

IBM WebSphere Commerce



连通性和通知：联机帮助文件

版本 5.4

IBM WebSphere Commerce



连通性和通知：联机帮助文件

版本 5.4

注意！

在使用本资料及其支持的产品之前，请务必阅读『声明』中的一般信息。

目录

第 1 章 程序适配器	1	第 6 章 启用出站消息传递发送 / 接收发送服务	35
CommandProperty 对象	1	启用消息传递系统传送适配器	35
设备格式算法	2	指定出错条件消息类型至传送方法	36
HTTP 上的 XML	2	激活商店的传送方法	36
MQSeries 作为中间件	3	将传送方法添加到商店	36
WebSphere Commerce MQSeries 适配器	3	配置商店的传送方法	37
MQSeries 适配器中的并行 — 串行消息处理	3	停用商店的传送方法	37
第 2 章 配置程序适配器	5	将传送方法指定到商店的消息类型	37
为 HTTP 上的 XML 请求启用程序适配器	5	激活站点的传送方法	38
添加适配器	5	添加传送方法到站点	38
下载并安装 MQSeries MA88 产品扩展包	6	将传送方法指定到站点的消息类型	38
启用 MQSeries 适配器	7	配置站点的传送方法	38
配置 MQSeries 的 JMS	8	停用站点的传送方法	39
更新 WebSphere Application Server 类路径变量	9	检查电子邮件传送方法的系统设置	39
使用 JMSAdmin 配置 JMS	9	激活通知	39
使用 MQSeries 适配器响应处理	11	启用错误通知	40
第 3 章 MQSeries 的程序适配器安全性	13	激活通知	40
HTTP 请求的程序适配器安全性	13	启用装运通知电子邮件	41
消息编辑服务	14	启用广播式消息	41
消息传递系统	14	启用订单状态通知	42
一般应用程序和系统错误 XML 消息	14	设置出站消息编辑	42
设置出站消息编辑	15	指定出错条件消息类型至传送方法	42
使用消息传递系统编辑服务的示例	15	启用订单状态通知	43
消息传递系统编辑服务中的出错处理	16	启用从管理控制台发送的消息	43
调用消息传递系统编辑方法	17	启用出站 OrderCreate 消息	44
第 4 章 OrderItemStatus 命令	19	第 7 章 消息类型	45
GetPickPackListDetail 命令	21	出站消息传递系统接口	46
BroadcastMessage 命令	21	出站消息扩展	47
OrderInvoiceStatus 命令	23	新出站消息支持	48
OrderShippingStatus 命令	23	出站消息的 UserData 元素	48
OrderStatus 命令	23	出站消息传递系统接口编程示例	48
ProductOfferPriceUpdate 命令	25	消息编辑模板	50
ProductInventoryUpdate 命令	26	初始化服务	51
SendWCSOrder 任务命令	27	消息内容设置服务	51
SendXMLOrder 命令	27	添加电子邮件部件或附件服务	51
NewInboundMessage 命令	28	可配置消息数据服务	52
OrderConfirmStatus 命令	28	发送服务	52
ProductListPriceUpdate 命令	28	其它服务	53
第 5 章 出站消息传递系统如何工作	31	第 8 章 入站后端集成消息	55
出站消息传递系统	31	入站实现集成消息	56
出站消息传递系统管理	32	第 9 章 添加新的入站 XML 消息	59
出站后端集成消息	32	添加新的 DTD 文件至系统	59
出站实现集成消息	33	添加至入站消息 DTD 文件列表	59
出站消息传递系统商店管理	34	入站消息扩展	60
出站消息传递系统站点管理	34	入站消息的 UserData 元素	60
向传送适配器添加新消息	34		

第 10 章 消息映射器	61	Update_WCS_InventoryReceipt 消息	101
XML 消息映射器	61	Create_WCS_ShipmentConfirmation 消息	101
旧消息映射器	61	Create_WCS_Customer 消息	102
入站消息模板定义文件	62	Update_WCS_Customer 消息	103
除去消息映射器	62	Update_WCS_ProductInventory 消息	105
添加消息映射器	63	“产品价格更新”消息	106
新入站消息支持	63	“产品数量更新”消息	107
		“订单创建”消息	107
第 11 章 定制 NewInboundMessage 命令	65	顾客新消息	109
		顾客更新消息	110
第 12 章 消息映射器配置	67	Update_NC_Customer 消息	111
使用模板定义文件进行 XML 分析	67	Update_NC_OrderStatus 消息	113
sys_template.xml 文件	67	Create_NC_Customer 消息	115
user_template.xml 文件	68	Update_NC_ProductInventory 消息	116
ec_template.dtd 文件	68	Update_NC_ProductPrice 消息	116
模板定义文件的 TemplateDocument 元素	69	Report_NC_PurchaseOrder 消息	117
模板定义文件的 TemplateTag 元素	70	CUSTOMER_NEW_HDR010_DATA	125
		CUSTOMER_UPDATE_HDR010_DATA	130
第 13 章 消息传递系统后端集成消息	73	ORDER_CREATE_HDR010_DATA	134
		ORDER_STATUS_UPDATE_HDR010_DATA	135
第 14 章 实现集成消息	75	PRODUCT_PRICE_UPDATE_HDR010_DATA	136
		PRODUCT_QUANTITY_UPDATE_HDR010_DATA	139
第 15 章 定制 NewInboundMessage 命令	77	ORDER_CREATE_HDR020_DATA	139
		ORDER_CREATE_HDR030_DATA	140
第 16 章 集成消息 DTD 文件	79	ORDER_CREATE_HDR040_DATA	141
后端集成旧的消息	80	ORDER_CREATE_ITM010_DATA	143
后端集成 XML 消息	81	ORDER_STATUS_UPDATE_ITM010_DATA	145
使用实现集成消息的样本方案	82	ORDER_CREATE_PROLOG_DATA	147
ReleaseShipNotify 消息	83	ORDER_STATUS_UPDATE_PROLOG_DATA	147
Response_WCS_ExpectedInvRecord 消息	83	PRODUCT_PRICE_UPDATE_PROLOG_DATA	148
Response_WCS_PickBatch 消息	84	PRODUCT_QUANTITY_UPDATE_PROLOG_DATA	149
Report_WCS_PickPackListDetail 消息	85	CUSTOMER_NEW_PROLOG_DATA	149
Response_WCS_CreateInvReceipt 消息	88	CUSTOMER_UPDATE_PROLOG_DATA	149
Response_WCS_UpdateInvReceipt 消息	89	出站消息的 DATUSR_DATA	149
Response_WCS_CreateShipConfirm 消息	90	入站消息的 DATUSR_DATA	150
Update_WCS_OrderStatus 消息	91	NCCustomer_10.mod 文件	150
“订单状态更新”消息	94	NCCCommon.mod — 源文件	152
Update_WCS_ProductPrice 消息	96	NCCustomer_10.mod — 源文件	153
Create_WCS_ExpectedInventoryRecord 消息	98	调用消息传递系统编辑方法	153
Create_WCS_PickBatch 消息	98	SendXMLOrder 命令	154
Inquire_WCS_PickPackListDetail 消息	99		
Create_WCS_InventoryReceipt 消息	100	第 17 章 实现集成消息	155
		声明	157

第 1 章 程序适配器

程序适配器允许外部系统通过使用 HTTP 协议传递 XML 请求与 WebSphere Commerce 通信。程序适配器为外部系统（例如，采购系统）提供一种通过 HTTP 与 WebSphere Commerce 通信的通用方法，允许 WebSphere Commerce 作为这些系统的供应商以进行买方 / 供应商交易。程序适配器通过执行以下操作处理进入的 XML 请求：

- 识别请求并验证其是否为 XML 请求。如果支持请求的以下三个属性，则其可以识别为 XML 请求。
 - 内容类型
 - 方法
 - 字符编码

受支持的请求属性在适配器配置中指定。

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

每一请求都当作独立的会话来处理。消息凭证在消息的控制区域指定。缺省情况下，程序适配器检查用户标识和密码参数以确定请求的真实性。因为旧的消息不支持凭证的规格，所以程序适配器不支持旧的消息。

程序适配器的有效期在 WebSphere Commerce 实例中始终存在。它在实例启动时初始化，除非除去其配置参数或未启用此适配器，并在实例运行时驻留。

关于 WebSphere Commerce 如何处理从设备接收的请求的结构体系信息，请参阅《*WebSphere Commerce 程序员指南*》。

CommandProperty 对象

`CommandProperty` 对象是控制器命令中的典型。对象包含要执行的命令名、执行命令时的命令属性以及命令的参数。消息映射器的目的是将入站请求消息转换为将由适配器执行的控制器命令。尽管所有的 WebSphere Commerce 组件都可以使用消息映射器将数据映射为扩展的 `TypedProperty` 对象，消息映射器的主要目的是将 XML 消息转换为表示控制器命令的公共 Java 对象。

`CommandProperty` 数据类型由以下三部分构成：

- `commandName`：要执行的命令的名称，以字符串形式表示。
- `requestProperties`：执行命令时的命令属性，以 `TypedProperty` 对象形式表示。
- `executionProperties`：执行命令的控制数据，以 `TypedProperty` 对象形式表示。

设备格式算法

设备格式算法用于确定适当的 JSP 以用作特定请求的响应。当接收到请求并且消息映射器用于将请求转换到 `CommandProperty` 对象时，用于处理请求的消息映射器和适配器确定相应的设备格式以生成响应的内容。消息映射器的标识添加到适配器的设备格式标识以确定响应的整体设备格式标识。整体设备格式标识和 `VIEWNAME` 用于从 `VIEWREG` 表获取相应的生成响应内容的 JSP。

接受请求和使用消息映射器的每一适配器都获得一个设备格式标识。此标识在适配器的 `instance_name.xml` 配置文件中定义。此标识在 `instance_name.xml` 配置文件中的适配器配置中定义。使用消息映射器的每一接收适配器的缺省设备格式标识以 `-10000` 为间隔。程序适配器的设备格式标识为 `-10000`，MQSeries 适配器的设备格式标识为 `-20000`。支持旧消息的适配器使用的设备格式标识为 `-30000`。当确定相应的 JSP 和视图命令以调用 JSP 时，消息映射器标识将添加到适配器的设备格式标识以确定响应视图的设备格式标识。但是，如果计算出的设备格式标识的视图不存在，则将使用适配器的间隔缺省视图。这样，对于采用 HTTP 上的 XML 的请求，缺省设备格式标识是 `-10000`，而对于 MQSeries 适配器请求，缺省设备格式标识是 `-20000`。需要区分响应视图，这样 HTTP 响应将使用 `HttpForwardViewCommand` 接口，而来自 MQSeries 适配器的响应将使用 `MessagingViewCommand` 接口。

此算法允许在一个适配器中定义最多 9999 个可能的消息映射器。每一消息映射器的配置文件包含一个间隔为 1 的标识号。要配置使用消息映射器的附加适配器，应当提供间隔为 10000 的设备格式标识。例如，新的请求机制的设备格式间隔可以为 40000。

注意 1: 如果无法找到计算出的设备格式标识，将使用适配器的缺省设备格式标识以获得结果。可以在 HTTP 适配器配置中配置此数字。

注意 2: 间隔大小并不重要。例如，如果只存在两个消息映射器，则间隔大小可以是 3。缺省情况下，间隔是 10000。

示例 1: 入站消息由标识为 `-1` 的消息映射器和设备格式标识为 `-10000` 的程序适配器处理。根据设备格式算法，为响应入站消息而生成的设备格式标识将为 `-10001`。为 `VIEWNAME` 在 `VIEWREG` 表中定义的 JSP 文件和设备格式标识 `-10001` 用于创建响应。

示例 2: 入站消息也可以由相同的标识为 `-1` 的消息映射器处理，但是再由设备格式标识为 `-20000` 的 MQSeries 适配器处理。则响应此消息而生成的设备格式标识将为 `-20001`。这样，为 `VIEWNAME` 在 `VIEWREG` 表中定义的 JSP 文件和设备格式标识 `-20001` 用于创建响应。注意，即使此请求可以使用相同的 JSP 以响应请求，但是用来调用 JSP 的类可能不同。

HTTP 上的 XML

通过使用程序适配器，WebSphere Commerce 可以在 HTTP 上接收入站 XML 消息。以下步骤说明了 HTTP 上的 XML 请求的总体流程：

1. 外部系统在 HTTP 上将 XML 消息发送到 WebSphere Commerce。
2. 此请求映射到程序适配器。
3. 程序适配器传递此 XML 请求到相应的消息映射器。

4. 消息映射器将 XML 请求转换为 CommandProperty 对象并将其传递回程序适配器。
5. 程序适配器准备执行命令并将其传递给 Web 控制器以执行。
6. 程序适配器生成正确的 XML 响应并将其返回给发出此请求的外部系统。

当程序适配器接收到 XML 请求时，它必须验证发送请求的外部系统的凭证。并非所有 XML 请求都可以处理。即使可以将 XML 请求映射到 WebSphere Commerce 命令，还必须作某些验证以确保应当处理此请求。关于更多信息，请参阅程序适配器的 HTTP 请求安全性。

MQSeries 作为中间件

MQSeries 适配器允许您使用 MQSeries 作为中间件将后端和外部系统与 WebSphere Commerce 集成在一起。MQSeries 适配器允许 WebSphere Commerce 从后端系统和外部系统接收消息。支持软件为 MQSeries Version 5.2 或更高版本，带有 MA88 产品扩展。

可以用以下两种方式之一使用 MQ Java 将 MQSeries 设置为中间件：

- **绑定方式**

WebSphere Commerce 安装在与 MQSeries 服务器相同的机器上，且它与 MQSeries 服务器通过 MQSeries Java 使用 Java 本地接口（JNI）相连接。由于通信是通过直接 JNI 调用而非通过网络传送到队列管理器的，但绑定方式使用网络连接提供了比客户机方式更好的性能。

- **客户机方式**

WebSphere Commerce 安装在一台机器上，而 MQSeries 服务器安装在后端系统上。

要验证 MQSeries 连接、队列和通道，可以运行测试程序在 MQSeries 对列中放入和获取消息。关于详细信息，请参阅 MQSeries 文档。

WebSphere Commerce MQSeries 适配器

WebSphere Commerce MQSeries 适配器（或简称为 MQSeries 适配器）是 WebSphere Commerce 的一个组件，可以通过 MQSeries 处理入站消息启用与后端系统集成。MQSeries 适配器是 JMS-MQ CCF 连接器的组合以检索 MQ 消息，程序适配器则被调用以执行这些消息。

MQSeries 适配器具有一组预定义的消息，这些消息有助于将 WebSphere Commerce 商务处理与后端或外部系统处理集成在一起。每条进入消息都可以激活 WebSphere Commerce 中的进程，来更新数据库表或执行其它操作。关于所提供消息的更多详细信息，请参阅后端集成和实现集成消息。除现有的预定义消息外，该适配器支持消息扩展和新消息。

MQSeries 适配器中的并行 — 串行消息处理

MQSeries 适配器可以用两种方式处理入站消息：串行或并行。串行处理即：每条消息放在一行或一个队列中，对它们依次处理。在这种方式中，每条消息必须等待前一消息处理完成之后才能处理。而并行处理是指同时处理许多消息。这些消息多数都可以同步运行，而不是每条消息都要等待前一条完成。

虽然并行处理通常会使得吞吐更快，但它并非对所有类型的请求都适用。在一些情况下事务的序列性必须保留。例如，如果一名新顾客在您的商店注册，之后更正了他们的地址信息，此后又进行了购买订购，则您会希望在处理时保存这些交易的订单。如果没有创建帐户，则您无法执行地址修改或购买订购。同样，您也不会希望在没有正确的装运信息的情况下便填写购买订单。

虽然通常倾向于在可能的地方使用并行处理，但您仍需决定在何处使用这种方法处理数据比较合适。

第 2 章 配置程序适配器

使用 *instance_name.xml* 配置文件配置程序适配器。程序适配器条目用于定义适配器。ProgramAdapter 节点之间的信息定义程序适配器的配置（定义它使用哪些消息映射器）、支持的内容类型和区分 HTTP 上的 XML 请求的其它请求属性。在 *instance_name.xml* 配置中，程序适配器的条目应看似如下：

```
<HttpAdapters display="false">
  <HttpAdapter deviceFormatTypeId="-10000"
    enabled="true"
    deviceFormatId="-10000"
    deviceFormatType="XmlHttp"

    factoryClassname="com.ibm.commerce.programadapter.HttpProgramAdapterImpl"
    name="XML/HTTP">
    <ProgramAdapter>
    <SessionContext
class="com.ibm.commerce.messaging.programadapter.security.
CredentialsSpecifiedProgramAdapterSessionContextImpl">
    <SessionContextConfig />
    </SessionContext>
    <Configuration supportedMethods="POST, M-POST"
      supportedContentTypes="text/xml, text/xml-SOAP"
      supportedMessageMappers="WCS.INTEGRATION"
      supportedCharacterEncoding="ISO8859-1, UTF-8" />
    </ProgramAdapter>
    </HttpAdapter>
</HttpAdapters>
```

为 HTTP 上的 XML 请求启用程序适配器

当创建实例时，程序适配器被禁用。为了通过 HTTP 支持 XML，必须使用 *instance_name.xml* 配置文件启用程序适配器。要启用程序适配器以通过 HTTP 支持 XML 请求，请执行以下操作：

1. 打开 *instance_name.xml* 配置文件。
2. 定位 HttpAdapters 部分，并将 enabled 参数设置为 “true”。

添加适配器

要添加新的适配器，必须在 *instance_name.xml* 配置文件中手工将它添加到适配器组。要添加新的适配器，请执行以下操作：

1. 打开 *instance_name.xml* 配置文件。
2. 定位 HttpAdapters XML 节点，并添加与以下类似的 XML 语法以定义适配器：

```

<HttpAdapter
enabled="true/false"
deviceFormatType="--device format--"
deviceFormatId="#"
name="--name--"
factoryClassname="-- class implementing HttpAdapterFactory --">
< --- free range format of XML to contain adapter configuration information
-->

```

注意： 以上语法用于添加程序适配器。要添加不同类型的适配器，请相应地修改类实现。在以上示例中，HttpAdapterFactory 是类实现。

下载并安装 MQSeries MA88 产品扩展包

要安装 MQSeries MA88 产品扩展包，请执行以下操作：

注意： 此信息在更新 MQSeries 安装过程时会得到更改。



- 从以下 URL 下载对操作系统适当的 ma88 产品扩展包：
<http://www.ibm.com/software/ts/mqseries/txppacs/ma88.html>。此文件是压缩格式的文件。
 - 从同一 URL 处下载 PDF 格式的文档 *MQSeries Using Java*。
- 解压缩并安装 SupportPac。
- 按照安装指导，将产品扩展安装在 `MQ_install_path\java` 目录，其中 `MQ_install_path` 是 MQSeries 的安装路径。
- 更新在以下目录中找到的 `admin.config` 文件：

```

▶ 2000 drive:\Program Files\WebSphere\CommerceServer\bin
▶ NT drive:\WebSphere\CommerceServer\bin
▶ AIX /usr/WebSphere/CommerceServer/bin
▶ Solaris /opt/WebSphere/CommerceServer/bin
▶ 400 /QIBM/Proddata/WebCommerce/bin

```

- 将 `MQ_install_path\java\lib` 目录添加到 `com.ibm.ejs.sm.util.process.Nanny.path` 变量。关于对 iSeries 类库的更改，请参阅更新 WebSphere Application Server 类路径变量。



- 从以下 URL 下载对操作系统适当的 ma88 产品扩展包：
<http://www.ibm.com/software/ts/mqseries/txppacs/ma88.html>。此文件是压缩格式的文件。确保获取名称为 `ma88_iSeries.zip` 的文件。
 - 从同一 URL 处下载 PDF 格式的文档 *MQSeries Using Java*。
- 使用 InfoZip 的 Unzip 解压缩。这将创建文件 `ma88_400.sav`。
- 在 iSeries 400 的适当库中创建称为 MA88 的保存文件，例如 `QGPL CRTSAVF FILE(QGPL/MA88)`

4. 将 `ma88_iSeries.sav` 作为二进制图像传送到此保存文件中。如果使用 FTP 执行此操作，则输入命令应当类似于：
`PUT C:\TEMP\MA88_iSeries.SAV QGPL/MA88`
5. 使用 `RSTLICPGM` 安装 Java 的 MQSeries 类，产品标识为 `5648C60`：
`RSTLICPGM LICPGM(5648C60) DEV(*SAVF) SAVF(QGPL/MA88)`
6. 删除在步骤 2 中创建的保存文件：
`DLTF FILE(QGPL/MA88)`

启用 MQSeries 适配器

使用以下检查表启用 MQSeries 适配器消息。

1. 安装 MQSeries 版本 5.2。关于如何设置 MQSeries 绑定方式或 MQSeries 客户机方式配置的信息，请参阅文档 *MQSeries Using Java*。对于 e-Integrator 版本 3.0，请使用绑定方式。
 -    用户登录标识必须具有读写已定义的队列管理器和队列的权限。对于 e-Integrator 版本 3.0，请使用绑定方式。
 -  实例用户简表必须具有读写已定义的队列管理器和队列的权限。要定义此权限，请使用 `GRTMQMAUT` 命令。
2. 请确保已定义以下 MQSeries 对象：
 - 队列管理器
 - 入站消息队列
 - 出站消息队列
 - 传输队列
 - 错误队列
 - 并行队列
 - 串行队列
3.    设置 MQSeries 队列管理器代码字符集标识为 1208 (UTF8)。在命令行上运行以下 MQSeries 命令：

```
strmqm YourQueueManagerName runmqsc YourQueueManagerName alter qmgr ccsid(1208) end
```


其中 *YourQueueManagerName* 是 MQSeries 队列管理器的名称。
4. 如果正在使用 MQSeries 客户机方式，则必须定义所有必需的通道，并且必须标识 MQSeries 客户机用于与 MQSeries 服务器通信的通道名称。
5. 下载并安装 MQSeries MA88 产品扩展包。
此产品扩展包含 MQSeries 适配器用来与 MQSeries 通信的 Java 消息服务 (JMS) API。
6. 配置 MQSeries 的 JMS。
必须创建映射至相应 MQSeries 对象的 JMS QueueConnectionFactory 和“JMS 队列”。这允许 MQSeries 适配器通过 JMS 访问 MQSeries 实体。
7. 使用 JMSAdmin 配置 JMS。
8. 启用消息传递系统传送适配器。

9. 更新 WebSphere Application Server 类路径变量。


注意: 要使用 MQSeries 适配器, 请确保在启动 WebSphere Commerce Server 和实例之前已启动队列管理器。

配置 MQSeries 的 JMS


要配置消息传递系统以处理 JMS (Java 消息传递服务), 请执行以下操作:



1.  iSeries QShell 提供了一个仿真器, 在这里 Unix 命令可以在 WebSphere Commerce 机器上执行。您必须使用 Java 1.3 来执行剩余的命令。要设置用户简表以使用 Java 版本 1.3, 请将行 `java.version=1.3` 添加到用户 `SystemDefault.properties` 文件。关于更多信息, 请参阅 iSeries Java 文档。必须在启动 QShell 之前执行此操作。

2.  通过在 CL 提示符下输入以下内容启动 iSeries QShell: STRQSH。

3.    在 WebSphere Commerce 机器上, 更新您的 `classpath` 变量:

a. 在一行上, 输入以下命令:

```
 set  
classpath=%classpath%;MQ_install_path\java\lib\com.ibm.mqjms.jar;  
MQ_install_path\java\lib\com.ibm.mq.jar;WAS_install_path\lib\ns.jar
```

```
  export CLASSPATH=$CLASSPATH:  
MQ_install_path/java/lib/com.ibm.mqjms.jar:  
MQ_install_path/java/lib/com.ibm.mq.jar;WAS_install_path/lib/ns.jar  
export CLASSPATH=$CLASSPATH:WAS_install_path/lib/ujc.jar:  
WAS_install_path/lib/ejs.jar; WAS_install_path/lib/sslight.jar 注意: 此  
语句太长, 无法在命令提示符下作为一条语句添加。必须添加为两段。
```



其中

`WAS_install_path` 是 WebSphere Application Server 的安装路径

其中 `MQ_install_path` 是 MQSeries 的安装路径。

b. 通过输入以下命令添加名为 `MQ_JAVA_INSTALL_PATH` 的新环境变量:


```
 set MQ_JAVA_INSTALL_PATH=MQ_install_path\java
```


```
  export  
MQ_JAVA_INSTALL_PATH=MQ_install_path/java
```

其中 `MQ_install_path` 是 MQSeries 的安装路径。

通过输入以下命令, 更新环境以使用 WebSphere Application Server 所带的 JDK:

```
 set PATH=WAS_install_path\Java\bin;%PATH%
```

```
 export PATH=WAS_install_path/java/jre/sh:$PATH
```


```
 export PATH=WAS_install_path/java/jre/bin:$PATH
```

4. 使用 JMSAdmin 配置 JMS。

更新 WebSphere Application Server 类路径变量

要更新实例的 WebSphere Application Server classpath 变量，请执行以下操作：

1. 打开 WebSphere Application Server 高级管理控制台。
2. 选择正在运行 WebSphere Commerce 实例的主机。
3. 选择 **WebSphere 管理域**。
4. 选择节点。
5. 选择您的主机名。
6. 选择应用程序服务器
7. 选择 WebSphere Commerce Server *instance_name*，其中 *instance_name* 是 WebSphere Commerce 实例的名称。对于 iSeries，选择 *instance_name* — WebSphere Commerce Server。
8. 转至实例的“JVM 设置”表。
9. 选择添加添加新的系统属性。
10. 输入以下系统属性：
name = ws.ext.dirs value = MQJAVA/lib
对于 iSeries: value=/QIBM/ProdData/mqm/java/lib
11. 单击应用应用更改。
12. 对每个与 MQSeries 一起使用的 WebSphere Commerce 实例，重复步骤 2-7。
13. 关闭 WebSphere 高级管理控制台。

 对 iSeries 执行以上的步骤 1-11，然后执行以下操作：

1. 单击 **JVM**。
2. 在“系统属性”框的右侧，单击添加。新的系统属性显示在列表中。
3. 在名称字段中输入 java.library.path。
4. 在值字段中输入 /QSYS.LIB/QMQMJAVA.LIB。
5. 单击应用应用更改。
6. 对每个与 MQSeries 一起使用的 WebSphere Commerce 实例，重复步骤 2-7。
7. 关闭 WebSphere 高级管理控制台。

