **createonly** 方法创建大量装入数据（MLD）文件，而无需向数据库装入数据。使用此方法创建的文件（**.mld** 和 **.cmd** 文件）将放在名为 **MassLoadOutputFiles** 的目录中。该目录是作为您运行装入命令的目录（您的工作目录）下的子目录而创建的。这样，您的工作目录必须是可写入的目录。

这里是使用 **createonly** 方法运行装入命令的示例：

```
massload -dbname mall -dbuser db2admin -dbpwd db2admin -infile
WC_installdir\data\example.xml
-method createonly
```

您可以稍后使用本机数据库装入实用程序来装入 MLD 文件，该文件是通过使用如下所述的 **loadonly** 方法运行装入命令而在 WebSphere Commerce 数据库中创建的。

**注：** 程序会获得关于本机数据库装入实用程序的信息，该实用程序是数据库产品从定制程序属性文件中使用的。

## loadonly

使用 `loadonly` 方法可装入使用前面所述的 `createonly` 方法创建的 MLD 文件。在使用 `loadonly` 方法时，您还必须使用 `-directory` 参数，否则会导致错误。

**注：**如果您正在使用除 `loadonly` 之外的任何方法，那么 `-directory` 参数会替换您指定的 `-infile` 参数。

对于 `-directory` 参数的值，您必须指定 `MassLoadOutputFiles` 目录的全限定路径，该目录是使用 `createonly` 方法创建的。

这里是使用 `loadonly` 方法（以及必需的 `-directory` 参数）运行装入命令的示例：

```
massload -dbname mall -dbuser db2admin -dbpwd db2admin -method  
loadonly -directory WC_install_dir\bin\MassLoadOutputFiles  
-schemaname wcsadmin
```

在使用此方法运行装入命令时，请始终通过使用 `-schemaname` 参数指定目标数据库模式的名称。否则，程序会使用在最初创建 `MassLoadOutputFiles` 目录及其文件时获得的数据库模式名。在使用 `loadonly` 方法时，错误和其它消息将保存在以 `.log` 为扩展名的文件中。这些日志文件会写到为 `-directory` 参数指定的 `MassLoadOutputFiles` 目录。

`loadonly` 方法只用于创建实例。如果在其它时间使用它，结果会很不理想。

## -noprimary

当输入文件中记录的主键丢失时，装入程序必须执行的操作

- `error` 选项指示它应当将丢失主键报告为错误并终止。
- `skip` 选项跳过输入文件中没有主键的任何记录。
- `insert` 选项尝试处理（插入或删除）数据。

此参数是可选的。缺省操作是 `error`。

## -commitcount

在使用操作的 `SQL import` 方法时，数据库提交发生前处理的记录数目。此参数是可选的。缺省数目是 1。

## -maxerror

操作的 `SQL import` 方法中的错误数目，在这些错误发生后装入程序将终止。此参数是可选的。

## -customizer

要使用的定制程序属性文件名。此参数是可选的。定制程序属性文件设置装入程序运作的方式。缺省文件为 `MassLoadCustomizer.properties`。定制程序属性文件可以如以下示例所示来指定：

```
-customizer WC_install_dir\my_directory\file_name.properties
```

如果在当前目录中存在此文件，则可以如以下示例所示指定相同的文件：

```
-customizer file_name.properties
```

如果在类路径系统环境变量中指定的目录中存在此文件，则可以如以下示例所示指定相同的文件：

```
-customizer file_name
```

其中，*my\_directory* 是用户定义的目录，而 *file\_name* 是您将要使用的属性文件的名称。关于创建和指定新定制程序属性文件的更多信息，请参阅最新版本的 WebSphere Commerce 产品和开发联机帮助。

### **-schemaname**

目标数据库模式的名称。此参数是可选的。

如果在运行命令时未指定此参数，则此命令在指定了 SchemaName 的值的定制程序属性文件中查找 name=value 对。如果此对出现在属性文件中，则命令使用指定的值。如果此参数的命令行或属性文件规范都不存在，则命令缺省为数据库中 KEYS 表的模式名称。

### **装入命令的方法：**

在装入数据之前，应当确定可用的处理方法中哪一种将产生最佳结果。


#### **装入方法：**

在任何以下情况中，请考虑装入方法：



- 源数据是干净的，数据库不包含任何数据

**注：**干净的数据是不违反正装入到的表的任何约束的数据。





- 源数据是干净的，数据库不包含正被装入的数据
- 源数据是干净的，一个或多个目标表不包含主键，数据库不包含正被装入的数据
- 数据库是本地的 DB2 数据库
- 数据库是本地或远程 Oracle 数据库
- 当发生装入时，数据库未由其它用户或应用程序访问

 使用装入方法时，数据被装入到数据库中。如果数据已经存在，则命令由于重复的键错误而失败，且显示“重复错误”消息。

对于使用装入方法存在以下限制：

-  装入方法无法插入或更新位数据字段或 DBCLOB 字段中的数据。只有新记录才会插入到数据库中；现有记录将导致错误。
-  利用装入方法，只有新的记录才会插入到数据库中；不会更新现有的记录。装入方法仅可用于本地（而不是远程）DB2 数据库。

#### **导入方法：**

    通过使用 DB2 的导入方法，数据也可以装入到数据库中。如果数据已经存在，则不删除它，而是用新值更新它。在任何以下情况中，请考虑此方法：

- 数据库管理是 DB2
- 您不知道数据是否干净
- 您必须在列级别更新许多组同类数据
- 正将数据导入其中的所有表都具有主键

▶ 400 使用导入方法时，数据也被装入到数据库中。如果数据已经存在，则不删除它，而是用新值更新它。在任何以下情况中，请考虑此方法：

- 您不知道数据是否干净
- 数据已经存在于数据库中
- 正将数据导入其中的所有表都具有主键

对于使用导入方法存在以下限制：

- 数据库管理系统必须是 DB2 以便使用导入方法。

- ▶ 400 导入方法无法插入或更新位数据字段或 DBCLOB 字段中的数据，且只能在本地数据库上使用
- 使用导入方法时，装入程序仅插入或更新已对其定义了主键的表；导入方法无法在不具有主键的表中插入或更新数据。如果输入记录仅对是主键的列具有值，则拒绝该记录。

*SQL 导入方法：*

使用 SQL 导入方法时，JDBC 或 SQL 用于向数据库更新或插入数据。如果数据尚未存在则插入数据，并更新现有的数据。在任何以下情况中，请考虑此方法：

- 您正在更新现有的数据并需要列级别的更新
- 一些数据不干净
- 数据库不是本地的

**注：**如果正在使用产品顾问搜索空间同步，则对于装入数据必须使用 SQL 导入方法。

*删除方法：*

删除方法用于从数据库中删除在输入 XML 文档中的数据。元素必须包含主键的值或表的唯一索引。如果正被删除的数据具有另一表中的数据，而该表又以启用了“级联删除”的方式依赖于这些数据，则还将删除相关的数据。

*用于 AIX、Linux、Solaris 和 Windows 系统的 createonly 方法：*

要改进实例创建期间的性能，请使用 createonly 方法。使用 createonly 方法创建大量装入数据（MLD）文件，而无需向数据库装入数据。您可以稍后使用本机数据库装入实用程序来装入 MLD 文件，该文件是通过使用 loadonly 方法来运行装入命令而在 WebSphere Commerce 数据库中创建的。

*用于 AIX、Linux、Solaris 和 Windows 系统的 loadonly 方法：*

使用 loadonly 方法来装入使用 createonly 方法创建的 MLD 文件。loadonly 方法只用于创建实例。如果在其它时间使用它，结果会很不理想。

*比较方法：*

- **SQL 导入和装入方法的比较**

SQL 导入方法检查数据一致性（包括外键引用），并让您能够更新现有的数据。而装入方法则不。

- **导入和 SQL 导入方法的比较**

导入和 SQL 导入方法执行类似的功能。导入方法通常更快，但是它需要磁盘空间来存放临时文件。

导入方法仅可插入或更新对其定义了主键的表；而 SQL 导入方法则不要求对表定义了主键。

- **基于所使用数据库产品的方法比较**

导入和装入方法使用对 DB2 优化的本机实用程序，而 SQL 导入方法使用 JDBC 调用（通用于许多数据库产品）。

#### 性能注意事项:

当使用装入程序将大文档装入数据库时，请考虑以下事项:

- **Java 虚拟机 (JVM) 堆大小**

缺省情况下，分配给 JVM 堆的最大内存是 64 MB。如果未增加此数值，则在装入过程中 JVM 最终可发生内存不足。可以通过在 Java 命令中使用 JVM `-mx` 选项来更改分配给 Java 堆的最大内存。

- **跟踪记录**

装入大的 XML 文档时，跟踪记录程序可耗尽 JVM 堆。跟踪信息最常用于在运行失败的情况下对运行进行调试。如果对装入过程的跟踪是不必要的，则应当关闭跟踪。当关闭跟踪时将获得显著的性能提高。通过修改记录配置 XML 文档来关闭跟踪。关于修改记录配置 XML 文档的信息，请参阅 WebSphere Commerce 产品和开发联机帮助。

- **提交计数**

装入程序在以 SQL 导入方式运作时的缺省提交计数是 1。因此，缺省情况下每次对数据库进行更新或插入时将提交事务。为了改善装入程序在装入大文档时的性能，应增加提交计数。考虑 `input.xml` 文件的大小后，您可以使用大于文件中记录数的提交计数。这将在发生错误时启用整个 `input.xml` 文件的回滚。

使用装入命令的 `-commitcount count` 选项来更改装入程序的提交计数（其中 `count` 是提交事务前执行的语句数）。

- **记录配置**

通常装入数据时进展缓慢可产生于以下情况之一:

- 调用装入程序的用户不具有许可权以写入在记录配置文档中所指定的目录或更新在记录配置文档中所指定的文件。
- 在记录配置文档中指定为文件位置的目录不存在。
- 在记录配置文档中指定为文件位置的驱动器不具有足够的空间。

当更正所有这些问题时，可能需要通过修改记录配置文档（缺省情况下是 `WCALoggerConfig.xml`）来更改文件的指定位置。关于修改记录配置 XML 文档的信息，请参阅 WebSphere Commerce 产品和开发联机帮助。

关于设置和定制用于运行此命令的文件的详细信息，请参阅 WebSphere Commerce 产品和开发联机帮助。

## 用于转换和抽取数据的装入程序软件包命令

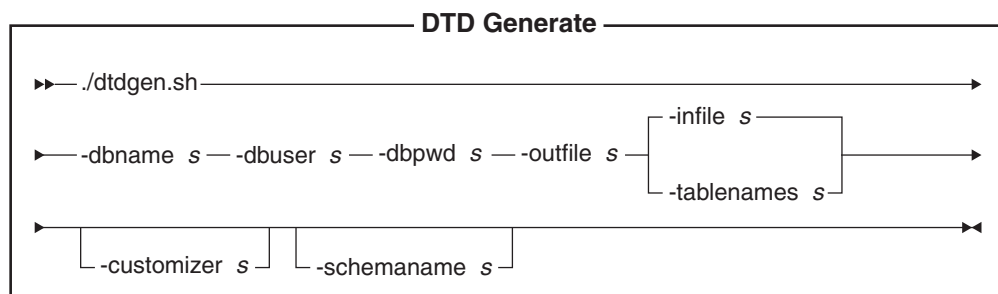
### DTD 生成命令

此命令创建用于装入程序软件包的 DTD。在整个数据装入过程中将使用此 DTD。根据调用命令的方式，DTD 生成程序可单独生成一个 DTD 或随 XML 模式一起生成一个 DTD。

DTD 生成程序可基于 WebSphere Commerce 数据库模式创建 DTD。如果使用随样本商店归档文件提供的 DTD 且未修改数据库模式，则无需使用 DTD 生成程序生成 DTD。所提供的 DTD 位于 `WC_installdir/xml/sar` 目录中。

建议您使用所提供的 DTD。但是，如果定制数据库模式，则必须编辑所提供的 DTD 以与所作更改相匹配，或者创建新的 DTD。

▶ AIX   ▶ Linux   ▶ 400   ▶ Solaris




注:

1. 上图主要用作命令参数的参考。
2. 指定为此命令的参数的文件名可以放置在相对或绝对路径之后。

参数值:


### -dbname

   目标数据库名称

 这是显示在关系数据库目录（WRKRDBDIRE）中的数据库名称

### -dbuser

   连接到数据库的用户名称

 它通常与实例用户名相同

### -dbpwd

连接至数据库的用户的密码

### -outfile

输出 DTD 文件的名称（首选使用 .dtd 扩展名）

**-infile** 每行包含了数据库表名称的输入文件的名称

### -tablenames

以逗号分隔且以引号（"）括起的表的名称

### -customizer

要使用的定制程序属性文件名。此参数是可选的。定制程序属性文件设置 DTD 生成程序起作用的方式。缺省文件为:

   DB2ConnectionCustomizer.properties

 ISeries\_GENWCSDTD\_Customizer.properties

如果已配置实例来使用工具箱驱动程序，则使用为工具箱驱动程序提供的 Toolbox\_GENWCSDTD\_Customizer 定制程序文件。还必须指定 -dbname 参数的主机名。以下是调用 dtdgen.sh 脚本的示例:

```
./dtdgen.sh -dbname MY.HOSTNAME.CA -dbuser instance -dbpwd mypass
-outfile /path/out.dtd
  method sqlimport -customizer Toolbox_GENWCSDTD_Customizer
-infile /path/file.xml
```

定制程序属性文件可以如以下示例所示来指定:

-customizer *WC\_installdir/my\_directory/file\_name*.properties

-customizer *WC\_installdir/my\_directory/file\_name*



-customizer *WC\_userdir/my\_directory/file\_name*.properties

-customizer *WC\_userdir/my\_directory/file\_name*

如果在当前目录中存在此文件，则可以如以下示例所示指定相同的文件:

-customizer *file\_name*.properties

如果在类路径系统环境变量中指定的目录中存在此文件，则可以如以下示例所示指定相同的文件:



-customizer *file\_name*

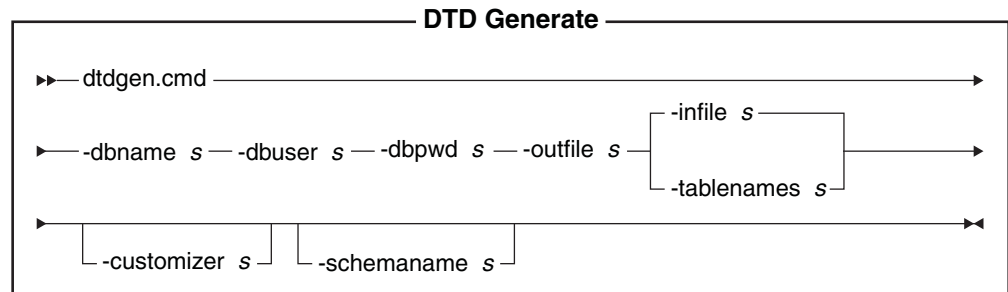
其中, *my\_directory* 是用户定义的目录, 而 *file\_name* 是您将要使用的属性文件的名称。关于创建和指定新定制程序属性文件的更多信息, 请参阅 WebSphere Commerce 产品和开发联机帮助。

### **-schemaname**

目标数据库模式的名称。此参数是可选的。

如果在运行命令时未指定此参数, 则此命令在指定了 SchemaName 的值的定制程序属性文件中查找 name=value 对。如果此对出现在属性文件中, 则命令使用指定的值。如果此参数的命令行或属性文件规范都不存在, 则命令缺省为数据库中表的模式所有者。 400 如果此参数的命令行或属性文件规范都不存在, 则命令缺省为数据库用户的名称。

Windows



注: 指定为此命令的参数的文件名可以放置在相对或绝对路径之后。

参数值:

### **-dbname**

目标数据库的名称

### **-dbuser**

连接至数据库的用户的名称

### **-dbpwd**

连接至数据库的用户的密码

### **-outfile**

输出 DTD 文件的名称

**-infile** 每行包含了数据库表名称的输入文件的名称

### **-tablenames**

以逗号分隔的表的名称

### **-customizer**

要使用的定制程序属性文件名。此参数是可选的。定制程序属性文件设置 DTD 生成程序起作用的方式。DB2ConnectionCustomizer.properties 是缺省文件。定制程序属性文件可以如以下示例所示来指定:

```
-customizer WC_installdir\my_directory\file_name.properties
```

```
-customizer WC_installdir\my_directory\file_name
```

如果在当前目录中存在此文件，则可以如以下示例所示指定相同的文件：

```
-customizer file_name.properties
```

如果在类路径系统环境变量中指定的目录中存在此文件，则可以如以下示例所示指定相同的文件：

```
-customizer file_name
```

其中，*my\_directory* 是用户定义的目录，而 *file\_name* 是您将要使用的属性文件的名称。关于创建和指定新定制程序属性文件的更多信息，请参阅 WebSphere Commerce 产品和开发联机帮助。

### **-schemaname**

目标数据库模式的名称。此参数是可选的。如果在运行命令时未指定此参数，则此命令在指定了 SchemaName 的值的定制程序属性文件中查找 name=value 对。如果此对出现在属性文件中，则命令使用指定的值。如果此参数的命令行或属性文件规范都不存在，则命令缺省为数据库中表的模式所有者。

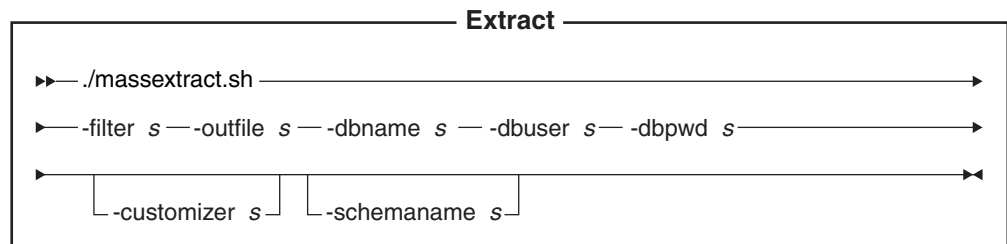
关于设置和定制用于运行此命令的文件的详细信息，请参阅 WebSphere Commerce 产品和开发联机帮助。

## **抽取命令**

此命令以 XML 文件格式从数据库中抽取选定的数据子集。

要使用抽取程序从数据库抽取数据，必须使用抽取过滤文件指定要抽取的数据。使用的抽取过滤取决于希望抽取的数据的类型。

**AIX** **Linux** **400** **Solaris**



注：

1. 上图主要用作命令参数的参考。
2. 指定为此命令的参数的文件名可以放置在相对或绝对路径之后。

参数值：

**-filter** 抽取过滤文件的名称

**-outfile**

将存储抽取后的数据的输出 XML 文件的名称

**-dbname**

**AIX** **Linux** **Solaris** 正被抽取数据的数据库的名称

**400** 这是显示在关系数据库目录（WRKRDBDIRE）中的数据库名称

**-dbuser**

▶ AIX ▶ Linux ▶ Solaris 正被抽取数据的数据库的数据库用户名

▶ 400 它通常与实例用户名相同

### -dbpwd

从其中抽取数据的数据库的用户名关联密码

### -customizer

要使用的定制程序属性文件名。定制程序属性文件设置抽取程序起作用的方式。缺省文件为:

▶ AIX ▶ Linux ▶ Solaris DB2ConnectionCustomizer.properties

▶ 400 ISeries\_EXTWCSDTA\_Customizer.properties 如果已配置实例来使用工具箱驱动程序, 则使用为工具箱驱动程序提供的 Toolbox\_EXTWCSDTA\_Customizer 定制程序文件。还必须指定 `-dbname` 参数的主机名。以下是调用 `massextract.sh` 脚本的示例:

```
./massextract.sh -dbname MY.HOSTNAME.CA -dbuser instance -dbpwd mypass  
-filter/path/filter.xml  
-outfile /path/file.xml -customizer Toolbox_EXTWCSDTA_Customizer
```

定制程序属性文件可以如以下示例所示来指定:

▶ AIX ▶ Linux ▶ Solaris DB2ConnectionCustomizer.properties

```
-customizer WC_installdir/my_directory/file_name.properties  
-customizer WC_installdir/my_directory/file_name
```

▶ 400

```
-customizer WC_userdir/my_directory/file_name.properties  
-customizer WC_userdir/my_directory/file_name
```

如果在当前目录中存在此文件, 则可以如以下示例所示指定相同的文件:

```
-customizer file_name.properties
```

如果在类路径系统环境变量中指定的目录中存在此文件, 则可以如以下示例所示指定相同的文件:

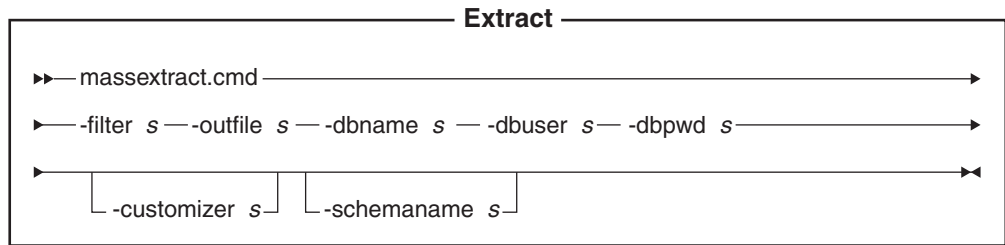
```
-customizer file_name
```

其中, `my_directory` 是用户定义的目录, 而 `file_name` 是您将要使用的属性文件的名称。关于创建和指定新定制程序属性文件的更多信息, 请参阅 [WebSphere Commerce 产品和开发联机帮助](#)。

### -schemaname

从中抽取数据的数据库模式的名称。此参数是可选的。

如果在运行命令时未指定此参数, 则此命令在指定了 `SchemaName` 的值的定制程序属性文件中查找 `name=value` 对。如果此对出现在属性文件中, 则命令使用指定的值。如果此参数的命令行或属性文件规范都不存在, 则命令缺省为数据库中表的模式名。▶ 400 如果 `-schemaname` 参数的命令行或属性文件规范都不存在, 则命令缺省为 `-dbuser` 参数的值。



注：指定为此命令的参数的文件名可以放置在相对或绝对路径之后。

#### 参数值:

**-filter** 抽取过滤文件的名称

#### **-outfile**

将存储抽取后的数据的输出 XML 文件的名称

#### **-dbname**

从其中抽取数据的数据库的名称

#### **-dbuser**

从其中抽取数据的数据库的数据库用户名

#### **-dbpwd**

从其中抽取数据的数据库的用户名关联密码

#### **-customizer**

要使用的定制程序属性文件名。定制程序属性文件设置抽取程序起作用的方式。DB2ConnectionCustomizer.properties 是缺省文件。定制程序属性文件可以如以下示例所示来指定:

```
-customizer WC_installdir\my_directory\file_name.properties
```

```
-customizer WC_installdir\my_directory\file_name
```

如果在当前目录中存在此文件，则可以如以下示例所示指定相同的文件:

```
-customizer file_name.properties
```

如果在类路径系统环境变量中指定的目录中存在此文件，则可以如以下示例所示指定相同的文件:

```
-customizer file_name
```

其中，*my\_directory* 是用户定义的目录，而 *file\_name* 是您将要使用的属性文件的名称。关于创建和指定新定制程序属性文件的更多信息，请参阅 WebSphere Commerce 产品和开发联机帮助。