使用 JMSAdmin 配置 JMS

要映射在 WebSphere Commerce 名称空间中创建的队列管理器和队列，请在 QShell 中执行以下操作：

1. 确保 WebSphere Application Server 正在运行，并确保已经设置环境变量和类路径。
2. 更改到以下目录：

 `MQ_install_path\java\bin`

 `MQ_install_path/java/bin`

 `MQ_install_path/java/bin`

 `/QIBM/ProdData/WebCommerce/bin`

其中 *MQ_install_path* 是 MQSeries 的安装路径。

3. 在文本编辑器中打开 JMSAdmin.config 文件。

确保以下三个变量已设置为显示的值:

```
INITIAL_CONTEXT_FACTORY=com.ibm.ejs.ns.jndi.CNInitialContextFactory
```

```
PROVIDER_URL=iiop://host_name:was_port SECURITY_AUTHENTICATION=none
```

其中

host_name

实例主机名

was_port

用于配置实例的 WebSphere Application Server 管理端口。

4. 从命令行中运行 JMSAdmin 程序:

```
Windows JMSAdmin -cfg JMSAdmin.config -t -v
```

```
AIX ./JMSAdmin -cfg JMSAdmin.config -t -v
```

```
Solaris ./JMSAdmin -cfg JMSAdmin.config -v
```

```
400 ./JMSAdmin -cfg JMSAdmin.config -v
```

等待装入管理命令行界面，并出现 Initctx> 提示符。

5. 向 WebSphere Application Server 名称空间中的队列管理器注册队列连接工厂:

```
define qcf(JMSQueueConnectionFactory) qmanager(YourQueueManagerName)
```

其中

JMSQueueConnectionFactory

它在 QueueConnectionFactory ConnectionSpec 属性（可在 MQSeries 的 JMS 配置中找到）中定义。这可在配置管理器的“ConnectionSpec-JMS 接口 CCF 连接”页面上找到。

YourQueueManagerName

MQSeries 队列管理器的名称。

6.    设置此代码字符集为 1208 (UTF8)。

```
alter qcf(JMSQueueConnectionFactory) ccsid(1208)
```

其中

JMSQueueConnectionFactory

MQQueueConnectionFactory JMS 对象的名称。

7. 定义以下 JMS 队列。要定义队列类型，请在一行上输入适当的命令:

JMSSerialInboundQueue — serial inbound queue

```
define
```

```
q(JMSSerialInboundQueue)qmanager(YourQueueManagerName)queue(YourSerialInboundQueueName)
```

例如:

```
define q(JMSSSerialInboundQueue)qmanager(WCSQMGR)queue(JMSSIBQ)
```

- **JMSParallelInboundQueue** — 并行入站队列

```
define
```

```
q(JMSParallelInboundQueue)qmanager(YourQueueManagerName)queue(YourParallelInboundQueueName)
```

- **JMSInboundQueue** — 入站队列

```
define
```

```
q(JMSInboundQueue)qmanager(YourQueueManagerName)queue(YourInboundQueueName)
```


- **JMSOutboundQueue** — 出站队列

```
define
```

```
q(JMSOutboundQueue)qmanager(YourQueueManagerName)queue(YourOutboundQueueName)
```

- **JMSErrorQueue** — 错误队列

```
define q(JMSErrorQueue)qmanager(YourQueueManagerName)queue(YourErrorQueueName)
```

YourQueueManagerName

MQSeries 队列管理器的名称。

YourSerialInboundQueueName

为串行入站队列创建的 MQSeries 队列名称。

YourParallelInboundQueueName

为并行入站队列创建的 MQSeries 队列名称。

YourInboundQueueName

为入站消息队列创建的 MQSeries 队列名称。

YourOutboundQueueName

为出站队列创建的 MQSeries 队列名称。

YourErrorQueueName

为错误队列创建的 MQSeries 队列名称。

队列的缺省名称与配置管理器中“ConnectionSpec-JMS 接口 CCF 连接”值使用的名称相同。如果更改了缺省名称，则必须同时在配置管理器中更改该名称，或为了出站消息传递，可在管理控制台中完成该名称更改。

8. 出站队列和错误队列需要设置目标客户机，使其指示 JMS 正在与本地 MQSeries 应用程序交互。请运行以下命令：

```
alter q(JMSOutboundQueue) targclient(MQ) alter q(JMSErrorQueue) targclient(MQ)
```

9.     如果正在使用 MQSeries 客户机 / 服务器设置，其中 MQSeries 客户机和 WebSphere Commerce Server 在同一台机器上，请运行以下命令：

```
alter qcf(JMSQueueConnectionFactory) transport(CLIENT) alter qcf(JMSQueueConnectionFactory) hostname(YourMQServerHostName)
```

其中 *YourMQServerHostName* 是 MQSeries 服务器的名称。

此命令连接到远程 MQSeries 服务器并配置客户机。

10. 输入 `end` 退出管理命令行界面。

使用 MQSeries 适配器响应处理

MQSeries 适配器可以用来作为到 WebSphere Commerce 系统和功能的备用接口。MQSeries 适配器不仅处理入站请求而且可以提供对请求的应答。任何产生响应的命令都可以被配置来通过 MQSeries 适配器提供响应。

当 MQSeries 适配器接收到一条入站消息时，此入站消息被映射到一个命令，而调用该命令时使用的参数是在入站消息中指定的。如果要从该入站请求中生成响应，则需要 VIEWREG 表中的条目。关于如何使用 VIEWREG 表的更多信息，请参阅《WebSphere Commerce 程序员指南》。如果响应是由命令生成的，则该响应将根据接收入站请求的连接器的名称，被放置在并行或串行连接器的出站队列中。

如果入站请求发生错误，则生成错误的消息将被放置在错误队列中。要生成对出站队列的错误响应，则需要 VIEWREG 表中的另一条目来使用正确的设备格式映射错误响应。

要设置 MQSeries 适配器组件以提供对入站消息的响应，请执行以下操作：

1. 通过确保为入站并行连接器包含了与以下类似的语法，来验证 *instance_name.xml* 文件是否为入站并行和串行连接器包含了出站队列：

```
<EditableProperty Admin="outQueue"
  editable="Yes"
  name="setOutboundQueue"
  display="false"
  value="JMSParallelOutboundQueue" />
```

2. 使用 JMSAdmin 来定义串行和并行出站队列的 JMS 名称。
3. 在 VIEWREG 表中添加一个条目来定义入站请求需要的响应。此步骤对现有的 WebSphere Commerce 命令和新命令而言都是必需的。使用以下值：

VIEWNAME

与常规 HTTP 响应相同的值

INTERFACENAME

com.ibm.commerce.messaging.viewcommands.MessagingViewCommand

CLASSNAME

com.ibm.commerce.messaging.viewcommands.MessagingViewCommandImpl

PROPERTIES

docname=*JSP filename*

DEVICEFMT_ID

- XML 入站消息为 -20000
- 旧的入站消息为 -30000
- 为新入站消息添加 -5

第 3 章 MQSeries 的程序适配器安全性

要更改安全性级别，必须在 MQSeries 适配器配置的 `SessionContext` 节点中更改类属性的值。

Web 控制器执行请求之前，`WebController` 确定请求必须在其下执行的凭证。这由指定类确定，它使用 `CommandProperty` 对象确定请求的凭证。

有两种级别的安全性可用：

- 级别 1：有限安全性。
- 级别 2：每次请求时需要用户标识和密码。

也可以创建您自己的定制安全性类。唯一限制是必须实现 `ProgramAdapterSessionContext` 接口。

安全性级别 1：有限安全性

此级别的安全性在安装时缺省启用。它假定所有请求都将由缺省用户标识处理，除非特别指定。如果请求由另一个用户标识处理，则此用户标识在请求中指定。不需要指定用户标识的密码，请求由 `WebSphere Commerce` 使用指定用户的凭证处理。

实现此安全性级别的类为

`messaging.programadapter.security.DefaultCredentialsProgramAdapterSessionContextImpl`。

安全性级别 2：每次请求的登录和密码

此级别的安全性发送到包含用户标识和密码的 `WebSphere Commerce` 的请求。如果密码与指定用户标识不匹配，则发出安全性异常且拒绝请求。如果请求中没有指定用户标识和密码，则请求作为临时用户处理。这意味着执行之前需要认证的命令必需拥有请求中指定的凭证。

实现此安全性级别的类为

`messaging.programadapter.security.CredentialsSpecifiedProgramAdapterSessionContextImpl`。

凭证的确认使用的技术与顾客通过浏览器与 `WebSphere Commerce` 交互时使用的技术相同。不管用户注册是否由 `WebSphere Commerce` 或第三方软件处理，这都在考虑之中。

所有 `WebSphere Commerce` 支持的 XML 集成消息支持此级别的安全性，尽管凭证规格不是必需的。

HTTP 请求的程序适配器安全性

此级别的安全性发送到包含用户标识和密码的 `WebSphere Commerce` 的请求。如果密码与指定用户标识不匹配，则发出安全性异常且拒绝请求。如果请求中没有指定用户标识和密码，则请求作为临时用户处理。这意味着执行之前需要认证的命令必需拥有请求中指定的凭证。此安全性表单以与来自浏览器客户机的 HTTP 请求相似的方式执行。

实现此安全性级别的类为

`messaging.programadapter.security.CredentialsSpecifiedProgramAdapterSessionContextImpl`。

凭证的确认使用的技术与顾客通过浏览器与 WebSphere Commerce 交互时使用的技术相同。不管用户注册是否由 WebSphere Commerce 或第三方软件处理，这都在考虑之中。

所有 WebSphere Commerce 支持的 XML 集成消息支持此级别的安全性，尽管凭证规格不是必填的。此安全性实现不支持旧的消息。

消息编辑服务

一些 WebSphere Commerce 消息使用消息编辑服务。通过使用 JSP 模板，编辑服务在出站消息通过传送发送之前生成一个出站消息。如果编辑服务用于某消息，它将运行 JSP，向它发送诸如订单号码或商店号码的信息。执行模板时，JSP 可以使用数据 Bean 检索任何数据库中的消息所需的附加信息。之后生成输出，并通过传送机制发送格式化消息。您可以像修改其它 JavaServer Pages 一样修改消息模板。

您可能想要修改的生成的消息中的一些特征包括：

- 消息的布局。
- 从数据库中检索出来并显示在页面上的有关商店、订单或顾客的信息。
- 显示给顾客的消息的文本，例如“感谢您订购我们的产品”。
- 生成输出的格式。例如，您可能希望发送 HTML 格式、纯文本格式或 XML 格式的消息。

消息传递系统

WebSphere Commerce 消息传递系统为 WebSphere Commerce 提供与其外部环境通信的能力。此通信包含向后端系统或外部系统发送消息和接收消息，以及向顾客和管理员发送通知，告诉他们在 WebSphere Commerce 中发生的事件。这通过两个子系统完成：管理来自后端和外部系统的入站消息的入站系统，和允许您向用户发送通知及向后端和外部系统发送出站消息的出站消息传递系统。

例如，您可设置消息传递系统使其发送电子邮件消息，通知顾客他们的订单已装运。消息传递系统提供将 WebSphere Commerce 和后端系统集成的机制。您可配置 WebSphere Commerce，使得无论何时在商店中创建了订单时，它都可以向后端系统发送出站消息。后端系统可以用此订单信息进行必要的订单实现处理。之后，后端系统可以将订单状态消息发送回 WebSphere Commerce，指示已经交付了订单，或发出了订单发票。也可发送电子邮件使顾客了解最新情况。

一般应用程序和系统错误 XML 消息

在 WebSphere Commerce 中，会发生一般应用程序和系统错误。这些消息以 XML 格式显示，并发送到出站队列。

如果一般应用程序错误消息与用户相关，则其发送到出站队列。当用户在 XML 消息中输入无效参数时，将抛出 `ECApplicationException`。然后此消息送到出站队列，日志文件将记录这个异常。

注意：当抛出此类异常时，Web 控制器将不重试此命令，即使被指定为可重试的命令。

如果检测到运行时异常或 WebSphere Commerce 配置错误（诸如空指针和翻译回滚异常），则发送一般系统错误消息到出站队列。

一般 XML 消息的内容因入站 XML 消息的内容而异，但是，其格式与以下类似：

```
<?xml version="1.0" encoding="UTF-8"?>

<WCS_Error type="GenericApplicationError"> (或者 <WCS_Error
type="GenericSystemError">)

<RequestAttributes>

...(Error parameters and data of inbound XML message)

<excMsg>(Error message key, e.g. _ERR_REMOTE_EXCEPTION)</excMsg>

...

</RequestAttributes>

</WCS_Error>
```

设置出站消息编辑

要为出站消息设置并使用编辑服务，请执行以下操作：

1. 使用这些方法中的一个，将传送指定给相应的消息类型：
 - 将传送方法指定到站点的消息类型
 - 为商店消息类型指定传送方法
必须为使用的每个传送指定一个有效的设备格式（如 DEVICEFMT 表中所指定的）。
2. 参阅主题出站消息传递系统接口中的信息，使用 SendMsgCmd 任务命令创建消息传递系统对象。使用 setMsgType() 和 setStoreId() 初始化服务。
3. 调用消息传递系统撰写方法。

使用消息传递系统编辑服务的示例

以下是您可能如何使用消息传递系统编辑服务的一个示例。如果您有一个名为 DemoStore 的商店，而且您想将电子邮件和文件这两种传送指定为 OrderAuthorized 消息类型，请执行以下操作：

1. 向 JSP 文件的 VIEWREG 表添加一个条目，用于撰写此出站消息。VIEWREG 表的键是视图名称、商店标识和设备格式标识。有关如何使用 VIEWREG 表的更多信息，请参阅《WebSphere Commerce 程序员指南》。

注意事项：供“消息传递系统”撰写服务使用而创建的每个视图必须对接口和类名字段使用“消息传递视图命令”。它还必须在 docname 字段中包含 JSP 文件的名称。总结如下：

INTERFACENAME

```
com.ibm.commerce.messaging.viewcommands.MessagingViewCommand
```

CLASSNAME

```
com.ibm.commerce.messaging.viewcommands.MessagingViewCommandImpl
```

PROPERTIES

使用以下格式指向 JSP 文件 “docname=jsp file”。

DEVICEFMT_ID

代表设备格式，且应当使用值 -3（标准设备格式），除非对您的应用程序使用定制设备格式。在 VIEWREG 条目中指定的 DEVICEFMT_ID 必须与对传送指定消息类型时选择的设备类型相对应。

例如，如果 DemoStore 的商店标识是 5，而 viewname 为 OrderAuthorized，您可以使用以下 SQL 语句插入记录：

```
insert into viewreg (VIEWNAME,STOREENT_ID,DEVICEFMT_ID,INTERFACENAME,
CLASSNAME,PROPERTIES) values ('OrderAuthorizedView',5,-3,
'ibm.commerce.messaging.viewcommands.MessagingViewCommand',
'ibm.commerce.messaging.viewcommands.MessagingViewCommandImpl',
'docname=OrderAuthorized.jsp');
```

2. 使用管理控制台将电子邮件和文件这两种传送指定为 OrderAuthorized 消息并配置设置。可以使用站点或商店级别的管理权限进行此操作。在站点级别创建设置将使所有商店都可访问该设置。
3. 在命令的实现中，使用 OrderAuthorized 消息类型的消息标识和 DemoStore 的商店标识，来实例化 SendMsgCmd 命令以使用消息传递服务以及调用 setMsgType() 和 setStoreID() 方法。如果您需要使用站点级别的配置，请指定 0 作为商店标识，并将 "&storeId=no" 附加在 JSP 名称的结尾处。否则，使用您的商店标识。（如果您的商店不存在配置，则“消息传递系统”的缺省值自动设置为站点级别配置。）
4. 调用出站消息传递系统接口的组成方法，并以 TypedProperty 对象形式传送任何附加参数。通过指定一个视图名称，您可以覆盖在撰写消息时使用的消息类型的缺省视图名称。
5. 如果希望立即或在传送成功提交之后发送消息，则请调用 SendMsgCmd 的 sendImmediate 或 sendTransacted。请参阅“消息传递系统”文档以获取使用每种方法的更多说明。
6. 调用 SendMsgCmd 的执行方法执行发送。

消息传递系统编辑服务中的出错处理

当处理 JavaServer Pages 出现错误时，通常情况下，页面的结果包含由运行时生成的扩展信息。如果这不是 JavaServer Pages 故障的期望结果，则有两种潜在的方法可以使此行为变得更加可预测。

在第一种方法种，您可以在 JavaServer Pages 中指定一个错误页面，如果遇到未预料错误（例如一个未被捕捉到的异常），该页面就会运行。要使用此方法，您的主页中需要包含以下行：

```
<%@ page errorPage="YourError.jsp" %>
```

YourError.jsp 中包含以下行：

```
<%@ page isErrorPage="true" %>
```

注意：在错误的事件中，消息编辑的结果就是 `YourError.jsp` 的结果。更多的详细信息请参阅 JSP 1.0 规范。

在第二种方法中，不将错误消息作为出站消息处理，您可以使用一个错误 JavaServer 页面，它生成的输出将作为异常处理。为此，可以使用 `ComposerError.jsp` 作为处理错误的基础。可按如上面所述的方式进行，但 `YourError.jsp` 的开头应从 `ComposerError.jsp` 的前两行开始。这样就允许编辑运行时检测到出现错误，并引发一个异常。可以在以下目录中找到 `ComposerError.jsp`：

```
▶ 2000 drive:\Program Files
\WebSphere\AppServer\installedApps\WC_Enterprise_App_<instance_name>.ear\wcstores.war
▶ NT
drive:\WebSphere\AppServer\installedApps\WC_Enterprise_App_<instance_name>.ear
\wcstores.war
▶ AIX
/usr/WebSphere/AppServer/installedApps/WC_Enterprise_App_<instance_name>.ear
/wcstores.war
▶ Solaris
/opt/WebSphere/AppServer/installedApps/WC_Enterprise_App_<instance_name>.ear
/wcstores.war
▶ 400
/QIBM/Userdata/WebASAdv4/<WAS_instance_name>/installedApps/WC_Enterprise_App_
<instance_name>.ear/wcstores.war
```

`ComposerError.jsp` 的前两行如下：

```
ERROR <%@ page isErrorPage="true" %>
```

`errorPage="ComposerError.jsp"` 指示它是错误页面。如果需要额外信息，可以将此文件复制为另一个文件名（例如 `YourError.jsp`）的文件，并将额外的错误信息放置在新文件中的这两行之后。然后在 JSP 主页面中设置 `errorPage="YourError.jsp"`。前两行之后的任何指定的额外输出都将是编辑服务发生的异常的文本的一部分。

调用消息传递系统编辑方法

要调用出站消息传递系统接口的 `compose()` 方法，请指定以下参数：

- **viewname:** 要使用的编辑视图名称，如 VIEWREG 表中现有记录的 VIEWNAME 列中指定的那样。关于如何使用 VIEWREG 表的更多信息，请参阅《*WebSphere Commerce 程序员指南*》。

重要信息: 在 VIEWREG 表参阅的记录中，INTERFACENAME 和 CLASSNAME 列的值必须包含与所有 WebSphere Commerce 消息传递系统是命令关联的接口和类的名称。接口名称必须是

`com.ibm.commerce.messaging.viewcommands.MessagingViewCommand`。类的名称必须是 `com.ibm.commerce.messaging.viewcommands.MessagingViewCommandImpl`。

- **CommandContext:** 关于 CommandContext 接口或实现该接口的 CommandContextImpl 类的信息，请参阅《*WebSphere Commerce 程序员指南*》。
- **TypedProperty:** 已输入属性中的值必须是字符串或实现 toString() 方法的对象。关于 TypedProperty 的更多信息，请参阅《*WebSphere Commerce 程序员指南*》。

compose() 方法为每一个已经启用并已指定为管理控制台中当前消息类型的传送运行一个视图命令。此方法执行以下过程:

- 它使用 viewname 参数和每个传送的 storeId 和设备格式，这已在管理控制台中定义。这些值用于查找 VIEWREG 表中的视图命令。
- 它运行视图命令，将在 TypedProperty 参数中指定的值传送给该命令。运行命令时，系统使用 viewname、storeId 和设备格式标识查找 VIEWREG 表 PROPERTIES 列中的 JSP 模板。运行 JSP 模板并传送 TypedProperty 参数中的值。
- JSP 编辑消息，并在对象中调用发送方法时通过相应的传送发送它。可以使用在消息传递对象（在该对象上运行编辑）上处理的、立即的或回复进行发送。

第 4 章 OrderItemStatus 命令

当从后端系统接收到 Update_WCS_OrderStatus、Update_NC_OrderStatus 或“订单状态更新”消息时，OrderStatus、OrderConfirmStatus、OrderInvoiceStatus 和 OrderShippingStatus 命令将在内部调用 OrderItemStatus 命令。此命令将更新有关某个商品的现有订单状态的信息。

参数

以下是命令参数的列表。每个参数都对应 ORDISTAT 表中的一个字段：

versioning: 字符串“TRUE”或者“FALSE”。可选，缺省值是 FALSE。

如果为 TRUE，则启用 versioning。

orderItemId: 整型

WebSphere Commerce 订购商品引用号，已在表 ORDISTAT 的 ORDERITEMS_ID 中定义。这是外键，该键引用 ORDERITEMS 表中的列 ORDERITEMS_ID。

merchantItemNumber: 字符串

在后端系统中生成的订购商品号，如 OIMITEM 列中定义。

PartNumber: 字符串

商品的产品号/库存号，已在 PARTNUMBER 列中定义。

UnitOfMeasure: 字符串

商品计量单位，已在 OIUOFM 列中定义。

RequestQuantity: 整型

请求的商品数量，已在 OIQTREQUEST 列中定义。

ConfirmQuantity: 整型

确认的商品数量，已在 OIQTCONFIRM 列中定义。

ShipQuantity: 整型

装运的商品数量，已在 OIQTSHIP 列中定义。

Currency: 字符串

表示价格的 ISO 4217 货币类型，已在 OICPCUR 列中定义。

UnitPrice: BigDecimal(20,4)

产品单价，已在 OIUNPRC 列中定义。

PriceTotal: BigDecimal(20,4)

商品的总产品价格，已在 OIPRTOT 列中定义。

TaxTotal: BigDecimal(20,4)

商品的总税款，已在 OITXTOT 列中定义。

ShippingTotal: BigDecimal(20,4)

商品的总装运费用，已在 OISHTOT 列中定义。

ShippingTaxTotal: BigDecimal(20,4)

商品装运费用的总税款，已在 OISHTXTOT 列中定义。

Status: 字符串

商品的状态, 已在 OISTATUS 列中定义。

PlaceDateTime: 时间戳记

实际放置商品的时间, 已在 OIPLTIME 列中定义。

RequestShipDateTime: 时间戳记

请求装运商品的日期, 已在 OIRSTIME 列中定义。

ScheduleShipDateTime: 时间戳记

调度装运商品的日期, 已在 OISSTIME 列中定义。

ActualShipDateTime: 时间戳记

实际装运商品的日期, 已在 OIASTIME 列中定义。

InvoiceDateTime: 时间戳记

商品开发票的日期, 已在 OIINVTIME 列中定义。

InvoiceValue: BigDecimal(20,4)

商品开发票的净值, 已在 OIINVVAL 列中定义。

itemShipCondition: 字符串

在 OISCOND 列中定义的、关于是否接收商品的部份装运的指定代码。‘SC’: 全部装运, ‘SP’: 部分装运。

itemComment: 字符串

关于商品状态的注释, 已在 OICMNT 列中定义。

field1: 整型

商品状态定制字段 1, 已在 FIELD1 列中定义。

field2: BigDecimal(15,2)

商品状态定制字段 2, 已在 FIELD2 列中定义。

field3: 字符串

商品状态定制字段 3, 已在 FIELD3 列中定义。

行为

- 订单的第一个订购商品状态必须指定 orderItemId 和 merchantItemNumber。
- 检查 orderItemId 中指定的订购商品是否在 ORDERITEMS 表中存在。
- 在后面的每个订购商品状态（在 ORDISTAT 表中已经有了条目）中, 允许通过现有行商品的后端系统创建换行项而不需 orderItemId。
- 使用提供的信息在 ORDISTAT 表中创建或更新一行。
- 如果启用了 versioning, 且这是订单的第一个订购商品状态, 将在 ORDISTAT 表中创建新的行。否则将建立上次订购商品状态的拷贝, 且将其版本 (OIVERNBR) 设置为现有版本的最大数目 +1。将创建新的行, 该行包含所有提供的信息, 其版本 (OIVERNBR) 设置为 0。

异常情况

- 指定的 orderItemId 不是 ORDERITEMS 表中的有效订单号。
- 在第一个订购商品状态中没有对特定订购商品指定 orderItemId 或 merchantItemNumber。
- orderItemId 和 merchantItemNumber 与 ORDISTAT 表中已有的订单号（对应特定的订购商品）不匹配, 该订购商品正在进行后面的订购商品状态更新。

GetPickPackListDetail 命令

此命令检索提货单和 Inquire_WCS_PickPackListDetail XML 消息的装货列表信息。

参数

pickBatchId

提货批量的引用号，如 PICKBATCH 表的 PICKBATCH_ID 列中所定义的。

行为

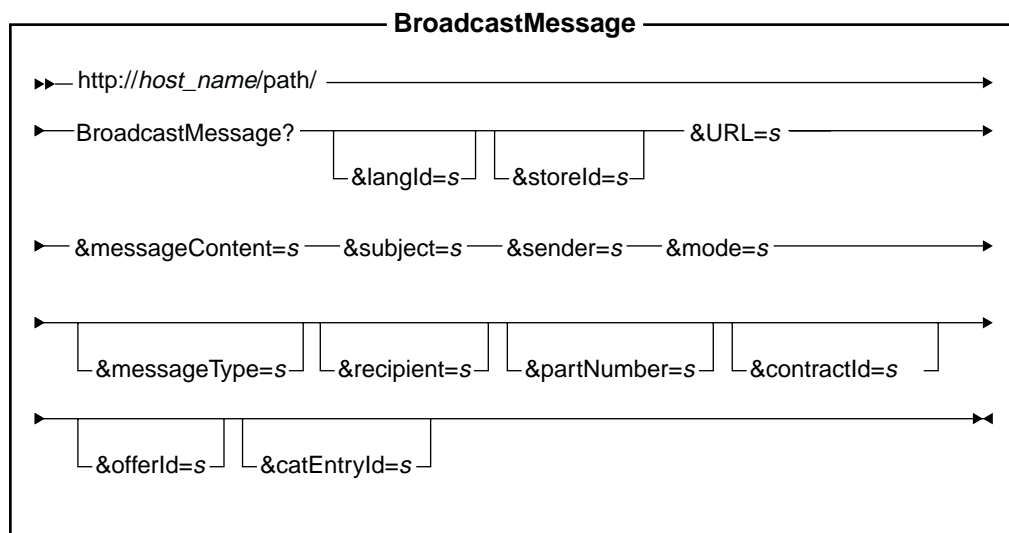
- 检查 PICKBATCH 表中是否存在 pickBatchId。
- 检查 ORDRELEASE 表中对该 pickBatchId 是否存在至少一条记录。
- 从 PICKBATCH 表中检索提货单和给定 pickBatchId 的装货列表 XML，然后将它们转发到视图任务以撰写 Report_WCS_PickPackListDetail 消息。
- 发送包含提货单和装货列表信息的 Report_WCS_PickPackListDetail 消息作为响应。