#### **-schemaname**

从中抽取数据的数据库模式的名称。此参数是可选的。

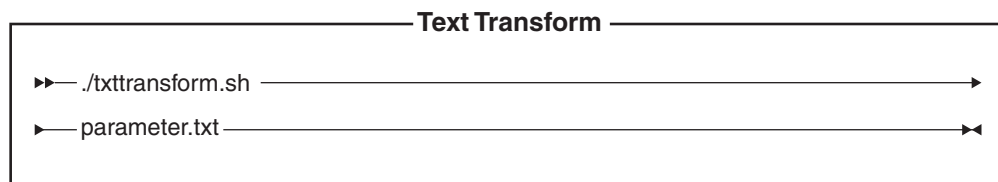
如果在运行命令时未指定此参数，则此命令在指定了 SchemaName 的值的定制程序属性文件中查找 name=value 对。如果此对出现在属性文件中，则命令使用指定的值。如果此参数的命令行或属性文件规范都不存在，则命令缺省为数据库中表的模式名。

关于此命令的更多信息，请参阅 WebSphere Commerce 产品和开发联机帮助。

## 文本转换命令

此命令在字符定界的变量格式与 XML 格式之间转换数据。

▶ AIX ▶ Linux ▶ 400 ▶ Solaris



注：上图主要用作命令参数的参考。

### 参数值：

在参数文件（*parameter.txt*）中指定了以下值且以逗号作了分隔：

#### 输入文件

要转换的文件的名称

#### 模式文件

要用于转换的 XML 模式文件的名称

#### 输出文件

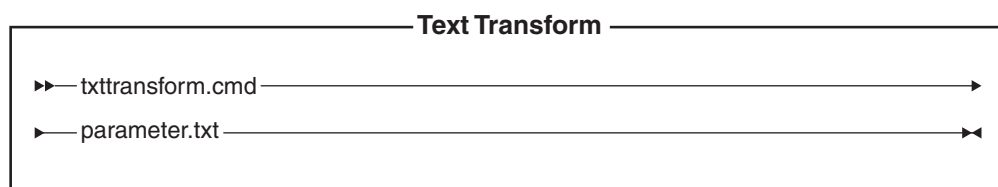
将存储转换后的数据的输出文件的名称

#### 转换方法

要用于向输出文件添加数据的方法。如果要创建新文件，则指定**创建**；如果要在现有的数据文件后附加输出数据，则指定**附加**。

此文件也称为“清单”或“命令”文件。它可包含多行，每行有四个参数。

▶ Windows



### 参数值：

在参数文件（*parameter.txt*）中指定了以下值且以逗号作了分隔：

#### 输入文件

要转换的文件的名称

#### 模式文件

要用于转换的 XML 模式文件的名称

#### 输出文件

将存储转换后的数据的输出文件的名称

## 转换方法

要用于向输出文件添加数据的方法。如果要创建新文件，则指定**创建**；如果在现有的数据文件后附加输出数据，则指定**附加**。

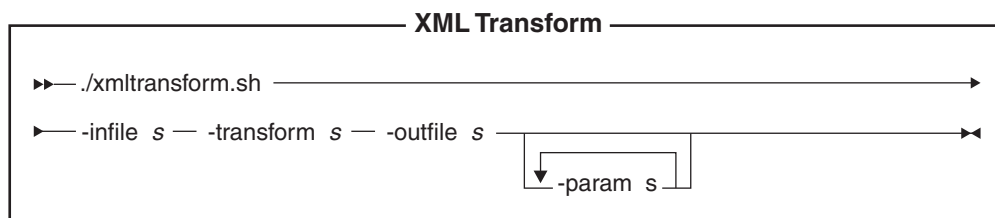
**注：**此文件也称为“清单”或“命令”文件。它可包含多行，每行有四个参数。

关于此命令的更多信息，请参阅 WebSphere Commerce 产品和开发联机帮助。

## XML 转换命令

此命令将 XML 文件转换为备用 XML 格式。

► AIX ► Linux ► 400 ► Solaris



**注：**

1. 上图主要用作命令参数的参考。
2. 指定为此命令的参数的文件名可以放置在相对或绝对路径之后。

**参数值：**

**-infile** 要转换的文件的名称

**-transform**

转换 XSL 映射规则文件的名称

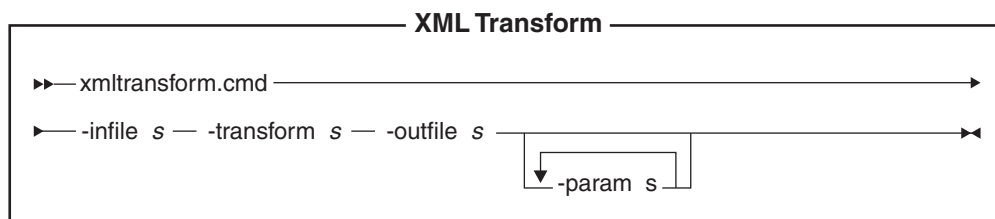
**-outfile**

将存储转换后的数据的输出 XML 文件的名称

**-param**

要传递给 XSL 映射规则文件的参数。**此参数是可选的。**此参数可以被指定多次以传递多个 name=value 对。

► Windows



**注：**指定为此命令的参数的文件名可以放置在相对或绝对路径之后。

**参数值：**

**-infile** 要转换的文件的名称

**-transform**

转换 XSL 映射规则文件的名称

**-outfile**

将存储转换后的数据的输出 XML 文件的名称

**-param**

要传递给 XSL 映射规则文件的参数。**此参数是可选的。**此参数可以被指定多次以传递多个 name=value 对。

关于此命令的更多信息，请参阅 WebSphere Commerce 产品和开发联机帮助。

## 装入程序软件包命令的相关工具

### 文本转换工具

文本转换工具有助于使用“文本转换”命令来处理字符定界的变量格式和 XML 格式之间的数据转换。提供了以下视图：

1. “文本模式编辑”视图让您能够创建和修改要用于转换的 XML 模式文件。
2. “转换命令编辑”视图让您能够创建和修改用于运行转换过程的实际命令。
3. “转换命令过程”视图让您能够启动转换过程。

关于此工具的更多信息，请参阅 WebSphere Commerce 产品和开发联机帮助。

### XSL 编辑器

XML 转换程序使用 XSL 定义将一个 XML 文件转换为另一 XML 文件的规则。XSL 编辑器中的映射功能提供了可视界面，您可使用此界面建立从源 DTD 中的元素至目标 DTD 中的元素的关联。给定了两个 DTD，您可开发 XML 规则，该规则确定如何将符合第一个（源）DTD 的 XML 文件转换为符合第二个（目标）DTD 的文件。

关于此工具的更多信息，请参阅 WebSphere Commerce 产品和开发联机帮助。



---

## 装入商店数据

此部分提供了如何使用装入程序软件包命令行实用程序将商店数据装入 WebSphere Commerce Server 数据库的示例。

### 注:

1. 此部分中的示例在 Windows 环境中执行。关于在其它环境中运行这些命令的信息, 请参阅 WebSphere Commerce 产品和开发联机帮助。
2. 尽管装入程序软件包命令行实用程序支持 DB2、DB2 iSeries 版, 以及 Oracle 数据库, 但是以下示例中仅包含了 DB2 的命令和选项。如果正在使用的数据库不是 DB2, 请确保如 WebSphere Commerce 产品和开发联机帮助中所述修改定制程序属性文件。

关于 WebSphere Commerce 数据库有用资源组的装入过程的信息, 请参阅第 367 页的第 38 章, 『装入 WebSphere Commerce 数据库有用资源组』和第 373 页的『装入数据库有用资源组』。


## 使用装入程序软件包命令和脚本

要运行装入程序软件包命令，请使用 WebSphere Commerce 中的 `WC_installdir/bin` 目录中提供的脚本或目录。

脚本和命令如下：

- ▶ AIX ▶ Linux ▶ 400 ▶ Solaris
  - dtdgen.sh** DTD 生成外壳程序脚本
  - idresgen.sh** 标识解析外壳程序脚本
  - massload.sh** 装入外壳程序脚本
  - massextract.sh** 抽取外壳程序脚本
  - txttransform.sh** 文本转换外壳程序脚本
  - xmltransform.sh** XML 转换外壳程序脚本
- ▶ Windows
  - dtdgen.cmd** DTD 生成命令
  - idresgen.cmd** 标识解析命令
  - massload.cmd** 装入命令
  - massextract.cmd** 抽取命令
  - txttransform.cmd** 文本转换命令
  - xmltransform.cmd** XML 转换命令

## 解析标识的示例

本部分中描述的标识解析的示例使用  “多乐五金店” 样本商店的商店有用资源文件。

因为此示例基于将新数据装入到 WebSphere Commerce Server 数据库中，所以我们将使用装入方法。

如果您稍后需要修改 XML 文档内的某些元素，您可以使用更新方法来原因。更新方法应比装入方法运行更快，因为使用更新方法不分配新标识。使用更新方法，将执行数据库查询以定位标识并在未找到标识的情况下报告错误。关于此过程如何工作的更多信息，请参阅从第 333 页开始的讨论。

如果输入 XML 文件包含已存在于数据库中的元素，也包含未存在于数据库中的元素，则使用混合方法。使用混合方法时，首先执行数据库查找，并在未找到记录的情况下将标识指定给元素。如果不能确定，请使用混合方法。尽管装入和更新方法提供了比混合方法更快的性能，但是通过使用混合方法产生的已解析 XML 文件更有可能做到装入无错误。

关于标识解析器如何工作的讨论，请参阅第 333 页的『标识解析命令的方法』。

### 用内部别名在 XML 文件中解析标识

要在将数据装入到 WebSphere Commerce Server 数据库之前通过使用内部别名来解析标识，请按以下示例所示运行标识解析命令。

**注：**此示例假设 WebSphere Commerce 安装在 Windows 机器上。如果您将 WebSphere Commerce 安装在不同的操作系统上，请用您的操作系统的相应值来替代。

#### 1. 创建工作目录。

对于此示例，请创建 `WC_installdir/test` 目录。

**注：**如果不使用 `WC_installdir/test` 作为工作目录，则用您实际使用的工作目录的名称和路径来替代在本章余下部分中显示的示例中的 `WC_installdir/test`。

#### 2. 确保输入 XML 文件和所有被引用的 DTD 文件位于标识解析器可找到的位置。

对于此示例，请执行以下操作：

##### a. 从 Windows 命令提示符，输入以下命令：

```
copy WC_installdir\samplstores\  
B2BDirect\B2BDirect.sar WC_installdir\test
```

这就将 `B2BDirect.sar` 文件复制到 `WC_installdir\test`。

- b. 从 Windows 命令提示符，输入以下命令：

```
cd to WC_installdir\test
```

- c. 请执行以下操作之一：

- 如果您已安装 Java，请从 Windows 命令提示符下输入以下命令：

```
jar -xvf B2BDirect.sar
```

这就将商家直销“多乐五金”样本商店 XML 文件抽取到 drive:\WebSphere\CommerceServer55\test 中。

- 使用任何最新的解压缩程序产品（例如 WinZip 或 PKZIP）将 WC\_installdir\samplstores\B2BDirect\B2BDirect.sar 的完整内容抽取到 WC\_installdir\test。

这就将  “多乐五金”样本商店 XML 文件抽取到 WC\_installdir\test\WEB-INF\stores\BusinessDirect\data\ToolTech\data 中。

- d. 从 Windows 命令提示符，输入以下命令：

```
copy WC_installdir\xml\sar\store.dtd  
WC_installdir\test\WEB-INF\stores\BusinessDirect\data\ToolTech\data
```

这就将 store.dtd 文件复制到 WC\_installdir\test\WEB-INF\stores\BusinessDirect\data\ToolTech\data。

- e. 从 Windows 命令提示符，输入以下命令：

```
copy WC_installdir\xml\sar\DBLoadMacros.dtd  
WC_installdir\test\WEB-INF\stores\BusinessDirect\data\ToolTech\data
```

这就将 DBLoadMacros.dtd 文件复制到 WC\_installdir\test\WEB-INF\stores\BusinessDirect\data\ToolTech\data 中。

- f. 从 Windows 命令提示符，输入以下命令：

```
copy WC_installdir\xml\sar\fulfillment.dtd  
WC_installdir\test\WEB-INF\stores\BusinessDirect\data\ToolTech\data
```

这就将 fulfillment.dtd 文件复制到 WC\_installdir\test\WEB-INF\stores\BusinessDirect\data\ToolTech\data 中。

3. 通过创建适当的 WebSphere Commerce Server 数据库实例，确保 WebSphere Commerce 模式已与必要的引导程序数据一起装入到数据库中。

**注：**关于创建实例的信息，请参阅对应于您的操作系统的《WebSphere Commerce 安装指南》。

此示例使用的 WebSphere Commerce Server 数据库实例称为 *mall*。将从此数据库的 KEYS 和 SUBKEYS 表中获取主键和外键；因此，标识解析器在未正确装入数据库的情况下将无法解析标识。

4. 要解析 fulfillment.xml 文件的标识，请执行以下操作：

- a. 编辑 fulfillment.xml 文件以包含以下内容：

```
<?xml version="1.0" encoding="UTF-8"?>  
<!DOCTYPE fulfillment-asset SYSTEM "fulfillment.dtd">  
<fulfillment-asset>  
</fulfillment-asset>
```

fulfillment.xml 应看似如下：

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE fulfillment-asset SYSTEM "fulfillment.dtd">
<fulfillment-asset>
<!--defaultshipoffset can be overridden in the STORITMFFC table.-->
<!--Now in ToolTech STORITMFFC.shippingoffset is set to 86400
seconds which is one day-->
<ffmcenter
  ffmcenter_id="@ffmcenter_id_1"
  member_id="&MEMBER_ID;"
  name="ToolTech Home"
  defaultshipoffset="0"
  markfordelete="0"
/>
</fulfillment-asset>

```

- b. 编辑 DBLoadMacros.dtd 文件以包含以下内容（如果缺少的的话）：

```

<!ENTITY MEMBER_ID "-2001">

```

- c. 输入以下命令以对照 fulfillment.xml 文件（其中定义了 FFMCENTER 表）运行标识解析器：

```

idresgen -dbname mall -dbuser db2admin -dbpwd db2admin
-infile WC_installdir\test\WEB-INF\stores\BusinessDirect\data\ToolTech\data\
fulfillment.xml
-outfile WC_installdir\test\WEB-INF\stores\BusinessDirect\data\ToolTech\data
\fulfillment1.xml
-method load

```

其中

- 如果不使用购物中心，那么 *mall* 应切换到目标数据库的名称。
- 如果不使用 *db2admin*，那么第一个 *db2admin* 就应更改为与数据库连接的用户名。
- 如果不使用 *db2admin*，那么第二个 *db2admin* 就应更改为与数据库连接的用户名和密码。

fulfillment1.xml 输出文件中的已解析元素与以下相似：

```

<fulfillment-asset>
<ffmcenter
  FFMCENTER_ID="10001"
  MEMBER_ID="-2001"
  NAME="ToolTech Home"
  DEFAULTSHIPOFFSET="0"
  MARKFORDELETE="0"
/>
</fulfillment-asset>

```

**注：**这是一个示例。输出文件可能包含不同的值。

5. 要解析 store.xml 文件的标识，请执行以下操作：

- a. 编辑 store.xml 文件以包含以下内容：

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE store-asset SYSTEM "store.dtd">
<store-asset>
</store-asset>

```

- b. 从生成的输出文件（fulfillment1.xml）中获取 FFMCENTER\_ID 键，并在 *WC\_installdir\test\WEB-INF\stores\BusinessDirect\data\ToolTech\data* 中用该键来替代出现在 store.xml 的工作副本中的所有 @ffmcenter\_id\_1

- c. 编辑 DBLoadMacros.dtd 文件以包含以下内容（如果缺少的的话）：

```

<!ENTITY MEMBER_ID "-2001">
<!ENTITY STORE_IDENTIFIER "ToolTech">
<!ENTITY STORE_DIR "ToolTech">

```

d. 输入以下命令:

```

idresgen -dbname mall -dbuser db2admin -dbpwd db2admin
-infile WC_installdir\test\WEB-INF\stores\
BusinessDirect\data\ToolTech\data\store.xml-outfile
WC_installdir\test\WEB-INF\stores\
BusinessDirect\data\ToolTech\data\store1.xml -method load

```

其中

- 如果不使用购物中心, 那么 *mall* 应切换到目标数据库的名称。
- 如果不使用 *db2admin*, 那么第一个 *db2admin* 就应切换到与数据库连接的用户名。
- 如果不使用 *db2admin*, 那么第二个 *db2admin* 就应切换到与数据库连接的用户密码。

*store1.xml* 输出文件中的完全解析元素与以下相似:

```

<store-asset>
<storeent
  STOREENT_ID="10151"
  MEMBER_ID="-2001"
  TYPE="S"
  IDENTIFIER="ToolTech"
  SETCCURR="USD"
/>
<store
  STORE_ID="10151"
  DIRECTORY="ToolTech"
  FFMCENTER_ID="10001"
  LANGUAGE_ID="-1"
  STOREGRP_ID="-1"
  ALLOCATIÖNGOODFOR="43200"
  BOPMPADFACTOR="0"
  DEFAULTBOFFSET="2592000"
  FFMSELECTIONFLAGS="0"
  MAXBOFFSET="7776000"
  REJECTEDORDEXPIRY="259200"
  RTNFFMCTR_ID="10001"
  PRICEREFFLAGS="0"
  STORETYPE="B2B"
/>
<vendor
  VENDOR_ID="10001"
  STOREENT_ID="10151"
  VENDORNAME="Tooltech Vendor"
  MARKFORDELETE="0"
/>
<dispentrel
  AUCTIONSTATE="0"
  CATENTRY_ID="0"
  CATENTTYPE_ID="ProductBean"
  DEVICEFMT_ID="-1"
  DISPENTREL_ID="10001"
  MBRGRP_ID="0"
  PAGENAME="CatalogProductDisplay.jsp"
  STOREENT_ID="10151"
  RANK="0"
/>
<dispentrel
  AUCTIONSTATE="0"
  CATENTRY_ID="0"

```

```

        CATENTTYPE_ID="ItemBean"
        DEVICEFMT_ID="-1"
        DISPENTREL_ID="10002"
        MBRGRP_ID="0"
        PAGENAME="CatalogItemDisplay.jsp"
        STOREENT_ID="10151"
        RANK="0"
    />
<dispcgpre1
    CATGROUP_ID="0"
    DEVICEFMT_ID="-1"
    DISPCGPREL_ID="10001"
    MBRGRP_ID="0"
    PAGENAME="CatalogCategories.jsp"
    STOREENT_ID="10151"
    RANK="0"
/>
<invadjcode
    ADJUSTCODE="PCNT"
    INVADJCODE_ID="10001"
    MARKFORDELETE="0"
    STOREENT_ID="10151"
/>
<invadjcode
    ADJUSTCODE="SPLG"
    INVADJCODE_ID="10002"
    MARKFORDELETE="0"
    STOREENT_ID="10151"
/>
<invadjcode
    ADJUSTCODE="DISC"
    INVADJCODE_ID="10003"
    MARKFORDELETE="0"
    STOREENT_ID="10151"
/>
<rtnreason
    REASONTYPE="C"
    RTNREASON_ID="10001"
    STOREENT_ID="10151"
    MARKFORDELETE="0"
    CODE="WPR"
/>
<rtnreason
    REASONTYPE="B"
    RTNREASON_ID="10002"
    STOREENT_ID="10151"
    MARKFORDELETE="0"
    CODE="DEF"
/>
<rtnreason
    REASONTYPE="M"
    RTNREASON_ID="10003"
    STOREENT_ID="10151"
    MARKFORDELETE="0"
    CODE="ERR"
/>
<rtnreason
    REASONTYPE="M"
    RTNREASON_ID="10004"
    STOREENT_ID="10151"
    MARKFORDELETE="0"
    CODE="WPS"
/>
</store-asset>

```

注：这是一个示例。输出文件可能包含不同的值。

6. 在 store.xml 文件中, 将找到以下元素:

```
<storeent
  STOREENT_ID="@storeent_id_1"
  MEMBER_ID="@seller_b2b_mbr_id"
  TYPE="S"
  IDENTIFIER("&STORE_IDENTIFIER"
  SETCURR="USD"
/>
```

store.xml 的此元素映射到数据库中的 storeent 表; 它的 STOREENT\_ID、MEMBER\_ID、TYPE、IDENTIFIER 和 SETCURR 属性映射到该表的列。storeent\_id\_1 规范是 STOREENT\_ID 属性值的内部别名; &MEMBER\_ID; 是实体参数。在可使用装入程序装入实体 &MEMBER\_ID; 的值之前, 必须对它作替换。&MEMBER\_ID; 的值在 DBLoadMacros.dtd 宏文件中定义; 此值是从该文件中替换的。当标识解析器遇到 @storeent\_id\_1 时, 它查看主表高速缓存以了解 storeent 是否存在。因为它是主表, 因此 storeent 存在。标识解析器读取该表的计数器, 将之递增, 然后用结果来替换内部别名。以同样方式处理 store.xml 文件中的所有其它此类条目。

7. 确保路径包含了目录, 该目录包含如第 356 页的『使用装入程序软件包命令和脚本』中所列的适当的标识解析命令或脚本。

对于此示例, 从 Windows 命令提示符下输入以下命令:

```
cd WC_installdir\bin
```

其中, 如果标识解析命令 idresgen.cmd 不处在您的系统上的 WC\_installdir\bin 中, 则应将 WC\_installdir\bin 更改为包含该标识解析命令的目录的名称。

8. 从 Windows 命令提示符, 输入以下命令:

```
idresgen -dbname mall -dbuser wcs -dbpwd wcs1 -infile
WC_installdir\test\WEB-INF\stores\BusinessDirect\data\ToolTech\
data\store.xml -outfile
WC_installdir\test\WEB-INF\stores\BusinessDirect\data\ToolTech\
data\store1.xml-method load
```

其中

- 如果不使用购物中心, 那么 mall 应切换到目标数据库的名称。
- 如果不使用 db2admin, 那么第一个 db2admin 就应切换到与数据库连接的用户名。
- 如果不使用 db2admin, 那么第二个 db2admin 就应切换到与数据库连接的用户的密码。

store1.xml 中第一个输出 XML 片段与以下相似:

```
<storeent
  STOREENT_ID="10001"
  MEMBER_ID="-2001"
  TYPE="S"
  IDENTIFIER="ToolTech"
  SETCURR="USD"
/>
```

**注:** 这是一个示例。输出文件可能包含不同的值。

store1.xml 中第二个 XML 片段与以下相似:

```
<store
  STORE_ID="10001"
  DIRECTORY="ToolTech"
  FFMCENTER_ID=""
```



```

LANGUAGE_ID="-1"
STOREGRP_ID="-1"
ALLOCATIONGOODFOR="43200"
BOPMPADFACTOR="0"
DEFAULTBOFFSET="2592000"
FFMCSELECTIONFLAGS="0"
MAXBOFFSET="7776000"
REJECTEDORDEXPIRY="259200"
RTNFFMCTR_ID=""
PRICEREFFLAGS="0"
STORETYPE="B2B"
/>