异常情况

- pickBatchId 无效，或者在 PICKBATCH 或 ORDRELEASE 表中未找到 pickBatchId。

BroadcastMessage 命令

此命令向顾客广播一份电子邮件。



参数值

`http://host_name/path/`

Websphere Commerce Server 的全限定名称和配置路径。

langId 设置或重新设置会话持续时间内使用的首选语言，商店的支持语言可在 STORELANG 表中找到。

storeId

正在发送消息的商店引用号。

URL 在命令成功完成时要调用的 URL。

messageContent

当前发送的消息内容。如果已指定，此内容将覆盖消息模板的内容。如果指定了 `messageType` 参数，则必须指定此参数。

subject

电子邮件中“主题”行的内容。如果此主题未指定，则将使用消息类型配置中指定的缺省主题。

sender

电子邮件中“发件人”行的内容。如果发件人未指定，则将使用消息类型配置中指定的缺省发件人。

mode 可能的值:

1. 将消息发送给在指定商店中购买了商品并表明愿意接受促销电子邮件的所有购物者。
2. 将消息发送给指定商店中购买了标明库存标识（SKU）的商品并表明愿意接受促销电子邮件的所有购物者。
3. 将消息发送给表明愿意接受促销电子邮件的所有购物者。

messageType

对 `MSGTYPES` 表的 `MSGTYPE_ID` 列中定义的预定义消息模板的引用。如果 `messageType` 未指定，则将使用广播消息的站点级消息类型。缺省情况下，站点级别的消息类型没有关联的消息模板，因此推荐商店具有自己的广播式消息设置。有关如何设置新消息类型的信息，请参阅消息记录。

recipient

以逗号分隔的列表表示的电子邮件的其他收件人。

partNumber

方式 2 中与电子邮件关联的商品的部件号。

contractId

方式 2 中在此合同下有过购买行为的目标成员

offerId

方式 2 中对此报价有过购买行为的目标成员

catEntryId

在方式 2 中购买过此产品目录条目的目标成员

行为

- 调用 `GetRecipientsCmd` 任务命令来创建电子邮件广播收件人列表。
- 调用另一个任务命令，通过提供电子邮件回信地址、消息主题以及以逗号分隔的收件人参数列表来发送电子邮件消息。
- 将指定消息类型对应的模板用于消息主体。

异常情况

如果不能确定收件人列表或传递消息时发生错误，将调用 `BoardcastMessageErrorView` 错误任务。

OrderInvoiceStatus 命令

当从后端系统接收 OrderStatusType 标记值为 OrderInvoice 的 Update_WCS_OrderStatus XML 消息时，才使用 OrderInvoiceStatus 命令。此命令更新有关现有订单发票状态的信息。它的参数、行为和异常情况与 OrderStatus 命令相同。这两个命令之间仅有的区别在于 OrderInvoiceStatus 命令将 orderStatus 参数的值设置为缺省值 I，即如果未在消息中提供订单，就认为已给订单开了发票。

关于 OrderInvoiceStatus 命令参数、行为和异常处理的详细信息，请参阅 OrderStatus 命令。

OrderShippingStatus 命令

当从后端系统接收 OrderStatusType 标记值为 OrderShipping 的 Update_WCS_OrderStatus XML 消息时，就使用 OrderShippingStatus 命令。此命令更新有关现有订单装运状态的信息。它的参数、行为和异常情况与 OrderStatus 命令相同。这两个命令之间仅有的区别在于 OrderShippingStatus 命令将 orderStatus 参数的值设置为缺省值 S，即如果未在消息中提供订单，就认为订单已装运。

关于 OrderShippingStatus 命令参数、行为和异常处理的详细信息，请参阅 OrderStatus 命令。

OrderStatus 命令

当从后端系统接收到 Update_WCS_OrderStatus、Update_NC_OrderStatus 或“订单状态更新”消息时，就使用 OrderStatus 命令。此命令更新有关现有订单状态的信息。

参数

sequenceNumber

字符串。串行化的 WCS 订单状态 sequenceNumber。

lastUpdateTimestamp:

时间戳记格式的字符串。串行化的 WebSphere Commerce 订单状态最后更新时间戳记。

versioning: 字符串“TRUE”或者“FALSE”。缺省值为 FALSE。

如果为 TRUE，则启用 versioning。

下面列出的参数对应于 ORDSTAT 表中的列。

orderId: 整数。

WebSphere Commerce 订单引用号，已在表 ORDSTAT 的 ORDERS_ID 列中定义。这是外键，该键引用 ORDERS 表中的列 ORDERS_ID。

merchantOrderNumber: 字符串，必需。

后端系统中生成的订单号，已在 OSMORDER 列中定义。

currency: 字符串

表示价格的 ISO 4217 货币类型，已在 OSCPCUR 列中定义。

priceTotal: BigDecimal(20,4)

订单的总产品价格，已在 OSPRTOT 列中定义。

taxTotal: BigDecimal(20,4)

订单的总税款，已在 OSTXTOT 列中定义。

shippingTotal: BigDecimal(20,4)

订单的总装运费用，已在 OSMORDER 列中定义。

shippingTaxTotal: BigDecimal(20,4)

订单的装运费用总税款，已在 OSSHTXTOT 列中定义。

orderStatus: 字符串

订单状态，已在 OSSTATUS 列中定义。

placeDateTime: 时间戳记

实际订购日期，已在 OSPLTIME 列中定义。

requestShipDateTime: 时间戳记

请求交付订单的日期，已在 OSRSTIME 列中定义。

requestShipDateTime: 时间戳记

调度交付订单的日期，已在 OSSSTIME 列中定义。

requestShipDateTime: 时间戳记

实际交付订单的日期，已在 OSASTIME 列中定义。

invoiceDateTime: 时间戳记

订单开发票的日期，已在 OSINVTIME 列中定义。

invoiceValue: BigDecimal(20,4)

订单开发票的净值，已在 OSINVVAL 列中定义。

shipCondition: 字符串

关于是否接受订单部份出货的指定代码，已在 OSSCOND 列中定义。'SC': 全部装运，'SP': 部分装运。

shippingModeFlag: 字符串

指示送货地址和装运方式是订单级别或者订购商品级别的代码，已在 OSSMFLAG 列中定义。'O': 订单级别；'I': 商品级别。

comment: 字符串

关于订单状态的注释，已在 OSCMNT 列中定义。

field1: 整型

订单状态定制字段 1，已在 FIELD1 列中定义。

field2: BigDecimal(15,2)

订单状态定制字段 2，已在 FIELD2 列中定义。

field3: 字符串

订单状态定制字段 3，已在 FIELD3 列中定义。

items: 散列表向量。

每个散列表都代表一个商品的参数。

行为

- 订单的第一个订单状态必须指定 orderId 和 merchantOrderNumber。
- 检查 orderId 中指定的订单是否在 ORDERS 表中存在。
- 如果串行化信息可用，例如 sequenceNumber 和 lastUpdateTimestamp 参数，请检查这些值是否比 ORDSTAT 表中已有的值更新，如果不是，将不执行更新。

- 使用所有提供的信息在 ORDSTAT 表中创建或更新一行。
- 如果启用了 versioning，且这是订单的第一个订单状态，将在 ORDSTAT 表中创建新的行，否则将建立上次订单状态的拷贝，且将其版本（OIVERNBR）设置为现有版本的最大数目 +1。将创建新的行，该行包含所有提供的信息，其 OSVERNBR 值设置为 0。
- 将表 ORDERS 的 STATUS 列中的值更新为 “G”。

异常情况

- 指定的 orderId 不是 ORDERS 表中的有效订单号。
- 第一个订购商品状态中没有对特定订单指定 orderId 或 merchantOrderNumber。
- orderId 和 merchantOrderNumber 与特定订单（正在进行后面的订单状态更新）的 ORDSTAT 表中已有的订单号不匹配。
- 串行化信息已提供，且订单状态消息失序。

ProductOfferPriceUpdate 命令

ProductOfferPriceUpdate 命令用于更新 Update_WCS_ProductPrice、Update_NC_ProductPrice 和 “产品价格更新” 消息的产品价格信息。

参数

offerId

标识报价的引用号。

partNumber

产品目录条目的部件号，已在 CATENTRY 表的 PARTNUMBER 列中定义。

memberId

标识产品目录条目所有者的引用号。

catEntryId

供销售用的产品目录条目。

currency

表示价格的 ISO 4217 货币类型。此值必需。

offerPrice

此次更新所用的报价。

precedence

此次更新所用的优先级。

tradingPositionContainerId

以报价为其部件的 TradingPositionContainer。

startDateTime

“报价” 有效时间范围的开始时间。

endDateTime

“报价” 有效时间范围的结束时间。

minimumQuantity

在此报价下的单一订单中可以购买的最低数量。

maximumQuantity

在此报价下的单一订单中可以购买的最高数量。

quantityUnit

minimumQuantity 和 maximumQuantity 的计量单位。

行为

- 此命令更新 OFFERPRICE 表中的记录。
- 此过程（通过它确定主键）有轻微变化，这取决于执行命令的入站消息的格式。关于如何确定主键的描述，请参阅指定的入站消息。

异常情况

- currency 参数为空。
- offerId 参数值无法在 OFFERPRICE 表中找到。
- 无法使用 memberId（它拥有商店及 partNumber）找到产品目录条目。
- tradingPositionContainerId 参数值无法在相匹配的 offerId 的 OFFER 表中找到。
- precedence 参数值超出最大值。precedence 必须小于 10^{16} 。

ProductInventoryUpdate 命令

ProductInventoryUpdate 命令用于更新 Update_NC_ProductInventory 消息、Update_WCS_ProductInventory 消息或“产品数量更新”消息的产品库存。

参数

catEntryId

WebSphere Commerce 产品目录条目，已在表 INVENTORY 的 CATENTRY_ID 列中定义。如果产品目录条目为空，那么将使用部件号和成员标识（拥有商店和产品目录）的组合获取产品目录条目。

partNumber

产品目录条目的部件号，已在 CATENTRY 表的 PARTNUMBER 列中定义。它将与拥有产品目录的成员标识一起用于获取表 CATENTRY 中的键 catEntryId（如果 catEntryId 参数为空）。

storeId

引用表 INVENTORY 中的列 STORE_ID 的商店标识。若与产品目录条目和缺省实现中心一起，就是表 INVENTORY 中的行的键。

inventoryQuantity

数量，已在表 INVENTORY 的 QUANTITY 列中定义。

fulfillmentCenterID

引用表 INVENTORY 中的列 FFMCENTER_ID 的实现中心标识。这是 FFMCENTER 数据库表的键。

行为

- 此命令更新 INVENTORY 表中的记录。
- 商店标识（引用 INVENTORY 表中的 STORE_ID）是必需的。
- catalogEntryId 和 storeId 用于更新 INVENTORY 表中的行。

- 如果没有 catEntryId, 那么就使用 storeId (STORE_ID) 获取成员标识 (引用 CATENTRY 表中的 MEMBER_ID)。成员标识必须与产品目录所有者相同。成员标识 (MEMBER_ID) 与 partNumber (引用 CATENTRY 表中的 PARTNUMBER) 一起用于获取 catEntryId。catEntryId 与商店标识和 (该产品目录条目的) 实现中心标识一起用于更新 INVENTORY 表中的行。
- 如果表 INVENTORY 中的该行不存在, 就会出现错误。

异常情况

如果遇到以下异常, 此命令将在错误日志中生成一个条目:

- storeId 不存在。
- 无法使用成员标识 (拥有商店及 partNumber) 找到 catEntryId。
- catEntryId 及 storeId 和实现中心标识无法在表 INVENTORY 中找到匹配行。

SendWCSOrder 任务命令

SendWCSOrder 命令由 WebSphere Commerce 系统使用, 将“订单创建”的旧消息发送到后端系统。

行为

- 通过将此任务命令分配到 OrderProcess 命令中的 OrderMessagingCmd 接口可启用该命令。一旦启用, 就在调用它之后完成 OrderProcess 命令的处理。
- 将订单引用号作为其输入参数使用, 它就可收集所有必需的订单信息。
- 根据可用的订单信息, 它将把“订单创建”的旧消息构建为“字符串”, 然后使用出站消息传递系统设置服务将其存储在消息中。
- 如果消息创建成功, 该命令将试图使用出站消息传递系统发送服务发送此消息。

异常情况

如果遇到异常, 此命令将在错误日志中生成一个条目。

SendXMLOrder 命令

SendXMLOrder 命令由 WebSphere Commerce 出站消息传递系统使用, 将 Report_NC_PurchaseOrder XML 消息发送至后端系统。此命令使用消息编辑模板生成 XML 消息, 然后出站消息传递系统将其发送至后端系统。

行为

- 通过将此任务命令分配到 OrderProcess 命令中的 OrderMessagingCmd 接口可启用该命令。
- 一旦启用, 就在调用它之后完成 OrderProcess 命令的处理。
- 此任务命令调用了消息传递系统编辑服务, 该服务使用 OrderCreateXML.jsp 编辑模板收集必需的订单信息, 并构建 Report_NC_PurchaseOrder 出站 XML 消息。
- 如果编辑成功, 该命令将试图使用出站消息传递系统发送服务发送此消息。

异常情况

如果遇到异常, 此命令将在错误日志中生成一个条目。

NewInboundMessage 命令

NewInboundMessage 命令用于未使用 user_template.xml 入站消息模板定义文件实现的定制入站消息。当适配器未识别人站消息时，将运行此命令。这意味着它既不是旧的消息，也不是在 sys_template.xml 或 user_template.xml 入站消息模板定义文件中定义的 XML 消息。开始时，NewInboundMessage 命令不包含编程语句，因此您必须自己定制此命令。

行为

- 将入站消息存储在 String 缓冲区，该缓冲区可以使用 getMessage() 方法检索。

异常

如果 performExecute() 方法未执行，缺省情况下它就生成一个异常。

OrderConfirmStatus 命令

当从后端系统接收 OrderStatusType 标记值为 OrderConfirm 的 Update_WCS_OrderStatus XML 消息时，将使用 OrderConfirmStatus 命令。此命令更新有关现有订单确认状态的信息。它的参数、行为和异常情况与 OrderStatus 命令相同。这两个命令之间仅有的区别在于 OrderConfirmStatus 命令将 orderStatus 参数的值设置为缺省值 C，这意味着如果未在消息中提供订单，就认为订单已确认。

关于 OrderConfirmStatus 命令参数、行为和异常处理的详细信息，请参阅 OrderStatus 命令。

ProductListPriceUpdate 命令

ProductListPriceUpdate 命令用于更新 Update_WCS_ProductPrice XML 消息的产品价格信息。此命令用于更新 WebSphere Commerce 数据库中有关产品标价的信息。

参数

partNumber

产品目录条目的部件号，已在 CATENTRY 表的 PARTNUMBER 列中定义。

memberId

标识产品目录条目所有者的引用号。

catEntryId

供销售用的产品目录条目。

currency

表示价格的 ISO 4217 货币类型。此值必需。

listPrice

用于更新 LISTPRICE 表的产品标价。

行为

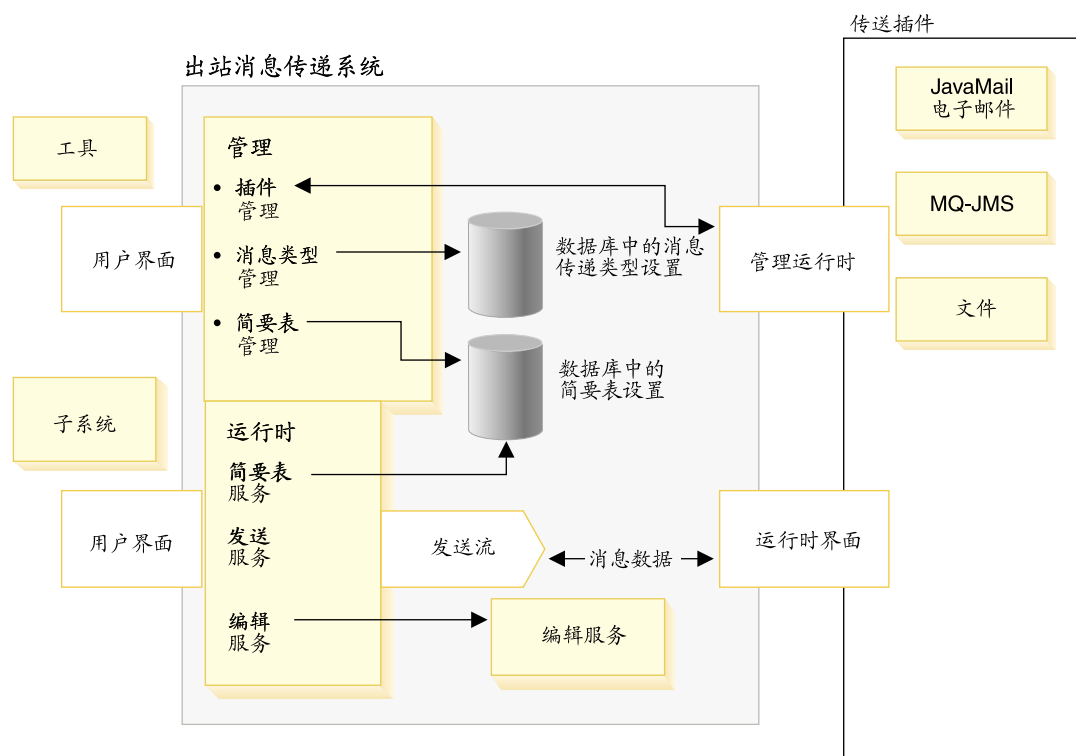
- 此命令更新 LISTPRICE 表中的记录。
- partNumber 与 memberId 一起用于获取产品目录条目键（表 CATENTRY 中的 CATENTRY_ID）。

- 如果该产品目录条目与 LISTPRICE 表中现有的产品目录条目匹配，但货币类型与具有该产品目录条目的任何记录中的货币类型不匹配，将在 LISTPRICE 表中创建一个新记录。这允许您对相同的产品目录条目指定不同的货币价格。

异常情况

- currency 参数为空。
- 无法使用 memberId（它拥有商店及 partNumber）找到产品目录条目。

第 5 章 出站消息传递系统如何工作



消息传递系统使用实现“公共连接器框架”（CCF）的插件提供系统与各种传送方法之间的公共接口。在系统站点管理期间，商店管理员可以执行以下任务：

- 添加、启用和配置传送方法。管理员使用管理控制台创建设置。消息传递系统和传送方法之间的通信在单一的管理接口上发生。
- 维护简要表、指定到个人消息类型的传送和指示每一类型将使用的设置。管理员使用管理控制台执行以上操作。

在运行期间，当 WebSphere Commerce 子系统生成消息时，将发生以下事件：

1. 为消息类型检索相应的简要表。如果该消息不存在商店简要表，则使用站点简要表。简要表用来确定使用何种传送方法和设置。
2. 如果消息使用编辑服务，则模板用于生成消息。
3. 消息通过运行时接口发送至给出通知的传送方法。

公共接口与外部传送的使用可以将实现传送的详细信息与消息传递系统相独立。使用此结构便可以插入在 CCF 接口上附加的额外传送方法。

出站消息传递系统

WebSphere Commerce 消息传递系统允许您管理定义和发送 WebSphere Commerce 内生成的消息的所有方面。他允许您控制通知管理员、顾客、后端系统和实现中心系统各种事件的方法，例如顾客订单或系统错误。

要配置出站消息系统，请使用管理控制台。消息传递系统可以使用某些传送方法发送消息，例如使用 SMTP 的电子邮件和使用 UTF-8 编码的文件。对于电子邮件，它的支持出站协议是 SMTP，消息编码依赖于指定的语言。您也可以选择配置消息传递系统，以使用 MQSeries 将消息发送到后端系统或实现中心系统。

出站消息系统的运行时环境提供了高度定制的消息传递环境。这些特征包括：

- 编辑服务
使用预定义的 JSP 模板定制消息。
- 多条消息传送支持
允许通过多种传送方法发送单一消息。
- 使用相同传送方法的多条通知消息
这对于向多个收件人发送电子邮件广播很有用。
- 对三种处理类型的支持：
 - 已处理
在成功完成当前处理之后应当发送的消息的用法。
 - 立即
应当在 WebSphere Commerce 中发生事件时发送的消息的用法。无论处理提交与否都要发送消息。
 - 回复
需要得到后端系统响应消息的消息的用法。

注意：请确保传送方法属性，例如电子邮件地址地址和文件位置是否有效。“消息传递系统”不提供属性确认；不正确的属性将导致发送消息失败。

出站消息传递系统管理

出站消息系统的管理可以分为两个主要的类别：站点管理和商店管理。站点管理员负责提供站点中所有商店都会使用的基本框架，例如可以使用何种传送方法。例如，商店可能不将电子邮件作为传送方法，除非先前已经过站点管理员配置并启用。这样商店管理员就可以接受站点的级别做出的设置，或自己为他们的商店进行修改。

出站后端集成消息

出站后端消息是 WebSphere Commerce 生成的请求，可以将它发送到外部系统。可以配置 WebSphere Commerce 来生成 Report_NC_PurchaseOrder XML 出站消息，您可以使用此消息与后端系统通信，通知它有顾客下了一份订单。之后生成 XML 消息，并将此消息发送到出站消息系统，格式采用 UTF-8 格式。也可以使用旧的“订单创建”消息，它可以执行相似的功能。

出站消息包含从 WebSphere Commerce Server 发送到外部系统的订单信息，在此订单接受进一步的处理。要启用出站消息，首先需要选择使用哪个，Report_NC_PurchaseOrder XML 消息或是旧的“订单创建”消息。这两条消息不能同时启用。

如果选择使用 Report_NC_PurchaseOrder XML 消息，请使用以下 SQL 语句更新 CMDREG 数据库表。

```
update cmdreg set
```

```
classname='com.ibm.commerce.messaging.commands.SendXMLOrderCmdImpl' where  
interfacename='com.ibm.commerce.order.commands.OrderMessagingCmd'
```

这指定 “SendXMLOrderCmdImpl 任务命令”，它生成消息并将其发送给 OrderProcess 命令的 OrderMessagingCmd 接口。

如果选择使用旧的“订单创建”消息，请使用以下 SQL 语句更新数据库 CMDREG 表：

```
update cmdreg set
```

```
classname='com.ibm.commerce.messaging.commands.SendWCSOrderCmdImpl' where  
interfacename='com.ibm.commerce.order.commands.OrderMessagingCmd'
```

这指定 “SendWCSOrderCmdImpl 任务命令”，它生成消息并将其发送给 OrderProcess 命令的 OrderMessagingCmd 接口。您需重新启动 WebSphere Commerce 实例，以便使更改生效。

也可以创建新的后端集成消息。

出站实现集成消息

出站实现集成消息是由 WebSphere Commerce 生成的请求，可以将其发送到实现中心系统。可以配置 WebSphere Commerce 以生成响应入站消息的出站消息，此出站消息包含将与实现中心系统通信的信息。

出站 XML 消息由出站消息系统送出，采用 Unicode UTF-8 格式编码。关于生成以下出站消息的 JSP 文件的信息，请参阅消息编辑模板。

Response_WCS_ExpectedInvRecord 出站消息允许您在预期库存记录的请求已创建时响应实现中心系统。此消息包含生成的 RA_ID 和 RADETAIL_ID。出站消息由 Create_WCS_ExpectedInventoryRecord 消息调用，由 RACreateResult.jsp 文件生成。

Response_WCS_PickBatch 出站消息允许您与已创建提货批量的实现中心系统通信。出站消息由 Create_WCS_PickBatch 消息调用，由 PickBatchResult.jsp 文件生成。

Report_WCS_PickPackListDetail 出站消息由 Inquire_WCS_PickPackListDetail 消息调用，允许您通过发送指定 PICKBATCH_ID 的详细信息列表来响应实现中心请求。此消息包含给定 PICKBATCH_ID 的 ORDRELEASE 表的提货单和装货列表。

Response_WCS_CreateInvReceipt 出站消息允许您与已创建库存接收的实现中心系统通信。出站消息由 Create_WCS_InventoryReceipt 消息调用，由 CreateInvReceiptOK.jsp 文件调用。

Response_WCS_UpdateInvReceipt 出站消息允许您响应实现中心系统，通知它们在获得接收时已更新了库存。出站消息由 Update_WCS_InventoryReceipt 消息调用，由 UpdateInvReceiptOK.jsp 文件生成。

Response_WCS_CreateShipConfirm 出站消息由 Create_WCS_ShipmentConfirmation 消息调用，允许您响应实现中心系统，确认装运确认已创建。

Release_WCS_ShipmentNotify 出站消息是个出站电子邮件消息，允许您使用在订单已装运时通知顾客的选项。

出站消息传递系统商店管理

商店管理员负责启用商店使用的传送方法。商店管理员可以添加、激活、禁用和配置商店的传送方法，并为消息类型指定传送方法。商店管理员可以选择接受站点管理员所创建的设置，也可将它们覆盖。以下是商店管理所涉及到的任务列表：

- 添加传送方法。
- 激活或禁用传送方法。
- 配置传送方法。
- 为消息类型指定传送方法。

一旦商店管理员重设了站点的设置，则以后站点管理员对特定设置的任何更改都不会对该商店生效。对其它站点级别设置的更改，而非商店管理员进行的修改仍会对该商店生效。例如，如果在 SMTP 主机上为站点配置了电子邮件，smtp.host1.com，但商店 A 已指定了 smtp.host2.com，则以后对电子邮件站点级别的更改将不会对商店 A 的设置生效。

出站消息传递系统站点管理

站点管理员可以确定站点支持哪种传送方法，并在站点范围的基础上对其进行配置。站点级别管理提供缺省设置，该设置可以由商店级别管理设置覆盖。

站点管理员可以激活和配置站点的传送方法和消息类型，也让商店管理员指定他们自己的设置。以下是站点管理涉及到的任务列表：

- 添加传送方法。
- 激活或禁用传送方法。
- 配置传送。这提供可以由商店管理员覆盖的缺省配置。
- 为消息类型指定传送方法。这些指定可以由商店管理员覆盖。

只有站点管理员可以执行以下任务：

- 启用错误通知向管理员发送电子邮件消息
- 启用 MQSeries JMS 传送向后端系统发送消息
- 启用订单状态通知及时告知顾客或管理员现有订单的状态

向传送适配器添加新消息

WebSphere Commerce 允许您扩展传送适配器以处理附加消息。这包括创建一个消息的 XML 模版以及将消息映射至控制器命令中。要完成这些，需要具有对 XML 和 WebSphere Commerce 控制器命令深入的了解。如果您的消息需要创建新的控制器命令，则还需具有对 WebSphere Commerce 数据库模式和 Java 编程较深入的了解。

第 6 章 启用出站消息传递发送 / 接收发送服务

WebSphere Commerce 可以通过出站消息传递发送 / 接收发送服务与其它系统交互。这允许 WebSphere Commerce 将消息发送给另一个系统并等待应答。除了“发送 / 接收”等待来自接收其消息请求的系统的返回应答之外，它的行为类似于 `SendImmediate` 方法。

在发送请求消息之后，WebSphere Commerce 侦听入站队列并等待，直至带有与请求消息的消息标识相等的相关标识的应答消息被放置在队列中。

您应当对“发送 / 接收”使用单独队列，它来自用于 MQSeries 适配器的队列。

要启用 WebSphere Commerce 以使用“发送 / 接收”消息服务，请执行以下操作：

1. 为用于“发送 / 接收”的新消息在 `MSGTYPES` 表中创建一个消息类型。
`VIEWNAME` 列中的值是用于生成要发送的消息的 `VIEWNAME`。
2. 创建使用 `sendReceiveImmediate` 发送模式和在步骤 1 中创建的消息类型的命令。
它用于调用 `SendMsg` 接口。
3. 要生成出站消息，请在 `VIEWREG` 表中创建一个条目以将步骤 1 中创建的 `VIEWNAME` 与 JSP 文件关联起来。条目必须使用以下值：

INTERFACENAME

`com.ibm.commerce.messaging.viewcommands.MessagingViewCommand`

CLASSNAME

`com.ibm.commerce.messaging.viewcommands.MessagingViewCommandImpl`

PROPERTIES

`docname=JSP filename`

4. 创建 JSP 文件以生成消息发送到出站队列。
5. 使用管理控制台确保指定给“发送 / 接收”的传送处于活动状态。
6. 使用管理控制台定义每个在步骤 1 中创建的消息类型。在 **方式** 字段中指定 0。这表是通信的“发送 / 接收”方式。

启用消息传递系统传送适配器

要启用消息传递系统传送适配器，请执行以下操作：

1. 启动配置管理器。
2. 选择 **主机名** → **实例**，然后打开 **组件** 文件夹。
3. 选择 **传送适配器**。
4. 确保已激活 **启用组件** 旁边的复选框，并单击 **应用**。
5. 退出配置管理器。
6. 重新启动 WebSphere Application Server。

指定出错条件消息类型至传送方法

确保指定错误消息至此站点而不是个别商店。要指定出错条件消息类型至传送方法，请执行以下操作：

1. 打开管理控制台并以站点管理员身份登录。
2. 从**配置**菜单中，选择**消息类型**。显示“消息类型配置”页面。
3. 单击**新建**。“消息传送指定”页面打开。
 - a. 在**消息类型**下拉列表上，选择 WebSphere Commerce 中发生的“错误条件的描述”。
 - b. 在**消息严重性**字段中，指定 0 0。
 - c. 从**传送**下拉列表中，选择电子邮件。
 - d. 从**设备格式**下拉列表中，选择“标准设备格式”。
4. 单击**下一步**。显示“消息传送指定”参数页面。
 - a. 在**主机**字段中，输入邮件服务器的全限定名称。
 - b. 在**协议**字段中，输入 smtp。
 - c. 在**收件人**字段中，指定应当接收错误通知消息的管理员。如果指定了多个收件人，则请用逗号分隔收件人姓名。
 - d. 在**发件人**字段中，指定消息的发件人。
此文本出现在电子邮件消息的“发件人”字段中。
 - e. 在**主题**字段中，指定消息的主题。
此文本出现在电子邮件消息的“主题”字段中。
5. 单击**完成**。

激活商店的传送方法

要激活商店现有的传送方法，请执行以下操作：

1. 打开管理控制台并以商店管理员身份登录。
2. 从**配置**菜单中，选择**传送**。显示“传送配置”页面。
3. 单击要激活的传送旁边的复选框。
4. 单击**更改状态**。页面重新装入，状态也随之更改。

将传送方法添加到商店

要添加新的传送方法至商店，请执行以下操作：

1. 打开管理控制台并以商店管理员身份登录。
2. 从**配置**菜单中，选择**传送**。显示“传送配置”页面。
3. 单击**添加**。“添加传送”对话框打开。
4. 选择要添加到商店的传送旁边的复选框。可以通过选择左上方顶部复选框来选择所有传送方法。如果没有可用的传送，则已经添加了站点管理员使其可用的所有传送。
5. 单击**添加**来添加传送，或者单击**取消**来返回到“传送配置”页面。

配置商店的传送方法

要配置商店的传送方法，请执行以下操作：

1. 打开管理控制台并以商店管理员身份登录。
2. 从**配置**菜单中，选择**传送**。显示“传送配置”页面。
3. 选择要配置的方法旁边的复选框。
4. 单击**配置**。“传送配置参数”页面打开。选定的传送的名称显示在参数表的左上方。
5. 在相应的字段中提供传送方法的信息。
6. 单击**确定**来接受更改，或者单击**取消**来在不实现更改的情况下返回到“传送配置”菜单。

注意：除非不希望设置生效，否则不要单击**确定**。一旦单击**确定**，站点管理员对此传送方法配置的更改将不再影响此商店。如果没有进行任何更改或不希望此设置生效，单击**取消**。

停用商店的传送方法

要停用商店现有的传送方法，请执行以下操作：

1. 打开管理控制台并以商店管理员身份登录。
2. 从**配置**菜单中，选择**传送**。显示“传送配置”页面。
3. 单击要禁用的传送的复选框。
4. 单击**更改状态**。页面重新装入，指示传送状态现在为“禁用”。

将传送方法指定到商店的消息类型

商店管理员可以接受站点管理员做出的传送方法指定，或者也可以为他们的商店将其覆盖。要指定将用于他们商店的特定消息类型的传送方法，请执行以下操作：

1. 打开管理控制台并以商店管理员身份登录。
2. 从**配置**菜单中，选择**消息类型**。显示“消息类型配置”页面。
3. 单击要向其指定传送方法的消息类型旁边的复选框，并单击**更改**。如果消息类型不在列表中，请单击**新建**。打开“消息传送指定”页面。
4. 如果这是新的传送指定，请从**消息类型**下拉列表中选择向其指定传送的消息类型。
5. 在相应的字段中输入传送配置值。一般的，建议使用**消息严重性 0,0**和“标准设备格式”。
6. 单击**下一步**为指定的消息类型配置传送方法参数。
7. 输入为此消息类型选择的传送的属性。
8. 单击**完成**保存更改或者**取消**来返回到“消息类型配置”页面。

注意：除非希望设置生效，不要单击**完成**。一旦单击**完成**，站点管理员对此传送方法配置的更改将不再影响此商店。如果没有进行任何更改或不希望此设置生效，单击**取消**。

激活站点的传送方法

要激活站点现有的传送方法，请执行以下操作：

1. 打开管理控制台并以站点管理员身份登录。
2. 从**配置**菜单中，选择**传送**。显示“传送配置”页面。
3. 单击要激活的传送旁边的复选框。
4. 单击**更改状态**。页面重新装入，指示传送状态现在为激活。

添加传送方法到站点

要添加新的传送方法至站点，请执行以下操作：

1. 打开管理控制台并以站点管理员身份登录。
2. 从**配置**菜单中，选择**传送**。显示“传送配置”页面。
3. 单击**添加**来打开“添加传送”页面。
4. 选择要添加到站点的传送旁边的复选框。可以通过选择顶部复选框来选择所有传送方法。
5. 单击**添加**来接受更改，或者单击**取消**来返回到“传送配置”页面。
将传送方法添加到站点时，它被自动激活。

将传送方法指定到站点的消息类型

要指定用于站点的指定消息类型的传送方法，请执行以下操作：

1. 打开管理控制台并以站点管理员身份登录。
2. 从**配置**菜单中，选择**消息类型**。显示“消息类型配置”页面。
3. 单击要向其指定传送方法的消息类型旁边的复选框，并单击**更改**。如果消息类型不在列表中，请单击**新建**。“消息传送指定”页面打开。
4. 如果这是新的传送指定，请从**消息类型**下拉列表中选择向其指定传送的消息类型。
5. 在相应的字段中输入传送配置值。一般的，建议使用**消息严重性 0,0**，和“标准设备格式”。
6. 单击**下一步**为指定的消息类型配置传送方法参数。
7. 输入为此消息类型选择的传送的属性。
8. 单击**完成**，或单击**取消**返回到“消息类型配置”页面。

配置站点的传送方法

要配置站点的传送方法，请执行以下操作：

1. 打开管理控制台并以站点管理员身份登录。
2. 从**配置**菜单中，选择**传送**。显示“传送配置”页面。
3. 选择要配置的方法旁边的复选框。
4. 单击**配置**。“传送配置参数”对话框打开。选定的传送方法的名称出现在参数表的左上方。
5. 输入传送方法使用的值。
6. 单击**确定**来接受更改，或者单击**取消**来返回到“传送配置”页面。

停用站点的传送方法

要禁用站点现有的传送方法，请执行以下操作：

1. 打开管理控制台并以站点管理员身份登录。
2. 从**配置**菜单中，选择**传送**。显示“传送配置”页面。
3. 单击要停用的传送旁边的复选框。
4. 单击**更改状态**。重新装入页面，状态也随之更改。

检查电子邮件传送方法的系统设置

确保电子邮件传送方法的缺省设置通过执行以下操作已经完成设置：

1. 启动配置管理器。
2. 选择实例然后打开**传送** → **出站** → **JavaMail** → **ConnectionSpec** 文件夹。
3. 在右框架中单击**高级**选项卡。
4. 将**主机**的值设置为 SMTP 邮件服务器。
5. 将**协议**的值设置为 smtp。
6. 单击**应用**。
7. 重新启动 Websphere Commerce Server。

注意： 这提供了电子邮件传送的系统范围缺省设置。在管理控制台中创建的那些设置将覆盖此设置。

激活通知

一旦已经设置了电子邮件传送方法的系统缺省设置，请通过执行以下操作激活通知：

1. 通过进行以下设置编辑 `<instance_name>.xml`：

```
<Notification display="false">
  <order enabled="true" />
  <error enabled="true" />
  <pwdreset enabled="true" />
</Notification>
```

2. 重新启动 WebSphere Commerce Server。

注意： 当为错误通知发送电子邮件时，如果遇到问题，可能需要采取以下操作：

- 检查配置值是否正确。
- 通过删除所有在商店级别上配置传送的条目，来清除所有配置数据。以下 SQL 语句将删除所有条目：

```
delete from cseditatt where store_id is not null
```

- 打开管理控制台，然后检查所有消息类型是否配置正确并且没有在商店级别上配置传送。

启用错误通知

要启用电子邮件错误通知，请执行以下操作：

1. 确保已经设置了电子邮件传送方法的系统缺省设置
2. 激活通知给错误消息。
3. 指定出错条件消息类型至传送。
4. 将传送方法指定到站点的消息类型。

在“消息传送指定”页面上，输入这些值：

- a. 在消息类型下拉列表上，选择 WebSphere Commerce 中发生的“错误条件的描述”。
- b. 在消息严重性字段中，指定 0 0。
- c. 从传送下拉列表中，选择电子邮件。
- d. 从设备格式下拉列表中，选择“标准设备格式”。

在“消息传送指定”参数页面上，使用以下值：

- a. 在主机字段中，输入邮件服务器的全限定名称。
- b. 在协议字段中，输入 smtp。
- c. 在收件人字段中，指定应当接收错误通知消息的管理员。如果指定了多个收件人，则请用逗号分隔收件人姓名。
- d. 在发件人字段中，指定消息的发件人。
此文本出现在电子邮件消息的“发件人”字段中。
- e. 在主题字段中，指定消息的主题。
此文本出现在电子邮件消息的“主题”字段中。

注意：当为错误通知发送电子邮件时，如果遇到问题，可能需要采取以下操作：

- 检查配置值是否正确。
- 通过删除所有在商店级别上配置传送的条目，来清除所有配置数据。以下 SQL 语句将删除所有条目：

```
delete from ccreditatt where store_id is not null
```

- 打开管理控制台，然后检查所有消息类型是否配置正确并且没有在商店级别上配置传送。

激活通知

一旦已经设置了电子邮件传送方法的系统缺省设置，请通过执行以下操作激活通知：

1. 通过进行以下设置编辑 `<instance_name>.xml`：

```
<Notification display="false">
  <order enabled="true" />
  <error enabled="true" />
  <pwdreset enabled="true" />
</Notification>
```

2. 重新启动 WebSphere Commerce Server。

注意: 当为错误通知发送电子邮件时, 如果遇到问题, 可能需要采取以下操作:

- 检查配置值是否正确。
- 通过删除所有在商店级别上配置传送的条目, 来清除所有配置数据。以下 SQL 语句将删除所有条目:

```
delete from cseditatt where store_id is not null
```

- 打开管理控制台, 然后检查所有消息类型是否配置正确并且没有在商店级别上配置传送。

启用装运通知电子邮件

要启用为顾客订单已装运时发送给顾客的装运通知电子邮件, 请执行以下操作:

- 更新 CMDREG 数据库表, 使用以下 SQL 语句

```
update cmdreg set
```

```
classname='com.ibm.commerce.messaging.commands.ReleaseShipNotifyCmdImpl' where  
interfacename='com.ibm.commerce.messaging.commands.ReleaseShipNotifyCmd'
```

注意: 缺省情况下, Release_WCS_ShipmentNotify 是禁用的, 在 CMDREG 数据库表中
使用 ReleaseShipNotifyDummyImpl 为 CLASSNAME。ReleaseShipNotifyDummyImpl 是
个哑元实现, 不执行任何操作。

启用广播式消息

要发送广播式电子邮件消息, 请执行以下操作:

1. 创建名为 BroadcastMessage.jsp 的 JSP 文件。
2. 将文件放置在站点或商店目录中。
3. 打开管理控制台并以站点管理员或商店管理员身份登录。
 - a. 从**配置**菜单中, 选择**消息类型**。显示“消息类型配置”页面。
 - b. 单击**新建**。打开“消息传送指定”页面。
 - 1) 在**消息类型**下拉列表中, 选择广播式消息。
 - 2) 在**消息严重性**字段中, 指定 0 0。
 - 3) 从**传送**下拉列表中, 选择电子邮件。
 - 4) 从**设备格式**下拉列表中, 选择“标准设备格式”。
 - c. 单击**下一步**。显示“消息传送指定”参数页面。
 - 1) 在**主机**字段中, 输入邮件服务器的全限定名称。
 - 2) 在**协议**字段中, 输入 smtp。
 - 3) 在**收件人**字段中, 指定缺省收件人。如果指定了多个收件人, 则请用逗号分隔收件人姓名。收件人姓名将在运行时替换为顾客电子邮件地址。
 - 4) 在**发件人**字段中, 指定消息的发件人。此文本作为电子邮件中“发件人”行的内容出现。
 - 5) 在**主题**字段中, 指定消息的主题。此文本作为电子邮件中“主题”行的内容出现。
 - d. 单击**完成**。

- e. 要发送消息，请遵循 `BroadcastMessage` 命令参考文件的指导。
例如，要将消息发送给所有从站点中的商店购买了部件号“sku1234”的顾客，请在浏览器的地址行上输入以下信息：

```
BroadcastMessage?subject=testing&messageContent=this+is+a+test
&sender=example%40ca.ibm.com&mode=2&partNumber=sku1234&URL=BroadcastMessage.jsp
```

启用订单状态通知

要启用订单状态消息通知，请执行以下操作：

1. 在 WebSphere Commerce 机器上打开数据库命令窗口。
2. 通过输入以下 SQL 语句注册订单状态通知命令：

```
update cmdreg set
classname='com.ibm.commerce.messaging.commands.OrderStatusNotifySendCmdImpl'where
interfacename='com.ibm.commerce.messaging.commands.OrderStatusNotifySendCmd'
```
3. 停止并启动 WebSphere Commerce Server。
4. 如有必要，添加方法到站点或商店。
5. 将“订单状态的通知消息”类型指定给传送。请确保选择了**设备格式**下拉列表中的**HTTP 浏览器**。

设置出站消息编辑

要为出站消息设置并使用编辑服务，请执行以下操作：

1. 使用这些方法中的一个，将传送指定给相应的消息类型：
 - 将传送方法指定到站点的消息类型
 - 为商店消息类型指定传送方法
必须为使用的每个传送指定一个有效的设备格式（如 `DEVICEFMT` 表中所指定的）。
2. 参阅主题 出站消息传递系统接口中的信息，使用 `SendMsgCmd` 任务命令创建消息传递系统对象。使用 `setMsgType()` 和 `setStoreId()` 初始化服务。
3. 调用消息传递系统撰写方法。

指定出错条件消息类型至传送方法

确保指定错误消息至此站点而不是个别商店。要指定出错条件消息类型至传送方法，请执行以下操作：

1. 打开管理控制台并以站点管理员身份登录。
2. 从**配置**菜单中，选择**消息类型**。显示“消息类型配置”页面。
3. 单击**新建**。“消息传送指定”页面打开。
 - a. 在**消息类型**下拉列表上，选择 WebSphere Commerce 中发生的“错误条件的描述”。
 - b. 在**消息严重性**字段中，指定 0 0。
 - c. 从**传送**下拉列表中，选择电子邮件。
 - d. 从**设备格式**下拉列表中，选择“标准设备格式”。
4. 单击**下一步**。显示“消息传送指定”参数页面。

- a. 在**主机**字段中，输入邮件服务器的全限定名称。
 - b. 在**协议**字段中，输入 `smtp`。
 - c. 在**收件人**字段中，指定应当接收错误通知消息的管理员。如果指定了多个收件人，则请用逗号分隔收件人姓名。
 - d. 在**发件人**字段中，指定消息的发件人。
此文本出现在电子邮件消息的“发件人”字段中。
 - e. 在**主题**字段中，指定消息的主题。
此文本出现在电子邮件消息的“主题”字段中。
5. 单击**完成**。

启用订单状态通知

要启用订单状态消息通知，请执行以下操作：

1. 在 WebSphere Commerce 机器上打开数据库命令窗口。
2. 通过输入以下 SQL 语句注册订单状态通知命令：

```
update cmdreg set  
classaname='com.ibm.commerce.messaging.commands.OrderStatusNotifySendCmdImpl'where  
interfacename='com.ibm.commerce.messaging.commands.OrderStatusNotifySendCmd'
```
3. 停止并启动 WebSphere Commerce Server。
4. 如有必要，添加方法到站点或商店。
5. 将“订单状态的通知消息”类型指定给传送。请确保选择了**设备格式**下拉列表中的**HTTP 浏览器**。

启用从管理控制台发送的消息

如果您的商店是根据流行时尚样本商店创建的，则您可以使用 WebSphere Commerce 管理控制台来使客户服务代表可以向顾客发送消息。要启用从 WebSphere Commerce 管理控制台发送的消息，请执行以下操作：

1. 打开管理控制台并以站点管理员或商店管理员身份登录。
2. 从**配置**菜单中，选择**消息类型**。显示“消息类型配置”页面。
3. 单击**新建**。打开“消息传送指定”页面。
 - a. 在**消息类型**下拉列表中，选择由客户服务代表发送的“订单”相关消息。
 - b. 在**消息严重性**字段中，指定 `0 0`。
 - c. 从**传送**下拉列表中，选择电子邮件。
 - d. 从**设备格式**下拉列表中，选择“标准设备格式”。
4. 单击**下一步**。显示“消息传送指定”参数页面。
 - a. 在**主机**字段中，输入邮件服务器的全限定名称。
 - b. 在**协议**字段中，输入 `smtp`。
 - c. 在**收件人**字段中，指定缺省收件人。如果指定了多个收件人，则请用逗号分开收件人名称。收件人名称将在运行时替换为顾客电子邮件地址。
 - d. 在**发件人**字段中，指定消息的发件人。
此文本出现在电子邮件消息的“发件人”字段中。（在“商店服务”中输入的电子邮件地址值将覆盖此值。）

- e. 在**主题**字段中，指定消息的主题。
此文本出现在电子邮件消息的“主题”字段中。（在“商店服务”中输入的值将覆盖此值。）
5. 单击**完成**。

启用出站 **OrderCreate** 消息

OrderCreate 出站消息可以 XML 格式或现有格式启用。通过以下任务命令可生成这两个格式：

1. **SendXMLOrder** 生成 **Report_NC_PurchaseOrder** XML 消息。
2. **SendWCSOrder** 生成“订单创建”的旧消息。

除非您是从 **WebSphere Commerce** 的先前版本迁移过来的，并想要保持现有的格式，否则建议您使用 XML 格式。您无法同时启用这两者。

要启用 **Report_NC_PurchaseOrder** XML 消息，可使用以下 SQL 语句更新数据库 **CMDREG** 表：

```
update cmdreg set  
classname='com.ibm.commerce.messaging.commands.SendXMLOrderCmdImpl' where  
interfacename='com.ibm.commerce.order.commands.OrderMessagingCmd'
```

要启用“订单创建”的旧消息，请使用以下 SQL 语句更新数据库 **CMDREG** 表：

```
update cmdreg set  
classname='com.ibm.commerce.messaging.commands.SendWCSOrderCmdImpl' where  
interfacename='com.ibm.commerce.order.commands.OrderMessagingCmd'
```

第 7 章 消息类型

WebSphere Commerce 出站消息传递系统可以处理不同消息类型。每种消息类型都因响应 WebSphere Commerce 系统内发生的特定类型事件而发送到消息传递系统。消息传递系统根据管理控制台指定的消息类型和消息设置来处理消息。以下表显示了由出站消息传递系统支持的消息类型。

消息类型 在 MSGTYPE 表中	管理控制台中的名称	用法
ErrorMessage	对 WebSphere WebSphere Commerce 中发生的错误条件的描述	配置该消息类型在 WebSphere Commerce 中发生错误时使管理员能够接收电子邮件消息。 要启用此消息，请参阅启用错误通知。
OrderCreateFixFormat	WebSphere Commerce 订单创建的出站消息	指示 WebSphere Commerce 中已经创建一个订单。可以使用消息将出站 WebSphere Commerce 订单创建消息发送到后端系统。 请参阅启用出站 OrderCreate 消息。
OrderCreateXMLFormat	WebSphere Commerce XML 创建的出站消息	指示 WebSphere Commerce 中已经创建一个订单。可以使用消息将出站 WebSphere Commerce 订单创建消息发送到后端系统。 请参阅启用出站 OrderCreate 消息。
OrderStatusNotify	订单状态的通知消息	指示订单已经更改的状态 请参阅启用订单状态通知。
OrderAuthorized	关于已授权订单的消息	指示已经授权一个订单。 要发送已授权订单消息，您需要创建一个 JSP 模板。 请参阅 OrderProcess 命令。
OrderReceived	关于已接收订单的消息	指示已经接收一个订单。 要发送已接收的订单消息，您将需要创建一个 JSP 模板。 请参阅 OrderProcess 命令。

消息类型 在 MSGTYPE 表中	管理控制台中的名称	用法
OrderRejected	关于已拒绝订单的消息	指示已经拒绝一个订单。 要发送已拒绝的订单消息，您需要创建一个 JSP 模板。 请参阅 OrderProcess 命令。
OrderCancel	关于已取消订单的通知消息	指示已经取消一个订单。 要发送已取消订单消息，您需要创建一个 JSP 模板。 请参阅 OrderProcess 命令。
PasswordNotify	关于密码重新设置的通知消息	配置该消息类型以启用要发送给顾客的电子邮件消息，该消息指示已重新设置了顾客的密码。
BroadcastMessage	广播式消息	配置该消息类型以发送广播式消息给顾客。 要发送广播式消息，您需要创建一个广播式消息 JSP 模板。 请参阅 BroadcastMessage 命令。
MerchantOrderNotify	用于给商家通知订单信息的信息	与 OrderProcess 命令的 NotifyMerchant 参数有关。 要发送通知消息，您需要创建一个 JSP 模板。 请参阅 OrderProcess 命令。
AdminOrderComment	客户服务代表发送的订单相关消息	配置该消息类型以使客户服务代表能够从 WebSphere 贸易加速器发送顾客电子邮件消息。
NotifyReleaseShip	消息，将其发送给顾客通知他们其订单已经装运。	配置此消息类型以将有关订单的装运确认信息的电子邮件消息发送给顾客。 请参阅 ReleaseShipConfirm 命令。

出站消息传递系统接口

可以通过 **SendMsgCmd** 任务命令进行与出站消息传递系统的交互。此任务命令向消息传递系统提供的方法赋予所有可用接口，设置必需的参数，从而构造和使用消息传递系统对象。以下是 **SendMsgCmd** 命令接口中所有可用服务的列表：

- 初始化服务

- 消息内容设置服务
- 添加电子邮件部件或附件服务
- 可配置消息数据服务
- 发送服务
- 其它服务

出站消息扩展

Report_NC_PurchaserOrder 消息包含 UserData XML 元素，它又包含 UserDataField 元素。您可以定制 UserData 元素，以通过扩展 SendXMLOrderMsg 命令和实现 getHeaderExtensionRecords() 方法或 getItemExtensionRecords() 方法来传送 Report_NC_PurchaserOrder 出站消息中不包含的额外信息，而这取决于您是在页眉级别还是在商品级别添加信息。缺省情况下，getHeaderExtensionRecords() 和 getItemExtensionRecords() 方法都返回空 String 值。这两种方法实现之后，都应返回 String 对象，该对象包含一系列如下 UserDataField 元素：

```
<UserDataField name="field_name">field_value</UserDataField>
.
  UserDataField repeated loop
.
<UserDataField name="field_name">field_value</UserDataField>
```

关于如何扩展命令的更多信息，请参阅《WebSphere Commerce 程序员指南》。下表概括了为 Report_NC_PurchaseOrder 消息的 XML 元素调用的方法：

消息	XML 元素	已调用方法
Report_NC_PurchaserOrder	Report_PO_Header	getHeaderExtensionRecords()
Report_NC_PurchaserOrder	Report_PO_Item	getItemExtensionRecords()

“订单创建”消息包括 USRLST 记录，它包含 DATUSR_DATA 数据段。DATUSR 允许向消息中添加可选字段。您可以定制 DATUSR 记录，以通过扩展 SendWCSOrderMsg 命令和实现 getHeaderExtensionRecords() 方法或 getItemExtensionRecords() 方法传送“订单创建”出站消息中不包含的额外信息，这取决于您想在页眉级别还是在商品级别添加信息。缺省情况下，getHeaderExtensionRecords() 和 getItemExtensionRecords() 方法都返回空 String 值。这两种方法实现之后，都应返回 String 对象，该对象包含一系列如下 DATUSR 记录：