```

**注：**这是一个示例。输出文件可能包含不同的值。

可以使用以下选项之一来解析标识：

- 选项 1:
  - a. 通过添加任何在 fulfillment.xml 中唯一的内容（包含对 fulfillment.dtd 的引用）来将 fulfillment.xml 和 store.xml 文件合并到 store.xml，确保下面所显示的 ffmcenter 元素在 store 元素之前。
 

```

<ffmcenter
  FFMCENTER_ID="@ffmcenter_id_1"
  MEMBER_ID="&MEMBER_ID;"
  NAME="ToolTech Home"
  DEFAULTBOFFSET="0"
  MARKFORDELETE="0"
/>

```
  - b. 对照合并后的文件运行标识解析器。
- 选项 2: 使用第 373 页的『装入数据库有用资源组』中所述的过程来装入商店有用资源数据组。

## 使用标识解析器指定属性文件

可通过使用 `-propfile` 参数修改标识解析器解析标识的方式。缺省属性文件为 `IdResolveKeys.properties`。要修改并使用 `IdResolveKeys.properties`，请将此文件复制到用户定义的目录中，进行所需的更改，然后在调用标识解析命令时指定该新文件。缺省 `IdResolveKeys.properties` 文件位于 `WC_installdir/properties` 目录中：

属性文件规范具有比内部别名的使用更高的优先级。

此处是 store.xml 文件中的样本 XML 片段：

```

<store
  STORE_ID="@storeent_id_1"
  DIRECTORY="ToolTech"
  FFMCENTER_ID="@ffmcenter_id_1"
  LANGUAGE_ID="&en_US;"
  STOREGRP_ID="-1"
  ALLOCATIONGOODFOR="43200"
  BOPMPADFACTOR="0"
  DEFAULTBOFFSET="2592000"
  FFMSELECTIONFLAGS="0"
  MAXBOFFSET="7776000"
  REJECTEDORDEXPIRY="259200"
  RTNFFMCTR_ID="@ffmcenter_id_1"
  PRICEREFFLAGS="0"
  STORETYPE="B2B"
/>

```

如果运行标识解析器时将 `-profile` 指定为 `WC_installdir\test\WEB-INF\stores\BusinessDirect\data\ToolTech\data\myPropFile`，且指定的文件 `myPropFile.properties` 包含以下条目：

```
NAMEDELIMITER=@
SELECTDELIMITER=:
FFMCENTER=@FFMCENTER_ID@MEMBER_ID:10051 -2001
```

当处理商店元素时，标识解析器使用 `where` 子句 `10051` 和 `-2001` 在数据库中查询 `FFMCENTER` 表。然后使用对此值返回的索引为 `FFMCENTER_ID` 解析标识。

关于使用此命令的更多信息，请参阅第 327 页的『标识解析命令』。

## 装入数据的示例

当解析了 XML 文件中的标识（如果需要）时，即已准备好将数据装入 WebSphere Commerce Server 数据库。

**注：**如果您正确解析了 XML 数据中的标识，则源 XML 文件不应包含以下任何内容：

- 以 `at`（`）` 符号开头的词
- 以 `ampersand`（`&`）符号开头的词
- 带有空引号（`""`）的标识

如果出现了这些符号中的任何一个则表示尚未准备好装入 XML 文件。

本部分中描述的装入数据的示例使用在第 357 页的『解析标识的示例』中解析的 `fulfillment1.xml` 文件。

要将数据装入 WebSphere Commerce Server 数据库，请如以下示例所示运行装入命令：

### 1. 创建工作目录。

对于此示例，使用您在第 357 页的『解析标识的示例』中创建的名为 `WC_installdir\test\WEB-INF\stores\BusinessDirect\data\ToolTech\data\` 的目录。

### 2. 确保输入 XML 文件位于装入程序可找到的位置。

对于此示例，请确保您在第 357 页的『解析标识的示例』中创建的 `fulfillment1.xml` 输出文件位于 `WC_installdir\test\WEB-INF\stores\BusinessDirect\data\ToolTech\data\` 中。

### 3. 确保备份了 WebSphere Commerce Server 数据库，以便在发生不可恢复的错误时可以从备份中恢复数据库。

**注：**请参阅随数据库产品提供的备份和恢复文档，以获得备份数据库的信息。

### 4. 确保路径包含了目录，该目录包含如第 356 页的『使用装入程序软件包命令和脚本』中所列的适当的装入命令或脚本。

对于此示例，从 Windows 命令提示符下输入以下命令：

```
cd WC_installdir\bin
```

其中，如果装入命令 `massload.cmd` 不处在您的系统上的 `WC_installdir\bin` 中，则应将 `WC_installdir\bin` 更改为包含该装入命令的目录的名称。

### 5. 对您解析的 XML 文件运行装入命令以将数据装入到目标数据库。

对于此示例，从 Windows 命令提示符下输入以下命令：

```
massload -dbname mall -dbuser db2admin -dbpwd db2admin -infile
WC_installdir\test\WEB-INF\stores\BusinessDirect\data\ToolTech
\data\fulfillment1.xml -method sqlimport -commitcount 50
```

其中

- 如果不使用购物中心，那么 *mall* 应切换到目标数据库的名称。
- 如果不使用 *db2admin*，那么第一个 *db2admin* 就应切换到与数据库连接的用户名。
- 如果不使用 *db2admin*，那么第二个 *db2admin* 就应切换到与数据库连接的用户的密码。

即使要装入的元素少于 50 个，此示例对 `-commitcount` 指定了值 50。这是出于性能原因。缺省情况下，提交计数是 1。使用此缺省值将引起对每个写入数据库的记录都执行提交操作。上述示例中将数值设置为 50 确保了仅当装入成功才发生数据库 I/O 且在发生错误的情况下不向数据库写入任何东西。然而，如果有大量数据要装入，建议不要将提交计数值设置得与元素数目一样大，原因如下：

- 高的提交计数值将引起高的内存消耗。
- 当提交计数值小于元素的数目时，至少会向数据库写入一些数据。根据 `-maxerror` 的值，较小的 `-commitcount` 值确保了在超出了最大错误数且工具终止之前，有一些数据写入了数据库。`-maxerror` 的缺省值是 1。

`-noprimary` 选项的缺省值是错误，这在遗漏主键时工具会报告错误并终止。

由于这些示例不按由管理控制台使用且在第 367 页的『数据库有用资源装入顺序』中描述的顺序来装入商店有用资源，因此第 357 页的『解析标识的示例』中创建的 `store1.xml` 文件可能违反某些表的完整性约束。如果尝试使用装入方法装入 `store1.xml` 而不加修改，则约束违反将引起数据库进入悬挂状态。因此为了简单，使用装入命令的此示例是基于 `fulfillment.xml` 文件的已解析版本的，该文件唯一的外键是在样本商店中定义的 `MEMBER_ID` 的外键。此示例装入已解析的 `fulfillment1.xml` 文件（该文件是第 357 页的『解析标识的示例』中的输出），并使用 SQL 导入方法。当您不能确定 XML 文件的内容是否干净时，请如此示例中所使用带有适当设置的 `-commitcount` 和 `-maxerror` 参数的 SQL 导入方法，以便报告所有对数据库约束的违反，而不改变数据库或危及数据库的完整性。

运行此命令时，缺省情况下会在执行子目录（在上例中是 `WC_installdir \bin`）中创建一个跟踪文本文件（`trace.txt`）。对于 ，`trace.txt` 在缺省情况下是创建在 `QIBM/Userdata/CommerceServer55/instances/instance_name/logs` 中的。

如果 `WCALoggerConfig.xml` 记录配置文件已更改为将 `trace.txt` 放置在另一位置，那么请转至该位置以检查该文件。关于定制 `WCALoggerConfig.xml` 的更多信息，请参阅 WebSphere Commerce 产品和开发联机帮助。

`trace.txt` 文件包含命令执行的操作及其结果的列表。如果对上例中显示的命令使用 SQL 导入方法，那么 `trace.txt` 的结束处会包含一个指示已提交记录数的条目。

关于使用此命令的更多信息，请参阅第 336 页的『装入命令』。



---

## 第 38 章 装入 WebSphere Commerce 数据库有用资源组

如果您不希望创建所有数据库有用资源并在发布之前将它们封装在商店归档文件中，则可以使用 WebSphere Commerce 装入程序软件包来装入数据库有用资源组。

本章的第一部分解释 WebSphere Commerce 数据库有用资源组，以及如何确定分组。第二部分描述将这些数据库有用资源组装入到 WebSphere Commerce 数据库的过程。在阅读本部分之前，应当彻底地复查第 323 页的第 37 章，『装入商店数据的概述』中的信息，这些信息将有助于您了解使用装入程序软件包装入数据库有用资源组时所需的知识。

---

### 数据库有用资源组

将数据库有用资源分成组以简化创建和装入过程。这些数据库有用资源组包含一个逻辑上相关的一组表。组织数据库有用资源组的顺序对于装入很重要，因为在装入数据之间的关系之前，数据必须已存在。


要装入商店的整个数据库有用资源集，需要遵循『数据库有用资源装入顺序』。要装入单组数据库有用资源，您需要确保该组在逻辑上是完整的。例如，在发布一个商店归档文件时，您可以选择忽略产品目录数据库有用资源，它可以稍后再发布。在这种情况下，任何依赖于产品目录的数据库有用资源（库存、价格列表以及一些送货和税务数据）也保持未发布。要发布省略的数据，请确保产品目录数据库有用资源在逻辑上是完整的：即，必须提供基本商品、产品目录条目和属性等。您还必须发布相关的数据库有用资源（它们必须在逻辑上对自身是完整的）。换句话说，每个库存标识都必须定义了适当的库存、价格、送货和税款。在此情况下，逻辑上完整的相关产品目录数据作为一个整体称为产品目录数据库有用资源组。

本指南前几章中描述的 WebSphere Commerce 数据库有用资源可以安排为几个组。一个组是一组逻辑上完整的数据，它可以独立装入。每个数据库有用资源组由 WebSphere Commerce 数据库表构成，并具有如第 419 页的附录 C，『数据库有用资源组』中所述的外部相关性。表列表是基于 WebSphere Commerce 样本商店的，不过该列表适用于任何类属商店。请记住每个数据库有用资源组的表列表并非详尽无遗的，而是作为总体指南提供。您可能要根据商店的特定需要，包含或排除一些表。

### 数据库有用资源装入顺序

要遵循特定顺序才能成功地装入数据库有用资源组。每个组都被看作是结构上完整的并独立于其它数据库有用资源组。然而，数据库有用资源组中有外键关系。这样的关系（带有来自其它组的数据）被称为数据库有用资源组的外部相关性。

将数据库有用资源组装入 WebSphere Commerce 数据库之前，必须满足该组的外部相关性。任何定义为给定数据库有用资源组的外部相关性的组都必须首先装入。可在第 419 页的『数据库有用资源组相关性』中找到外部相关性和相关表的列表。

**注：** WebSphere Commerce 商店需要一个商店所有者。您可以使用缺省组织（它可以作为缺省所有者可用）。  要装入此组，请创建新的组织，而不是使用缺省组织。

按照以下顺序装入数据库有用资源组:

1. 仅依赖于引导程序数据的数据库有用资源组。
  - a. 首先装入**组织**数据库有用资源。
2. 依赖于供货所有者的数据库有用资源组。
  - a. **供货**数据库有用资源。除了组织数据库有用资源组之外, 其它几个数据库有用资源组对此组中定义的数据都具有直接或间接的外部相关性。
3. 依赖于商店所有者组织的数据库有用资源组。
  - a. **访问控制**数据库有用资源依赖于商店所有者组织 (ORGENITY\_ID)。其它所有数据库有用资源组对该组中定义的数据都不具有相关性, 这意味着可以随时装入访问控制数据库有用资源。然而, 访问控制所有者必须与商店所有者相同。
  - b. **商店**数据库有用资源依赖于商店所有者组织 (ORGENITY\_ID)。  
商店可以引用一个供货中心。商店所有者组织还可以是供货中心的所有者组织。
4. 依赖于商店数据库有用资源的数据库有用资源组。可以按任何顺序装入以下组:
  - a. **竞销**数据库有用资源。
  - b. **命令**数据库有用资源。
  - c. **货币**数据库有用资源。
  - d. **策略**数据库有用资源。
  - e. **送货**数据库有用资源。
  - f. **税款**数据库有用资源。
5. 其它数据库有用资源组。
  - a. **产品目录**数据库有用资源依赖于送货和税款数据库有用资源组。
  - b. **商店缺省**数据库有用资源对送货数据库有用资源组具有外部相关性。如果送货数据库有用资源不存在, 则无需填充此组。
  - c. **合同**数据库有用资源依赖于组织有用资源。不直接装入合同数据库有用资源。关于更多信息, 请参阅第 378 页的『发布合同有用资源』。您应该在其它数据库有用资源组之后装入合同有用资源。

要查看 WebSphere Commerce 样本商店所形成的数据库有用资源组的内容, 请参阅第 419 页的附录 C, 『数据库有用资源组』。

---

## 装入商店

为了帮助您装入数据库有用资源, WebSphere Commerce 5.5 中提供了样本商店。要将整个商店的 XML 数据装入 WebSphere Commerce 数据库中, 请执行以下操作:

1. 复查以下信息:
  - a. 第 417 页的附录 B, 『创建数据』。
  - b. 第 419 页的附录 C, 『数据库有用资源组』, 因为您需要知道哪些 WebSphere Commerce 数据库有用资源文件和数据库表会受影响。
  - c. 第 323 页的第 37 章, 『装入商店数据的概述』, 它提供装入程序软件包的背景信息。

2. 计划您针对整个商店数据库有用资源集的装入过程。无论您是希望如第 373 页的『装入数据库有用资源组』中所示装入单个数据库有用资源组，还是希望装入整个商店，基本的过程都是相同的。在下一步中，将使用或创建以下文件用于装入过程：
  - a. 每个组的一个或多个数据库有用资源文件。当您装入整个商店时，需要所有已创建的数据库有用资源文件。例如，您将需要 *database asset.xml* 文件（如 *campaign.xml*、*catalog.xml* 或 *currency.xml*），以及独立的特定于您的商店支持的语言环境的 *database asset.xml* 文件。此类文件的示例包含在 WebSphere Commerce 样本商店归档文件中。样本商店归档文件是按 *WC\_installdir/samplestores* 目录中的业务模型来组织的。注意，不是所有的数据库有用资源组都需要特定于语言环境的信息。
  - b. 一个新的 XML 文件（它巩固了商店所有的 XML 数据库有用资源文件）包含 XML 实体引用，并包含整个商店的根元素。这称为主数据库有用资源组 XML 文件。您可以在样本数据包中找到此文件，它名为 *store-data-assets.xml*。
  - c. 一个新的 DTD 文件（它定义了数据库有用资源组中的 XML 文件所需的所有数据类型），它称为主数据库有用资源组 DTD 文件。您可以在样本数据包中找到此文件，它名为 *store-data-assets.dtd*。
  - d. 第二个 DTD 文件，它定义了外部相关性。您可能需要在主数据库有用资源组 DTD 文件中包含此文件。您可以在样本数据包中找到此文件，它名为 *ForeignKeys.dtd*。
  - e. 第三个 DTD 文件，它包含所有 WebSphere Commerce 表的定义。*wcs.dtd* 文件已存在于 WebSphere Commerce 中，位于 *WC\_installdir/schema/xml* 目录中。您可能需要在主数据库有用资源组 DTD 文件中包含此文件。如果您尚未定制 WebSphere Commerce 模式，则可以使用此文件而无需修改。
3. 如本指南先前的各章中所指导，创建数据库有用资源 XML 文件。如果您完成了有用资源章节中的任务，则这些 XML 文件已存在。数据库有用资源文件不得在文件开始处包含任何 DTD 声明或页面伪指令，因为这可能在连接文件时引起冲突。必须具有根元素的唯一文件是主数据库有用资源组 XML 文件。

**注：**如果有多种语言的数据库有用资源文件，则每个文件必须以 `<?xml encoding = locale specific encoding>` 开头。例如，英语数据库有用资源文件应该指定 `<?xml encoding = "UTF-8"?">`，而法语文件应该指定 `<?xml encoding = "ISO-8859-1"?">`。

4. 为整个商店数据集创建主数据库有用资源组 XML 文件。此文件包含引用实体以包含商店的各种数据库有用资源 XML 文件。通过使用外部引用实体，您可以连接 XML 文件以简化标识解析命令和装入过程。而且当同时装入多个组时，每个 XML 文件中使用的内部别名相对于该组中其它 XML 数据库有用资源文件或其它组而言可以是外部的。XML 分析程序将在外部引用的地方替换以由外部引用实体引用的文件的内容。

将以下用于装入整个商店数据集的示例用作指南，可基于此摘要创建您的数据库有用资源组文件：

```
<?xml version="1.0"?>
<!DOCTYPE import SYSTEM "store-data-assets.dtd">
<import>
<!-- Fulfillment data group -->
&fulfillment.xml;

<!-- Store data group -->
```

```

&store.xml;
&en_US_store.xml;
&fr_FR_store.xml;

<!-- Tax data group -->
&tax.xml;
&en_US_tax.xml;
&fr_FR_tax.xml;
&taxfulfill.xml;

<!-- Shipping data group -->
&shipping.xml;
&en_US_shipping.xml;
&fr_FR_shipping.xml;
&shipfulfill.xml;

<!-- Catalog data group -->
&catalog.xml;
&en_US_catalog.xml;
&fr_FR_catalog.xml;
&storecatalog.xml;
&storefulfill.xml;
&offering.xml;
&store-catalog-tax.xml;
&store-catalog-shipping.xml;

<!-- Currency data group -->
&currency.xml;
&en_US_currency.xml;
&fr_FR_currency.xml;

<!-- Campaign data group -->
&campaign.xml;
&en_US_campaign.xml;
&fr_FR_campaign.xml;

<!-- Business policy data group -->
&businesspolicy.xml;
&en_US_businesspolicy.xml;
&fr_FR_businesspolicy.xml;

<!-- Access control data group -->
&accesscontrol.xml;
&en_US_accesscontrol.xml;
&fr_FR_accesscontrol.xml;

<!-- Other data groups -->
&command.xml;
&store-default.xml;
</import>

```

其中

- `import` 是 XML 文档的根元素。根元素已在随 WebSphere Commerce 提供的 `wcs.dtd` 文件中定义，并包含所有 WebSphere Commerce 数据库表的定义。然而，如果定制了 WebSphere Commerce 模式，则可能需要使用不同的根元素。您可以生成新的反映定制模式的 DTD 文件，或者可以更新现有的 `wcs.dtd` 文件。
- `store-data-assets.dtd` 是指将在下一步中创建的主数据库有用资源组 DTD 文件的名称。注释的文本将商店的不同数据库有用资源组分隔开。
- `&database asset.xml`；是对数据库有用资源 XML 片段文件的 XML 实体引用。数据库有用资源的路径和位置在数据库有用资源组 DTD 文件中定义。`&database asset.xml`；文件的名称将更改，以匹配已为每个组创建的数据库有用资源。



- `&locale_database_asset.xml`; 对商店支持的每种语言而言都是必需的。如果您的商店只使用一种语言, 则将只引用一个文件。如果商店支持多种语言, 则对每种语言都需要一个引用。上面的摘要假定商店支持英语和法语。
5. 创建一个主数据库有用资源组 DTD 文件, 它定义以上实体以及数据库有用资源所需的其它 DTD 文件。

使用以下用于整个商店数据库有用资源集的示例作为指南, 您可以创建主数据库有用资源组 DTD 文件:

```
<!ENTITY % wcs.dtd SYSTEM "absolute path for WebSphere Commerce wcs.dtd file">
%wcs.dtd;

<!ENTITY % ForeignKeys.dtd SYSTEM "ForeignKeys.dtd">
%ForeignKeys.dtd;
<!ENTITY fulfillment.xml SYSTEM "data/fulfillment.xml">
<!ENTITY en_US_fulfillment.xml SYSTEM "data/en_US/fulfillment.xml">
<!ENTITY fr_FR_fulfillment.xml SYSTEM "data/fr_FR/fulfillment.xml">

<!ENTITY store.xml SYSTEM "data/store.xml">
<!ENTITY en_US_store.xml SYSTEM "data/en_US/store.xml">
<!ENTITY fr_FR_store.xml SYSTEM "data/fr_FR/store.xml">

<!ENTITY tax.xml SYSTEM "data/tax.xml">
<!ENTITY en_US_tax.xml SYSTEM "data/en_US/tax.xml">
<!ENTITY fr_FR_tax.xml SYSTEM "data/fr_FR/tax.xml">
<!ENTITY taxfulfill.xml SYSTEM "data/taxfulfill.xml">

<!ENTITY shipping.xml SYSTEM "data/shipping.xml">
<!ENTITY en_US_shipping.xml SYSTEM "data/en_US/shipping.xml">
<!ENTITY fr_FR_shipping.xml SYSTEM "data/fr_FR/shipping.xml">
<!ENTITY shipfulfill.xml SYSTEM "data/shipfulfill.xml">

<!ENTITY catalog.xml SYSTEM "data/catalog.xml">
<!ENTITY en_US_catalog.xml SYSTEM "data/en_US/catalog.xml">
<!ENTITY fr_FR_catalog.xml SYSTEM "data/fr_FR/catalog.xml">
<!ENTITY store-catalog.xml SYSTEM "data/store-catalog.xml">
<!ENTITY storefulfill.xml SYSTEM "data/storefulfill.xml">
<!ENTITY offering.xml SYSTEM "data/offering.xml">
<!ENTITY store-catalog-tax.xml SYSTEM "data/store-catalog-tax.xml">
<!ENTITY store-catalog-shipping.xml SYSTEM "data/store-catalog-shipping.xml">

<!ENTITY currency.xml SYSTEM "data/currency.xml">
<!ENTITY en_US_currency.xml SYSTEM "data/en_US/currency.xml">
<!ENTITY fr_FR_currency.xml SYSTEM "data/fr_FR/currency.xml">

<!ENTITY campaign.xml SYSTEM "data/campaign.xml">
<!ENTITY en_US_campaign.xml SYSTEM "data/en_US/campaign.xml">
<!ENTITY fr_FR_campaign.xml SYSTEM "data/fr_FR/campaign.xml">

<!ENTITY businesspolicy.xml SYSTEM "data/businesspolicy.xml">
<!ENTITY en_US_businesspolicy.xml SYSTEM "data/en_US/businesspolicy.xml">
<!ENTITY fr_FR_businesspolicy.xml SYSTEM "data/fr_FR/businesspolicy.xml">

<!ENTITY accesscontrol.xml SYSTEM "data/accesscontrol.xml">
<!ENTITY en_US_accesscontrol.xml SYSTEM "data/en_US/accesscontrol.xml">
<!ENTITY fr_FR_accesscontrol.xml SYSTEM "data/fr_FR/accesscontrol.xml">

<!ENTITY command.xml SYSTEM "data/command.xml">
<!ENTITY store-defaults.xml SYSTEM "data/store-defaults.xml">
```

其中

- `wcs.dtd` 是指包含了定义在其数据库有用资源组之外的数据的 DTD 文件。此文件是随 WebSphere Commerce 提供的, 它还定义了数据库有用资源 XML 文件中使用的根元素。