有关如何扩展命令的更多信息，请参阅《WebSphere Commerce 程序员指南》。下表概括了为“订单创建”消息的部分调用的方法：

消息	USRLST 部分	已调用方法
订单创建	<HEADER>	getHeaderExtensionRecords()
订单创建	<ITMDAT>	getItemExtensionRecords()

新出站消息支持

除了支持的 XML 和 WebSphere Commerce 出站消息以外，您还可以对新的出站消息添加支持。要添加新出站消息，您必须编写新的控制器命令，构建新出站消息的内容，并使用出站消息传递系统的发送服务将该消息发送到后端系统。

构建新出站消息内容的方法有两种。第一种方法是在您自己的 String 缓冲区构建新出站消息，并通过使用它的消息内容设置服务（直接设置消息内容）将它分配到出站消息传递系统。选择此方法需要在您编写的控制器命令中包含构建消息的逻辑。第二种方法是通过使用出站消息传递系统编辑服务构建新出站消息。通过使用 JSP 模板，编辑服务将根据在 JSP 模板中定义的消息布局和内容生成出站消息。

关于如何使用出站消息传递服务的更多信息，请参阅“出站消息传递系统接口”部分。关于如何写命令的更多信息，请参阅《WebSphere Commerce 程序员指南》。

出站消息的 UserData 元素

出站 XML 消息 Report_NC_PurchaseOrder 包含 UserData XML 元素，将其作为可选元素。包括此消息中要发送附加数据的 UserDataField 元素。可以定制字段，传送未包括在这些消息中的额外数据。

要添加的新字段名称应是 UserDataField 元素的属性名。

以下的 DTD 描述了 UserData 元素：

```
<!ELEMENT UserData (UserDataField+)> <!ELEMENT UserDataField (#PCDATA)>
<!ATTLIST UserDataField
name CDATA #REQUIRED>
```

以下是 UserData 元素的示例：

```
<UserData>
  <UserDataField name="field_name">field_value</UserDataField>
  .
  .UserDataField repeated loop
  .
</UserData>
```

出站消息传递系统接口编程示例

以下 Java 代码段显示与出站消息传递系统间的交互如何发生。示例 1 显示如何构建新的 XML 消息，以及如何通过出站消息传递系统发送它。示例 2 显示如何构建电子邮件消息，并通过出站消息传递系统将其发送出去：

示例 1

```
try { com.ibm.commerce.messaging.commands.SendMsgCmd api =
(com.ibm.commerce.messaging.commands.SendMsgCmd)
CommandFactory.createCommand(SendMsgCmd.NAME, getStoreId()); // Assume you have
set the msgType in the MSGTYPES table to 100 and you are using // storeId of
```

```

1. api.setMsgType(new Integer(100)); api.setStoreID(new Integer(1)); // You
have to choice on how to build the msg: // First choice: build your xml msg
in a String object and then use the setContent(). String OrderCreateMsg = new
String("<?xml version="1.0" encoding="UTF-8"?> ...");
api.setContent(OrderCreateMsg); // Or, use the message composition services
(compose()) by passing the template/view name // This view name should be
registered in VIEWREG and MSGTYPES tables referring to // a JSP message layout
template. String viewName = new String("OrderCreateMsgView"); TypedProperty
tp = new TypedProperty(); // get the orderRefNumber and put it into tp
tp.put("ORDER_REF_NUMBER", getOrderRn().toString()); // get the languageId and
put it into tp tp.put("LANGUAGE_ID", getCommandContext().getLanguageId()); //
Pass the viewName, command Context and parameters stored in tp to compose
services. // Upon successful completion, a message is build according to message
layout defined in the // JSP message layout template referred by viewName.
api.compose(viewName, getCommandContext(), tp); // Send out the message using
sendTransacted send service. api.sendTransacted(); // Set the command context
obtained from the controller command.
api.setCommandContext(getCommandContext()); // Run the outbound messaging system
services api.execute(); } catch (Exception ex ) {
ex.printStackTrace(System.err); }

```

示例 2

```

try { com.ibm.commerce.messaging.commands.SendMsgCmd api =
(com.ibm.commerce.messaging.commands.SendMsgCmd)
CommandFactory.createCommand(SendMsgCmd.NAME, getStoreId()); // Assume you have
set the msgType in the MSGTYPES table to 200 and you are using // storeId of
1. api.setMsgType(new Integer(200)); api.setStoreID(new Integer(1)); // You
have to choice on how to build the msg: // First choice: build your xml msg
in a String object and then use the setContent(). String OrderNotifyMsg = new
String("Your Order has been received. Thank You for Shopping with us.");
api.setContent(OrderNotifyMsg); // Or, use the message composition services
(compose()) by passing the template/view name // This view name should be
registered in VIEWREG and MSGTYPES tables referring to // a JSP message layout
template. String viewName = new String("OrderNotifyMsgView"); TypedProperty
tp = null; // Pass the viewName, command Context and null parameter stored in
tp to compose services. // Upon successful completion, a message is build
according to message layout defined in the // JSP message layout template
referred by viewName. api.compose(viewName, getCommandContext(), tp); // Set
the subject, recipient and sender information using Configurable message data
services api.setConfigData("subject","Your Order has been received");
api.setConfigData("recipient", getEmailAddress());
api.setConfigData("sender","storeAdmin@storeABC.com); // Send out the message
using sendImmediate send service. api.sendImmediate(); // Set the command
context obtained from the controller command.
api.setCommandContext(getCommandContext()); // Run the outbound messaging system
services api.execute(); } catch (Exception ex ) {
ex.printStackTrace(System.err); }

```

消息编辑模板

WebSphere Commerce 出站消息传递系统包含许多消息类型的 JavaServer Pages 编辑模板。当 WebSphere Commerce 内生成其中一种消息类型时，消息编辑服务使用相应的模板创建出站消息。创建了出站消息后，就可以通过指定给管理控制台中消息类型的任何传送将它发送。OrderCreateXMLFormat 是使用消息编辑模板的一个消息类型示例，它使用 OrderCreateXML.jsp 模板。可以在以下目录中找到此 JavaServer Pages 模板。

▶ 2000 `drive:\Program`

`Files\WebSphere\CommerceServer\installedApps\WC_Enterprise_App<instance_name>.ear\wcstores.war`

▶ NT

`drive:\WebSphere\CommerceServer\installedApps\WC_Enterprise_App<instance_name>.ear\wcstores.war`

▶ AIX

`/usr/WebSphere/CommerceServer/installedApps/WC_Enterprise_App<instance_name>.ear/wcstores.war`

▶ Solaris

`/opt/WebSphere/CommerceServer/installedApps/WC_Enterprise_App<instance_name>.ear/wcstores.war`

▶ 400

`/QIBM/Userdata/WebASAdv4/<WAS_instance_name>installedApps/WC_Enterprise_App<instance_name>.ear/wcstores.war`

下表显示了使用编辑模板的消息类型的示例，以及与它关联的 JSP 文件。

消息类型	模板
OrderCreateXMLFormat	OrderCreateXML.jsp
OrderStatusNotify	OrderStatusNotify.jsp
PasswordReset	PasswordResetNotification.jsp

您可以定制这些 JSP 文件。

以下表显示了使用编辑模板的其它消息类型，要使用这些消息类型创建您自己的带有缺省名称的 JSP 模板（缺省名称在 VIEWREG 表已注册）：

消息类型	缺省模板名称
OrderAuthorized	OrderAuthorized.jsp
OrderReceived	OrderReceived.jsp
OrderRejected	OrderRejected.jsp
BroadcastMessage	BroadcastMessage.jsp
MerchantOrderNotify	MerchantOrderNotification.jsp
OrderCancel	OrderCanceledNotification.jsp

初始化服务

这些方法设置了初始参数，该参数标识了哪个消息传递简要表用于当前的消息。这些参数检索在管理控制台中创建和维护的信息。

- `public void setMsgType(Integer msgType)` 此方法是必需的。它用于设置当前消息的“消息类型”。
- `public void setStoreID(Integer storeId)` 此方法是必需的。它用于检索此商店的消息简要表信息。要检索站点级信息，可以使用站点级商店的标识。消息传递系统将试图根据输入的商店检索简要表。如果不存在简要表，则它将试图根据缺省站点标识来检索简要表。
- `public void setPriority(Integer priority)` 此方法提供了可选的初始化信息。指定的 `priority` 整数限制了已检索的简要表。将只对当前消息检索那些简要表（其优先级范围包含此整数）。

消息内容设置服务

您可以使用编辑服务或直接设置消息内容。要使用消息传递系统编辑服务，请使用以下服务：

- 要使用消息传递系统编辑服务，请使用以下服务：
 - `public void compose(String viewName, CommandContext cmdContext, TypedProperty inParms)` 此方法访问了编辑器的功能。它允许用户通过使用 JSP 模板设置消息内容。关于此主题的更多信息，请参阅编辑器的文档。`cmdContext` 参数向编辑器提供了必需的上下文信息。`viewName` 参数允许用户确定 JSP，它将由编辑器访问。`inParms` 参数表示要传送到 JSP 的数据。关于可以在 `TypedProperty` 对象中放置何值，有一些控制规则。更多信息请参阅编辑器的文档。
- 要直接设置消息内容，请使用以下服务：
 - `public void setContent(Integer transportId, Integer languageId, byte[] msgContent)` 此方法允许您使用 `msgContent` 参数直接设置消息的内容。必须有另外两个参数，但可以为空。`transportID` 参数允许您为特定的传送设置内容。`languageID` 参数允许您为指定的语言设置内容。要在设置多种语言的内容方面取得最大的灵活性，建议使用撰写方法运行 JSP。
 - `public void setContent(Integer transportId, Integer languageId, String msgContent)` 除了允许您以“字符串”格式而不是作为字节数组输入内容，此 `setContent` 方法执行的功能与上面所描述的相同。

添加电子邮件部件或附件服务

标准电子邮件传送，以及其它一些传送，允许对消息添加附件。以下方法允许消息传递系统的用户将内容部分或附件连接到消息：

- `public void addContentPart(byte[] msgAttachment)` 此方法提供一种将内容部件添加到消息的简单方式。对于电子邮件传送，“内容部件”指的是一个附件。`msgAttachment` 参数表示要添加到消息的内容。请注意这必须是部件的实际内容，该内容已转换为字节格式。
- `public void addContentPart(byte[] msgAttachment, String partName, String partType)` 此方法在向消息添加内容方面给予您更大的灵活性。`msgAttachment` 参数

表示要添加到消息的内容。partName 参数表示用于内容部件的名称。partType 参数表示正在发送的部件的 MIME 类型。例如，MIME Email 的 partType 可以是“text/plain”。

可配置消息数据服务

使用以下类属方法配置用于消息的传送服务：

- `public void setConfigData(String key, String value)` 此类属方法允许用户配置用于消息的传送服务。key 参数引用用于标识要更改的属性的管理名称。value 参数是要指定的值。调用此方法将导致在这里指定的值覆盖管理控制台中指定的值。请参阅下表，该表包含“消息传递系统”可用的缺省传送，以及适用于每个传送的属性。关于设置电子邮件收件人的可选方式，请参阅下面的 `addMember` 方法。

“消息传递系统”可用的缺省传送包含以下属性（属性键是区分大小写的）：

传送方法	属性键	描述
电子邮件	subject	电子邮件主题。
	recipient	收件人的电子邮件地址。
	sender	发件人的电子邮件地址。
	host	用于发送消息的邮件主机。
	协议	用于连接到邮件主机的协议。
文件	location	待编写文件的位置。
	FileName	待编写文件的名称。
	mode	待执行编写的类型。 0 — 附加，或创建（如果文件不存在） 1 — 覆盖

发送服务

以下方法提供了出站消息传递系统发送服务：

- `public void sendImmediate()`
此方法立即将消息发送至收件人。调用者将一直受阻拦至发送出消息。
- `public void sendTransacted()`
此方法将消息存储在 MSGSTORE 数据库表中。在预定时间里，WebSphere Commerce 调度程序将调作业，该作业将以批处理方式发送所有消息。使用此方法可确保只有在调用者已成功提交或终止后才会进行发送。如果无法容忍使用 `sendImmediate()` 方法时阻拦调用，应当使用此方法。
- `sendReceiveImmediate()`
此方法用于执行请求回复发送。这个发送类型是与后端集成消息的 MQ-JMS 传送一起使用的。应答的内容是内部存储的，可以通过 `getReply()` 方法访问。
提示： 要使用 MQ-JMS 传送执行发送和接收，您必须确保已经使用管理控制台或可配置消息数据服务中的 `setConfigData()` 方法正确的设置了方式属性。
- `public byte[] getReply()`
此方法用于检索 `sendReceiveImmediate()` 方法的结果。要获取结果，应在 `performExecute()` 方法执行命令之后调用该结果。它将来自传送的响应作为字节数组返回。

其它服务

以下方法概括了出站消息传递系统提供的其它服务:

- `public void addMember(Long aMember)` 此方法表示将收件人数据输入消息传递系统（第一种方式是 `setConfigData`）的第二种方式。其中的参数表示有效成员。用户可以重复调用此方法，每次调用都将一个附加成员添加到收件人的列表。消息传递系统在内部抽取“成员”的相应地址。
注意 1: 确保对成员地址执行确认是重要的。
注意 2: 可以使用 `addMember` 或 `setConfigData`（但不能同时使用）设置收件人。如果两项同时输入，`addMember` 条目将覆盖 `setConfigData` 条目。还要注意，消息传递系统需要用户在设置内容前输入收件人。这是因为成员信息可能改变在消息中使用的内容。
- `public void remTransport(Integer aRemoveTransportID)` 此方法在使用哪种传送交付此方法方面给予您灵活性。其中的参数应是对此消息类型有效的传送标识。当调用此方法时，消息传递系统就从可能使用的传送的消息列表中除去此传送。换句话说，作为参数传递的传送将因此特定消息的调用而禁用。
- `public void setPartialSend(Boolean partialSend)` 此方法仅在 `addMember` 方法用于设置收件人时适用。其中的布尔参数表示是否允许部分发送。在消息传递系统中，部分发送指只要添加的成员中有一个具有首选电子邮件地址（与他们的简要表关联）就发送消息。以下是 `partialSend` 参数的说明：
 - **true:** 允许部分发送。将跳过那些尚未在适当位置设置地址的成员。
 - **false:** 全部发送或不发送。即使成员中有一个没有适当配置的电子邮件，也返回一个异常。
- **注意:** 消息传递系统没有执行任何验证以确保地址的格式正确。部分发送根据适当位置中值存在的原则进行操作。

第 8 章 入站后端集成消息

入站消息是 WebSphere Commerce 从外部应用程序接收到的请求。每条入站消息可以激活 WebSphere Commerce 中的执行某特定功能的命令。如果在处理入站消息时发生错误，则此失败消息将放入错误队列。

WebSphere Commerce 支持完成以下五种功能的入站后端消息：

- 创建顾客注册
- 更新顾客注册
- 更新订单状态
- 更新产品库存
- 更新产品价格

以上列出的每种功能都可以由 XML 格式的请求消息激活，其中某些则可以使用旧的消息格式激活。通常我们推荐使用 XML 格式。XML 消息用 UTF-8 格式编码。

要创建顾客注册，请使用 Create_WCS_Customer XML 消息。如果已经将顾客信息记录在现有的后端系统上，而不是从暂存区重新创建此信息，则使用此信息将顾客数据注册到 WebSphere Commerce 数据库中。这些消息将现有的顾客信息从后端服务器发送到 WebSphere Commerce 服务器。也可以使用 Create_NC_Customer XML 消息和“顾客更新”旧的消息，来执行相似的功能。然而，Create_WCS_Customer XML 消息为您提供更新更多顾客信息的能力。

要更新已经存在于 WebSphere Commerce 数据库中的顾客注册，请使用 Update_WCS_Customer XML 消息。使用此消息时，后端顾客管理系统更新顾客信息，并将消息发送到 WebSphere Commerce 以更新关于已注册购物者的信息。也可以使用 Update_NC_Customer XML 消息和“顾客更新”旧的消息，来执行相似的功能。然而，Update_WCS_Customer XML 消息为您提供更新更多顾客信息的能力。

要更新已经存在于 WebSphere Commerce 数据库中的订单状态，请使用 Update_WCS_OrderStatus XML 消息。使用此消息更新 WebSphere Commerce 数据库中的订单状态，其中这些订单由后端应用程序处理。例如，通过将此信息从后端系统发送到 WebSphere Commerce，可以在 WebSphere Commerce 系统中更新来自后端实现系统的订单的装运状态。也可以使用 Update_NC_OrderStatus XML 消息或“订单状态更新”旧的消息，来执行相似的功能。然而，Update_WCS_OrderStatus XML 消息为您提供更新更多订单状态信息的能力。

要更新已经存在于 WebSphere Commerce 数据库中的产品库存，请使用 Update_WCS_ProductInventory XML 消息。当外部或后端库存系统维护产品库存时，此消息对于实例是理想的。外部系统可以将消息发送给 WebSphere Commerce，来更新 WebSphere Commerce 数据库中的库存。也可以使用“产品数量更新”旧的消息来执行相同的功能。同样地，Update_NC_ProductInventory XML 消息执行相似功能。然而，Update_WCS_ProductInventory XML 消息为您提供更新更多库存信息的能力。

要更新已经存在于 Commerce Suite 数据库中的标价或产品报价，请使用 Update_WCS_ProductPrice XML 消息。当外部或后端系统维护产品信息（包含价格）

时，此消息对于实例是理想的。外部系统可以将其中的一种消息发送给 WebSphere Commerce，以便更新 WebSphere Commerce 数据库中产品的标价或报价。如果要更新产品报价，也可以使用 Update_NC_ProductPrice XML 消息或“产品价格更新”旧的消息，来执行相同的功能。然而，只有 Update_WCS_ProductPrice 消息可以用来更新标价信息。

入站实现集成消息

入站实现集成消息是 WebSphere Commerce 从实现中心系统接收到的请求。每条入站消息可以激活 WebSphere Commerce 中的执行某特定功能的命令。如果在处理入站消息时发生错误，则此失败消息将放入错误队列。

WebSphere Commerce 支持完成以下功能的入站集成消息：

- 创建预期库存记录
- 创建提货批量
- 请求提货单和装货列表
- 创建接收
- 调整产品的库存接收级别，该产品由 WebSphere Commerce 数据库在 RECEIPT 表中维护
- 由实现中心发出装运确认

以上列出的每种功能都可以由 XML 格式的消息激活。XML 消息用 UTF-8 格式编码。

以下表概括了所使用的入站实现集成消息以及它们调用的控制器命令。

XML 消息	描述	控制器命令
Create_WCS_ExpectedInventoryRecord	创建预期库存记录。	ExpectedInventoryRecordCreateCmd
Create_WCS_PickBatch	生成提货批量。	PickBatchGenerateCmd
Inquire_WCS_PickPackListDetail	请求由 CreatePickBatch 创建的详细信息。	GetPickPackListDetailCmd
Create_WCS_InventoryReceipt	创建商品的库存记录。	ReceiptCreateCmd
Update_WCS_InventoryReceipt	调整商品的库存。	InventoryAdjustCmd
Create_WCS_ShipmentConfirmation	发出商品的装运确认。	ReleaseShipConfirmCmd

创建预期库存记录

要在 WebSphere Commerce 数据库中创建预期库存，请使用 Create_WCS_ExpectedInventoryRecord XML 消息。如果实现中心管理库存和供应商的订购信息，此消息对于实例是理想的。通知 WebSphere Commerce 能否提供将来存货，并且 WebSphere Commerce 可以跟踪库存级别。如果产品库存级别低，实现中心可以发送这些消息到 WebSphere Commerce 以在 WebSphere Commerce 数据库中创建预期库存记录。此记录可以用于延迟交货。

创建提货批量

要在 WebSphere Commerce 数据库中创建提货批量，请使用 Create_WCS_PickBatch XML 消息。如果实现中心管理提货批量函数，此消息对于实例是理想的。实现中心可以发送其中一条消息到 WebSphere Commerce 以在 WebSphere Commerce 数据库中创建提货批量。提货批量将所有未付清的订单组合在一起，这些订单准备好发货到给定的实现中心和商店。提货批量中所有发货订单的提货单在 PICKBATCH 表中创建并存储。每个发货订单的装货列表在 ORDRELEASE 表中创建并存储。

获取提货装货列表的详细信息

要查询提货单的详细信息，请使用 Inquire_WCS_PickPackListDetail XML 消息。此消息与 Create_WCS_PickBatch 消息一起使用。实现中心发送 Create_WCS_PickBatch 消息到 WebSphere Commerce 之后，将创建提货批量，并返回新的提货批量标识到实现中心。实现中心则可以发送以提货批量标识作为参数 Inquire_WCS_PickPackListDetail 消息，以检索提货单和装货列表的详细信息。

创建库存接收

要创建已订购产品的接收，请使用 Create_WCS_InventoryReceipt XML 消息。此消息允许您创建供应商已订购产品的接收，帮助在 WebSphere WebSphere Commerce 库存数据库中更新当前可用的库存（在 RECEIPT 数据库表中）。实现中心可用发送其中一条消息到 WebSphere Commerce 以创建可用于跟踪已订购产品的接收。

更新库存级别

要解决物理库存计数和 WebSphere Commerce 中维护的库存级别间的任何差异，请使用 Update_WCS_InventoryReceipt XML 消息。当外部的实现中心系统管理库存装运时，将使用此消息。实现中心系统可以发送此消息到 WebSphere Commerce 以调节产品库存级别。

发出装运确认

要发出装运确认到 WebSphere Commerce 请使用 Create_WCS_ShipmentConfirmation XML 消息。实现中心可以发送其中一条消息到 WebSphere Commerce 以创建验证订单是否已装运的装运确认信息。此消息也给您在订单已装运时发送电子邮件通知给顾客的选项。关于启用顾客电子邮件通知的更多详细信息，请参阅启用装运通知电子邮件。

第 9 章 添加新的入站 XML 消息

要添加对新入站消息的支持，以下步骤是必需的：

1. 定义新 XML 消息的 DTD 文件。可使用现有的 XML 消息的 DTD 文件作为指南。缺省情况下，所有这些文件都位于以下目录中：

▶ 2000 *drive:\Program Files\WebSphere\CommerceServer\xml\messaging*

▶ NT *drive:\WebSphere\CommerceServer\xml\messaging*

▶ AIX */usr/WebSphere/CommerceServer/xml/messaging*

▶ Solaris */opt/WebSphere/CommerceServer/xml/messaging*

▶ 400 */QIBM/ProdData/WebCommerce/xml/messaging*

2. 添加新的 DTD 文件至系统。
3. 更新新消息的入站消息模板定义文件 `user_template.xml`。要完成此操作请参阅入站消息模板定义文件中概括的结构准则。

添加新的 DTD 文件至系统

要使新入站 XML 消息能被 XML 消息映射器识别和处理，请执行以下操作：

1. 将为新消息创建的 DTD 文件放置在与其它 DTD 文件相同的目录中。缺省情况下，此目录是：

▶ 2000 *drive:\Program Files\WebSphere\CommerceServer\xml\messaging*

▶ NT *drive:\WebSphere\CommerceServer\xml\messaging*

▶ AIX */usr/WebSphere/CommerceServer/xml/messaging*

▶ Solaris */opt/WebSphere/CommerceServer/xml/messaging*

▶ 400 */QIBM/ProdData/WebCommerce/xml/messaging*

2. 添加新 DTD 文件的名称至入站消息 DTD 文件列表中。

添加至入站消息 DTD 文件列表

要添加新的 DTD 文件至入站消息 DTD 文件列表，请执行以下操作：

1. 启动配置管理器。
2. 选择实例属性，然后打开消息传递文件夹。
3. 在“入站消息 DTD 文件”字段中，将新的 DTD 文件的名称添加至列表的结尾，并在新文件名的前面放置一个逗号。
4. 单击应用保存更改。
5. 在 WebSphere Application Server 管理控制台上，停止然后重新启动此实例。

入站消息扩展

WebSphere Commerce 允许您通过修改由每个消息运行的 WebSphere Commerce 控制器命令来修改或扩展所有入站消息的功能。您可以为所使用的任何入站消息命令提供附加的预处理或后处理语句，或者您可以完全覆盖现有的处理。若要如此，需要有关 Java 编程的一定知识。

当从后端系统接收到一条入站消息时，它的信息就处理为命令参数，并且 WebSphere Commerce 控制器命令与所有已提供的参数一起调用。运行此命令时，就调用 performExecute() 方法，而该方法又按以下次序调用三个方法：

1. doPreProcess()
2. doProcess()
3. doPostProcess()

第一次安装 WebSphere Commerce 时，只有 doProcess() 方法包含编程语句。您可以通过扩展命令和实现 doPreProcess() 方法来添加预处理语句，或者可以通过实现 doPostProcess() 方法添加后处理语句。同样的，可以实现 doProcess() 或 performExecute() 方法以覆盖整个处理。有关如何扩展命令的更多信息，请参阅《WebSphere Commerce 程序员指南》。

入站消息的 UserData 元素

所有入站消息都包含 UserData XML 元素，将其作为可选元素。包含要传送附加数据的 UserDataField 元素。可以定制字段，传送未包括在这些消息中的额外数据。

要添加的新字段名称应是 UserDataField 元素的属性名。

以下的 DTD 描述了 UserData 元素：

```
<!ELEMENT UserData (UserDataField+)>
<!ELEMENT UserDataField (#PCDATA)>
<!ATTLIST UserDataField
name CDATA #REQUIRED
>
```

以下是 UserData 元素的示例：

```
<UserData>
  <UserDataField name="field_name">field_value</UserDataField>
  .
  .UserDataField repeated loop
  .
</UserData>
```

第 10 章 消息映射器

消息映射器是获取 XML 消息并将其转换为 `CommandProperty` 对象的机制。它提供公共接口，以便可以将消息转换为 `CommandProperty` 对象并由所有的 WebSphere Commerce 组件使用。

支持的适配器（例如程序适配器）和 WebSphere Commerce 组件都可以调用消息映射器。两种情况下，消息映射器执行以下任务：

- 接收 XML 消息。
- 将消息转换为 `CommandProperty` 对象。
- 如果无法转换 XML 消息，则返回 `null`。

入站消息由 WebSphere Commerce 或外部系统发送到 WebSphere Commerce，以请求某类操作。WebSphere Commerce 要执行此操作，XML 消息必须由消息映射器处理以确定已请求什么操作。WebSphere Commerce 在安装时包含两种消息映射器：用于分析 XML 集成消息的 XML 消息映射器和用于分析后端集成旧的的消息的旧的消息映射器。

`CommandProperty` 对象表示 WebSphere Commerce 命令以满足支持的设备适配器的需求。其它组件也可以使用此消息映射器将消息转换为 `CommandProperty` 对象。