- `ForeignKeys.dtd` 指的是定义了除根元素之外的其它元素的 DTD 文件。此文件包含数据库有用资源组之外外部相关性所有的 XML 实体引用声明和定义。这样，XML 文件引用了外键值（这些外键值没有作为数据库有用资源组的一部分来创建，而它们必须在此组之前已装入到数据库）。

**注：** 确保正确标识了路径。在此示例中，文件位于与主数据库有用资源组 DTD 文件相同的目录中。

- `store.xml`、`en_US_store.xml` 和 `fr_FR_store.xml` 是在主数据库有用资源组 XML 文件中使用的外部引用实体（假设您的商店支持英语和法语）。要使用引用，请遵循实体引用约定：`&alias_name;`。
  - `database asset.xml` 是指从中装入数据库有用资源的 XML 文件的名称。此名称将更改以便与已为每个组创建的数据库有用资源文件相匹配。此类文件的示例包含在 WebSphere Commerce 样本商店归档文件。样本商店归档文件是按 `WC_installdir/samplestores` 目录中的业务模型来组织的。
  - `locale_database asset.xml` 文件对于商店支持的每种语言而言都是必需的，它们位于上述目录下。如果您的商店只使用一种语言，则将只引用一个文件。如果商店支持多种语言，则每种语言都将需要一个特定于语言环境的文件。上面的摘要假定商店支持英语和法语。
6. 每个数据库有用资源组都需要在其域之外或其数据集之外定义的信息，因为每个组都可能具有外部相关性。可在 DTD 文件中提供此数据。例如，商店数据库有用资源组具有以下外部相关性：

```
bootstrap.LANGUAGE.LANGUAGE_ID, bootstrap.MEMBER.MEMBER_ID,
bootstrap.SETCURR.SETCURR_ID, fulfillment.FFMCENTER.FFMCENTER_ID
```

当装入一个数据库有用资源组或整组商店有用资源时，必须从 WebSphere Commerce 数据库定义外部相关性。要使用此数据，请遵循对应的 XML 实体引用。例如，要使用由 `ffmcenter_id` 实体定义的数据，则要在 XML 文件中写入 `&ffmcenter_id;`。将以下商店数据库有用资源示例作为指南，您可基于此抽取创建 DTD 文件，名为 `ForeignKeys.dtd`：

```
<!ENTITY en_US "-1"><!ENTITY fr_FR "-2">
<!ENTITY de_DE "-3">
<!ENTITY it_IT "-4">
<!ENTITY es_ES "-5">
<!ENTITY pt_BR "-6">
<!ENTITY zh_CN "-7">
<!ENTITY zh_TW "-8">
<!ENTITY ko_KR "-9">
<!ENTITY ja_JP "-10">
<!ENTITY MEMBER_ID "-2000">
<!ENTITY ffmcenter_id "10001">
```

其中

- `MEMBER_ID` 是标识商店所有者的内部引用号。
- `ffmcenter` 是商店供货中心的引用号。因为商店可以使用多个供货中心，因此可在 `ForeignKeys.dtd` 文件中定义多个供货中心。
- `locale` 是每种语言环境的 WebSphere Commerce 引用号（由国家或地区和语言标识）。值位于 `LANGUAGE` 数据库表中。

**注：** 如果正在将现有的商店归档文件分割成数据库有用资源组，请确保（比如说）用对应的实体引用 `&ffmcenter_id;`；替换所有对别名 `@ffmcenter_id` 的引用。

- 一旦创建了所有数据文件，则如第 327 页的『标识解析命令』所述对照主数据库有用资源组 XML 文件运行 IDResolve 命令以解析数据。
- 如第 336 页的『装入命令』中所述，对已解析的数据文件运行装入命令。要验证您的装入过程，请参阅日志文件：

-  idresgen.db2.log 和 massload.db2.log
-  idresgen.oracle.log 和 massload.oracle.log

日志文件位于以下目录中：

-     WC\_installdir/logs
  -  WC\_userdir/instances/instance\_name/logs
-  如第 378 页的『发布商业帐户有用资源』中所述运行 AccountImport 命令。
  - 如果适用，如第 378 页的『发布合同有用资源』中所述发布合同。
  - 完成第 379 页的『通过复制到 WebSphere Commerce Server 发布商店前台有用资源和商店配置文件』中的任务

---

## 装入数据库有用资源组

要将单个数据库有用资源组的 XML 数据装入 WebSphere Commerce 数据库，请执行以下操作：

- 复查以下信息：
  - 第 417 页的附录 B，『创建数据』
  - 第 419 页的附录 C，『数据库有用资源组』，因为您需要知道哪些 WebSphere Commerce 有用资源文件和数据库表会受影响。
  - 第 323 页的第 37 章，『装入商店数据的概述』，它提供装入程序软件包的背景信息。
- 计划您的装入过程，并决定将要装入哪个数据库有用资源组。无论您是希望如第 368 页的『装入商店』中所示装入整个商店数据库有用资源集，还是希望装入单个数据库有用资源组，基本的过程都是相同的。在下一步中，将使用或创建以下文件用于装入过程：
  - 一个或多个数据库有用资源文件，这取决于您选择了哪个组。例如，如果装入商店数据库组有用资源，将需要 store.xml 文件和商店支持的每种语言环境的单独 store.xml 文件。此类文件的示例包含在 WebSphere Commerce 样本商店归档文件。样本商店归档文件是按 WC\_installdir/samplestores 目录中的业务模型来组织的。注意，不是所有的数据库有用资源组都需要特定于语言环境的信息。
  - 一个新的 XML 文件（它巩固了所有的 XML 数据库有用资源文件）包含 XML 实体引用，并包含数据库有用资源的根元素。这称为主数据库有用资源组 XML 文件。您可以在样本数据包中找到此文件，它名为 store-all-assets.xml。
  - 一个新的 DTD 文件（它定义了数据库有用资源组中的 XML 文件所需的所有数据类型），它称为主数据库有用资源组 DTD 文件。您可以在样本数据包中找到此文件，它名为 store-all-assets.dtd。

- d. 第二个 DTD 文件，它定义了外部相关性。您可能需要在主数据库有用资源组 DTD 文件中包含此文件。您可以在样本数据包中找到此文件，它称为非数据库有用资源组 ForeignKeys.dtd。
  - e. 第三个 DTD 文件，它包含所有 WebSphere Commerce 表的定义。wcs.dtd 文件已存在于 WebSphere Commerce 中，位于 WC\_installdir/schema/xml 目录中。您可能需要将此文件包含在主数据库有用资源组 DTD 文件中。如果您尚未定制 WebSphere Commerce 模式，则可以使用此文件而无需修改。
3. 如本指南先前的各章中所指导，为您将装入的组创建数据库有用资源 XML 文件。如果您完成了有用资源章节中的任务，则这些 XML 文件已存在。数据库有用资源文件不得在文件开始处包含任何 DTD 声明或页面伪指令，因为这可能在连接文件时引起冲突。而且为了简单，您也可能决定不创建任何根元素。必须具有根元素的唯一文件是主数据库有用资源组 XML 文件。

**注：**如果有多种语言的数据库有用资源文件，则每个文件必须以 `<?xml encoding = locale specific encoding>` 开头。例如，英语数据库有用资源文件应该指定 `<?xml encoding = "UTF-8"?>`，而法语文件应该指定 `<?xml encoding = "ISO-8859-1"?>`。必须确保您指定的编码与文件的实际编码相匹配。

4. 为每个您希望装入的组创建主数据库有用资源组 XML 文件。此文件包含引用实体以包含一个（或多个）数据库有用资源组中各种不同的 XML 文件。通过使用外部引用实体，您可以连接 XML 文件以简化标识解析命令和装入过程。而且当同时装入多个组时，每个 XML 文件中使用的内部别名相对于该组中其它 XML 数据文件或其它组而言可以是外部的。XML 分析程序将在外部引用的地方替换以由外部引用实体引用的文件的内容。

将以下用于装入单个商店数据库有用资源组的示例用作指南，可基于此摘要创建您的数据库有用资源组 XML 文件：

```
<?xml version="1.0"?>
<!DOCTYPE import SYSTEM "store-assets.dtd">
<import>
&store.xml;
&en_US_store.xml;
&fr_FR_store.xml;
</import>
```

其中

- `import` 是 XML 文档的根元素。根元素已在随 WebSphere Commerce 提供的 `wcs.dtd` 文件中定义，并包含所有 WebSphere Commerce 数据库表的定义。然而，如果定制了 WebSphere Commerce 模式，则可能需要使用不同的根元素。您可以生成新的反映定制模式的 DTD 文件，或者可以更新 `wcs.dtd` 文件。
  - `store-assets.dtd` 是指将在下一步中创建的主数据库有用资源组 DTD 文件的名称。
  - `&store.xml;` 是一个引用数据库有用资源组 XML 文件的 XML 实体。路径和位置在数据库有用资源组 DTD 文件中定义。此名称将更改以便与已为每个组创建的有用资源文件相匹配。
  - `locale_store.xml;` 对于商店支持的每种语言而言都是必需的。如果您的商店只使用一种语言，则将只引用一个文件。如果商店支持多种语言，则对每种语言都将需要一个引用。上面的摘要假定商店支持英语和法语。
5. 创建一个主数据库有用资源组 DTD 文件，它定义以上实体以及组所需的其它 DTD 文件。

将以下用于装入单个商店数据库有用资源组的示例用作指南，您可以为任何数据库组创建主数据库有用资源组 DTD 文件：

```
<!ENTITY % wcs.dtd SYSTEM "absolute path for WebSphere Commerce wcs.dtd file">
%wcs.dtd;

<!ENTITY % ForeignKeys.dtd SYSTEM "ForeignKeys.dtd">
%ForeignKeys.dtd;
<!ENTITY store.xml SYSTEM "store.xml">
<!ENTITY en_US_store.xml SYSTEM "en_US/store.xml">
<!ENTITY fr_FR_store.xml SYSTEM "fr_FR/store.xml">
```

其中

- `wcs.dtd` 是指包含了定义在其数据库有用资源组之外的数据的 DTD 文件。此文件是随 WebSphere Commerce 提供的，它还解析和定义了数据库有用资源组 XML 文件中使用的根元素。
- `ForeignKeys.dtd` 指的是定义了除根元素之外的其它元素的 DTD 文件。此文件包含数据库有用资源组之外外部相关性所有的 XML 实体引用声明和定义。这样，XML 文件引用了外键值（这些外键值没有作为数据库有用资源组的一部分来创建，而它们必须在此组之前已装入到数据库）。

**注：**确保正确标识了路径。在此示例中，文件位于与数据库有用资源组 DTD 文件相同的目录中。

- `store.xml`、`en_US_store.xml` 和 `fr_FR_store.xml` 是在数据库有用资源组 XML 文件中使用的 外部引用实体。要使用引用，请遵循实体引用约定：`&alias_name;`。
  - `store.xml` 是指从中装入数据库有用资源的组的数据文件。此名称将更改以便与已为每个组创建的数据库有用资源文件相匹配。请注意，特定于语言环境的 XML 文件位于 `WC_installdir/samplestores` 目录中。
  - `path_store.xml` 文件对于商店支持的每种语言而言都是必需的，它们位于上述目录下。如果您的商店只使用一种语言，则将只引用一个文件。如果商店支持多种语言，则每种语言都将需要一个特定于语言环境的文件。上面的摘要假定商店支持英语和法语。
6. 每个数据库有用资源组都需要在其域之外或其数据集之外定义的信息，因为每个组都可能具有外部相关性。可在 DTD 文件中提供此数据。例如，商店数据库有用资源组具有以下外部相关性：

```
bootstrap.LANGUAGE.LANGUAGE_ID, bootstrap.MEMBER.MEMBER_ID,
bootstrap.SETCURR.SETCURR_ID, fulfillment.FFMCENTER.FFMCENTER_ID
```

当装入一个数据组或整组商店数据时，必须从 WebSphere Commerce 数据库定义以下外部相关性。要使用此数据，请遵循对应的 XML 实体引用。例如，要使用由 `ffmcenter_id` 实体定义的数据，则要在 XML 文件中写入 `&ffmcenter_id;`。将以下商店数据库有用资源组示例作为指南，您可以基于此抽取创建 DTD 文件，名为 `非数据库有用资源组 ForeignKeys.dtd`：

```
<!ENTITY en_US "-1"><!ENTITY fr_FR "-2">
<!ENTITY de_DE "-3">
<!ENTITY it_IT "-4">
<!ENTITY es_ES "-5">
<!ENTITY pt_BR "-6">
<!ENTITY zh_CN "-7">
<!ENTITY zh_TW "-8">
```



```
<!ENTITY ko_KR "-9">
<!ENTITY ja_JP "-10">
<!ENTITY MEMBER_ID "-2000">
<!ENTITY ffmcenter_id "10001">
```

其中

- MEMBER\_ID 是标识商店所有者的内部引用号。
- ffmcenter 是商店供货中心的引用号。因为商店可以使用多个供货中心，因此可在 ForeignKeys.dtd 文件中定义多个供货中心。
- locale 是每种语言环境的 WebSphere Commerce 引用号（由国家或地区和语言标识）。值位于 LANGUAGE 数据库表中。

注:

- a. 如果正在将现有的商店归档文件分割成数据库有用资源组，请确保（比如说）用对应的实体引用 &ffmcenter\_id；替换所有对别名 @ffmcenter\_id 的引用。
  - b. 如果您正在引用已存在于数据库中的成员标识，则可以用 &MEMBER\_ID；来替换样本商店数据中使用的内部别名。如果未引用已存在的标识，则可以使用 @member\_id 来包含解析成员标识所需的 XML。
7. 一旦创建了所有数据文件，则如第 327 页的『标识解析命令』所述对照数据库有用资源组 XML 文件运行 IDResolve 命令以解析数据。
  8. 如第 336 页的『装入命令』中所述，对已解析的数据文件运行装入命令。要验证您的装入过程，请参阅日志文件:

-  idresgen.db2.log 和 massload.db2.log
-  idresgen.oracle.log 和 massload.oracle.log

日志文件位于以下目录中:

-     WC\_installdir/logs
  -  WC\_userdir/instances/instance\_name/logs
9.  如第 378 页的『发布商业帐户有用资源』中所述运行 AccountImport 命令。
  10. 如果适用，如第 378 页的『发布合同有用资源』中所述发布合同。
  11. 完成第 379 页的『通过复制到 WebSphere Commerce Server 发布商店前台有用资源和商店配置文件』中的任务

---

## 第 39 章 发布商业帐户和合同

一些商店数据库有用资源（商业帐户和合同）不能由装入程序软件包装入。可以通过使用管理控制台或从命令行来发布这些数据库有用资源，将其作为选择发布完整的商店的一部分（如第 311 页的第 36 章，『发布完整的商店』中所述），或者可使用商业帐户和合同所对应的命令来发布它们。这些命令如下：

- AccountImport - 从商店归档文件中的 businessaccount.xml 文件中创建商业帐户。
- ContractImportApprovedVersion - 从 contract.xml 文件中创建合同。如果合同处于活动状态，则此命令创建并部署合同。即使 contract.xml file 包含多个合同，也仅需要调用此命令一次。

**注：**关于这些命令的更多信息，请参阅 WebSphere Commerce 产品和开发联机帮助。

商业帐户有用资源以 XML 文件的格式包含在随 WebSphere Commerce 提供的一些样本商店归档文件中。然而，建议您使用所提供的工具创建商业帐户有用资源，而不是为这些有用资源创建 XML 文件。关于使用所提供的工具创建这些有用资源的更多信息，请参阅 WebSphere Commerce 产品联机帮助。用于发布商业帐户的指示信息包含在以下部分中，以备您选择发布随样本商店归档文件提供的对应的 XML 文件或创建自己的 XML 文件。

**注：**如果您未使用管理控制台来发布商业帐户或合同，则必须先发布商店和产品目录有用资源才能发布商业帐户和合同。特别的，您需要商店和产品目录标识，以及拥有商店的组织的标识，还有任何与该合同关联的买方组织的标识。如果合同的条款和条件未指定特定的产品目录，则您无需在发布商业帐户或合同之前发布产品目录。

如果使用管理控制台或命令行发布来发布这些有用资源，请确保选择了产品目录选项，或者商店已具有已发布的产品目录。如果使用对应的命令发布这些有用资源，请确保已将上面列出的有用资源装入数据库。

---

### 使用管理控制台或命令行来发布商业帐户和合同

您可以使用管理控制台或从命令行使用发布实用程序来发布商业帐户和合同。为了使用管理控制台或命令行来发布商业帐户和合同，必须以商店归档文件格式打包有用资源。关于将商店前台有用资源封装为商店归档文件的更多信息，请参阅第 301 页的第 9 部分，『封装商店』。

关于使用管理控制台或命令行发布有用资源的详细的分步说明，请参阅 WebSphere Commerce 产品联机帮助。

---

### 使用命令发布商业帐户和合同

如果不希望将有用资源封装为商店归档文件，则仍可使用对应的命令发布商业帐户和合同：

- AccountImport - 从商店归档文件中的 businessaccount.xml 文件中创建商业帐户。

- `ContractImportApprovedVersion` - 将已核准的或活动的合同从 XML 文件导入到 WebSphere Commerce Server。在导入合同之前，此命令确保正在导入的合同包含必要的条款和条件并且是有效的合同。

## 发布商业帐户有用资源

要发布从样本商店获得的商业帐户有用资源，请执行以下操作：

1. 将 `ForeignKeys.dtd` 复制到以下位置：
  - `WC_installdir/xml/trading/dtd`
  - `400 WC_userdir/instances/instance_name/xml/trading/xml`  
*ForeignKeys.dtd* 包含 *businessaccount.xml* 所引用的实体值。
2. 将 `businessaccount.xml` 复制到以下位置中：
  - `WC_installdir/xml/trading/xml`
  - `400 WC_userdir/instances/instance_name/xml/trading/xml`
3. 打开管理控制台。以管理员身份登录。
4. 在浏览器中，输入以下 URL：
  - `https://hostname:8002/webapp/wcs/admin/servlet/AccountImport?fileName=businessaccount.xml&URL=`  
成功完成时要重定向的 URL

注：关于命令语法和参数的更多信息，请参阅 WebSphere Commerce 产品和开发联机帮助。

## 发布合同有用资源

要发布从样本商店获得的合同有用资源，请执行以下操作：

1. 将 `ForeignKeys.dtd` 复制到以下位置：
  - `WC_installdir/xml/trading/dtd`
  - `400 WC_userdir/instances/instance_name/xml/trading/xml`  
*ForeignKeys.dtd* 包含 *businessaccount.xml* 所引用的实体值。
2. 将 `contract.xml` 复制到以下位置中：
  - `WC_installdir/xml/trading/xml`
  - `400 WC_userdir/instances/instance_name/xml/trading/xml`
3. 打开管理控制台。以管理员身份登录。
4. 在浏览器中输入以下内容：
  - `https://hostname:8002/webapp/wcs/admin/servlet/ContractImportApprovedVersion?fileName=contract.xml&xsd=false&URL=ContractDisplay`
5. 如果您的商店包含多个 `contract.xml` 文件（例如，特定于语言环境的合同文件），请对每个 `contract.xml` 文件重复步骤 1 到 4。



---

## 第 40 章 发布商店前台有用资源和商店配置文件

发布商店前台有用资源、HTML 和 JSP 文件、属性文件或资源束以及创建商店页面的图像和图形，是创建功能性商店过程的一部分。您可以使用管理控制台或从命令行（如第 311 页的第 36 章，『发布完整的商店』中所述）来发布商店前台有用资源，将其作为选择发布完整的商店的一部分，或者可通过简单地将有用资源复制到 WebSphere Commerce Server 上的指定位置来发布商店前台有用资源。

如果发布包含在样本商店中的 JSP 文件且您计划更改商店流程，则还将需要发布作为该商店归档文件一部分的商店配置文件。商店的 XML 配置文件在商店归档文件的以下目录中：

WEB-INF/xml/tools/stores/StoreDirectory/devtools/flow

“更改流程”工具的属性文件在商店归档文件的以下目录中：

StoreDirectory/devtools/flow/ui

---

### 使用管理控制台或命令行来发布商店前台有用资源和商店配置文件

您可使用管理控制台或从命令行使用发布实用程序来发布商店前台有用资源和商店配置文件。

为了使用管理控制台或命令行来发布商店前台有用资源和商店配置文件，必须以商店归档文件格式打包商店前台有用资源和商店配置文件。关于将商店前台有用资源打包为商店归档文件的更多信息，请参阅第 301 页的第 9 部分，『封装商店』。

---


### 通过复制到 WebSphere Commerce Server 发布商店前台有用资源和商店配置文件

如果不希望将有用资源打包为商店归档文件，您仍可通过将商店前台有用资源直接复制到 WebSphere Commerce Server 来发布它们。必须将 Web 有用资源（HTML、JSP 文件、图像和图形）复制到 Web 应用程序文档根路径。必须将资源绑定或属性文件复制到应用程序的属性路径。这些商店配置文件必须复制到这些商店或工具 Web 模块下的相应位置。

**注：**当您将样本商店归档文件解压缩到商店 Web 模块时，（保留路径结构）这些文件有用资源将被放置在正确的位置中。然而，由“更改流程”工具使用的属性文件（StoreDirectory/devtools/flow/ui/\*.properties）必须复制到 WebSphere Commerce 配置文件 instance\_name.xml 中定义的 ToolsStoresPropertiesPath 中。


要将商店前台有用资源和商店配置文件复制到 WebSphere Commerce Server，请执行以下操作：

1. 将 JSP 文件、HTML（包含文件、图像和图形）复制到商店 Web 应用程序文档根路径中的商店目录（*storedir*）：
  - `WAS_installdir/installedApps/cell_name/WC_instance_name.ear/Stores.war/storedir`


-  `WAS_userdir/WAS_instance_name/installedApps/  
cell_name/WC_instance_name.ear/Stores.war/storedir`

其中 `storedir` 是 STORE 数据库表中 DIRECTORY 列的值。

2. 将资源绑定和属性文件复制到应用程序属性路径:

- `WAS_installdir/installedApps/cell_name/  
WC_instance_name.ear/Stores.war/WEB-INF/  
classes/storedir`
-  `WAS_userdir/WAS_instance_name/InstalledApps/cell_name/  
WC_instance_name.ear/Stores.war/WEB-INF/  
classes/storedir`

3. 将商店配置文件复制到 WebSphere Commerce 配置文件 `instance_name.xml` 中定义的位置。此文件位于以下目录中:

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

商店配置文件被复制到以下位置:


- 商店配置 XML ( `WEB-INF/xml/tools/stores/storedirectory/devtools/flow` ) 将被复制到 `ToolsStoresXMLPath`。此路径是在 WebSphere Commerce 配置文件 `instance_name.xml` 中定义的。
- 商店配置属性文件将复制到 `ToolsStoresPropertiesPath`。此路径是在 WebSphere Commerce 配置文件 `instance_name.xml` 中定义的。

4. 使用以下一种方法启动商店:

- 使用 `StoreCatalogDisplay` 命令:  
`StoreCatalogDisplay?storeId=storeId&catalogId=catalogId&langId=langId`  
其中

- `storeId` 是位于 STORE 数据库表 STORE\_ID 列中的值。
- `catalogId` 是位于 CATALOG 数据库表 CATALOG\_ID 列中的值。
- `langId` 是给定语言环境的 LANGUAGE 数据库表 LANGUAGE\_ID 列的值。  
有关缺省 WebSphere Commerce 值的列表, 请参阅 LANGUAGE 数据库表。

- 如果您的商店基于 WebSphere Commerce 样本商店, 请通过编辑以下目录中的 `index.jsp` 文件来组合商店的 URL:

- `WAS_installdir/installedApps/instance_name/  
/WC_instance_name.ear/Stores.war/storedir`
-  `WAS_userdir/WAS_instance_name/  
installedApps/cell_name/WC_instance_name.ear/Stores.war/storedir`

为以下参数添加正确的值:

- `hostname` 是 WebSphere Commerce 机器的全限定名称,
- `storeId` 是位于 STORE 数据库表 STORE\_ID 列中的值。
- `catalogId` 是位于 CATALOG 数据库表 CATALOG\_ID 列中的值。
- `langId` 是给定语言环境的 LANGUAGE 数据库表 LANGUAGE\_ID 列的值。  
有关缺省 WebSphere Commerce 值的列表, 请参阅 LANGUAGE 数据库表。

要在浏览器中查看您的商店，请启动以下 URL: `http://host name  
/webapp/wcs/stores/servlet/storedir/index.jsp`



---

## 第 11 部分 向商店添加 WebSphere Commerce 功能部件

为了向您的商店添加一些在 WebSphere Commerce 中可用的功能，需要完成某些手工步骤。本部分中的各章讨论向您的商店添加以下功能：

- 第 385 页的第 41 章，『向商店添加客户关心』
- 第 407 页的第 42 章，『向商店添加电子广告位』



---

## 第 41 章 向商店添加客户关心

**Professional Business** 通过一个使用 Lotus® Sametime™ 服务器的同步文本界面，WebSphere Commerce 中的客户关心功能提供实时客户服务支持。在商店中启用客户关心时，客户可以进入商店、单击一个链接并连接到客户服务代表（CSR）。然后，客户可以通过因特网与 CSR 通信。

**注：** 本章讲述了如何在商店中启用客户关心。然而，在商店可以启用客户关心之前，必须首先安装 Sametime 服务器并配置它与 WebSphere Commerce 协同工作。关于更多信息，请参阅《WebSphere Commerce 附加软件指南》。如果 Sametime 服务器不使用与 WebSphere Commerce 相同的 LDAP 服务器，则您还必须在管理控制台中注册 CSR，以使它们能够使用客户关心。关于此任务以及客户关心的整体概念和 CSR 如何使用客户关心的更多信息，请参阅 WebSphere Commerce 联机帮助。

**注：**

如果您根据以下样本商店之一创建商店，则可以使用 WebSphere 贸易加速器在您的商店中快速且方便地启用客户关心：

- **Business** B2B 直销（多乐五金）
- 消费者直销（时尚潮流）

在管理控制台中使用发布实用程序发布商店之后，请打开 WebSphere 贸易加速器，然后选择商店菜单，再选择更改流程并启用客户关心功能部件。关于更多详细的指示信息，请参阅 WebSphere Commerce 产品联机帮助中的帮助主题“使用 WebSphere 贸易加速器更改商店流程”

但是，如果没有使用样本商店作为基础来创建商店，则将必须执行一些操作在商店中启用客户关心。本章的剩余部分讨论了在不是基于样本商店之一的商店中启用客户关心所需的概念和步骤。

**注：** 样本商店 **Business** “多乐五金”和“时尚潮流”演示了应如何实现客户关心，并提供了您可以在商店中用来启用客户关心的代码。本章将引用这两个商店的示例，以说明如何在商店中启用客户关心。请确保在阅读本章时您已有样本商店的最新版本。

**注：** 为了支持向后的兼容性以及随 Internet Explorer 和 Netscape V4.x 浏览器提供的内置 JVM，applet 代码是用 JDK1.1 和 AWT 组件开发的。因此，JDK 1.2 或更高版本提供的功能部件（包括语言支持、双向支持和辅助选项）将不可用或不完全受支持。Sun Java 插件和 Netscape 6 和 7 浏览器不受支持。

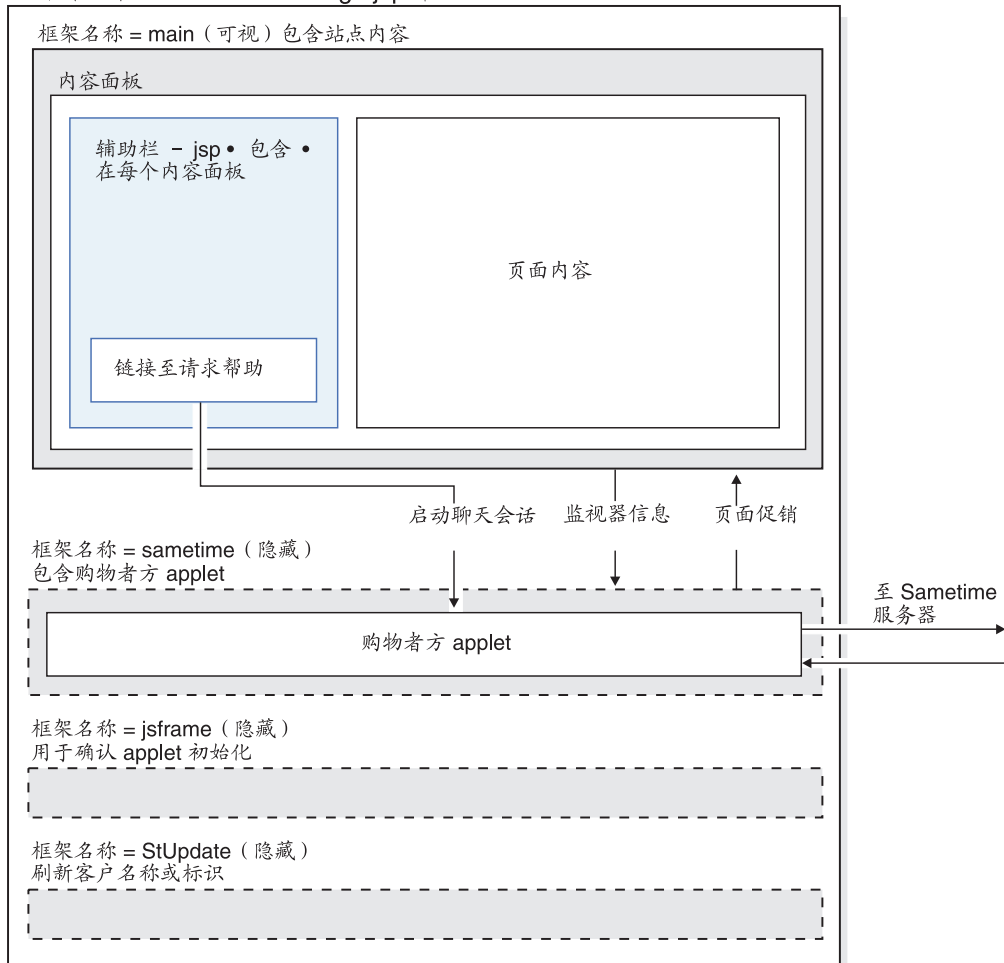
---

### 了解商店中的客户关心

当客户在启用了客户关心的商店里选择客户关心链接时（例如与客户助手实时聊天），会启动包含聊天窗口的小应用程序。此小应用程序是在隐藏的框架集中运行的，这样的框架集不影响站点的外观。

下图说明了框架集的构造。

框架集（在 storeFramesetPage.jsp 中）



框架集包含四个框架：

- **主框架：** 该框架包含商店的内容，包括创建商店页面的文件（这些文件是创建页面主体的文件、头和页脚文件以及辅助栏文件）。此框架的内容是对商店访问者可见的。请注意主框架包含与 the Sametime 框架的以下连接：指向客户关心和监视信息的链接。在第 396 页的『使用客户关心监视客户』中更详细地讨论了监视信息。
- **Sametime：** 包含客户关心小应用程序的框架。此框架对商店的访问者不可见。然而，当客户单击链接启动小应用程序时，客户将看到客户关心窗口。此框架通过页面“推”功能，还将信息推向主框架。
- **jsframe：** 确认小应用程序已正确装入的框架。此框架的内容不显示给客户。
- **StUpdate：** 该框架刷新客户的信息，其中包括客户姓名或标识。

## 使用框架集

在框架集中启动客户关心小应用程序会将小应用程序代码与商店页面中的代码分开。如上图说明，商店页面包含在框架集的主框架中，而小应用程序代码包含在 Sametime 框架中。由于 applet 只下载一次，因此通过将 applet 代码与商店页面分开，您可以减少网络流量。



使用框架集还让您保持与 Sametime 服务器的连接。如果小应用程序是每个页面的一部分而不是框架集中的部分，则每次客户访问新页面时将创建新的 Sametime 会话。由于客户关心 applet 是匿名登录到 Sametime 服务器的，因此每次客户访问新页面时创建一个新会话会使您无法跟踪客户在商店中的活动。通过使用框架集，客户的原始 Sametime 会话得到了保持，并且属性更改时客户活动将发送回 Sametime 服务器。

## 使用框架集的问题

虽然使用框架集是在商店中实现客户关心的建议方法，但您应了解使用框架集出现的以下问题：

- 单一入口点：只有客户在框架内浏览商店时，他们才可以客户关心。同样，CSR 也只能通过框架集来监视客户的行动。为确保客户通过框架集来浏览商店，他们必须通过单一入口点来访问站点。如果客户通过另一个页面访问商店（例如产品目录页面），则他们将不在框架集中。
- 放置书签：当使用框架集时，客户将只能将框架集的 URL 而不是单个的页面作为书签。
- 刷新：当客户在框架集中并单击“刷新”时，会按框架集的编码将他们带回原始页面，该页面可能与当前正在浏览器中显示的页面不同。
- 重新缩放浏览器窗口：如果客户在处于框架集中时重新缩放浏览器窗口，则浏览器可能自动重新装入条目地址。如果重新装入了条目地址，则可能终止与 Sametime 服务器的连接。不同的浏览器在此情况下的行为是不同的。
- 安全性：为了使工作正常，客户关心框架集中的页面必须能够相互通信。通过 JavaScript 函数调用可启用该通信。当客户通过框架集浏览站点时，每个单独的框架以及框架集（地址栏中的 URL）都会保持其自己的连接，无论是非安全的（http，缺省端口 80）还是安全的（https，缺省端口 443）。如果客户通过非安全连接浏览商店，则框架集内的所有框架都是 HTTP 的形式。在此情况下，没有 SSL 的问题。但是，如果客户浏览到安全页面（例如，注册页面），则框架集中的主框架将切换为 HTTPS，而剩余的框架仍保持为非安全的（http）。在此情况下，客户将不能启动客户关心小应用程序。浏览器不会授权启动 applet，因为调用 applet（安全，端口 443）的 JavaScript 函数看起来不是浏览器地址栏中的 URL 而是来自其它的服务器（HTTP，端口 80）。要解决该问题，您应该始终使用 StoreFramesetView 命令进入商店，因为它对框架集 URL 会强制使用 HTTPS 连接。
- 为了在框架之间进行通信，Java applet 和 JavaScript 函数也必须进行通信。由于 applet 代码库是在 Sametime 服务器上，而不是在 WebSphere Commerce Server 上，因此有些使用 Sun Java Plug-in 的浏览器（例如，Netscape V6 和 7）会阻止 JavaScript 与从其它主机装入的 Java applet 通信。

## 在没有框架集的情况下使用客户关心

客户关心将在没有框架集的情况下工作，然而所有监视客户活动对 CSR 都不可用。也就是说，从客户那里收集的信息只是在客户提交帮助请求的那一刻是精确的。如果后来客户更改了页面，或向购物车添加了更多的商品，则对于 CSR 将不会在监视列表中更新新的信息。

为了在没有框架集的情况下使用客户关心，请执行以下操作：

1. 将以下代码添加到包含启动交谈页面的链接的 JSP 文件：

```
<script>
function LaunchChat()
{
```

```

<%
String pname = request.getRequestURI();
int indpn = pname.lastIndexOf('/');
indpn = pname.lastIndexOf('/', indpn-1);
if(indpn != -1) { pname = pname.substring(indpn+1);}
String headerType = (String) request.getAttribute("liveHelpPageType");
if (headerType==null) { headerType="";}
if (headerType.equals("personal"))
{
%>
currentPageURL='PERSONAL_URL';
<% } else { %>
currentPageURL=escape(location.href);
<% } %>
currentpageDesc="<%=pname%>";
chatURL="<%=com.ibm.commerce.tools.util.UIUtil.getWebappPath(request)%>
CCChatPageView?"
+ "pageURL=" + currentPageURL
+ "&pageDesc=" + currentpageDesc;
WindowName="";
chatAttr="toolbars=no,location=no,directories=no,status=yes,
menubar=no,scrollbars=no,resizable=no,width=360,height=400"
window.open(chatURL,WindowName, chatAttr);
return true;
}
</script>

```

注：请确保宽度和高度与 CustomerCareChatSetup.jsp 的 APPLET 标记所定义的相同。

- 如果您向之添加以上代码的页面是个人页面，则请确保在此代码块之前使用以下语句定义该个人页面：`request.setAttribute("liveHelpPageType", "personal");`
- 您可能还希望另外在 `chatURL` 字符串中添加定制属性的参数值对，或者添加其它有用的信息，这些信息可以通过请求对象在 CustomerCareChatSetup.jsp 中获得。
- 添加以下代码可从链接或图像启动交谈页面：

```

<A HREF="javascript:void(LaunchChat());">
<FONT COLOR="#ffffff" STYLE="font-size : 8pt">
<%= tooltechtext.getString("LiveHelp")%></FONT></A>

```
- （可选）定制 CustomerCareChatSetup.jsp 文件，使之传递附加客户信息。以下 `applet` 参数允许您定制 CustomerCareChatSetup.jsp。

表 16.

可定制的参数名	描述	注释
CHAT_FONT_SIZE	要在交谈区使用的字体大小。	缺省值为 12。
CHAT_FONT_COLOR	在交谈区中用于进入消息的字体颜色。	缺省值为蓝色 (#0000FF)。
CHAT_NAME_LENGTH	为在交谈区中显示用户名而保留的字符长度。	缺省值为 15。
WAIT_RANGE_1	整数值，如果正在商店中等待的客户数少于该值，则显示等待消息 1。否则，等待消息根据 WAIT_RANGE_2 的设置显示。	当仅将显示消息 1 时使用 -1。

表 16. (续)

WAIT_RANGE_2	整数值，如果正在商店中等待的客户数少于该值但多于 WAIT_RANGE_1，则显示等待消息 2。否则，等待消息将根据 WAIT_RANGE_3 的设置显示。	如果 WAIT_RANGE_1 为 -1，则忽略它。使用 -1 可禁用此范围。
WAIT_RANGE_3	整数值，如果正在商店中等待的客户数少于该值但多于 WAIT_RANGE_2，则显示等待消息 3。否则，显示等待消息 4。	如果 WAIT_RANGE_2 为 -1，则忽略它。使用 -1 可禁用此范围。
contentFrame	用于常规 WebSphere Commerce 商店页面的框架的名称	缺省值为 “_blank”。当 CSR 发送回 URL 时，它总是会启动一个新的浏览器窗口。
COUNTER_UNIT_WAIT	整数值，指示等待计数器增加 1 的频率。	缺省值为 30 秒。请确保该值与商店的 CustomerCareMonitorList.jsp 文件中所定义的不同。
WIDTH	交谈框架的首选宽度（以像素为单位）。	缺省宽度为 360 个像素。邀请消息的长度将影响最终宽度。
HEIGHT	交谈框架的高度（以像素为单位）。	缺省值为 400 个像素。
QUEUE_ID	客户关心队列标识	提供有效的队列标识将把帮助请求直接放到队列中。
ML_ATTRIBUTES	一组定制的监视属性标识（由逗号分隔）。	示例：<PARAM name="ML_ATTRIBUTES" value="8001,9002">
ML_ATTRIBUTE_xxxx	提供定制监视属性 xxxx 的值（xxxxx 是属性标识）	示例 <PARAM name="ML_ATTRIBUTE_8001" value="A value string "> 请确保属性标识已在 ML_ATTRIBUTES 参数中定义，否则将忽略该参数。

## 定义客户关心

在商店中使用客户关心前，必须定义以下信息：

- 定义商店的监视列表
- 定义商店的主题列表
- 定义商店的 URL

以下各节通过使用样本商店的示例说明如何定义信息来解释这些概念。

### 定义商店的监视列表

每个商店都包含定义了要由 CSR 监视的商品的文件。该文件 CustomerCareMonitorList.jsp 在以下目录中：

- `WAS_installdir/installedApps/instance_name/WC_instance_name.ear/Stores.war/storedir/CustomerServiceArea/CollaborationSection`
- **400** `WAS_userdir/WAS_instance_name/installedApps/cell_name/WC_instance_name.ear/Stores.war/storedir/CustomerServiceArea/CollaborationSection`

`CustomerCareMonitorList.jsp` 在 WebSphere 贸易加速器调用 `CCMonitorListView` 命令时装入 CSR applet。该命令返回一个 JSP 文件，该文件返回以下 XML 字符串，这些字符串定义商店的监视信息：

```
<?xml version="1.0" encoding="UTF-8"?>
<WCS_CUSTOMERCARE>
  <MONITORING_LIST>
    <ATTR ID="4" LABEL="MonitoringVisitorsTableCurrentPage" ></ATTR>
    <ATTR ID="3010" LABEL="MonitoringVisitorsTableInSite" UNIT="30" ></ATTR>
    <ATTR ID="3011" LABEL="MonitoringVisitorsTableInPage" UNIT="30" ></ATTR>
    <ATTR ID="6" LABEL="MonitoringVisitorsTableCart" ></ATTR>
  </MONITORING_LIST>
</WCS_CUSTOMERCARE>
```

以上例举的属性元素（ATTR）定义客户的监视属性。

表 17.

属性名称	描述	注释
ID	用于跟踪该商品的 Sametime 属性标识。	必需
Label	属性文件中该商品描述的标签键。相应的描述将显示在 CSR applet 中。	必需
Unit	指示过多少秒之后计数器会增加 1 的整数。值。	可选。仅供预定义计数器属性使用。

当 CSR 启动 CSR applet 时，applet 会装入 XML 字符串，同时创建监视列表。监视列表显示了与相应的 Sametime 属性相关联的字符串值。如有必要，JavaScript 函数会从相应的商店页面检索属性值。

**注：**如果 XML 字符串包含错误，或无法找到 `CustomerCareMonitorList.jsp` 文件，则 CSR applet 将使用缺省监视列表，该列表仅包含客户的姓名。

**CustomerCareMonitorList.jsp 示例：** 以下是 B2B 直销样本商店“多乐五金”的代码示例：

```
<%
// Create XML string
String strCfg=ECLivehelpConstants.EC_CC_XML_HEADER
  + LiveHelpConfiguration.getOpenTagString(ECLivehelpConstants.EC_CC_XML_ROOT)
  + LiveHelpConfiguration.getOpenTagString(ECLivehelpConstants.
EC_CC_XML_MONITORING_LIST);
%>

<% // modify this block to customize monitoring list
  strCfg=strCfg + LiveHelpConfiguration.getMonitorAttributeElementString
(ECLivehelpConstants.EC_CC_ST_ATTR_PAGE_URL, "MonitoringVisitorsTableCurrentPage")
  + LiveHelpConfiguration.getCloseTagString(ECLivehelpConstants.
EC_CC_XML_MONITORING_ATTR);
  strCfg=strCfg + LiveHelpConfiguration.getMonitorCounterAttributeElementString
```

```

(ECLivehelpConstants.EC_CC_ST_ATTR_SITE_COUNTER,
"MonitoringVisitorsTableInSite","30")
+ LiveHelpConfiguration.getCloseTagString(ECLivehelpConstants.
EC_CC_XML_MONITORING_ATTR);
    strCfg=strCfg + LiveHelpConfiguration.getMonitorCounterAttribute
ElementString
(ECLivehelpConstants.EC_CC_ST_ATTR_PAGE_COUNTER,
"MonitoringVisitorsTableInPage", "30")
+ LiveHelpConfiguration.getCloseTagString(ECLivehelpConstants.
EC_CC_XML_MONITORING_ATTR);
    strCfg=strCfg + LiveHelpConfiguration.getMonitorAttributeElementString
(ECLivehelpConstants.EC_CC_ST_ATTR_CART_ITEMS,
"MonitoringVisitorsTableCart")
+ LiveHelpConfiguration.getCloseTagString
(ECLivehelpConstants.EC_CC_XML_MONITORING_ATTR);
%>

<%
strCfg=strCfg
+ LiveHelpConfiguration.getCloseTagString
(ECLivehelpConstants.EC_CC_XML_MONITORING_LIST)
+ LiveHelpConfiguration.getCloseTagString
(ECLivehelpConstants.EC_CC_XML_ROOT);
%>

```

该样本使用 LiveHelpConfiguration 类的实用程序方法来确保 XML 的正确性。为了避免语法分析问题，该样本还使用已编码的属性值。下表提供了有关方法的更多详细信息。

表 18.

方法	描述	注释
静态字符串 getOpenTagString(String tagName)	返回一个开始标记字符串	LiveHelpConfiguration. getOpenTagString("HELLO") 返回字符串 <HELLO>
静态字符串 getCloseTagString(String tagName)	返回一个结束标记字符串	LiveHelpConfiguration. getCloseTagString ("HELLO") 返回字符串 </HELLO>
静态字符串 getMonitorAttribute ElementString(String sAttributeId, String sLabelKey)	为监视列表返回 ATTR 元素字 符串	LiveHelpConfiguration. getMonitorAttribute ElementString("1234", "myLabel") 返回字符串 <ATTR ID="1234" LABEL="myLabel" >
静态字符串 getMonitorCounter AttributeElementString (String sAttributeId, String sLabelKey, String sUnit)	为监视列表返回计数器 ATTR 元素字符串	LiveHelpConfiguration .getMonitorCounter AttributeElementString ("1234", "myLabel", "60") 返回字 符串 <ATTR ID="1234" LABEL="myLabel" UNIT="60" >

下表列出了可以监视的预定义 Sametime 属性标识。

表 19.