消息映射器的有效期在 WebSphere Commerce 实例中始终存在。它在实例启动时初始化，并在实例运行时驻留。

XML 消息映射器

XML 消息映射器负责将入站 XML 消息的 XML 数据转换为 `CommandProperty` 对象。它是 ECSAX 分析程序的扩展。在 `instance_name.xml` 配置文件找到的 XML 消息映射器的 XML 配置节点应当与以下类似：

```
<MessageMapper messageMapperId="-1"
  classname="com.ibm.commerce.messaging.programadapter.messageadapter.ecsax.ECSAXMessageMapper"
  enable="true"
  name="WCS.INTEGRATION">
</configuration/>
</MessageMapper>
```

旧消息映射器

旧消息映射器负责将数据从入站旧的消息转换为 `CommandProperty` 对象。在 `instance_name.xml` 配置文件找到的旧消息映射器的 XML 配置节点应当与以下类似：

```
<MessageMapper messageMapperId="-2"
  classname="com.ibm.commerce.messaging.programadapter.messageadapter.nclegacy.NetCMessageMapper"
  enable="true">
```

```
        name="NC.LEGACY">
<configuration>
</MessageMapper>
```

入站消息模板定义文件

WebSphere Commerce 提供 XML 消息映射器，它可以用于根据入站 XML 消息模板定义文件，将入站 XML 消息映射至 WebSphere Commerce 命令接口。

无论入站 XML 消息何时发送到消息映射器，它都会将检查消息是否在模板定义文件中进行了定义。如果定义了，则它将检索这条消息的 WebSphere Commerce 控制器命令名和参数名，且分析此进入消息来获取参数值。消息分析之后，消息映射器会返回 CommandProperty 对象，此对象包含该命令的命令名和参数“名称值”对应的对象。

消息模板定义文件用于定义入站 XML 消息的 XML 分析信息。这些文件中定义的每条消息都具有以下两个基本元素：

- **TemplateDocument:** 定义消息所用的 DTD 文件、在接收到消息时调用的命令、要使用哪个标记映射，以及标记映射起始的 XML 元素。
- **TemplateTag:** 定义 DTD 文件中的 XML 元素到 WebSphere Commerce 中的命令的参数名的映射。模板标记元素对参数名进行定义，并告诉消息映射器在何处寻找进入消息的值。

WebSphere Commerce 提供了两个模板定义文件。sys_template.xml 文件是用于将现有 WebSphere Commerce 入站 XML 消息进行映射的模板定义。user_template.xml 用于添加附加 XML 消息。根据定义 DTD 文件中的 ec_template.dtd 模板定义，这两种文件都使用 XML 格式。

NT	drive:\WebSphere\CommerceServer\xml\messaging
2000	drive:\Program Files\WebSphere\CommerceServer\xml\messaging
AIX	/usr/WebSphere/CommerceServer/xml/messaging
Solaris	/opt/WebSphere/CommerceServer/xml/messaging
400	/QIBM/Proddata/WebCommerce/xml/messaging

重要信息: 为安全起见，必须确保只有已授权的人员可以访问和修改 user_template.xml 和 user_template.xml 消息模板定义文件。如果未授权用户也有写入此文件的访问权限，则他们也将能编写新的入站消息，并像站点管理员那样调用任何 WebSphere Commerce 命令。

除去消息映射器

要除去消息映射器，必须在 instance_name.xml 配置文件中手工将它从消息映射器组中除去。要除去消息映射器，请执行以下操作：

1. 打开 instance_name.xml 配置文件。
2. 定位名称为 MessageMapperGroup 的组件。
3. 定位希望除去的消息映射器所用的 XML 配置节点。它看似如下：

```
<MessageMapper messageId="#"
  classname="class
  implementing MessageMapper interface"
  enable="true"
  name="Name of Message Mapper">
  <configuration />
</MessageMapper>
```

4. 将 `enable` 参数更改为 `"false"`。这将禁用消息映射器，并使它不可用。

添加消息映射器

要添加新的消息映射器，必须在 `instance_name.xml` 配置文件中手工将它添加到消息映射器组。要添加新的消息映射器，请执行以下操作：

1. 打开 `instance_name.xml` 配置文件。
2. 定位名称为 `MessageMapperGroup` 的组件。
3. 在该组件的属性标记之间，添加以下 XML 节点以定义消息映射器：

```
<MessageMapper messageId="#"
  classname="class
  implementing MessageMapper interface"
  enable="true"
  name="Name of Message Mapper"> </MessageMapper>
```

关于这些参数的信息，请参阅消息映射器配置。

4. 在消息映射器的配置节点中，添加该消息映射器所需要的附加配置参数。这转换成 `TypedProperty` 对象，并传送到消息映射器的初始方法。以下是可能添加的附加参数的一个示例：

```
<configuration
EcSystemTemplateFile="mapping.xml"
EcInboundMessageDtdFiles="something.dtd"
EcTemplatePath="E:\users\user\test\map"
EcSaxParserClass="com.ibm.xml.parsers.ValidatingSAXParser"
EcInboundMessageDtdPath="E:\users\user\test\dtd"
EcSaxParserClass="com.ibm.xml.parsers.ValidatingSAXParser"
/>
```

新入站消息支持

除了支持的 XML 和 WebSphere Commerce 消息以外，您还可以对新的入站消息添加支持。有两种主要的添加新入站消息的方法。

建议您采用通过使用 `user_template.xml` 入站消息模板定义文件添加新入站 XML 消息的方法。在此文件中，可以指示新的入站消息调用的控制器命令，定义消息元素，并指示每个元素所对应的命令参数。接收消息时，XML 消息映射器指出要运行的命令和要使用的参数。然后使用站点管理员权限调用此命令。为安全起见，您必须确保只有获得授权的人员才可以访问和修改 `user_template.xml` 消息模板定义文件，否则未授权用户也将能编写新的入站消息并像站点管理员那样调用任何 WebSphere Commerce 命令。

如果您不想将入站 XML 消息模板定义文件与 XML 消息映射器一起使用，则您还可以实现 `NewInboundMessage` 命令来添加新消息。当消息映射器不能将消息识别为现有的旧消息或在入站 XML 消息模板定义文件中定义的 XML 消息时，将调用此命令。由于 `NewInboundMessage` 命令不是预编程的，所以一旦调用它，您将对发生的处理过程具有完全控制权。然而，此方法需要相当可观的编程工作量，特别是有很多新消息时。

第 11 章 定制 NewInboundMessage 命令

要定制 NewInboundMessage 命令以处理已创建的消息，请执行以下操作：

1. 扩展 NewInboundMessage 命令。关于如何完成此操作的详细信息，请参阅《*WebSphere Commerce 程序员指南*》。
2. 要接收入站消息，请使用此命令的 getMessage() 方法，它以字符串返回消息。
3. 实现此命令的 performExecute() 方法。在此方法中，放置处理入站消息的编程语句。
4. 通过更新 WebSphere Commerce 数据库中的 CMDREG 表，使用以下 SQL 语句注册新的扩展命令：

```
update cmdreg set classname='yourCommandClassName' where  
interfacename='com.ibm.commerce.messaging.commands.NewInboundMessageCmd'
```

第 12 章 消息映射器配置

instance_name.xml 配置文件列出了所有可能的消息映射器，并且包含了它们的名称、类、设备格式、是否可用以及特定配置参数。要查找某些消息映射器的参数，请查找 *instance_name.xml* 文件中名为 `MessageMapperGroup` 的组件。由于每个消息映射器可能含有不同的配置参数，每个映射器的 `MessageMapperGroup` 组件中含有一个 XML 配置信息的节点。配置参数节点格式的唯一请求是必须为“名称值”对。这就允许消息映射器配置参数比较容易地转换为 `TypeProperty` 对象。

下面列出了每个独立消息映射器的参数。

- `name`: 消息映射器的名称。
- `class`: 包含消息映射器接口实现的类。
- `messageMapperId`: 消息映射器的标识。每个消息映射器标识必须唯一。
- `enabled`: 指示是否应该使用并初始化消息映射器。如果此值设置为“true”，则消息映射器在实例启动时初始化并且可用。如果此值设置为“false”，则禁用消息映射器。

使用模板定义文件进行 XML 分析

当 XML 消息映射器分析入站 XML 消息时，它将从消息中的 XML 文档依次获取文档类型、版本（如果可用）和元素名称。分析映射器将查找模板定义文件的 `ECTemplate` 元素中定义的模板文档。模板文档向消息映射器提供以下信息：

- 应当从中启动标记映射的元素。
- 要使用的标记模板。
- 入站消息要调用的命令名称。此命令可能是现有的 `WebSphere Commerce` 控制器命令或您创建的新命令。

一旦启动标记映射，消息映射器就要查看 `TemplateTag`（它已在 `ECTemplate` 文件中定义以根据入站 XML 消息生成的 XPath 来确定字段名称和类型），然后设置该字段的值。字段和值对存储在 `commandProperty` 或 `messageProperty` 的 `TypedProperty` 散列表中，而这取决于该字段的 `FieldInfo` 属性中的值。在对整个入站消息分析成功之后，就返回一个 `PropertyCommand` 对象，该对象包含命令名称以及 `commandProperty` 和 `messageProperty` 对象。

sys_template.xml 文件

sys_template.xml 文件包含 `WebSphere Commerce` 支持的所有入站 XML 消息的概括。此文件定义了每个消息的数据字段，将消息映射至相应的 `WebSphere Commerce` 控制器命令，并将消息中的每个字段映射至该命令的相应参数。*sys_template.xml* 的结构基于 *ec_template.dtd* 文件，该文件定义了消息必须采用的格式。

不要向此文件添加新消息。要添加您自己的入站消息，请使用 *user_template.dtd* 文件。

所有 XML 文件都在以下目录中：

▶ 2000	<i>drive:\Program Files\WebSphere\CommerceServer\xml\messaging</i>
▶ NT	<i>drive:\WebSphere\CommerceServer\xml\messaging</i>
▶ AIX	<i>/usr/WebSphere/CommerceServer/xml/messaging</i>
▶ Solaris	<i>/opt/WebSphere/CommerceServer/xml/messaging</i>
▶ 400	<i>/QIBM/Proddata/WebCommerce/xml/messaging</i>

user_template.xml 文件

user_template.xml 是 XML 消息模板定义文件，它允许添加系统支持的新入站 XML 消息。应在您想要支持的每条新 XML 消息的这种文件中添加摘要。可以使用 sys_template.xml 文件作为如何使用或更新此模板文件的指南。

此概括应当指示要使用的标记模板、标记映射启动的起始元素、要调用的 WebSphere Commerce 控制器命令名称以及对应每个 XML 元素的 URL 元素。如果您正在使用新的入站 XML 消息来调用新的 WebSphere Commerce 命令，请参阅《WebSphere Commerce 程序员指南》以了解如何写和注册新的 WebSphere Commerce 控制器命令。

所有 XML 文件都在以下目录中：

▶ 2000	<i>drive:\Program Files\WebSphere\CommerceServer\xml\messaging</i>
▶ NT	<i>drive:\WebSphere\CommerceServer\xml\messaging</i>
▶ AIX	<i>/usr/WebSphere/CommerceServer/xml/messaging</i>
▶ Solaris	<i>/opt/WebSphere/CommerceServer/xml/messaging</i>
▶ 400	<i>/QIBM/Proddata/WebCommerce/xml/messaging</i>

在添加自己的任何消息之前，请确保此文件包含以下行：

```
<?xml version='1.0' encoding='UTF-8'?>
<!DOCTYPE ECTemplate SYSTEM 'ec_template.dtd' >
<ECTemplate>
</ECTemplate>
```

在模板定义文件中定义的每个消息包括两个基本元素：

- TemplateDocument
- TemplateTag

关于这些元素中每个元素的说明，请按照下面的链接进行查看。

ec_template.dtd 文件

ec_template.dtd 文件包含 sys_template.xml 文件中概括的所有入站消息的蓝图，它应当用于指导您在 user_template.xml 文件中创建任何新消息。

所有的 DTD 文件都位于以下目录中：

▶ 2000	<i>drive:\Program Files\WebSphere\CommerceServer\xml\messaging</i>
▶ NT	<i>drive:\WebSphere\CommerceServer\xml\messaging</i>

```

▶ AIX /usr/WebSphere/CommerceServer/xml/messaging
▶ Solaris /opt/WebSphere/CommerceServer/xml/messaging
▶ 400 /QIBM/Proddata/WebCommerce/xml/messaging

```

以下文本描述 ec_template.dtd 文件:

```

<!ELEMENT ECTemplate ((TemplateDocument | TemplateTag)*)> <!ATTLIST ECTemplate
version CDATA #FIXED "1.0"> <!ELEMENT TemplateDocument (DocumentType,
StartElement, TemplateTagName, CommandMapping?)> <!ELEMENT DocumentType
(#PCDATA)> <!ATTLIST DocumentType version CDATA #IMPLIED > <!ELEMENT
StartElement (#PCDATA)> <!ELEMENT TemplateTagName (#PCDATA)> <!ELEMENT
CommandMapping (Command+)> <!ELEMENT Command (Constant*) > <!ATTLIST Command
CommandName CDATA #REQUIRED Condition CDATA #IMPLIED TemplateTagName CDATA
#IMPLIED> <!ELEMENT Constant (#PCDATA)> <!ATTLIST Constant Field CDATA #REQUIRED
FieldInfo CDATA #IMPLIED > <!ELEMENT TemplateTag (Tag+)> <!ATTLIST TemplateTag
name CDATA #REQUIRED > <!ELEMENT Tag EMPTY> <!ATTLIST Tag XPath CDATA #REQUIRED
XPathType (PCDATA | EMPTY | REPEAT | VECTOR | ATTRIBUTE | USERDATA) "PCDATA"
Field CDATA #IMPLIED FieldType ( STRING | DATE ) "STRING" FieldInfo CDATA
#IMPLIED >

```

模板定义文件的 **TemplateDocument** 元素

对于每个人站 XML 消息，在消息中至少定义一个模板文档。这是定义在 TemplateDocument 元素中的。TemplateDocument 元素有四个子元素：

- **DocumentType** (带有可选的 **version** 属性)：这一部分指定了 XML 文档名称或根元素名称以及根元素的“version”属性（如果有根元素的话）。以下是 DocumentType 元素的一个示例：

```
<DocumentType version='1.0'>Reset_Password</DocumentType>
```

- **StartElement**：这一部分指定 XML 消息映射器开始标记映射的元素。这必须与该消息的 DTD 中的元素对应。标记映射生成名称值对，该值对作为命令参数使用。以下是 StartElement 用法的一个示例：

```
<StartElement>PasswordInfo</StartElement>
```

在此例中，该消息的 DTD 文件中应定义一个名为 PasswordInfo 的元素。例如，DTD 文件应包含定义 PasswordInfo 元素的行，例如以下行：

```
<!ELEMENT PasswordInfo (Password, ConfirmPassword)>
```

- **TemplateTagName**：这一部分指定将用于该消息的标记映射。标记映射是在 TemplateTag 元素下的模板定义文件中的别处定义的。这意味着 TemplateTagName 元素的值应当匹配模板定义文件中某处出现的 TemplateTag 元素的 name 属性。更多详细信息请参阅关于 TemplateTag 元素的部分。以下示例说明了 TemplateTagName 元素的使用：

```
<TemplateTagName>PasswordReset10Map</TemplateTagName>
```

在此例中，模板定义文件中的别处应有以下 TemplateTag 元素：

```
<TemplateTag name='PasswordReset10Map'> ... </TemplateTag>
```

在开始和结束标记之间的元素取决于该消息的映射。

- **CommandMapping**：此元素确定该消息调用哪个命令。该元素包含一个称为 Command 的子元素。Command 元素用于指示 WebSphere Commerce 控制器命令，当此消息映射成功时将运行该命令。它包含一个称为 CommandName 的强制属性，此属性用于指示

命令的名称。此属性的值必须与在 URLREG 表中已注册的现有命令对应。例如，您可以使用以下语法将消息与重新设置密码的命令关联：

```
<CommandMapping> <Command CommandName='ResetPassword' /> </CommandMapping>
```

您还可以将多个命令与同一消息关联。若要如此，需要使用 Command 元素的以下附加属性：

- **Condition 属性：**条件的语法如下（[] 表示可选，* 表示可以重复）：

```
fieldName ["fieldName "] [AND fieldName ["fieldName "]]*
```

fieldName

它应与 TemplateTag 定义中 Tag 元素的 Field 属性匹配。如果此标记的 XPath 属性在入站 XML 消息中，则该条件为真。

fieldName="fieldValue "

将 XML 消息中的一个值设置为 fieldName 中的字段。映射消息时，如果该值与 fieldValue 相同，该条件为真。

fieldName1="fieldValue1" AND fieldName1="fieldValue2"

分析此消息时，XML 消息中的值设置为字段 fieldName1 和 fieldName2。

如果这些值分别与 fieldValue1 和 fieldValue2 相同，则条件为真。

- **TemplateName 属性：**如果您指定此 Command 元素的 TemplateTagName 属性，无论何时 Condition 为真，在 TemplateTagName 中定义其名称的新标记模板都将用于入站 XML 消息的剩余部分。

- **Constant 元素：**将为该命令而放置到 TypedProperty 中的常量的列表。

- **Field：**将置入 TypedProperty 中的名称值对的字段名。

- **FieldInfo：**请参阅 TemplateTag 中的定义。

此值应当放置在 <Constant> 和 </Constant> 标记之间。

以下是如何使用多个 Command 元素以映射到多个命令的简单示例：

```
<CommandMapping>
```

```
  <Command CommandName='ResetPassword' Condition='Verb="Reset" AND Noun="Password"'/>
```

```
  <Command CommandName='AdminResetPassword' Condition='Verb="Reset" AND Noun="AdminPassword"'/>
```

```
</CommandMapping>
```

在此例中，您应当是在该消息的 DTD 文件中定义的 Noun 和 Verb 元素。

模板定义文件的 TemplateTag 元素

TemplateTag 元素与模板定义文件中的 TemplateDocument 元素一起使用。对于每个入站 XML 消息可以定义几个 TemplateTag 元素。然而，每个 TemplateTag 元素都必须使用 name 属性链接到一个 TemplateDocument 元素。与 name 属性关联的值应与 TemplateDocument 元素（已用模板定义文件定义）的 TemplateName 子元素中的值匹配。

每个 TemplateTag 元素包含 Tag 子元素中的标记定义的列表。Tag 元素可以包含以下五个属性：

- **XPath：**XML 元素的路径，它相对于 TemplateDocument 中指示的 StartElement。

XPath 元素是查找“名称值”对的字段名的键。XPath 的示例是：

- E1/E2: 元素 E2, 已在元素 E1 内嵌套。以下是此类型路径的示例:
XPath='Address/ZipCode'
- E1/E2/E3: 元素 E3, 已在元素 E2 (已在元素 E1 内嵌套。)内嵌套。以下是此类型路径的示例:
XPath='ContactInfo/Address/ZipCode'
- E3@a1: 元素 E3 中的属性 a1。以下是此类型路径的示例:
XPath='InvoiceInfo@InvoiceType'
在此情况下, InvoiceType 属性可用于指示此消息适用于各种发票类型之中的哪一类型。
- E4[1]: 元素 E4 的第一个实例。可能有 E4 的多个实例。例如, 在您的地址有多行处就可以使用此格式。地址的每一行都将包含在独立的 Tag 元素中。
- E5[@a2="value 2"]: 元素 E5 的一个实例, 其中 E5 的属性 a2 等于 "value 2"。例如, 您可以使用以下格式:
XPath='InvoiceInfo[@InvoiceType="ShippingInvoice"]
E5[@a1="value1"][@a2="value2"]: 元素 E5 的一个实例, 其中 E5 的属性 a1 等于 "value1", 而 E5 的属性 a2 等于 "value 2"。
- **XPathType:** XPath 中指示的元素的类型。此属性指示 XML 分析程序如何处理 XML 元素。受支持的类型有:
 - **PCDATA:** 元素或元素的属性包含原始的人站数据, 该数据将在“名称值”对的格式中进行处理并返回。此为 XPathType 的缺省值。
 - **EMPTY:** 元素为空或包含可以被忽略的数据。对此元素不返回任何“名称值”对。
 - **REPEAT:** 元素可以有多个实例。每个元素的 PCDATA 都以“名称值”对返回。
 - **ATTRIBUTE:** 如果元素字段名由属性的值确定, 此属性就应当有 ATTRIBUTE 类型。
 - **VECTOR:** 将向 Vector 附加一个新的散列表, 所有为子元素生成的“名称值”对都应放入新的散列表。
 - **USERDATA:** 指示用户定义的元素。此元素有称为 name 的属性, 该属性的值是字段名。元素的数据以“名称值”对返回。
- **Field:** 将置入 TypedProperty 的“名称值”对的字段名。这应与所调用命令使用的参数名称匹配。
- **FieldType:** 数据字段的类型。字段类型可以是 String 或 Date (ISO 8601 日期格式)。String 是缺省值。
- **FieldInfo:** 指示 TypedProperty (名称值对应当置入它)。Data 是缺省值。如果您想将名称值对置入一个以上 TypedProperty, 必须指定下面所列值 (由逗号分隔) 中的一个以上值:
 - **Data:** “名称值”对置入包含命令变量的 commandProperty 对象。
 - **Control:** “名称值”对将置入 messageProperty (它包含命令的控制信息, 例如 USERID 或 PASSWORD)。
 - **Command:** “名称值”对用于确定应调用哪个命令。生成的“名称值”对在 TemplateDocument 元素的 CommandMapping 元素中使用。

如果在标记模板中没有找到元素 XPath, XML 消息中的 XPath 将作为字段名使用, 而且 XPathType 为 PCDATA, 生成一个将 XPath 作为字段名使用的“名称值”对。

有关如何使用 `TemplateTag` 元素的示例，请参阅 `sys_template.xml` 文件。

第 13 章 消息传递系统后端集成消息

WebSphere Commerce 消息传递系统提供了通过使用入站和出站消息将 WebSphere Commerce 与后端系统集成的机制。入站消息用于根据来自后端系统的消息在 WebSphere Commerce 中运行命令。出站消息可以由出站消息传递系统生成，以使用发生的事件（例如新顾客订单）更新后端系统。要使用后端集成消息，您必须已安装了适配器并且已配置了消息传递系统以接收 XML 消息。

消息传递系统可以发送和接收许多 XML 格式的预定义的消息。这种格式提供了高度的可读性，使得修改和维护变得简单易行。您也可以使用旧的消息格式。然而我们推荐使用 XML 消息格式。关于每种消息的说明，请参阅关于入站和出站后端集成消息的部分。您也可以添加新消息。对于新的入站消息，您既可以将它们与现有 WebSphere Commerce 命令相关联，也可与您创建的命令相关联。

第 14 章 实现集成消息

WebSphere Commerce 提供一个使用入站和出站消息以和实现中心系统集成的机制。入站实现集成消息用于在 WebSphere Commerce 中根据从实现中心系统接收到的入站请求运行命令。出站消息可以由出站系统生成，以便用发生的事件更新实现中心系统，例如接收新存货或订单装运。要使用实现集成消息，您必须已安装了适配器并且已配置了消息传递系统以接收 XML 消息。

消息传递系统可以发送和接收许多 XML 格式的预定义的消息。这种格式提供了高度的可读性，使得修改和维护变得简单易行。关于每种消息的说明，请参阅关于入站和出站实现集成消息的部分。您也可以添加新消息。对于新的入站消息，您既可以将它们与现有 WebSphere Commerce 命令相关联，也可与您创建的命令相关联。

XML 消息的格式由在特定的 DTD 文件内定义的 XML 元素集合构成。每个 DTD 可能包含一个或多个公用文件，由文件扩展名 .mod 标识。另外，每条入站消息与 sys_template.xml 消息模板定义文件中的一条 WebSphere Commerce 控制器命令关联。所有 DTD、MOD 和 XML 文件都在以下目录中：

▶ 2000	<i>drive:\Program Files\WebSphere\CommerceServer\xml\messaging</i>
▶ NT	<i>drive:\WebSphere\CommerceServer\xml\messaging</i>
▶ AIX	<i>/usr/WebSphere/CommerceServer/xml/messaging</i>
▶ Solaris	<i>/opt/WebSphere/CommerceServer/xml/messaging</i>
▶ 400	<i>QIBM/ProdData/WebCommerce/xml/messaging</i>

第 15 章 定制 NewInboundMessage 命令

要定制 NewInboundMessage 命令以处理已创建的消息，请执行以下操作：

1. 扩展 NewInboundMessage 命令。关于如何完成此操作的详细信息，请参阅《*WebSphere Commerce 程序员指南*》。
2. 要接收入站消息，请使用此命令的 getMessage() 方法，它以字符串返回消息。
3. 实现此命令的 performExecute() 方法。在此方法中，放置处理入站消息的编程语句。
4. 通过更新 WebSphere Commerce 数据库中的 CMDREG 表，使用以下 SQL 语句注册新的扩展命令：

```
update cmdreg set classname='yourCommandClassName' where  
interfacename='com.ibm.commerce.messaging.commands.NewInboundMessageCmd'
```


第 16 章 集成消息 DTD 文件

所有支持的 WebSphere Commerce 集成 XML 消息由位于 DTD 文件中的信息构成。一些 DTD 文件使用来自公用文件 NCCCommon.mod 或其它 MOD 文件的信息。在这些 MOD 文件中描述了 DTD 文件的 XML 元素值的格式和源。

所有的 DTD 和 MOD 文件都位于以下目录:

- ▶ 2000 *drive:\Program Files\WebSphere\CommerceServer\xml\messaging*
- ▶ NT *drive:\WebSphere\CommerceServer\xml\messaging*
- ▶ AIX */usr/WebSphere/CommerceServer/xml/messaging*
- ▶ Solaris */opt/WebSphere/CommerceServer/xml/messaging*
- ▶ 400 */QIBM/Proddata/WebCommerce/xml/messaging*

消息	使用的 DTD 和 MOD 文件
Create_NC_Customer 消息	Create_NC_Customer_10.dtd NCCCommon.mod NCCustomer_10.mod
Update_NC_OrderStatus 消息	Update_NC_OrderStatus_10.dtd NCCCommon.mod
Update_NC_ProductInventory 消息	Update_NC_ProductInventory_10.dtd NCCCommon.mod
Update_NC_ProductPrice 消息	Update_NC_ProductPrice_10.dtd NCCCommon.mod
Report_NC_PurchaseOrder 消息	Report_NC_PO_10.dtd
Create_WCS_Customer 消息	Create_WCS_Customer_20.dtd NCCCommon.mod
Update_WCS_ProductPrice 消息	Update_WCS_ProductPrice_20.dtd NCCCommon.mod
Update_WCS_ProductInventory 消息	Update_WCS_ProductInventory_20.dtd NCCCommon.mod.
Update_WCS_Customer 消息	Update_WCS_Customer_10.dtd NCCCommon.mod NCCustomer_10.mod
Update_WCS_OrderStatus 消息	Update_WCS_OrderStatus_20.dtd NCCCommon.mod
Create_WCS_ExpectedInventoryCreate 消息	Create_WCS_ExpectedInventoryRecord_10.dtd
Create_WCS_PickBatch 消息	Create_WCS_PickBatch_10.dtd
Inquire_WCS_PickPackListDetail 消息	Inquire_WCS_PickPackListDetail_10.dtd
Create_WCS_InventoryReceipt 消息	Create_WCS_InventoryReceipt_10.dtd
Update_WCS_InventoryReceipt 消息	Update_WCS_InventoryReceipt_10.dtd

Create_WCS_ShipmentConfirmation 消息	Create_WCS_ShipmentConfirmation_10.dtd
Response_WCS_ExpectedInventoryRecord 消息	Response_WCS_ExpectedInventoryRecord_10.dtd
Response_WCS_PickBatch 消息	Response_WCS_PickBatch_10.dtd
Report_WCS_PickPackListDetail 消息	Report_WCS_PickPackListDetail_10.dtd
Response_WCS_CreateInvReceipt 消息	Response_WCS_CreateInvReceipt_10.dtd
Response_WCS_UpdateInvReceipt 消息	Response_WCS_UpdateInvReceipt_10.dtd
Response_WCS_CreateShipConfirm 消息	Response_WCS_CreateShipConfirm_10.dtd

后端集成旧的消息

WebSphere Commerce 提供了对使用旧消息格式的消息的支持。除非您正从先前版本的 WebSphere Commerce Suite 迁移，否则建议您使用 XML 消息，因为它们可实现相同的功能，且更易于阅读和维护。

WebSphere Commerce 消息的格式由两部分组成：消息描述符和应用程序数据。就入站消息而言，消息描述符包含进行操作所需的控制信息，例如消息身份和类型。应用程序数据包含将要处理的信息。所有的 WebSphere Commerce 消息都由按逻辑排列的一组标记和记录以及那些记录中定义的数据段组成。受支持消息中所包括的 <PROLOG>、<HDR>、<ITM> 和 <DATUSR> 记录遵循以下格式：

```
<TAG>DATA SEGMENT</TAG>
```

其中，数据段用 _DATA 后缀标识。例如，“订单创建”消息的记录可能如下：

```
<HDR010>ORDER_CREATE_HDR010_DATA</HDR010>
```

必须用特定消息的特定字段和数据库表信息替换每个数据段（此示例中为 ORDER_CREATE_HDR010_DATA）。

下表概括了六个受支持的消息及每个消息调用的控制器命令：

消息名	消息类型	数据段	控制器命令
顾客新建	入站	CUSTOMER_NEW_PROLOG_DATA CUSTOMER_NEW_HDR010_DATA 入站消息的 DATUSR_DATA	UserRegistration Add
顾客更新	入站	CUSTOMER_UPDATE_PROLOG_DATA CUSTOMER_UPDATE_HDR010_DATA 入站消息的 DATUSR_DATA	UserRegistration Update
订单创建	出站	ORDER_CREATE_PROLOG_DATA ORDER_CREATE_HDR010_DATA ORDER_CREATE_HDR020_DATA ORDER_CREATE_HDR030_DATA ORDER_CREATE_HDR040_DATA 出站消息的 DATUSR_DATA ORDER_CREATE_ITM010_DATA	SendWCSOrder

消息名	消息类型	数据段	控制器命令
订单状态更新	入站	ORDER_STATUS_UPDATE_PROLOG_DATA ORDER_STATUS_UPDATE_HDR010_DATA 入站消息的 DATUSR_DATA ORDER_STATUS_UPDATE_ITM010_DATA	OrderStatus
产品价格更新	入站	PRODUCT_PRICE_UPDATE_PROLOG_DATA PRODUCT_PRICE_UPDATE_HDR010_DATA	ProductOffer PriceUpdate
产品数量更新	入站	PRODUCT_QUANTITY_UPDATE_PROLOG_DATA PRODUCT_QUANTITY_UPDATE_HDR010_DATA	ProductInventory Update

后端集成 XML 消息

WebSphere Commerce 对使用 XML 格式的入站和出站消息提供支持。每个入站消息通过执行控制器命令在 WebSphere Commerce Server 中调用特定的行为。每个控制器命令依次在 WebSphere Commerce 数据库和子系统中执行操作。一些控制器命令可以被多个 XML 消息执行。另外，一些消息可能调用不同的命令，这依赖于消息的内容。

XML 消息的格式由在特定的 DTD 文件内定义的 XML 元素集合构成。每个 DTD 可能包含一个或多个公用文件，由文件扩展名 .mod 标识。另外，每条入站消息与 sys_template.xml 消息模板定义文件中的一条 WebSphere Commerce 控制器命令关联。所有 DTD、MOD 和 XML 文件都在以下目录中：

- ▶ 2000 *drive:\Program Files\WebSphere\CommerceServer\xml\messaging*
- ▶ NT *drive:\WebSphere\CommerceServer\xml\messaging*
- ▶ AIX */usr/WebSphere/CommerceServer/xml/messaging*
- ▶ Solaris */opt/WebSphere/CommerceServer/xml/messaging*
- ▶ 400 */QIBM/Proddata/WebCommerce/xml/messaging*

下表概括了所使用的入站消息以及它们调用的控制器命令：

XML 消息	描述	控制器命令
Create_WCS_Customer, Create_NC_Customer	为新用户创建注册记录，或为现有用户更新记录。	UserRegistrationAdd
Update_NC_Customer, Update_WCS_Customer	为现有用户更新注册记录。	UserRegistrationUpdate
Update_WCS_OrderStatus, Update_NC_OrderStatus	更新订单的常规状态。	OrderStatus
Update_WCS_OrderStatus	更新订单的许可状态。	OrderConfirmStatus
Update_WCS_OrderStatus	更新订单的装运状态。	OrderShippingStatus
Update_WCS_OrderStatus	更新订单的帐单状态。	OrderInvoiceStatus
Update_WCS_ProductPrice, Update_NC_ProductPrice	更新产品的报价信息。	ProductOfferPriceUpdate
Update_WCS_ProductPrice	更新订单的标价信息。	ProductListPriceUpdate
Update_NC_ProductInventory Update_WCS_ProductInventory	更新产品库存信息。	ProductInventoryUpdate

注意: 一些消息的名称中包含字母 NC, 其它的消息名称中包含名称 WCS。那些名称中包含字母 NC 的消息是来自 WebSphere Commerce 的先前版本的 XML 消息。那些名称中有 WCS 的消息使用提供更大的灵活性的已更新的格式。通常建议在具有选项时使用 WCS 版本。

以下表将概括出站消息系统所使用的后端集成消息以及生成它的命令。

XML 消息	描述	控制器命令
Report_NC_PurchaseOrder	发送消息到包含新订单信息的后端系统。	SendXMLOrder

使用实现集成消息的样本方案

实现集成消息允许 WebSphere Commerce 与实现中心系统通信。这允许站点管理员能够随时得到有关他们向顾客提供的产品的可用性信息。以下方案说明实现中心系统和 WebSphere Commerce 系统可以通过使用实现集成消息相互通信而协同工作的方法。

方案 1 — 预期库存与延迟交货的订单

实现中心系统发现一种商品的库存级别很低。它从供应商处订购更多的库存, 并使用 Create_WCS_ExpectedInventoryRecord XML 消息将新库存的预期接收报告给 WebSphere Commerce。

通过根据更多库存的预期接收允许延迟交货订单, 就算库存级别很低, WebSphere Commerce 也可以继续提供该商品以供销售。WebSphere Commerce 发送 Response_WCS_ExpectedInvRecord 消息作为响应, 此消息包含一个 WCSRDetailID 参数。

实现中心注意到此 WCSRDetailID 参数。当新的装运到达时, 实现中心发送 Create_WCS_InventoryReceipt XML 消息, 此消息包含 WCSRDetailID 参数供参考。如果没有任何与新库存关联的 WCSRDetailID, 则省略此 WCSRDetailID。

方案 2 — 库存更新

实现中心的工作人员发现系统中记录的库存级别与物理库存计数期间显示的实际库存之间存在差异。实现中心可以使用 Update_WCS_InventoryReceipt XML 消息将此差异通知 WebSphere Commerce。

WebSphere Commerce 相应地更新已记录的库存级别并用 Response_WCS_UpdateInvReceipt 消息响应。

方案 3 — 订单实现

要实现一个订单, 实现中心发送 Create_WCS_PickBatch XML 消息给 WebSphere Commerce, 初始化实现过程。

WebSphere Commerce 发送包含 PickBatchID 的响应消息, 此消息与“准备装运就绪”的商品列表组合在一起。

实现中心发送 Inquire_WCS_PickPackListDetail XML 消息请求提货单和装货列表详细信息, 以及其它装运信息。

WebSphere Commerce 用一个要选取并包装的商品列表作为响应。

实现中心准备装运并将其发送给顾客，同时还可以选择使用 Create_WCS_ShipmentConfirmation XML 消息将该装运通知 WebSphere Commerce。实现可能还使用电子邮件将该装运通知顾客。

ReleaseShipNotify 消息

ReleaseShipNotify 消息是一个出站电子邮件消息，当订单发货时发送此消息通知顾客。当 ORDRELEASE 表的 STATUS 列更新为 MNF 时，将发生此情况。此电子邮件消息由 ReleaseShipNotify 任务命令发送，它使用 ReleaseShipNotify.jsp 来撰写消息的内容。ReleaseShipNotify 任务命令由 ReleaseManifest 控制器命令调用。

不论实现中心系统是内部的还是外部的，都可以使用此消息。当使用内部 WebSphere Commerce 实现中心时，管理员可以在装运确认屏幕中选择发布载货单选项以触发此消息。当使用外部实现中心系统时，装运确认消息的 UpdateManifestStatus 属性设置为 1 以触发此消息。

此消息可以通过在商店级别覆盖 ReleaseShipNotify 任务命令来启用或禁用。缺省情况下，使用 ReleaseShipNotifyDummyImpl 作为 CMDREG 表的类名来禁用此消息。

Response_WCS_ExpectedInvRecord 消息

Response_WCS_ExpectedInvRecord 消息是包含了有关 WebSphere Commerce 预期库存记录信息的出站消息。WebSphere Commerce 生成此消息作为对入站 Create_WCS_ExpectedInventoryRecord 消息的响应。

如果入站消息包含有效的 StoreID 或者有效的 ExpectedDate，则该消息调用 ExpectedInventoryRecordCreate 命令，该命令在成功完成时重定向到 ExpectedInventoryRecordCreateRedirectView 视图任务，在命令失败时重定向到 ExpectedInventoryRecordCreateErrorView 视图任务。

ExpectedInventoryRecordCreateRedirectView 视图任务由 RACreateError.jsp 实现以撰写 Response_WCS_ExpectedInvRecord 响应消息。在发回实现中心的响应消息中，可以包含 BackendRaDetailID 参数，这样实现中心可以正确的将响应与它发送的原始消息关联。

注意：如果入站消息不包含有效的 StoreID 或者有效的 ExpectedDate，则 GenericApplicationError viewname 用于错误消息编辑。响应消息由 GenericApplicationErrorXML.jsp 生成。

Response_WCS_ExpectedInvRecord 消息使用 XML 消息格式并遵循 Response_WCS_ExpectedInvRecord_10.dtd。

下表描述了 Response_WCS_ExpectedInvRecord 消息的格式。关于数据库列的描述，请随着链接查看与其关联的表。除非另外注明，所有的字段都是可选的。用于类型 INT、BIGINT 和 DOUBLE 的数据库字段的 XML 消息中的标记值长度应当分别为 CHAR(10)、CHAR(19) 和 CHAR(16)。

	字段名	注释	表名	列名	注解
1	ResponseStatus	必填	N/A	N/A	命令失败时的错误注释

	字段名	注释	表名	列名	注解
2	status	必填	N/A	N/A	OK 或 ERROR (ResponseStatus 的一个属性)
3	code		N/A	N/A	出错码 (ResponseStatus 的一个属性, 仅当 status="ERROR" 时才存在)
4	BackendRaID		N/A	N/A	由原始 Create_WCS_ExpectedInventoryRecord 消息引用
5	StoreID		RA	STORE_ID	
6	VendorID		RA	VENDOR_ID	
7	OrderDate		RA	ORDERDATE	
8	WCSRaID		RA	RA_ID	
9	BackendRaDetailID		N/A	N/A	可以用作一个引用
10	ItemOwnerID		ITEMSPC	MEMBER_ID	
11	ProductSKU		ITEMSPC	PARTNUMBER	
12	WCSRaDetailID		RADETAIL	RADETAIL_ID	可以与 Create_WCS_InventoryReceipt 消息一起使用

Response_WCS_PickBatch 消息

Response_WCS_PickBatch 消息是一个出站消息, 它包含有关 WebSphere Commerce 提货批量的信息。WebSphere Commerce 生成此消息作为对入站 Create_WCS_PickBatch 消息的响应。如果入站消息包含有效的 StoreID, 则该消息调用 PickBatchGenerate 命令, 该命令在成功完成时重定向到 PickBatchGenerateRedirectView 视图任务, 在命令失败时重定向到 PickBatchGenerateErrorView 视图任务。对于响应处理, PickBatchGenerateRedirectView 由 PickBatchResult.jsp 实现。在 Response_WCS_PickPatch XML 消息中, 来自原始请求的后端 PickBatchID 以及新生成的 PickBatchID 作为响应发送回。如果命令失败, 则使用 PickBatchGenerateErrorView, 它由 PickBatchError.jsp 实现。

注意: 如果入站消息不包含有效的 StoreID, 则 GenericApplicationError viewname 用于错误消息编辑。响应消息由 GenericApplicationErrorXML.jsp 生成。

Response_WCS_PickBatch 消息使用 XML 消息格式并遵循 Response_WCS_PickBatch_10.dtd。

下表描述了 Response_WCS_PickBatch 消息的格式。关于数据库列的描述, 请随着链接查看与其关联的表。除非另外注明, 所有的字段都是可选的。用于类型 INT、BIGINT、和 DOUBLE 的数据库字段的 XML 消息中的标记值长度应当分别为 CHAR(10)、CHAR(19) 和 CHAR(16)。

级别	字段名	注释	表名	列名	注解
1	ResponseStatus	必填	N/A	N/A	status="ERROR" 时的错误注释

级别	字段名	注释	表名	列名	注解
2	status	必填	N/A	N/A	OK 或 ERROR (ResponseStatus 的一个属性)
3	code		N/A	N/A	出错码 (ResponseStatus 的一个属性, 仅当 status="ERROR" 时才存在)
4	MorePickBatch		N/A	N/A	YES 或 NO: 指示是否应当为更多的 PickBatch 再次提交另一个 Create_WCS_PickBatch 消息。
5	BackendPickBatchID		N/A	N/A	对原始 PickBatch 请求的引用
6	WCSPickBatchID		PICKBATCH	PICKBATCH_ID	如果没有 PickBatch 可用于指定的 StoreID 和 FulfillmentCenterID, 则可以认为 "NULL"。用在 Inquire_WCS_PickPackListDetail 消息中

Report_WCS_PickPackListDetail 消息

Report_WCS_PickPackListDetail 消息是一个出站消息, 它报告提货单和装货列表详细信息。WebSphere Commerce 生成此消息作为对进站 Inquire_WCS_PickPackListDetail 消息的响应。进站消息调用 GetPickPackListDetail 命令, 该命令在成功完成时重定向到 PickPackListRedirectView 视图任务, 在命令失败时重定向到 PickPackListErrorView 视图任务。对于响应处理, PickPackListRedirectView 由 PickPackListResult.jsp 实现。PickPackListErrorView 由 PickPackListError.jsp 实现。

Report_WCS_PickPackListDetail 消息包含两个单独的由 Create_WCS_PickPackListDetail_10.dtd 生成的固定 XML 消息片断。它们是提货单和装货列表。提货单和装货列表的属性在下表中描述。

Report_WCS_PickPackListDetail 消息使用 XML 消息格式并遵循 Report_WCS_PickPackListDetail_10.dtd。

下表描述了 Report_PickPackListDetail 消息的格式。关于数据库列的描述, 请随着链接查看与其关联的表。除非另外注明, 所有的字段都是可选的。用于类型 INT、BIGINT 和 DOUBLE 的数据库字段的 XML 消息中的标记值长度应当分别为 CHAR(10)、CHAR(19) 和 CHAR(16)。

级别	XML 元素	注释	表名	列名	注解
1	ResponseStatus	必填	N/A	N/A	status="ERROR" 时的错误注释

级别	XML 元素	注释	表名	列名	注解
2	status	必填	N/A	N/A	OK 或 ERROR (ResponseStatus 的一个属性)
3	code		N/A	N/A	出错码 (ResponseStatus 的一个属性, 仅当 status="ERROR" 时才存在)
4	PickPackListReport		N/A	N/A	仅当 status="OK" 时, 退出。包含提货单和装货列表。

提货单元素的定义遵循 PickTicket_10.dtd。此 DTD 文件的内容是固定的, 不可修改。

级别	XML 元素	注释	表名	列名	注解
1	Pickticket	必填	N/A	N/A	包含 Pickbatch_Information 和 Pickbatch_Items。一条消息中可以找到多个 Pickbatch_Items。
2	Pickbatch_Information	必填	N/A	N/A	带有以下属性: Store_Name、Fulfillment_Center 和 Pickbatch_Number
3	Store_Name	必填	STOREENTDS	DISPLAYNAME	Pickbatch_Information 的一个属性
4	Fulfillment_Center	必填	FFMCENTDS	DISPLAYNAME	Pickbatch_Information 的一个属性
5	Pickbatch_Number	必填	PICKBATCH	PICKBATCH_ID	Pickbatch_Information 的一个属性
6	Pickbatch_Items		N/A	N/A	带有以下属性: SKU、Product_Name、Product_Description 和 Quantity
7	SKU	必填	BASEITEM	PARTNUMBER	Pickbatch_Items 的一个属性
8	Product_Name	必填	BASEITMDSC	SHORTDESCRIPTION	Pickbatch_Items 的一个属性
9	Product_Description		BASEITMDSC	LONGDESCRIPTION	Pickbatch_Items 的一个属性
10	Quantity	必填	ORDERITEMS	QUANTITY	Pickbatch_Items 的一个属性

装货列表的定义遵循 PackSlip_10.dtd。此 DTD 文件的内容是固定的，不可修改。

级别	XML 元素	注释	表名	列名	注解
1	Packslip	必填	N/A	N/A	包含 Order_Information、Shipto 和 Order_Items。一条消息中可以找到多个装货单。
2	Order_Information	必填	N/A	N/A	带有以下属性： Store_Name、 Fulfillment_Center、 Order_Number、 Release_Number、 PickBatch_Number、 Order_Date、 Catalog_Name、 Shipping_Provider、 Customer_Number 和 Invoice_Method
3	Store_Name	必填	STOREENTDS	DISPLAYNAME	Order_Information 的一个属性
4	Fulfillment_Center	必填	FFMCENTDS	DISPLAYNAME	Order_Information 的一个属性
5	Order_Number	必填	ORDERITEMS	ORDERS_ID	Order_Information 的一个属性
6	Release_Number	必填	ORDERITEMS	ORDERRELEASENUM	Order_Information 的一个属性
7	PickBatch_Number	必填	PICKBATCH	PICKBATCH_ID	Order_Information 的一个属性
8	Order_Date	必填	ORDERS	TIMEPLACED	Order_Information 的属性
9	Shipping_Provider	必填	SHPMODEDSC	DESCRIPTION	Order_Information 的一个属性
10	Customer_Number	必填	ORDERS	MEMBER_ID	Order_Information 的一个属性
11	Invoice_Method		TERMCOND	STRINGFIELD1	这是一个字符串。3 个选项：两者、 电子邮件、打印。缺省值为 NULL。
12	Shipto	必填	N/A	N/A	带有以下属性： AddressID、First_Name、 Last_Name、Middle_Name、 Address_1、Address_2、 Address_3、City、 State、Zip 和 Country
13	AddressID	必填	ADDRESS	ADDRESS_ID	Shipto 的一个属性
14	First_Name		ADDRESS	FIRSTNAME	Shipto 的一个属性

级别	XML 元素	注释	表名	列名	注解
15	Last_Name	必填	ADDRESS	LASTNAME	Shipto 的一个属性
16	Middle_Name		ADDRESS	MIDDLENAME	Shipto 的一个属性
17	Address_1	必填	ADDRESS	MIDDLENAME	Shipto 的一个属性
18	Address_2		ADDRESS	ADDRESS2	Shipto 的一个属性
19	Address_3		ADDRESS	ADDRESS3	Shipto 的一个属性
20	City	必填	ADDRESS	CITY	Shipto 的一个属性
21	State		ADDRESS	STATE	Shipto 的一个属性
22	Zip		ADDRESS	ZIPCODE	Shipto 的一个属性
23	Country	必填	ADDRESS	COUNTRY	Shipto 的一个属性
24	Order_Items	必填	N/A	N/A	可以包含带有以下属性: SKU、Product_Name、 Product_Description、 Quantity、Catalog_Name、 Unit_Price、Total_Price、 Currency 和 Customer_Comments 的 多个组件。
25	SKU	必填	BASEITEM	PARTNUMBER	Order_Items 的一个属性
26	Product_Name	必填	BASEITMDSC	SHORTDESCRIPTION	Order_Items 的一个属性
27	Product_Description		BASEITMDSC	LONGDESCRIPTION	Order_Items 的一个属性
28	Quantity	必填	ORDERITEMS	QUANTITY	Order_Items 的一个属性
29	Catalog_Name	必填	CATENTDESC	NAME	Order_Items 的一个属性
30	Unit_Price	必填	ORDERITEMS	PRICE	Order_Items 的一个属性
31	Total_Price	必填	ORDERITEMS	TOTALPRODUCT	Order_Items 的一个属性
32	Currency	必填	ORDERITEMS	CURRENCY	Order_Items 的一个属性
33	Customer_Comments		ORDERITEMS	COMMENTS	Order_Items 的一个属性
34	Component		N/A	N/A	带有以下属性: SKU、Product_Name、 Product_Description 和 Quantity
35	SKU	必填	BASEITEM	PARTNUMBER	Component 的一个属性
36	Product_Name	必填	BASEITMDSC	SHORTDESCRIPTION	Component 的一个属性
37	Product_Description		BASEITMDSC	LONGDESCRIPTION	Component 的一个属性
38	Quantity	必填	ORDERITEMS	QUANTITY	Component 的一个属性

Response_WCS_CreateInvReceipt 消息

Response_WCS_CreateInvReceipt 消息是一个出站消息，它包含有关创建 WebSphere Commerce 库存接收的信息。WebSphere Commerce 生成此消息作为对入站 Create_WCS_InventoryReceipt 消息的响应。如果入站消息包含有效的 storeID 和有效的 ReceiptDate，则它调用视图任务 ReceiptCreateRedirectView，此命令使用 CreateInvReceiptOK.jsp 生成一个响应消息。在响应消息中，包含了 RECEIPT 和

RCPTAVAIL 表中创建的记录的新 receipt_id。如果此命令遇到错误，则使用的视图任务是 ReceiptCreateErrorView。此错误视图任务由 CreateInvReceiptError.jsp 实现。如果存在足够的库存来实现预期库存记录，则关闭该记录。

注意：如果入站消息不包含有效的 StoreID 或者有效的 ReceiptDate，则 GenericApplicationError viewname 用于错误消息编辑。响应消息由 GenericApplicationErrorXML.jsp 生成。

Response_WCS_CreateInvReceipt 消息使用 XML 消息格式并遵循 Response_WCS_CreateInvReceipt_10.dtd。

下表描述了 Response_WCS_CreateInvReceipt 消息的格式。关于数据库列的描述，请随着链接查看与其关联的表。除非另外注明，所有的字段都是可选的。用于类型 INT、BIGINT 和 DOUBLE 的数据库字段的 XML 消息中的标记值长度应当分别为 CHAR(10)、CHAR(19) 和 CHAR(16)。

级别	字段名	注释	表名	列名	注解
1	ResponseStatus	必填	N/A	N/A	错误注释
2	status	必填	N/A	N/A	OK 或 ERROR (ResponseStatus 的一个属性)
3	code		N/A	N/A	出错码 (ResponseStatus 的一个属性, 仅当 status="ERROR" 时才存在)
4	ItemOwnerID		ITEMSPC	MEMBER_ID	N/A
5	ProductSKU		ITEMSPC	PARTNUMBER	N/A
6	StoreID		RECEIPT	STORE_ID	N/A
7	FulfillmentCenterID		RECEIPT	FFMCENTER_ID	N/A
8	VendorID		RECEIPT	VENDOR_ID	N/A
9	QTYReceived		RECEIPT	QTYRECEIVED	N/A
10	ReceiptDate		RECEIPT	RECEIPTDATE	N/A

Response_WCS_UpdateInvReceipt 消息

Response_WCS_UpdateInvReceipt 消息是一个出站消息，它包含有关更新商品库存的信息。WebSphere Commerce 生成此消息作为对 Update_WCS_InventoryReceipt 消息的响应。如果入站消息包含有效的 StoreID，则它调用 InventoryAdjust 命令，该命令在成功完成时重定位到视图任务 InventoryAdjustRedirectView。在成功完成之上，InventoryAdjustRedirectView 使用 UpdateInvReceiptOK.jsp 生成响应消息。如果命令遇到错误，则使用的视图任务是 InventoryAdjustErrorView。此错误视图任务由 UpdateInvReceiptError.jsp 实现。

如果消息中的库存调整是正确的，则该命令在 RECEIPT 和 RCPTAVAIL 数据库表中创建一个新的行。如果调整是负的，则使用相应的选取方法标记 RECEIPT 表中的 QTYONHAND 行。

注意：如果入站消息不包含有效的 StoreID，则 GenericApplicationError viewname 用于错误消息编辑。响应消息由 GenericApplicationErrorXML.jsp 生成。

Response_WCS_UpdateInvReceipt 消息使用 XML 消息格式并遵循 Response_WCS_UpdateInvReceipt_10.dtd。

下表描述了 Response_WCS_UpdateInvReceipt 消息的格式。关于数据库列的描述，请随着链接查看与其关联的表。除非另外注明，所有的字段都是可选的。用于类型 INT、BIGINT 和 DOUBLE 的数据库字段的 XML 消息中的标记值长度应当分别为 CHAR(10)、CHAR(19) 和 CHAR(16)。

级别	字段名	注释	表名	列名	注解
1	ResponseStatus	必填	N/A	N/A	错误注释
2	status	必填	N/A	N/A	OK 或 ERROR (ResponseStatus 的一个属性)
3	code		N/A	N/A	出错码 (ResponseStatus 的一个属性, 仅当 status="ERROR" 时才存在)
4	ItemOwnerID		ITEMSPC	MEMBER_ID	N/A
5	ProductSKU		ITEMSPC	PARTNUMBER	N/A
6	StoreID		RECEIPT	STORE_ID	N/A
7	FulfillmentCenterID		RECEIPT	FFMCENTER_ID	N/A
8	QTYAdjusted		INVADJUST	QUANTITY	N/A
9	InvAdjCodeID		INVADJUST	INVADJCODE_ID	N/A