属性标识	描述	标签键字符串	注释
------	----	--------	----

表 19. (续)

ECLivehelpConstants. EC_CC_ST_ATTR_ PAGE_URL	购物者正在浏览的 Web 页面	"MonitoringVisitors TableCurrentPage"	需要从商店页面输入
ECLivehelpConstants. EC_CC_ST_ATTR_ CART_ITEMS	已经添加到购物车的 商品数	"MonitoringVisitors TableCart"	需要从商店页面输入
ECLivehelpConstants. EC_CC_ST_A TTR_SITE_COUNTER	客户已在商店站点中 逗留了多少时间	"Monitoring VisitorsTable InSite"	计数器属性，使用 UNIT 可设置更新频率
ECLivehelpConstants .EC_CC_ST_ATTR_ PAGE_COUNTER	客户已在此页面中逗 留了多少时间	"Monitoring VisitorsTable InPage"	计数器属性，使用 UNIT 可设置更新频 率，并且需要从商店 页面输入才能将监视 器复位
ECLivehelpConstants. EC_CC_ST_ATTR_ WAIT_COUNTER	客户已等待了 CSR 多 少时间	"Monitoring Visitors TableInWait"	计数器属性，使用 UNIT 可设置更新频 率。 通过客户提交帮助请 求或在交谈期间 CSR 将客户重新加入队列 时复位。 它还用于确定在 CSR 使用“服务下一位” 时，CSR 先为哪个客 户提供服务
ECLivehelpConstants. EC_CC_ST _ATTR_ REQ_QUEUE	客户请求所在队列的 名称	"Monitoring VisitorsTable Queue"	初始队列名，可能当 客户在商店页面中提 交帮助请求时提供
ECLivehelpConstants. EC_CC_ ST_ATTR_ REQ_CSR_NAME	与客户交谈的 CSR 的 姓名	"Monitoring Visitors TableCSR"	CSR 从队列接受请求 时将更新

注：可以使用大于 8000 的属性标识定义定制监视商品。例如，

```
<ATTR ID="8004" LABEL="MonitoringVisitorsTableItem8004"></ATTR>
```

## 定义商店的主题列表

每个商店都包含定义了与客户进行交谈会话期间可以由 CSR 使用的主题的文件。该文件 CustomerCareStoreQuestionList.jsp 在以下目录中：

- `WAS_installdir/installedApps/instance_name/  
WC_instance_name.ear/Stores.war/storedir/CustomerServiceArea//  
CollaborationSection`
- `400 WAS_userdir/WAS_instance_name/  
installedApps/cell_name/WC_instance_name.ear/  
Stores.war/storedir/CustomerServiceArea/CollaborationSection`

该 JSP 文件返回以下 XML 字符串，这些字符串定义商店的交谈主题：

```
<?xml version="1.0" encoding="UTF-8"?>
<WCS_CUSTOMERCARE>
  <QUESTION_LIST>
    <GROUP NAME="TopicGroupName" >
      <
        <QUESTION TITLE="TopicName" TEXT="Topic+Text" ></QUESTION>
      /GROUP>
    </QUESTION_LIST>
  </WCS_CUSTOMERCARE>
```

以上例举的组元素（GROUP）定义可以具有多个子主题的主题组。它有以下属性：

表 20.

属性名称	描述	注释
Name	如果组名重复，则将使用最新的组定义	必需

问题元素（QUESTION）定义单个主题。它有以下属性：

表 21.

属性名称	描述	注释
Title	如果某标题重复，则将使用最后的标题定义。	必需
Text	主题的内容。	必需

**CustomerCareStoreQuestionList.jsp 示例：** 以下是 B2B 直销样本商店“多乐五金” CustomerCareStoreQuestionList.jsp 中的代码：

```
<% // Create XML string
String strCfg=ECLivehelpConstants.EC_CC_XML_HEADER
  + LiveHelpConfiguration.getOpenTagString(ECLivehelpConstants.EC_CC_XML_ROOT)
  + LiveHelpConfiguration.getOpenTagString(ECLivehelpConstants.
EC_CC_XML_QUESTION_LIST);
%>
<% //unmark this block to add Topic group/topics
// start of Topic group block, repeat for more topic groups
strCfg=strCfg + LiveHelpConfiguration.getTopicGroupElementString
("TopicGroupName");
// start of Topic block, repeat for all topics in the same
group
strCfg=strCfg + LiveHelpConfiguration.getTopicElementString("TopicName",
Topic Text")
+ LiveHelpConfiguration.getCloseTagString(ECLivehelpConstants.
EC_CC_XML_QUESTION_QUESTION);
// end of Topic block
strCfg=strCfg + LiveHelpConfiguration.getCloseTagString
(ECLivehelpConstants.EC_CC_XML_QUESTION_GROUP);
// end of Topic group block
%>
<%
strCfg=strCfg + LiveHelpConfiguration.getCloseTagString
(ECLivehelpConstants.EC_CC_XML_QUESTION_LIST)
+ LiveHelpConfiguration.getCloseTagString
(ECLivehelpConstants.EC_CC_XML_ROOT);
%>
```

该样本使用 LiveHelpConfiguration 类的实用程序方法来确保 XML 的正确性。下表提供了有关方法的更多详细信息。

注:


1. 为了避免语法分析问题, 该样本还使用已编码的属性值。Unicode 字符串还用于使字符免遭损坏。
2. 要支持多种语言, 您可以对每种语言使用一个属性文件, 并根据在 CSR 的会话中选定的语言标识检索经翻译的名称或主题。

表 22.

方法	描述	注释
静态字符串 getOpenTagString (String tagName)	返回一个开始标记字符串	LiveHelpConfiguration. getOpenTagString ("HELLO") 返回字符串 <HELLO>
静态字符串 getCloseTagString(String tagName)	返回一个结束标记字符串	LiveHelpConfiguration. getCloseTagString ("HELLO") 返回字符串 </HELLO>
静态字符串 getTopicGroupElement String (String sGroupName)	返回主题列表的 GROUP 元素 字符串	LiveHelpConfiguration. getTopicGroupElementString ("myGroup") 返回字符串 <GROUP NAME = 'myGroup'>
静态字符串 getTopic ElementString (String sTitle, String sText)	返回主题列表的 QUESTION 元素字符串	LiveHelpConfiguration. getTopicElementString ("myTitle", "myText") 返回 <QUESTION TITLE ="myTitle" TEXT ="myText">

## 定义商店的 URL 列表

每个商店都包含定义了 CSR 可以在交谈会话期间发送到客户的浏览器的 URL 的文件。该文件 CustomerCareStoreURLList.jsp 在以下目录中:

- WAS\_installdir/installedApps/instance\_name  
/WC\_instance\_name.ear/Stores.war/storedir/CustomerServiceArea/  
CollaborationSection
-  400 WAS\_userdir/WAS\_instance\_name/installedApps/cell\_name  
/WC\_instance\_name.ear/Stores.war/storedir/CustomerServiceArea/  
CollaborationSection

该 JSP 文件返回以下 XML 字符串, 这些字符串包含商店的 URL 信息:

```
<?xml version="1.0" encoding="UTF-8"?>
<WCS_CUSTOMERCARE>
  <URL_LIST>
    <GROUP NAME="URLGroupName" >
      <PAGE NAME="IBM" URL="http%3A%2F%2Fwww.ibm.com" ></PAGE>
    </GROUP>
  </URL_LIST>
</WCS_CUSTOMERCARE>
```

以上例举的组元素 (GROUP) 定义可以具有多个 URL 的 URL 组。它有以下属性:

- Name: URL 组的名称。Name 是必需的属性。如果组名重复, 则使用最新的组定义。

页面元素 (PAGE) 定义单个 URL 地址。它有以下属性:



- Name: URL 页面的名称。Name 是必需的属性。如果页面名称重复, 则使用最新的页面定义。
- URL: 页面的 URL 地址。URL 是必需的属性。

**CustomerCareStoreURLList.jsp 示例:** 以下是 B2B 直销样本商店“多乐五金”CustomerCareStoreURLList.jsp 中的代码:

```
<% // Create XML string
String strCfg=ECLivehelpConstants.EC_CC_XML_HEADER
    + LiveHelpConfiguration.getOpenTagString(ECLivehelpConstants.
EC_CC_XML_ROOT)
    + LiveHelpConfiguration.getOpenTagString(ECLivehelpConstants.
EC_CC_XML_URL_LIST);
%>

<% //unmark following block to add URL group/pages
// start of URL group block, repeat for more URL groups
strCfg=strCfg + LiveHelpConfiguration.getURLGroupElementString("URLGroupName");
// start of URL pages block, repeat for all pages in the same group
strCfg=strCfg + LiveHelpConfiguration.getURLPageElementString
("IBM","http://www.ibm.com")
    + LiveHelpConfiguration.getCloseTagString
(ECLivehelpConstants.EC_CC_XML_URL_PAGE);
// end of URL pages block
strCfg=strCfg + LiveHelpConfiguration.getCloseTagString
(ECLivehelpConstants.EC_CC_XML_URL_GROUP);
// end of URL group block
%>
<%
strCfg=strCfg + LiveHelpConfiguration.getCloseTagString
(ECLivehelpConstants.EC_CC_XML_URL_LIST)
+ LiveHelpConfiguration.getCloseTagString(ECLivehelpConstants.EC_CC_XML_ROOT);
%>
```

该样本使用 LiveHelpConfiguration 类的实用程序方法来确保 XML 的正确性。下表提供了有关方法的更多详细信息。

**注:** 为了避免语法分析问题, 该样本还使用已编码的属性值。Unicode 字符串还用于使字符免遭损坏。

表 23.

方法	描述	注释
静态字符串 getOpenTagString(String tagName)	返回一个开始标记字符串	LiveHelpConfiguration.getOpenTagString("HELLO") 返回字符串 <HELLO>
静态字符串 getCloseTagString(String tagName)	返回一个结束标记字符串	LiveHelpConfiguration.getCloseTagString("HELLO") 返回字符串 </HELLO>
静态字符串 getTopicGroupElementString(String sGroupName)	返回 URL 列表的 GROUP 元素字符串	LiveHelpConfiguration.getTopicGroupElementString("myGroup") 返回字符串 <GROUP NAME = 'myGroup'>

表 23. (续)

静态字符串 <code>getUrlPageElementString</code> (String sName, String sURL)	返回 URL 列表的 PAGE 元素字符串	LiveHelpConfiguration. <code>getUrlPageElementString</code> ("myName", "myURL") 返回 <code>&lt;QUESTION TITLE = "myName" TEXT = "myURL"&gt;</code>
---	-----------------------	---

## 使用客户关心监视客户

当客户选择主框架中的链接时，将返回新的页面，同时设置以下活动中的事件链。

1. 此新的页面在其头部分中包含 `CustomerCareHeaderSetup.jsp` 文件。
2. `CustomerCareHeaderSetup.jsp` 在框架集页面中调用以下 JavaScript 函数：  
`parent.setPageParams()`。
3. `parent.setPageParams()` 更新变量以保存当前的商店页面信息，然后调用 `UpdateStInfo()`。
4. `UpdateStInfo()` 重新装入 `StUpdate` 框架，然后该框架调用 `CCShopperInfoUpdatePageView` 视图命令。
5. 当返回 `CCShopperInfoUpdatePageView` 时，它会装入 `CustomerCareInformationSetup.jsp` 文件，并收集包括客户姓名、标识、以及客户购物车中的商品在内的客户信息。
6. 然后 `CCShopperInfoUpdatePageView` 调用 `setCustomerName()`、`setShoppingCartItems()` 来更新此客户信息，接着调用 `changeSTAttributes()` 来更新 applet 中所有的客户属性。
7. `changeSTAttributes()` 调用一些 applet 方法来更新 CSR 正在监视的属性值。
8. 然后 `changeSTAttributes()` 调用 applet 的 `changeWCSAttrs()` 方法将所有已更新的属性值发送给 Sametime 系统，接着该系统通知 CSR applet 属性值已更改。

客户关心通过以下方式让您能够监视商店中正在与 CSR 通信的客户

- 获取客户的姓名或标识
- 确定客户正在浏览的页面
- 跟踪购物车中的商品
- 跟踪定制监视商品

将已定制代码添加到商店页面中以便获取这些信息。以下部分讨论了每个这样的监视功能是如何在样本商店中实现的。

### 获取客户的姓名或标识

一旦启动了客户关心小应用程序，且 CSR 已登录，CSR 就可以通过姓名或购物者标识来识别谁在使用小应用程序。样本商店中包含处理客户关心小应用程序的专门代码，以确定客户姓名或购物者标识。此代码确定客户是临时客户、购物车中有商品的临时客户还是注册客户，然后为该客户指定一个姓名或标识，并将此名称传递回客户关心小应用程序。然后这些名称显示给 CSR。例如，如果客户是没有在购物车中放任何商品的临时客户，则会为客户指定一个已生成的标识，购物者标识为 -1002。例如，如果客户是购物车中有商品的临时客户，则会显示购物者标识，且如果该客户已经注册，则他们的名字和姓氏也会显示。

样本商店通过在商店页面头文件中包含 CustomerCareInformationSetup.jsp 来获取客户的名称或标识。CustomerCareInformationSetup.jsp 文件中的以下代码将获取客户的姓名和标识:

```
<jsp:useBean id="userRDB" class="com.ibm.commerce.user.beans.
UserRegistrationDataBean" scope="page">
<% DataBeanManager.activate(userRDB, request); %>
</jsp:useBean>
<%
String customer_name="";
customer_name=userRDB.getUserId();
if (userRDB.findUser()){
    if (userRDB.getLastName() !=null && userRDB.getLastName().
length() > 0){
        if (locale.toString().equals("ja_JP")||locale.toString().
equals("ko_KR")||locale.toString().equals("zh_CN")||locale.toString().
equals("zh_TW"))
            {customer_name = userRDB.getLastName() + " " +
userRDB.getFirstName();}
        else
            {customer_name = userRDB.getFirstName() + " " +
userRDB.getLastName();}
    }
}
if (customer_name.equals("-1002")) {
    customer_name="";
}
else {
    // need to check order items
    ....
}
        customer_name=customer_name.trim();
%>
```

**注:** 每次客户在商店中浏览新页面时, 客户姓名或标识都会刷新。

CustomerCareInformationSetup.jsp 文件中的以下代码将更新客户的姓名和标识:

```
<script language="javascript">
....
function changeSTAttributes()
{
    if (typeof top.setCustomerName == 'function') {
        top.setCustomerName(<%=userRDB.getUserId()%>, '<%=customer_name%>');
        top.setShoppingCartItems(<%=shoppingCartItems%>);
        top.changeSTAttributes();
    }
}
</script>
```

在样本商店的“注销”页面中, 包含了更多的定制代码, 它们将客户姓名设置为已生成的标识, 并将购物车中的商品数量重新设置为零。“注销”页面是 UserLogoffRouter.jsp。定制代码如下:

```
<HTML>
<HEAD>
<SCRIPT language="javascript">
if (typeof parent.setCustomerName == 'function')
    parent.setCustomerName (parent.WCSGUESTID, '')
    if (typeof parent.setShoppingCartItems == 'function')
        parent.setShoppingCartItems(0);
</SCRIPT>
</HEAD>
</HTML>
```

## 确定客户正在浏览的页面

客户关心还让 CSR 能够确定商店中的客户当前正在浏览的页面。样本商店通过在头文件 (HeaderDisplay.jsp) 中包含 CustomerCareHeaderSetup.jsp 文件来确定客户正在访问哪些页面。CustomerCareHeaderSetup.jsp 文件中的以下代码将获取页面 URL 信息并更新购物者 applet:

```
<script language="javascript">
var PageName="";
var PersonalPage=false;
<%
    String pname = request.getRequestURI();
    int indpn = pname.lastIndexOf('/');
    indpn = pname.lastIndexOf('/', indpn-1);

    if(indpn != -1)
        pname = pname.substring(indpn+1);

String headerType = (String) request.getAttribute("liveHelpPageType");
if (headerType==null) headerType="";

// Determine if this is a personal page or not
if (headerType.equals("personal"))
{
    %>
    if (typeof parent.setPageParams == 'function') {
        PersonalPage=true;
        parent.setPageParams('PERSONAL_URL', '<%=pname%>');
    }
    <%
}
else
{
    %>
    if (typeof parent.setPageParams == 'function')
        parent.setPageParams(location.href, '%=pname%>');
    <%
}
    %>
    Pagename="<%=pname%>";
</script>
```

您应不允许 CSR 查看包含个性化信息的客户页面, 因为 CSR 所查看到的内容可能与客户查看到的内容不同。例如, CSR 不能有权访问竞销页面、包含由合同确定的价格的页面, 或包含用户标识的页面, 例如通讯录页面。应该把这些页面标为个人页面以避免在交谈会话期间误导 CSR。

为了将页面标为个人页面 (即对 CSR 不可用的页面), 样本商店在标题页面中包含了以下代码, 就在包含 CustomerCareHeaderSetup.jsp 文件的位置前。

```
<flow:ifEnabled feature="customerCare">
<%
request.setAttribute("liveHelpPageType", "personal");
%>
</flow:ifEnabled>
```

**注:** 尽管 CSR 无法看到标为个人页面的页面的内容, 但是 CSR 可以通过使用“查看客户页”按钮看到该页面的 URL。

## 跟踪购物车中的商品数

客户关心还让 CSR 能够随时跟踪客户的购物车中有多少商品。

CustomerCareInformationSetup.jsp 文件中的以下代码将获取购物车中商品的数目:

```

<%
JSPHelper jhelper = new JSPHelper(request);
String storeId = jhelper.getParameter("storeId");
int shoppingCartItems = 0;
%>
<
jsp:useBean id="userRDB" class="com.ibm.commerce.user.beans.
UserRegistrationDataBean" scope="page">
<% DataBeanManager.activate(userRDB, request); %></jsp:useBean>

<%
....
// need to check order items
OrderListDataBean orderListBean = new OrderListDataBean();
orderListBean.setStoreId(new Integer(storeId));
orderListBean.setOrderStatus("P");
orderListBean.setUserId(cmdcontext.getUserId());
DataBeanManager.activate(orderListBean, request);
Vector pendingOrders = orderListBean.getOrders();
for (int k=0; k< pendingOrders.size(); k++) {
  OrderAccessBean next_order = (OrderAccessBean) pendingOrders.
elementAt(k);
  OrderDataBean orderBean = new OrderDataBean();
orderBean.setOrderId(next_order.getOrderId());
DataBeanManager.activate(orderBean, request);
//Get items in the order
OrderItemDataBean [] orderItems = orderBean.getOrderItemDataBeans();
for (int i = 0; ((orderItems != null) &&
(i < orderItems.length)); i++)
{
  OrderItemDataBean orderItem = orderItems[i];
  shoppingCartItems += orderItem.getQuantityInEJBType().intValue();
}
}
....
%>

```

下面的 JavaScript 函数将向 Sametime applet 更新购物车信息:

```

<script language="javascript">
...
function changeSTAttributes()
{
  if (typeof top.setCustomerName == 'function') {
    top.setCustomerName(<%=userRDB.getUserId()%>,
'<%=customer_name%>');
    top.setShoppingCartItems(<%=shoppingCartItems%>);
    top.changeSTAttributes();
  }
}
</script>

```

**注:** CSR 可使用[查看购物车](#)按钮查看购物车的内容。关于更多信息, 请参阅 [WebSphere Commerce 联机帮助](#)。

样本商店通过上面的页面中添加以下代码来确定购物车中的商品数量:

- 首先定义一个 int 变量

```
int liveHelpShoppingCartItems= 0;
```
- 接下来, 无论何时在购物车中添加订购商品, 都使用以下代码行在 liveHelpShoppingCartItems 中添加数量:

```
liveHelpShoppingCartItems+= orderItem.getQuantityInEJBType().intValue();
```

- 然后，以下代码添加在页面的末尾，以便将客户姓名设置为临时购物者标识，并获取客户购物车中的商品数。

```
<script language="javascript">
  if (typeof parent.setShoppingCartItems == 'function')
    parent.setShoppingCartItems(<%=liveHelpShoppingCartItems%>);
</script>
```

以下代码用在空的购物车页面和订单确认页面中，以便将购物车中的商品数量重新设置为零：

```
<script language="javascript">
  if (typeof parent.setShoppingCartItems == 'function')
    parent.setShoppingCartItems(0);
</script>
```

## 跟踪定制的监视商品

为了跟踪定制的监视商品，请执行以下操作：

1. 定义监视属性标识（保留小于 8000 的属性标识）。关于更多信息，请参阅第 389 页的『定义商店的监视列表』。
2. 在 `Sametime.js` 文件中定义 JavaScript 变量以保存属性标识和值，并允许从所有框架访问。例如：`var myTrackAttributeId=8001; var myTrackAttributeValue="anything";`
3. 添加 JavaScript 代码，以便在商店页面中为此变量指定或更新值。例如：`top.myTrackAttributeValue="new Value";`
4. 调用 applet 的 `setWcsAttribute()` 方法更新 `Sametime.js` 文件中 `changeSTAttributes()` 函数中的属性：

```
function changeSTAttributes()
{
    if(CustomerAppletIsUp) {
        sametime.document.applets["InteractivePresenceApplet"].
setWcsAttribute(myTrackAttributeId,myTrackAttributeValue);
        sametime.document.applets["InteractivePresenceApplet"].
changeWCSAttrs();
    }
    else
        setTimeout("changeSTAttributes()",3000);
    // wait for 3 sec and try again
}
```

注：在 `changeWCSAttrs()` 方法之前插入您的代码。

## 将请求直接发送给客户关心队列

缺省情况下，当客户请求与 CSR 进行实时交谈时，客户的请求将被放置到缺省队列（`queueId=0`）中。然后由 CSR 负责将该请求路由到更加特定的队列。然而，您还可以将您的商店页面定制成将请求直接发送到已定义的客户关心队列中。要定制您的商店页面以将请求发送到已定义的队列，请执行以下操作：

1. 使用 WebSphere 贸易加速器为您的商店创建客户关心队列。关于更多信息，请参阅 WebSphere Commerce 产品联机帮助。
2. 将这些队列指定给 CSR。关于更多信息，请参阅 WebSphere Commerce 产品联机帮助。
3. 对您已创建的队列保留队列标识的记录。例如，队列标识为 10020。

4. 使用 JavaScript 更新商店 JSP 中的 reqQueue 变量。例如, top.reqQueue="10020";。

队列信息不会在 CSR 启动 CSR applet 后自动更新。如果将新队列添加到系统, 或现有的队列已更改, 则这些更改将到 CSR 再次启动 CSR applet 之后才会生效。

如果队列标识未由 CSR applet 识别, 则队列标识将放回 CSR 的缺省队列。此行为不会影响与客户关联的队列属性值。

## 定制客户关心

您可选择按以下方式定制客户关心:

- 定制 applet 参数
- 定制商店消息

### 定制 applet 参数

下表列出了可定制的 applet 参数, 这些参数使您能够通过修改 CustomerCareFrameSetup.jsp 文件来定制客户关心。

表 24.

可定制的参数名	描述	注释
CHAT_FONT_SIZE	要在交谈区使用的字体大小	缺省值为 12
CHAT_FONT_COLOR	用于交谈区中的进入消息的字体颜色。	缺省值为蓝色 (#0000FF)
CHAT_NAME_LENGTH	为在交谈区中显示用户名而保留的字符的长度。	缺省值为 15
WAIT_RANGE_1	整数值, 如果正在商店中等待的客户数少于该值, 则显示等待消息 1。否则, 等待消息根据 WAIT_RANGE_2 的设置显示。	当仅将显示消息 1 时使用 -1。
WAIT_RANGE_2	整数值, 如果正在商店中等待的客户数少于该值但多于 WAIT_RANGE_1, 则显示等待消息 2。否则, 等待消息将根据 WAIT_RANGE_3 的设置显示。	如果 WAIT_RANGE_1 为 -1, 则忽略它。使用 -1 可禁用此范围。
WAIT_RANGE_3	整数值, 如果正在商店中等待的客户数少于该值但多于 WAIT_RANGE_2, 则显示等待消息 3, 否则显示等待消息 4。	如果 WAIT_RANGE_2 为 -1, 则忽略它。使用 -1 可禁用此范围。
contentFrame	用于常规 WebSphere Commerce 商店页面的框架的名称	缺省值为 "main"
COUNTER_UNIT_PAGE	指示过多少时间页计数器会增加 1 的整数值	缺省值为 30 秒。请确保该值与商店的 CustomerCare MonitorList.jsp 文件中定义的值相同。

表 24. (续)

COUNTER_UNIT_SITE	指示过多少时间站点计数器会增加 1 的整数值	缺省值为 30 秒。请确保该值与商店的 CustomerCareMonitorList.jsp 文件中定义的值相同。
COUNTER_UNIT_WAIT	指示过多少时间等待计数器会增加 1 的整数值	缺省值为 30 秒。请确保该值与商店的 CustomerCareMonitorList.jsp 文件中定义的值相同。
WIDTH	交谈框架的首选宽度（以像素为单位）	缺省值为 360 个像素。邀请消息的长度将影响最终宽度。
HEIGHT	交谈框架的高度（以像素为单位）	缺省值为 400 <sup>®</sup> 个像素。

## 定制商店消息

当客户初始连接到 CSR 时显示给客户的消息（例如：“您好，我能为您做什么？”或“我们的办公时间从早上 9 点到晚上 9 点”）存储在 Sametime 服务器上的属性文件中。属性文件分为两种类型的文件：Customer.properties 和 Agent.properties。Customer.properties 文件包含显示给客户的消息，而 Agent.properties 文件包含显示给 CSR 的信息。对于安装在 WebSphere Commerce 实例中的每种语言环境，这两种文件也都具有对应的特定于语言环境的文件，例如 Customer\_de\_DE.properties 和 Agent\_de\_DE.properties。

要更改这些文件中的消息，请执行以下操作：

1. 在 Sametime 服务器上定位这些属性文件。缺省情况下，这些属性文件位于以下目录中：
  - *Customer\_Care\_installdir*\properties
2. 进行必要的更改。下表列出每条消息的消息键。

表 25.

消息键	描述	注释
WelcomeMessage	CSR 接受交谈请求之后将显示在客户的 applet 中的第一条消息。	
GoodbyeMessage	CSR 或客户结束交谈会话后马上显示在客户 applet 中的消息。	
PushPageMessage	CSR 向客户的浏览器发送 URL 后将显示在客户 applet 中的消息。	
CallCSRMessage	当客户提交一个帮助请求且 applet 正与 Sametime 服务器连接着时将显示的消息。	
NoConnectionMessage	当客户 applet 无法连接到 Sametime 服务器或断开与 Sametime 服务器的连接时将显示的消息。	



表 25. (续)

StoreCloseMessage	当没有 CSR 可为客户提供服务时将显示的消息。	总是与 StoreWorkingHour 消息一起显示。
StoreWorkingHour	当没有 CSR 可为客户提供服务时将显示的消息。描述商店的营业时间。	总是与 StoreCloseMessage 消息一起显示。
WaitingMessage	当客户提交请求且正在等待的客户总数少于 WAIT_RANGE_1 时将显示的消息。	
WaitingMessage1	当客户提交请求且正在等待的客户总数少于 WAIT_RANGE_2 但多于 WAIT_RANGE_1 时将显示的消息。	
WaitingMessage2	当客户提交请求且正在等待的客户总数少于 WAIT_RANGE_3 但多于 WAIT_RANGE_2 时将显示的消息。	
WaitingMessage3	当客户提交请求且正在等待的客户总数多于 WAIT_RANGE_3 时将显示的消息。	
CIWarningLabelLineOne	CSR 向某个客户启动交谈请求时显示的邀请消息的首行。	消息的显示长度将影响交谈框架的宽度。不应使用行分隔符。
CIWarningLabelLineTwo	CSR 向某个客户启动交谈请求时显示的邀请消息的第二行。	消息的显示长度将影响交谈框架的宽度。不应使用行分隔符。

注：应在消息中插入行分隔符“ ”来进行换行。否则，消息可能超出显示区的边界。

注：您可能需要针对不同的语言环境修改属性，以确保对消息的翻译是正确的。

3. 关闭并保存文件。

## 对商店添加客户关心

要对不是基于样本的商店添加客户关心，请执行以下操作：

### 第 1 部分：安装先决条件

为使客户关心可在商店中工作，必须执行以下操作：

- 安装 Sametime 服务器。关于更多信息，请参阅《WebSphere Commerce 附加软件指南》。
- 安装 WebSphere Commerce Sametime 集成软件包。关于更多信息，请参阅《WebSphere Commerce 附加软件指南》。

- 停止 WebSphere Commerce 实例，然后在配置管理器中启用 Sametime，再重新启动实例。关于更多信息，请参阅《WebSphere Commerce 附加软件指南》。
- 如果 Sametime 服务器所使用的 LDAP 服务器与 WebSphere Commerce 所使用的不同，则请使用管理控制台创建 CSR 并对客户关心注册该 CSR。关于更多信息，请参阅 WebSphere Commerce 产品联机帮助。

## 第 2 部分：从样本商店复制客户关心集成文件

样本商店“时尚潮流”和“多乐五金”包括以下文件，它们用来将客户关心集成到商店中：

- `Sametime.js`: 包含对所有框架均包括的 JavaScript 功能。此文件中的功能使用 `parent` 前缀从主框架的页面中调用，例如 `parent.setCustomerName`。
- `CustomerCareBlankSetup.jsp`: 空的 JSP 文件。`CCShopperBlankPageView` 命令将其作为框架的占位符调用。
- `CustomerCareFrameSetup.jsp`: 包含 JavaScript 函数并对商店前台嵌入客户关心 applet。它由 `CCShopperFramePageView` 命令调用，并在框架集中装入客户 applet。
- `CustomerCareAppletReadySetup.jsp`: 指出 applet 已正确装入。它由 `CCShopperReadyPageView` 命令调用，并指示对于 JavaScript 函数，客户 applet 已做好了准备。
- `CustomerCareHeaderSetup.jsp`: 将参数传递至 applet 的头文件，它指出包含此头的页面是否是个人页面。它必须包含在每个页面中，以便更新客户的页面 URL。
- `CustomerCareInformationSetup.jsp`: 更新客户的姓名信息和标识。它由 `CCShopperInfoUpdatePageView` 命令调用。
- `CustomerCareMonitorList.jsp`: 监视列表配置文件。它由 CSR 调用以装入监视列表。
- `CustomerCareStoreQuestionList.jsp`: 主题列表配置文件。它由 CSR 调用以装入商店主题列表
- `CustomerCareStoreURLList.jsp`: 商店级别 URL 列表配置文件。它由 CSR 调用以装入商店 URL 列表
- `CustomerCareChatSetup.jsp`: 用于在没有框架集的情况下使用客户关心的安装文件。它由 `CCChatPageView` 命令调用，并在非框架集配置中启动客户 applet。
- `EnvironmentSetup.jsp`: 由所有客户关心 JSP 文件使用，用于商店级别的配置（例如，资源束）。
- `index.jsp`: 根据客户关心商店流程设置，将客户的浏览器重定向至框架集页面或常规产品目录页面的入口页面。
- `StoreFramesetPage.jsp`: 由 `StoreFramesetView` 命令调用以构造客户关心的框架集。

要将 Sametime 集成文件从样本商店复制到您的商店，请执行以下操作：

1. 为消费者直销（“时尚潮流”商店）或 B2B 直销（“多乐五金”）商店定位商店归档文件。商店归档文件位于以下目录中：
  - `WC_installdir/samplestores`
2. 打开 `ConsumerDirect` 或 `B2BDirect` 文件夹，然后选择 `consumerdirect.sar` 或 `B2Bdirect.sar`。
3. 使用 WinZip 或类似的工具打开商店归档文件。
4. 选择上面列出的文件：

5. 将这些文件解压缩到包含商店 Web 有用资源的目录中。要保持与样本商店一样的目录结构, 您可能希望为以下文件创建子目录:

- ../CustomerServiceArea/CollaborationSection
  - CustomerCareAppletReadySetup.jsp
  - CustomerCareBlankSetup.jsp
  - CustomerCareBlankSetup.jsp
  - CustomerCareFrameSetup.jsp
  - CustomerCareInformationSetup.jsp
  - CustomerCareMonitorList.jsp
  - CustomerCareStoreQuestionList.jsp
  - CustomerCareStoreURLList.jsp
  - CustomerCareChatSetup.jsp
- /include
  - CustomerCareHeaderSetup.jsp
  - EnvironmentSetup.jsp
- ..\
  - index.jsp
  - Sametime.js
  - StoreFramesetPage.jsp

### 第 3 部分: 添加可确定客户正在浏览哪些页面的代码

要确定客户正在浏览的页面, 请执行以下操作:

1. 在商店的头文件中包含 CustomerCareHeaderSetup.jsp 文件, 例如:

```
<%@ include file="include\CustomerCareHeaderSetup.jsp" %>
```

2. 将以下代码添加到任何应标记为个人页面 (这样 CSR 就无法访问) 的页面中。确保在您包含 CustomerCareHeaderSetup.jsp 文件前添加以下代码。

```
<flow:ifEnabled feature="customerCare">
<%
request.setAttribute("liveHelpPageType", "personal");
%>
</flow:ifEnabled>=incfile%>"flush="true"/>
```

### 第 4 部分: 添加指向客户关心的链接

要让客户能够在商店中访问客户关心, 请执行以下操作:

1. 确定希望在何处放置指向客户关心的链接。例如, 您可能希望在导航栏中放置该链接, 以便客户始终可以使用它, 或者放置在商店的某些特定页面中。

2. 将以下代码复制到将要包含该链接的页面中:

**注:** 您可能需要将 infashiontext 替换为该商店资源所使用的对象名。

```
<a href="javascript:if((parent.sametime != null))
top.interact();"><%=infashiontext.getString("LiveHelp")%> </a>
```

## 第 5 部分: 创建将重定向至客户关心框架集页面的入口页面

因为大多数客户关心功能部件都需要框架集才能正常运行, 所以客户必须调用 `StoreFramesetView` 命令来激活框架集。有关示例, 请参阅消费者直销或 B2B 直销样本商店的 `index.jsp`。

---

## 第 42 章 向商店添加电子广告位

电子广告位在商店页面上保留一定的空间，在其中显示竞销活动的个性化市场营销内容。当客户请求页面时，该页面上显示的任何电子广告位将与规则服务器进行通信以处理与该广告位关联的基于规则的代码。每个电子广告位关联一个或多个活动。关于竞销和竞销活动的更多信息，请参阅第 199 页的第 20 章，『竞销有用资源』和 WebSphere Commerce 联机帮助。

为了在商店页面上正确显示竞销活动，必须向 JSP 文件添加一个电子广告位，然后使用 WebSphere 贸易加速器在数据库中注册它。本章将讨论如何向商店的 JSP 文件添加电子广告位。关于使用 WebSphere 贸易加速器在数据库中注册电子广告位的更多信息，请参阅 WebSphere Commerce 联机帮助。

---

### 电子广告位

以下是电子广告位的示例。

```
<!-- =====
/**-----
/** The sample contained herein is provided to you "AS IS".
/**
/** It is furnished by IBM as a simple example and has not been thoroughly tested
/** under all conditions. IBM, therefore, cannot guarantee its reliability,
/** serviceability or functionality.
/**
/** This sample may include the names of individuals, companies, brands and products
/** in order to illustrate concepts as completely as possible. All of these names
/** are fictitious and any similarity to the names and addresses used by actual persons
/** or business enterprises is entirely coincidental.
/**-----
/**
===== -->
<%
/**
 * START - the following code should exist only once in a page, it initialize the
 *       command context and store data bean.
 */

// create the store bean to get the store directory
String collateralPath = "/webapp/wcs/stores/";
com.ibm.commerce.command.CommandContext emsCommandContext =
    (com.ibm.commerce.command.CommandContext)
    request.getAttribute(com.ibm.commerce.server.ECConstants.EC_COMMANDCONTEXT);
com.ibm.commerce.common.beans.StoreDataBean storeDataBean =
    new com.ibm.commerce.common.beans.StoreDataBean();
storeDataBean.setStoreId(emsCommandContext.getStoreId().toString());
com.ibm.commerce.beans.DataBeanManager.activate(storeDataBean, request);
if (storeDataBean.getDirectory() != null) {
    collateralPath += storeDataBean.getDirectory() + "/";
}

/**
 * END - the following code should exist only once in a page, it initialize the
 *       command context and store data bean.
 */
%>

<!-- =====
// The following HTML form submits the request on the e-Marketing Spot to the ClickInfo
// command which captures the campaign statistics, and redirects to the location specified
// by the URL parameter.
===== -->
<form name="storeEmsForm" method="POST" action="/webapp/wcs/stores/servlet/ClickInfo">
  <input type="hidden" name="evtype">
  <input type="hidden" name="mpe_id">
  <input type="hidden" name="intv_id">
```

```

<input type="hidden" name="URL">
</form>

<%
/**
 * START - the following code can be used to drop multiple e-Marketing Spots onto the page.
 * Customize the appropriate EMarketingSpot instance name and the e-Marketing Spot
 * name before use. Duplicate this code if more than 1 spot is needed, but do not
 * use the same spot name.
 */

// create the e-Marketing Spot
com.ibm.commerce.marketing.beans.EMarketingSpot eMarketingSpot =
    new com.ibm.commerce.marketing.beans.EMarketingSpot();

// IMPORTANT - set the correct name here
String emsName = request.getParameter("emsName");
if (emsName == null) {
    emsName = "defaultEMSName";
}
eMarketingSpot.setName(emsName);

// Set the catalog entry ID that is currently displaying on the page. This is required if the
// up-sell/cross-sell initiative is based on the content of the current page.
String sourceCatEntryId = request.getParameter("sourceCatEntryId");
if (sourceCatEntryId != null) {
    eMarketingSpot.setSourceCatalogEntryId(sourceCatEntryId); // use this method to set single ID
    //eMarketingSpot.setMultipleSourceCatalogEntryId(sourceCatEntryId2);
    // use this method to set multiple IDs
}

// the maximum number of products/categories/ad copies that display through this
// e-Marketing Spot can be set here
eMarketingSpot.setMaximumNumberOfCatalogEntries(20);
eMarketingSpot.setMaximumNumberOfCategories(20);
eMarketingSpot.setMaximumNumberOfCollateral(20);
eMarketingSpot.setMaximumNumberOfAssociateCatalogEntries(20);

// instantiate the bean
com.ibm.commerce.beans.DataBeanManager.activate(eMarketingSpot, request);
%>

<%
// The following block is used to display the up-sell/cross-sell products associated with
// this e-marketing spot. The product display page which shows the selected product in the
// campaign will be referenced through the submission of the HTML form attached above.

if (eMarketingSpot.getAssociateCatalogEntries() != null
    && eMarketingSpot.getAssociateCatalogEntries().length < 0) {
%>
<TABLE>
<%
    for (int i=0; i<eMarketingSpot.getAssociateCatalogEntries().length; i++) {
        String associateCatalogEntryThumbNail = null;
        String associateCatalogEntryShortDescription = null;
        try {
            associateCatalogEntryThumbNail =
                eMarketingSpot.getAssociateCatalogEntries()[i].getDescription(emsCommandContext.
                    getLanguageId()).getThumbNail();
            associateCatalogEntryShortDescription =
                eMarketingSpot.getAssociateCatalogEntries()[i].getDescription(emsCommandContext.
                    getLanguageId()).getShortDescription();
        }
        catch (Exception e) {
            // no description defined for the current language
        }
%>
<TR>
<TD>
<A HREF="/webapp/wcs/stores/servlet/ClickInfo?evtype=CpgnClick&mpe_id=<%=
eMarketingSpot.getId() %>&intv_id=<%=
eMarketingSpot.getAssociateCatalogEntries()[i].getInitiativeId() %>
&URL=/webapp/wcs/stores/servlet/ProductDisplay&<%=
com.ibm.commerce.server.ECConstants.EC_STORE_ID %>=<%=
emsCommandContext.getStoreId().toString() %>&<%=
com.ibm.commerce.server.ECConstants.EC_PRODUCT_ID %>=<%=
eMarketingSpot.getAssociateCatalogEntries()[i].getCatalogEntryID() %>&<%=
com.ibm.commerce.server.ECConstants.EC_LANGUAGE_ID %>=<%=
emsCommandContext.getLanguageId().toString() %>">
<IMG SRC="<%= collateralPath + associateCatalogEntryThumbNail %>"

```

```

        ALT="<%= associateCatalogEntryShortDescription %>" BORDER=0 WIDTH=60>
    </A>
</TD>
    <TD><%= associateCatalogEntryShortDescription %></TD>
</TR>
<% } %>
</TABLE>
<% } %>

<%
// The following block is used to display the advertisements associated with this
// e-marketing spot. The URL link defined with an advertisement can be referenced through
// the submission of the HTML form attached above.

if (eMarketingSpot.getCollateral() != null && eMarketingSpot.getCollateral().length > 0) {
%>
<TABLE>
<% for (int i=0; i<eMarketingSpot.getCollateral().length; i++) { %>
<TR>
<% if (eMarketingSpot.getCollateral()[i].getTypeName().equals("Image")) { %>
<TD>
    <A HREF="javascript:document.storeEmsForm.evtype.value='CpgnClick';
    document.storeEmsForm.mpe_id.value='<%= eMarketingSpot.getId()
    %>';document.storeEmsForm.intv_id.value='<%=
    eMarketingSpot.getCollateral()[i].getInitiativeId()
    %>';document.storeEmsForm.URL.value='<%=
    eMarketingSpot.getCollateral()[i].getUrlLink()
    %>';document.storeEmsForm.submit();">
    <IMG SRC="<%= collateralPath + eMarketingSpot.getCollateral()[i].getLocation() %>">
    </A>
</TD>
<TD>
    <%= eMarketingSpot.getCollateral()[i].getMarketingText() %>
</TD>
<% } else if (eMarketingSpot.getCollateral()[i].getTypeName().equals("Flash")) { %>
<TD>
    <EMBED src="<%= collateralPath + eMarketingSpot.getCollateral()[i].getLocation()
    %>" quality=high bgcolor=#FFFFFF WIDTH=120 HEIGHT=90
    TYPE="application/x-shockwave-flash"></EMBED>
</TD>
<TD>
    <A HREF="javascript:document.storeEmsForm.evtype.value='CpgnClick';
    document.storeEmsForm.mpe_id.value='<%= eMarketingSpot.getId()
    %>';document.storeEmsForm.intv_id.value='<%=
    eMarketingSpot.getCollateral()[i].getInitiativeId()
    %>';document.storeEmsForm.URL.value='<%=
    eMarketingSpot.getCollateral()[i].getUrlLink()
    %>';document.storeEmsForm.submit();">
    <%= eMarketingSpot.getCollateral()[i].getMarketingText() %>
    </A>
</TD>
<% } %>
</TR>
<% } %>
</TABLE>
<% } %>

<%
// The following block is used to display the categories associated with this e-marketing
// spot. The category display page which shows the selected category in the campaign will
// be referenced through the submission of the HTML form attached above.

if (eMarketingSpot.getCategories() != null && eMarketingSpot.getCategories().length > 0) {
%>
<TABLE>
<%
for (int i=0; i<eMarketingSpot.getCategories().length; i++) {
    String catalogGroupName = null;
    String catalogGroupLongDescription = null;
    try {
        catalogGroupName = eMarketingSpot.getCategories()[i].getDescription
            (emsCommandContext.getLanguageId()).getName();
        catalogGroupLongDescription = eMarketingSpot.getCategories()[i].getDescription
            (emsCommandContext.getLanguageId()).getLongDescription();
    }
    catch (Exception e) {
        // no description defined for the current language
    }
%>
<TR>

```

```

<TD>
  <A HREF="/webapp/wcs/stores/servlet/ClickInfo?evtype=CpgnClick&mpe_id=<%=
    eMarketingSpot.getId() %>&intv_id=<%=
    eMarketingSpot.getCategories()[i].getInitiativeId()
    %>&URL=/webapp/wcs/stores/servlet/CategoryDisplay&<%=
    com.ibm.commerce.server.ECConstants.EC_STORE_ID %>=<%=
    emsCommandContext.getStoreId().toString() %>&<%=
    com.ibm.commerce.server.ECConstants.EC_CATEGORY_ID %>=<%=
    eMarketingSpot.getCategories()[i].getCategoryId() %>&<%=
    com.ibm.commerce.server.ECConstants.EC_CATALOG_ID %>=<%=
    eMarketingSpot.getCategories()[i].getCatalogId() %>&<%=
    com.ibm.commerce.server.ECConstants.EC_LANGUAGE_ID %>=<%=
    emsCommandContext.getLanguageId().toString() %>">
    <%= catalogGroupName %>
  </A>
</TD>
<TD><%= catalogGroupLongDescription %></TD>
</TR>
<% } %>
</TABLE>
<% } %>

<%
// The following block is used to display the products associated with this e-Marketing
// Spot. The product display page which shows the selected product in the campaign will
// be referenced through the submission of the HTML form attached above.

if (eMarketingSpot.getCatalogEntries() != null &&eMarketingSpot.getCatalogEntries().length > 0) {%>
<TABLE>
<%
for (int i=0; i<eMarketingSpot.getCatalogEntries().length; i++) {
  String catalogEntryThumbNail = null;
  String catalogEntryShortDescription = null;
  try {
    catalogEntryThumbNail =
      eMarketingSpot.getCatalogEntries()[i].getDescription(emsCommandContext.
        getLanguageId()).getThumbNail();
    catalogEntryShortDescription =
      eMarketingSpot.getCatalogEntries()[i].getDescription(emsCommandContext.
        getLanguageId()).getShortDescription();
  }
  catch (Exception e) {
    // no description defined for the current language
  }
%>
<TR>
<TD>
  <A HREF="/webapp/wcs/stores/servlet/ClickInfo?evtype=CpgnClick&mpe_id=<%=
    eMarketingSpot.getId() %>&intv_id=<%=
    eMarketingSpot.getCatalogEntries()[i].getInitiativeId()
    %>&URL=/webapp/wcs/stores/servlet/ProductDisplay&<%=
    com.ibm.commerce.server.ECConstants.EC_STORE_ID %>=<%=
    emsCommandContext.getStoreId().toString() %>&<%=
    com.ibm.commerce.server.ECConstants.EC_PRODUCT_ID %>=<%=
    eMarketingSpot.getCatalogEntries()[i].getCatalogEntryID()
    %>&<%= com.ibm.commerce.server.ECConstants.EC_LANGUAGE_ID %>=<%=
    emsCommandContext.getLanguageId().toString() %>">
    <IMG SRC="<%= collateralPath + catalogEntryThumbNail %>"
      ALT="<%= catalogEntryShortDescription %>" BORDER=0 WIDTH=60>
  </A>
</TD>
<TD><%= catalogEntryShortDescription %></TD>
</TR>
<% } %>
</TABLE>
<% } %>

<%
/**
 * END - the following code is used to drop multiple e-Marketing Spots onto the page.
 *       Customize the appropriate e-Marketing Spot name before use.
 *       Duplicate this code if more than 1 spot is needed, but do not use the same spot name.
 */
%>

```

前述电子广告位支持 4 种竞销活动:

- 产品推荐



- 类别推荐
- 推广广告
- 销售关联

注：关于每个活动的更多详细信息，请参阅第 199 页的第 20 章，『竞销有用资源』。

## 电子广告位 bean

电子广告位使用电子广告位 bean 返回当前调度到广告位的竞销活动的结果。使用 bean 的不同属性将允许您定制电子广告位和相应的竞销活动。关于电子广告位 bean 及其属性的更多信息，请参阅 WebSphere Commerce 联机帮助。

---

## 向商店页面添加电子广告位

为了向商店页面添加电子广告位，请执行以下操作：

1. 确定广告位将在哪个 JSP 文件上显示。广告位可以添加到多个 JSP 文件。
2. 确定在 JSP 文件上何处放置广告位。
3. 将第 407 页的『电子广告位』中的样本电子广告位复制到商店 Web 应用程序中的新 JPS 文件。
4. 定制样本电子广告位以符合 JSP 文件的布局。
5. 在电子广告位代码中，对电子广告位给定一个名称。

注：电子广告位的命名应该是描述性的，以便包含它们的位置（例如 HomePageAd 或 CheckOutPageRecommendation）。这有助于减小有关广告位出现位置以及广告位包含内容的混乱。如果必要，可以向名称添加数字以区分同一页面上出现的两个电子广告位。电子广告位名称必须是有效的 Java 标识。在使用 WebSphere 贸易加速器在数据库中注册电子广告位时，您必须使用此同一名称。

6. 通过使用 `<jsp:include>` 标记动态地包含电子广告位，将电子广告位添加到 JSP 文件中。
7. 如果要求每个 JSP 文件有一个以上的电子广告位，则重复步骤 2 到 6。
8. 使用 WebSphere 贸易加速器在数据库中注册电子广告位。关于详细指示信息，请参阅 WebSphere Commerce 联机帮助。

注：

- a. 如果您计划使用以下约定 `"langId=<%= languageId %>"` 向 URL 添加商店标识、产品目录标识或语言标识，请注意电子广告位所嵌入的 JSP 必须使相应的标识可用。还可以通过命令上下文来检索标识，例如 `getCommandContext().getLanguageId(?)`。
- b. URL 参数 `CatalogDisplay` 应该以 `"&"` 开头而不是 `"?"`，因为此代码不直接引用命令。
- c. 通过使用 Java 动态 include 标记将电子广告位添加到商店 JSP，可以启用新的 Dynacache 功能，这样将会对 JSP 的内容（不包括电子广告位）进行高速缓存。每次访问时电子广告位均会刷新，从而显示正确的动态内容。这里是一个示例，说明如何能使用动态 include 来将电子广告位添加到名为 `ESpot.jsp` 的 JSP 文件中：

```
<jsp:include page="ESpot.jsp" flush="true">
<jsp:param name="emsName" value="StoreHomePage" />
</jsp:include>
```

- d. 如果正在将某个销售关联活动调度到电子广告位，且关联的源基于页面的内容，则可以使用在 `com.ibm.commerce.marketing.beans.EMarketingSpot` 中提供的 `setSourceCatalogEntryId(String source)` 和 `setMultipleCatalogEntryId(String source)` 方法对该销售关联进行设置。例如，在“产品显示”页面，如果页面中显示的产品将用作关联的源，则会调用以下方法：

```
eSpot.setSourceCatalogEntryId(productId)
```

其中 *eSpot* 是 `com.ibm.commerce.marketing.beans.EMarketingSpot` 类的一个实例，而 *productId* 是源产品的标识。

---

## 第 12 部分 附录



## 附录 A. UML 图注

统一建模语言是用于表示软件设计的不同元素的标准图形语言。以下示例是一些最常用的 UML 元素。关于正式规范的进一步详细信息，请参阅 <http://www.rational.com> 和 <http://www.omg.org>。

UML 图由以下项组成：

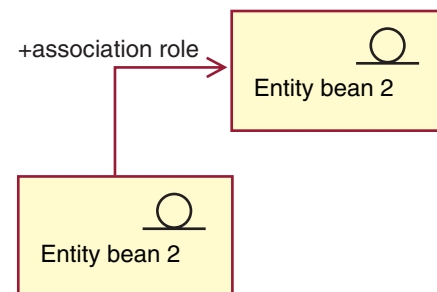
- 框：框代表对象的类。类名出现在框的顶部。属性如果显示，则出现在类名下方。类名和属性用一条线分开。
- 线：线代表两个类的对象之间的可能关系。线一端上类的对象可与线另一端上的类相“关联”。
- 实心菱形：线末端的实心菱形指示按值保存。线另一端上类的对象是菱形触及的一个且仅一个类的对象的组成部分。
- 空心菱形：线末端的空心菱形指示按引用保存。线的菱形末端上的对象可以视为线另一端上类的分组对象。
- 基数：这些基数出现在关系线的末端以指示基数限制。下表总结了基数限制：

基数	关系类型
1	一种且仅一种
0..1	零种或一种
0..n	零种或多种
1..n	一种或多种

如果未显示任何基数限制，则假定基数为 0..n，除非关系线末端出现实心菱形。在那种情况下，基数必须为 1。

- 加号：出现在关系线末端的加号指示线末端上类的对象在关系中充当了角色。加号后面的文本指示对象在关系中的角色。
- 箭头：关系线末端的箭头指示两个对象之间关系的方向是沿着箭头方向的。关系线上没有箭头时指示对象之间的关系通常是双向的。

下图说明了以上概念：



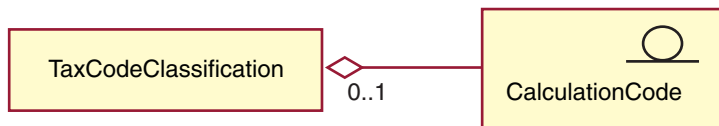
此图显示了两个实体 bean，它们以固定的装饰符号指示 Enterprise Java Bean。存在一个从第一个 bean 到第二个实体 bean 的单向关联。加号后面跟着描述实体 bean 2 在关联充当什么角色的文本。



在此图中，StoreEntity 有且仅有一个所有者，即 Member。Member 可以有零个或多个 StoreEntity。加号指示该 Member 在关系中充当了角色。在此情况下，Member 是 StoreEntity 的所有者。箭头指示您将通常通过向 StoreEntity 询问其所有者，而不是向 Member 询问其拥有的所有 StoreEntity，来找到 StoreEntity 的所有者。



在此图中，OrderItem 总是一个且仅一个 Order 的组成部分。Order 有零个或多个 OrderItem。



此图指示 CalculationCode 按零个或一个 TaxCodeClassification 分组，而 TaxCodeClassification 对零个或多个 CalculationCode 分组。

---

## 附录 B. 创建数据

在以 XML 文件格式创建商店数据之前，请执行以下操作：

- 请复查第 323 页的第 37 章，『装入商店数据的概述』中的信息。
- 确定正在创建的信息顺序。每个商店数据章节中的信息提示您创建数据的顺序，但是在创建 XML 文件时请切记父表的信息必须在子表的信息之前。关于装入有用资源的顺序的更多信息，请参阅第 367 页的第 38 章，『装入 WebSphere Commerce 数据库有用资源组』。

---

### 创建样本商店的数据

样本商店归档文件中的数据采用组织良好的、对装入程序软件包有效的 XML 文件。商店归档 XML 文件计划成为可移植文件并且不应包含特定于数据库特定实例的已生成主键。相反，它们使用发布时由标识解析器解析的内部别名。这些约定的使用允许对样本商店归档文件进行多次复制和发布。



以 XML 文件格式创建商店的商店数据时不必使用这些约定，除非您打算创建将用于生成多个商店的样本商店归档文件，或者除非您希望创建的商店归档文件可移植，或者可以发布到另一个 WebSphere Commerce 实例中。

因此，样本商店归档文件使用以下约定：

- @，如 `ffmcenter_id="@ffmcenter_id_1"` 中的情况。@ 符号的使用称为内部别名解析。标识解析器（装入程序软件包的一个实用程序）为需要标识的 XML 元素生成标识。标识解析器使用的技术之一是内部别名解析。使用内部别名解析时，以别名替换 XML 文档中的主键（标识）。然后在 XML 文件的其它位置使用该别名指代该元素。这样就无需了解构建 XML 文件所需的唯一索引。在管理控制台中发布期间，或使用装入程序软件包时，标识解析器会以独特值替换 @ 符号。请参阅来自某 XML 文件的以下示例：

– 使用标识解析器前

```
<catalog
catalog_id="@catalog_id_1"
member_id("&MEMBER_ID;"
identifer="FashionFlow"
description="FashionFlow Catalog"
tpclevel="0"/>
```

– 使用标识解析器后

```
<catalog
catalog_id="10001"
member_id="-2000"
identifer="FashionFlow"
description="FashionFlow Catalog"
tpclevel="0"/>
```

其中 10001 是由标识解析器指定的唯一标识，而 -2000 则是用户在管理控制台中选择的成员标识。然后使用装入程序软件包装入产生的 XML 文件。通过标识解析器运行文件确保可从一组 XML 文件创建多个商店。





---

## 附录 C. 数据库有用资源组

所有 WebSphere Commerce 数据库有用资源都分到不同的组中以供创建和装入。这些组是逻辑上相关的一组表。组织这些数据库有用资源组的顺序对于数据装入很重要，因为在装入对象之间的关系之前某些对象必须存在。

当装入 XML 格式的数据库有用资源时，可选择仅装入选定的组。这些组由先前几章中所创建的数据库有用资源（例如产品目录或实现）所组成。在如第 373 页的『装入数据库有用资源组』中所指导装入数据组之前，请执行以下操作：

- 确定正在装入哪个数据库有用资源组。每个组都包含一些相关性，在可装入有用资源之前必须满足这些相关性。请复查『数据库有用资源组相关性』中的信息。
- 确保已对选定的数据库有用资源组创建或更新了 XML 文件。每个有用资源章中的信息建议了创建数据库有用资源的顺序，但是在创建或更新 XML 文件时，请切记父表的信息必须在子表的信息之前。

---

### 数据库有用资源组相关性

每个数据库有用资源组都从 WebSphere Commerce 数据库表中抽取它的信息。在它们自己的组中，数据库有用资源具有相关性。即，数据库有用资源组无法从不同的数据组中抽取其它 XML 文件中的数据，且每个组都是独立的（除了外键之外）。然而，如果数据库有用资源需要引用其它组中定义的外部数据，则您需要手工提供该数据。这意味着，一个组中的数据对在其域之外定义的数据（即，其它数据库有用资源组上的数据）具有外部相关性。当数据库有用资源组对其它组中的表的主键具有外键关系时，则发生外部相关性。要装入数据库有用资源，必须满足它的外部相关性。要使用以下图表中的示例，商店数据库有用资源组的一个外部相关性是 fulfillment.FFMCENTER.FFMCENTER\_ID，它指示在可以装入商店数据库有用资源组之前，实现数据库有用资源必须已存在于 WebSphere Commerce 数据库中。

在您开始装入过程之前，请考虑以下图表。每组数据库有用资源都依赖于其它数据库表（数据从这些数据库表中装入）。


要记住的一些要点：

- 单个组可能不满足一些外部相关性。每个商店使用的站点范围内的数据库有用资源或常规数据库有用资源在实例创建时在 *bootstrap* 中已预填充，并已准备就绪可以被访问。数据库有用资源组中包含的表具有对此类型数据的外键引用。引导程序数据被分为公共数据和特定于语言环境的数据。如果您具有多语言的商店，则需要选择公共的引导程序数据和特定于语言环境的引导程序数据。例如，您需要语言和成员引导程序数据。实例创建过程用商店支持的 WebSphere Commerce 语言填充 LANGUAGE 表，并创建根组织（MEMBER.MEMBER\_ID=-2001）和缺省组织（MEMBER.MEMBER\_ID=-2000）。您必须在需要的地方使用根组织，但是应该创建一个商店所有者组织，而不是使用缺省组织。关于组织及其层次结构的更多信息，请参阅 WebSphere Commerce 联机帮助。
- 外部相关性列下列出的文件使用以下命名结构：*database asset group.database table.database column*。使用 store.STOREENT.STOREENT\_ID 文件作为示例，数据取自

*store* 数据库有用资源组、*STOREENT* 表和 *STOREENT\_ID* 列。以 *bootstrap* 开头的文件名表示数据是在 WebSphere Commerce 实例创建期间填充的。

- 在**外部相关性**下列出的文件包含了对**相关表**的外键引用。必须首先填充这些表。
- 仅为了显示的目的，这些表被分割以表示包含有多语言信息（例如，产品描述）的特定于语言环境的表。
- 图表中的表代表 WebSphere Commerce 样本商店中的数据库有用资源。根据商店大小、功能和需要，这些表可能有所不同。根据您的商店的需要，确保包含了所有含有商店有用资源的数据库表，即使该特定表未列在下面。

访问控制数据库有用资源		
外部相关性	数据库有用资源 XML 文件中的相关表	数据库有用资源 XML 文件中有关特定于语言环境的表
bootstrap.LANGUAGE.LANGUAGE_ID（根和商店所有者组织）， bootstrap.MEMBER.MEMBER_ID（根和商店所有者组织）	accesscontrol.xml ACACTACTGP, ACACTGRP, ACACTION, ACPOLICY, ACRESCGRY, ACRESGPRES, ACRESGRP	accesscontrol.xml ACACGPDESC, ACACTDESC, ACPOLDESC, ACRSCGDES, ACRESGPDES
业务策略数据库有用资源		
外部相关性	数据库有用资源 XML 文件中的相关表	数据库有用资源 XML 文件中有关特定于语言环境的表
bootstrap.LANGUAGE.LANGUAGE_ID, bootstrap.MEMBER.MEMBER_ID, store.STOREENT.STOREENT_ID（商店所有者组织）	businesspolicy.xml POLICY, POLICYCMD	businesspolicy.xml POLICYDESC
竞销数据库有用资源		
外部相关性	数据库有用资源 XML 文件中的相关表	数据库有用资源 XML 文件中有关特定于语言环境的表
store.STOREENT.STOREENT_ID	campaign.xml CAMPAIGN, COLLATERAL, EMSPOT	campaign.xml COLLDESC
产品目录数据库有用资源		
外部相关性	数据库有用资源 XML 文件中的相关表	数据库有用资源 XML 文件中有关特定于语言环境的表

bootstrap.LANGUAGE.LANGUAGE_ID, bootstrap.MEMBER.MEMBER_ID (商店所有者组织), store.STOREENT.STOREENT_ID, shipping.CALCODE.CALCODE_ID, tax.CALCODE.CALCODE_ID	<b>catalog.xml</b> BASEITEM, CATALOG, CATENTREL, CATENTRY, CATGROUP, CATGRPREL, CATTOGRP, ITEMSPC, ITEMVERSN, RA, RADETAIL, STOREITEM, STORITMFFC, VERSIONSPC <b>offering.xml</b> CATGRPTPC, MGPTRDPSCN, OFFER, OFFERPRICE, TRADEPOSCN <b>storefulfill.xml</b> INVENTORY <b>store-catalog.xml</b> DISPCGPREL, DISPENTREL, STORECAT, STORECENT, STORECGRP <b>store-catalog</b> <b>-shipping.xml</b> CATENTCALCD, CATENTSHIP <b>store-catalog-tax.xml</b> CATENTCALD	<b>catalog.xml</b> ATTRIBUTE, ATTRVALUE, BASEITMDSC, CATALOGDSC, CATENTDESC, CATGRPDESC, PKGATTR, PKGATTRVAL,
命令数据库有用资源		
外部相关性	数据库有用资源 <b>XML</b> 文件中的相关表	数据库有用资源 <b>XML</b> 文件中有关特定于语言环境的表
store.STOREENT.STOREENT_ID	<b>command.xml</b> CMDREG, VIEWREG	N/A
 <b>合同数据库有用资源</b>		
外部相关性	数据库有用资源 <b>XML</b> 文件中的相关表	
store.STOREENT.STOREENT_ID	合同数据库表不是直接装入的, 且遵循与其它 WebSphere Commerce 数据组不同的过程。关于更多信息, 请参阅第 378 页的『发布合同有用资源』。	
货币数据库有用资源		

外部相关性	数据库有用资源 <b>XML</b> 文件中的相关表	数据库有用资源 <b>XML</b> 文件中有关特定于语言环境的表
store.STOREENT.STOREENT_ID	<b>currency.xml</b> CURCVLIST	<b>currency.xml</b> CURCONVERT, CURLIST
供货数据库有用资源		
外部相关性	数据库有用资源 <b>XML</b> 文件中的相关表	数据库有用资源 <b>XML</b> 文件中有关特定于语言环境的表
bootstrap.LANGUAGE.LANGUAGE_ID, bootstrap.MEMBER.MEMBER_ID (商店所有者组织)	<b>fulfillment.xml</b> FFMCENTER, STADDRESS	<b>fulfillment.xml</b> FFMCENTDS
组织数据库有用资源		
外部相关性	数据库有用资源 <b>XML</b> 文件中的相关表	数据库有用资源 <b>XML</b> 文件中有关特定于语言环境的表
bootstrap.LANGUAGE.LANGUAGE_ID (根和商店所有者组织), bootstrap.MEMBER.MEMBER_ID (根和商店所有者组织)	<b>organization.xml</b> ADDRBOOK, ADDRESS, MBRREL, MEMBER, ORGENTITY	N/A
送货数据库有用资源		
外部相关性	数据库有用资源 <b>XML</b> 文件中的相关表	数据库有用资源 <b>XML</b> 文件中有关特定于语言环境的表
bootstrap.LANGUAGE.LANGUAGE_ID, bootstrap.MEMBER.MEMBER_ID (商店所有者组织), fulfillment.FFMCENTER.FFMCENTER_ID, store.STOREENT.STOREENT_ID	<b>shipping.xml</b> CALCODE, CALRULE, CRULESCALE, JURST, JURSTGPREL, JURSTGROUP, SHIPMODE, STENCALUSG <b>shipping.xml</b> SHPJCRULE, SHPARRANGE	<b>shipping.xml</b> CALCODEDSC, CALRANGE, CALRLOOKUP, CALSCALE, SHPMODEDSC
商店数据库有用资源		
外部相关性	数据库有用资源 <b>XML</b> 文件中的相关表	数据库有用资源 <b>XML</b> 文件中有关特定于语言环境的表

bootstrap.LANGUAGE.LANGUAGE_ID, bootstrap.MEMBER.MEMBER_ID (商店所有者 组织), bootstrap.SETCURR.SETCURR_ID, fulfillment.FFMCENTER.FFMCENTER_ID	<b>store.xml</b> INVADJCODE, RTNREASON, STORE, STORENT, STORELANG, VENDOR	<b>store.xml</b> FFMCENTDS, INVADJDESC, RTNRSNDESC, STADDRESS, STOREENTDS, STORLANGDS, VENDORDESC
商店缺省数据库有用资源		
外部相关性	数据库有用资源 <b>XML</b> 文件中的相关表	数据库有用资源 <b>XML</b> 文 件中有关特定于语言环境 的表
shipping.SHIPMODE.SHIPMODE_ID (如果适 用, 必须首先装入此文件), contract.CONTRACT.CONTRACT_ID, store.STOREENT.STOREENT_ID	<b>store-default.xml</b> STOREDEF	N/A
税款数据库有用资源		
外部相关性	数据库有用资源 <b>XML</b> 文件中的相关表	数据库有用资源 <b>XML</b> 文 件中有关特定于语言环境 的表
bootstrap.LANGUAGE.LANGUAGE_ID, bootstrap.MEMBER.MEMBER_ID (商店所有者组 织), store.STOREENT.STOREENT_ID, fulfillment.FFMCENTER.FFMCENTER_ID, store.STOREENT.STOREENT_ID	<b>tax.xml</b> CALCODE, CALRANGE, CALRLOOKUP, CALRULE, CALSCALE, CRULESCALE, JURST, JURSTGROUP, JURSTGPREL, STENCALUSG, TAXCGRY, TAXJCRULE <b>taxfulfill.xml</b> TAXJCRULE	<b>tax.xml</b> CALCODEDSC, CALSCALEDS, TAXCGRYDS



---

## 声明

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

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

IBM Director of Licensing  
IBM Corporation  
500 Columbus Avenue  
Thornwood, NY 10594  
U.S.A.

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

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

**本条款不适用英国或任何这样的条款与当地法律不一致的国家或地区：**

国际商业机器公司以“按现状”的基础提供本出版物，不附有任何形式的（无论是明示的，还是默示的）保证，包括（但不限于）对非侵权性、适销性和适用于某特定用途的默示保证。某些国家或地区在某些交易中不允许免除明示或默示的保证。因此本条款可能不适用于您。

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

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

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

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

IBM Canada Ltd.  
Office of the Lab Director  
8200 Warden Avenue Markham, Ontario L6G 1C7  
Canada

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

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

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

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

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

显示的所有 IBM 的价格均是 IBM 当前的建议零售价，可随时更改而不另行通知。经销商的价格可与此不同。

本信息仅为了规划目的。其中的信息在描述的产品可以使用之前会得到更改。

本信息包含日常业务经营中使用的数据和报告示例。为尽可能表述完整，这些示例包括了个人姓名、公司、品牌和产品的名称。所有这些名称均系虚构，如与实际企业所用的名称和地址雷同，纯属巧合。

版权许可：

本信息包括源语言形式的样本应用程序，这些样本说明不同操作平台上的编程方法。如果是为按照在编写样本程序的操作平台上的应用程序编程接口（API）进行应用程序的开发、使用、经销或分发为目的，您可以任何形式对这些样本程序进行复制、修改、分发，而无须向 IBM 付费。这些示例并未在所有条件下作全面测试。因此，IBM 不能担保或暗示这些程序的可靠性、可维护性或功能。用户如果是为了按照 IBM 应用程序编程接口开发、使用、经销或分发应用程序，则可以任何形式复制、修改和分发这些样本程序，而无须向 IBM 付费。

凡这些样本程序的每份拷贝或其任何部分或任何衍生品，都必须包括如下版权声明：

©Copyright International Business Machines Corporation 2001. 此部分代码是根据 IBM 公司的样本程序衍生出来的。©Copyright IBM Corp. 2000, 2001. All rights reserved.

如果您正以软拷贝格式查看本信息，图片和彩色图例可能无法显示。

本产品中提供的信用卡图像、商标和商品名称，仅供信用卡标记拥有者特许的商家用于接受该信用卡付款。



---

## 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	IBM	WebSphere
AS/400®	IBM Payment Manager	
DB2	iSeries	
DB2 Universal	OS/400	
Database		
eServer	VisualAge®	

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

Oracle 和 Oracle8 分别是 Oracle Corporation 的注册商标和商标。

SET Secure Electronic Transaction™、SET™ 和 SET 徽标是归 SET Secure Electronic Transaction LLC 所有的商标。严禁未经 SET Secure Electronic Transaction LLC 书面许可而使用这些商标。

Solaris、Solaris Operating Environment、Java、JavaServer Pages、JavaBeans、和所有基于 Java 的商标和徽标是 Sun Microsystems, Inc. 的商标或注册商标。

UNIX® 是 The Open Group 在美国和其它国家或地区的注册商标。

其它公司、产品和服务名称可能是其它公司的商标或服务标记。



中国印刷