Response_WCS_CreateShipConfirm 消息

Response_WCS_CreateShipConfirm 消息是一个出站消息，它包含有关创建订单的装运确认的信息。WebSphere Commerce 生成此消息作为对入站 Create_WCS_ShipmentConfirmation 消息的响应。如果入站消息包含有效的 ActualShipDate，则它调用 ReleaseShipConfirm 命令，该命令在成功完成时重定向到视图任务 ReleaseShipConfirmRedirectView。Response_WCS_CreateShipConfirm 响应消息由 CreateShipConfirmOK.jsp 生成。此命令更新必需的数据库，将商品的实现状态更改为确认装运。它通过密钥管理器从 MANIFEST 表获取新的 manifest_id，并用输入的数据传播 MANIFEST 表。在 Release_WCS_ShipmentNotify 消息已启用的情况下，如果此命令执行成功而且 UpdateManifestStatus 为 1，则缺省的 ReleaseShipNotify.jsp 生成一个通知电子邮件。

如果此命令遇到错误，则它重定向到视图任务 ReleaseShipConfirmErrorView。此错误视图任务由 CreateShipConfirmError.jsp 实现。

注意：如果入站消息不包含有效的 ActualShipDate，则 GenericApplicationError viewname 用于错误消息编辑。响应消息由 GenericApplicationErrorXML.jsp 生成。

Response_WCS_CreateShipConfirm 消息使用 XML 消息格式并遵循 Response_WCS_CreateShipConfirm_10.dtd。

下表描述了 Response_WCS_CreateShipConfirm 消息的格式。关于数据库列的描述，请随着链接查看与其关联的表。除非另外注明，所有的字段都是可选的。用于类型 INT、BIGINT 和 DOUBLE 的数据库字段的 XML 消息中的标记值长度应当分别为 CHAR(10)、CHAR(19) 和 CHAR(16)。

级别	字段名	注释	表名	列名	注解
1	ResponseStatus	必填	N/A	N/A	status="ERROR" 时的错误注释
2	status	必填	N/A	N/A	OK 或 ERROR (ResponseStatus 的一个属性)
3	code		N/A	N/A	出错码 (ResponseStatus 的一个属性, 仅当 status="ERROR" 时才存在)
4	ShipModeID		MANIFEST	SHIPMODE_ID	N/A
5	OrderNumber		MANIFEST	ORDERS_ID	N/A
6	OrderReleaseNum		MANIFEST	ORDERRELEASENUM	N/A

Update_WCS_OrderStatus 消息

Update_WCS_OrderStatus 消息是包含 WebSphere Commerce 订单状态信息的入站消息。此消息有四种可能的形式。虽然每种消息形式都具有相同的 XML 元素，但每种形式都与不同的命令相关联。消息表单至命令的映射如下：

订单状态消息类型	命令
OrderConfirm	OrderConfirmStatus
OrderShipping	OrderShippingStatus
OrderInvoice	OrderInvoiceStatus
OrderStatus	OrderStatus

当订单由后端系统接收时，它生成此消息，包含任何订单实现状态信息，并将它发送到 WebSphere Commerce 入站消息队列中，WebSphere Commerce 系统在该队列中接收此消息。WebSphere Commerce 调用相应的控制器命令用新的订单状态信息更新表 ORDSTAT 和 ORDISTAT。

Update_WCS_OrderStatus 消息使用 XML 消息格式并遵循 Update_WCS_OrderStatus_20.dtd。

下表描述 Update_WCS_OrderStatus 消息的格式。除非特别指出，这四种订单状态消息类型都遵循相同的格式。下表中描述了 XML 元素值的格式和源。关于数据库列的描述，请随着链接查看与其关联的表。除非另外注明，所有的字段都是可选的。用于类型 INT、BIGINT 和 DOUBLE 的数据库字段的 XML 消息中的标记值长度应当分别为 CHAR(10)、CHAR(19) 和 CHAR(16)。

级别	字段名	注释	表名	列名	注解
1	<i>OrderStatusType</i>	必填	N/A	N/A	OrderConfirm、 OrderShipping、 OrderInvoice 或 OrderStatus
1.1	SerializationInfo		N/A	N/A	如果已提供，则用它来对 消息检查序列化信息。
1.1.1	SequenceNumber		ORDSTAT	OSSEQNUM	
1.1.2	LastUpdate Timestamp		ORDSTAT	OSUPDTIME	
1.2	OrderStatus Header	必填	N/A	N/A	
1.2.A1	Versioning	属性	N/A	N/A	“TRUE” 或 “FALSE” 值。如果 “TRUE” 则启 用 versioning。
1.2.1	OrderNumber		ORDSTAT	ORDERS_ID/ OSMORDER	如果 type= ByWCS 则 ORDERS_ID（它是 WebSphere Commerce 订单引用号），否则 OSMORDER（后端系统 生成的订单引用号）。
	type	属性			ByWCS 或 ByBackend
1.2.2	TotalPriceInfo		N/A	N/A	
1.2.2.A1	currency	属性	ORDSTAT	OSCPCUR	
1.2.2.1	TotalNetPrice		ORDSTAT	OSPRTOT	
1.2.2.2	TotalTaxPrice		ORDSTAT	OSTXTOT	
1.2.2.3	TotalShippingPrice		ORDSTAT	OSSHOTOT	
1.2.2.4	TotalTaxOn ShippingPrice		ORDSTAT	OSSHOTXTOT	
1.2.3	Status		ORDSTAT	OSSTATUS	缺省值： “C” = “已确认”（对 于 OrderConfirm） “S” = “已装运”（对 于 OrderShipping） “I” = “已开票”（对于 OrderInvoice）
1.2.4	PlacedDate	必填	ORDSTAT	OSPLTIME	
1.2.5	ShippingInfo		N/A	N/A	
1.2.5.A1	ShipCondition	属性	ORDSTAT	OSSCOND	指示是否允许订单部份装 运的代码 “SC” = 完全装运， “SP” = 部分装运。

级别	字段名	注释	表名	列名	注解
1.2.5.A2	ShipModeFlag	属性	ORDSTAT	OSSMFLAG	指示送货地址和装运方式是否为订单级别或者订购商品级别的代码。 “O” = 订单级别 “I” = 订购商品级别。
1.2.5.1	RequestShipDate		ORDSTAT	OSRSTIME	
1.2.5.2	ScheduledShipDate		ORDSTAT	OSSSTIME	
1.2.5.3	ActualShipDate		ORDSTAT	OSASTIME	
1.2.6	InvoiceInfo				
1.2.6.1	InvoiceDate		ORDSTAT	OSINVTIME	
1.2.6.2	InvoiceValue		ORDSTAT	OSINVVAL	
1.2.7	Comment		ORDSTAT	OSCMNT	
1.2.8	CustomerField	首先出现	ORDSTAT	FIELD1	
1.2.8	CustomerField	第二出现	ORDSTAT	FIELD2	
1.2.8	CustomerField	第三出现	ORDSTAT	FIELD3	
1.2.9	UserData		N/A	N/A	
1.3	OrderStatusItem	可重复	N/A	N/A	向量
1.3.A1	Versioning	属性	N/A	N/A	“TRUE” 或 “FALSE” 值。如果 “TRUE” 则启用 versioning。
1.3.1	ItemNumber		ORDISTAT	ORDERITMES_ID/ OIMITEM	如果 type = ByWCS, 则 ORDERITMES_ID, 如果 type = ByBackend 则 OIMITEM
1.3.1.A1	type	属性	N/A	N/A	ByWCS 或 ByBackEnd。
1.3.2	ProductNumber ByMerchant		ORDISTAT	PARTNUMBER	
1.3.3	QuantityInfo		N/A	N/A	
1.3.3.1	RequestedQuantity		ORDISTAT	OIQTREQUEST	
1.3.3.2	ConfirmedQuantity		ORDISTAT	OIQTCONFIRM	
1.3.3.3	ShippedQuantity		ORDISTAT	OIQTSHIP	
1.3.4	ItemUnitPrice		ORDISTAT	OIUNPRC	
1.3.5	TotalPriceInfo		N/A	N/A	
1.3.5.A1	currency	属性	ORDISTAT	OICPCUR	
1.3.5.1	TotalNetPrice		ORDISTAT	OIPRTOT	
1.3.5.2	TotalTaxPrice		ORDISTAT	OITXTOT	
1.3.5.3	TotalShippingPrice		ORDISTAT	OISHTOT	
1.3.5.4	TotalTaxOn ShippingPrice		ORDISTAT	OISHTXTOT	

级别	字段名	注释	表名	列名	注解
1.3.6	Status		ORDISTAT	OISTATUS	
1.3.7	PlacedDate		ORDISTAT	OIPLTIME	
1.3.8	ShippingInfo		N/A	N/A	
1.3.8A1	ShipCondition		ORDISTAT	OISCOND	指示是否允许行商品部份装运的代码。“SC” = 完全装运 “SP” = 部分装运
1.3.8A2	ShipModeFlag		N/A	N/A	
1.3.8.1	Requested ShipDate		ORDISTAT	OIRSTIME	
1.3.8.2	Scheduled ShipDate		ORDISTAT	OISSTIME	
1.3.8.3	ActualShipDate		ORDISTAT	OIASTIME	
1.3.9	InvoiceInfo		N/A	N/A	
1.3.9.1	InvoiceDate		ORDISTAT	OIINVTIME	
1.3.9.2	InvoiceValue		ORDISTAT	OIINVVAL	
1.3.10	Comment		ORDISTAT	OICMNT	
1.3.11	CustomerField1	首先出现	ORDISTAT	FIELD1	
1.3.11	CustomerField2	第二出现	ORDISTAT	FIELD2	
1.3.11	CustomerField3	第三出现	ORDISTAT	FIELD3	
1.3.12	UserData		N/A	N/A	

“订单状态更新” 消息

“订单状态更新” 消息是一个入站消息，它包含 WebSphere Commerce 订单的状态消息。一个后端应用程序生成此消息并将其发送到 WebSphere Commerce 入站消息队列，WebSphere Commerce 系统在此队列中接收消息。WebSphere Commerce 使用 OrderStatus 命令用新的订单状态信息更新表 ORDSTAT 和 ORDISTAT。

“订单状态更新” 消息支持两种应用程序数据集合：“订单状态更新” 版本 01 和“订单状态更新” 版本 02。版本 02 包括版本 01 中数据的超集。

“订单状态更新” 消息使用 WebSphere Commerce 消息格式，它由一组在缓冲区中按顺序排列的记录组成。下列数据描述了“订单状态更新” 消息：

```
<ECEDOC>
<PROLOG>ORDER_STATUS_UPDATE_PROLOG_DATA</PROLOG>
<HEADER>
  <HDR010>ORDER_STATUS_UPDATE_HDR010_DATA</HDR010>
  <USRLST>
    <DATUSR>DATUSR_DATA</DATUSR>
```

```

                .DATUSR repeated loop
                .
                <DATUSR>DATUSR_DATA</DATUSR>
            </USRLST>
        </HEADER>
    <ITMLST>
        <ITMDAT>
            <ITM010>ORDER_STATUS_UPDATE_ITM010_DATA</ITM010>
            <USRLST>
                <DATUSR>DATUSR_DATA</DATUSR>
                .
                .DATUSR repeated loop
                .
                <DATUSR>DATUSR_DATA</DATUSR>
            </USRLST>
        </ITMDAT>
        .
        .ITEM repeated loop
        .
        <ITMDAT>

    <ITM010>ORDER_STATUS_UPDATE_ITM010_DATA</ITM010>
    <USRLST>

    <DATUSR>DATUSR_DATA</DATUSR>
    .
    .DATUSR repeated loop
    .
    <DATUSR>DATUSR_DATA</DATUSR>
    </USRLST>
    </ITMDAT>
</ITMLST>
</ECEDOC>

```

注意:

- 所有记录在缓冲区中按顺序排列。此处使用缩进是为了可读性；缓存区中并不会出现缩进。
- 在缓冲区中，数据段中的所有字段是左对齐的，右边用空格填充。

订单状态更新的数据段

- **ORDER_STATUS_UPDATE_PROLOG_DATA**
指定应用程序数据定义的消息类型。在此情况下，消息为“订单状态更新”。
- **ORDER_STATUS_UPDATE_HDR010_DATA**
指定“订单状态更新”消息中的订单信息。
- **DATUSR_DATA**
指定要添加到“订单状态更新”消息的可选信息。DATUSR_DATA 在此消息的 <HDR> 和 <ITM> 部分出现。

- ORDER_STATUS_UPDATE_ITM010_DATA
指定“订单状态更新”消息中的商品或产品装运信息。

Update_WCS_ProductPrice 消息

Update_WCS_ProductPrice 消息是包含产品价格信息的入站消息。后端应用程序生成此消息并将其发送到 WebSphere Commerce 入站消息队列。当 WebSphere Commerce 系统接收此消息时，它就运行一个控制器命令。运行的控制器命令依赖于消息的类型。以下表显示了可以发送的两种类型的消息，以及与它们关联的命令。

产品价格消息类型	控制器命令
OfferPrice Update	ProductOfferPriceUpdate
ListPrice Update	ProductListPriceUpdate

Update_WCS_ProductPrice 消息使用 XML 消息格式并遵循 Update_WCS_ProductPrice_20.dtd。

OfferPrice Update

下表描述 Update_WCS_ProductPrice 消息的 OfferPrice Update 变体格式。关于数据库列的描述，请随着链接查看与其关联的表。除非另外注明，所有的字段都是可选的。用于类型 INT、BIGINT 和 DOUBLE 的数据库字段的 XML 消息中的标记值长度应当分别为 CHAR(10)、CHAR(19) 和 CHAR(16)。

级别	XML 元素	注释	表名	列名
1	OfferPriceInfo		N/A	N/A
1.1	ProductNumberByMerchant	必填	CATENTRY	PARTNUMBER
1.2	MerchantID	必填	CATENTRY	MEMBER_ID
1.2.A1	type	属性	N/A	N/A
1.3	Precedence		OFFER	PRECEDENCE
1.4	TradingPositionContainerID		OFFER	TRADEPOSCN_ID
1.5	Currency	必填	OFFERPRICE	CURRENCY
1.6	ItemUnitPrice		OFFERPRICE	PRICE
1.7	StartTimeStamp		OFFER	STARTDATE
1.8	EndTimeStamp		OFFER	ENDDATE
1.9	MinimumQuantity		OFFER	MINIMUMQUANTITY
1.10	MaximumQuantity		OFFER	MAXIMUMQUANTITY
1.11	QuantityUnit		OFFER	QTYUNIT_ID
1.12	Published		OFFER	PUBLISHED
1.13	PriceCustomField	首先出现	OFFER	FIELD1
1.13	PriceCustomField	第二出现	OFFER	FIELD2
1.14	UserData		N/A	N/A

OfferPrice Update 行为:

- 货币类型（引用 OFFERPRICE 表中的 CURRENCY）是必需的，且必须以 ISO 4217 格式指定。
- 部件号（引用 CATENTRY 表中的 PARTNUMBER）和成员标识（引用 CATENTRY 表中的 MEMBER_ID）的组合将用于获取产品目录（CATENTRY_ID）。该值将与优先级（PRECEDENCE）或贸易状态容器（TRADEPOSCN_ID）一起用于获取产品价格引用号（OFFER_ID）。该产品价格引用号与货币类型一起作为更新 OFFERPRICE 表中某行的键使用。
- 如果该产品价格引用号（OFFER_ID）与数据库中现有的产品价格引用号匹配，但货币类型与具有该产品价格引用号的任何记录中的货币类型不匹配，则将在 OFFERPRICE 表中创建一个新记录。这就允许您对相同报价以不同货币类型指定价格。
- 如果未指定优先级（PRECEDENCE），那么 ProductOfferPriceUpdate 命令将找出与此给定值（没有优先级）匹配的所有先前记录。取这些值中的最大值，并加 1。如果前一记录不存在，那么将优先级值设置为 1。以新的优先级值在 OFFERPRICE 表插入一行。此优先级的值必须低于 10^{16} 。如果到达最大的值，那么将拒绝新的更新。

ListPrice Update

下表描述 Update_WCS_ProductPrice 消息的 ListPrice Update 变体格式。关于数据库列的描述，请随着链接查看与其关联的表。除非另外注明，所有的字段都是可选的。用于类型 INT、BIGINT 和 DOUBLE 的数据库字段的 XML 消息中的标记值长度应当分别为 CHAR(10)、CHAR(19) 和 CHAR(16)。

1	ListPriceInfo		N/A	N/A
1.1	ProductNumberByMerchant		CATENTRY	PARTNUMBER
1.2	MerchantID		CATENTRY	MEMBER_ID
1.2.A1	type	属性	N/A	N/A
1.3	Currency		LISTPRICE	CURRENCY
1.4	ItemUnitPrice		LISTPRICE	LISTPRICE
1.5	UserData		N/A	N/A

ListPrice Update 行为:

- 此命令更新 LISTPRICE 表中的记录。
- partNumber 与 memberId 一起用于获取产品目录条目键（表 CATENTRY 中的 CATENTRY_ID）。
- 如果该产品目录条目与 LISTPRICE 表中现有的产品目录条目匹配，但货币类型与具有该产品目录条目的任何记录中的货币类型不匹配，将在 LISTPRICE 表中创建一个新记录。这允许您对相同的产品目录条目指定不同的货币价格。

Create_WCS_ExpectedInventoryRecord 消息

Create_WCS_ExpectedInventoryRecord 消息是一个入站消息，它包含有关在 WebSphere Commerce 数据库中创建期望的库存记录的信息。实现中心应用程序生成此请求并将其发送到 WebSphere Commerce 入站消息队列，WebSphere Commerce 系统在此队列中接收它。在 WebSphere Commerce 处理此消息之后，调用 ExpectedInventoryRecordCreate 命令。该命令在 RA 表中创建一条记录，并为此消息中的每个 RADETAIL 组件创建一个 RADETAIL 记录。

注意：如果入站消息不包含有效的 StoreID 或者有效的 ExpectedDate，则 GenericApplicationError viewname 用于错误消息编辑，而不调用 ExpectedInventoryRecordCreate 命令。响应消息由 GenericApplicationErrorXML.jsp 生成。

Create_WCS_ExpectedInventoryRecord 消息使用 XML 消息格式并遵循 Create_WCS_ExpectedInventoryRecord_10.dtd。

下表描述了 Create_WCS_ExpectedInventoryRecord 消息的格式。关于数据库列的描述，请随着链接查看与其关联的表。除非另外注明，所有的字段都是可选的。用于类型 INT、BIGINT 和 DOUBLE 的数据库字段的 XML 消息中的标记值长度应当分别为 CHAR(10)、CHAR(19) 和 CHAR(16)。

	字段名	注释	表名	列名	注解
1	BackendRaID		N/A	N/A	这可以用在响应消息中，以与原始请求关联
2	StoreID	必填	RA	STORE_ID	N/A
3	VendorID	必填	RA	VENDOR_ID	N/A
4	OrderDate	必填	RA	ORDERDATE	N/A
5	ExternalID		RA	EXTERNALID	N/A
6	FulfillmentCenterID	必填	RADETAIL	FFMCENTER_ID	N/A
7	ItemOwnerID	必填	ITEMSPC	MEMBER_ID	N/A
8	ProductSKU	必填	ITEMSPC	PARTNUMBER	ProductSKU 与 ItemOwnerID 一起用于标识商品规格
9	ExpectedDate	必填	RADETAIL	EXPECTEDDATE	ISO 8601 日期格式
10	QuantityOrdered	必填	RADETAIL	QTYORDERED	N/A
11	注释		RADETAIL	RADETAILCOMMENT	N/A
12	BackendRaDetailID		N/A	N/A	这可以用在响应消息中，以与原始请求关联。

Create_WCS_PickBatch 消息

The Create_WCS_PickBatch 消息是一个入站消息，它包含有关生成 WebSphere Commerce 提货批量的信息。实现中心应用程序生成此请求并将其发送到 WebSphere Commerce 入站消息队列。在 WebSphere Commerce 处理此消息之后，调用 PickBatchGenerate 控制器命令。此命令通过密钥管理器管理器获取新的 PICKBATCH_ID，并在 ORDRELEASE

数据库表中选择 STATUS 列的值为 SHIP 的所有行，然后生成 XML 提货单以及输入的 StoreID 和 FulfillmentCenterID 的装货列表，最后将它们相应的保存到 PICKBATCH 和 ORDRELEASE 表。发送 Response_WCS_PickBatch 消息以作为响应。

注意：如果入站消息不包含有效的 StoreID，则 GenericApplicationError viewname 用于错误消息编辑，而不调用 PickBatchGenerate 命令。响应消息由 GenericApplicationErrorXML.jsp 生成。

Create_WCS_PickBatch 消息使用 XML 消息格式并遵循 Create_WCS_PickBatch_10.dtd。

以下表描述了 Create_WCS_PickBatch 消息的格式。关于数据库列的描述，请随着链接查看与其关联的表。除非另外注明，所有的字段都是可选的。用于类型 INT、BIGINT 和 DOUBLE 的数据库字段的 XML 消息中的标记值长度应当分别为 CHAR(10)、CHAR(19) 和 CHAR(16)。

级别	字段名	注释	表名	列名	注解
1	BackendPickBatchID		N/A	N/A	这可以用在 Response_WCS_PickBatch 消息中
2	StoreID	必填	ORDERITEMS	STORE_ID	N/A
3	FulfillmentCenterID	必填	ORDERITEMS	FFMCENTER_ID	N/A

Inquire_WCS_PickPackListDetail 消息

Inquire_WCS_PickPackListDetail 消息是一个入站消息，它请求由先前 Create_WCS_PickBatch 消息创建的提货单和装货列表信息。实现中心应用程序生成此请求并将其发送到 WebSphere Commerce 入站消息队列。处理时，入站消息调用 GetPickPackListDetail 命令，该命令在成功完成时重定向到 PickPackListResult 视图任务，在命令失败时重定向到 PickPackListErrorView 视图任务。成功完成时，该命令从 PICKBATCH 表中检索给定 PICKBATCH_ID 的提货单 XML，从 ORDRELEASE 表检索给定 PICKBATCH_ID 的装货列表 XML。PickPackListResult.jsp 文件生成包含提货单和装货列表的出站 Report_WCS_PickPackListDetail 消息作为响应。

Inquire_WCS_PickPackListDetail 消息使用 XML 消息格式并遵循 Inquire_WCS_PickPackListDetail_10.dtd。

下表描述了 Inquire_PickPackListDetail 消息的格式。关于数据库列的描述，请随着链接查看与其关联的表。除非另外注明，所有的字段都是可选的。用于类型 INT、BIGINT 和 DOUBLE 的数据库字段的 XML 消息中的标记值长度应当分别为 CHAR(10)、CHAR(19) 和 CHAR(16)。

级别	XML 元素	注释	表名	列名	注解
1	PickBatchID	必填	PICKBATCH	PICKBATCH_ID	N/A

Create_WCS_InventoryReceipt 消息

Create_WCS_InventoryReceipt 消息是一个入站消息，它包含有关创建 WebSphere Commerce 库存记录的信息。实现中心应用程序生成此请求并将其发送到 WebSphere Commerce 入站消息队列，WebSphere Commerce 系统在此队列中接收消息。当 WebSphere Commerce 系统接收此消息时，它就调用 ReceiptCreate 控制器命令。如果命令执行成功，则发送 Response_WCS_CreateInvReceipt 消息作为响应。如果存在足够的库存来实现期望的库存记录，则关闭该记录。

注意：如果入站消息不包含有效的 StoreID 或者有效的 ReceiptDate，则 GenericApplicationError viewname 用于错误消息编辑，而不调用 ReceiptCreate 命令。响应消息由 GenericApplicationErrorXML.jsp 生成。

Create_WCS_InventoryReceipt 消息使用 XML 消息格式并遵循 Create_WCS_InventoryReceipt_10.dtd。

下表描述了 Create_WCS_InventoryReceipt 消息的格式。关于数据库列的描述，请随着链接转至与其关联的表。除非另外注明，所有的字段都是可选的。用于类型 INT、BIGINT 和 DOUBLE 的数据库字段的 XML 消息中的标记值长度应当分别为 CHAR(10)、CHAR(19) 和 CHAR(16)。

级别	字段名	注解	表名	列名	注解
1	ItemOwnerID	必需	ITEMSPC	MEMBER_ID	N/A
2	ProductNumberbyMerchant	必需	ITEMSPC	PARTNUMBER	N/A
3	VersionName		ITEMVERSN	VERSIONNAME	保留供 IBM 内部使用。
4	StoreID	必需	RECEIPT	STORE_ID	N/A
5	FulfillmentCenterID	必需	RECEIPT	FFMCENTER_ID	N/A
6	VendorID	必需	RECEIPT	VENDOR_ID	N/A
7	Cost	必需	RECEIPT	COST	N/A
8	Currency	必需	RECEIPT	SETCCUR	N/A
9	QTYReceived	必需	RECEIPT	QTYRECEIVED	N/A
10	ReceiptDate	必需	RECEIPT	RECEIPTDATE	ISO 8601 日期格式
11	WCSRaDetailID		RECEIPT	RADETAIL_ID	记录在 RA 表中的预期库存记录。如果不存在此记录，则接收为特别类型。
12	ReceiptComment		RECEIPT	COMMENT1	N/A
13	QualityComment		RECEIPT	COMMENT2	N/A

Update_WCS_InventoryReceipt 消息

Update_WCS_InventoryReceipt 消息是一个入站消息，它包含有关调节商品库存的信息。实现中心应用程序生成此消息并将其发送到 WebSphere Commerce 入站消息队列，WebSphere Commerce 系统在此队列中接收消息。在 WebSphere Commerce 处理此消息之后，调用 InventoryAdjust 命令。

注意：如果入站消息不包含有效的 StoreID，则 GenericApplicationError viewname 用于错误消息编辑，而不调用 InventoryAdjust 命令。响应消息由 GenericApplicationErrorXML.jsp 生成。

Update_WCS_InventoryReceipt 消息使用 XML 消息格式并遵循 Update_WCS_InventoryReceipt_10.dtd。

下表描述了 Update_WCS_InventoryReceipt 消息的格式。关于数据库列的描述，请随着链接查看与其关联的表。除非另外注明，所有的字段都是可选的。用于类型 INT、BIGINT 和 DOUBLE 的数据库字段的 XML 消息中的标记值长度应当分别为 CHAR(10)、CHAR(19) 和 CHAR(16)。

级别	字段名	注释	表名	列名	注解
1	ItemOwnerID	必填	ITEMSPC	MEMBER_ID	N/A
2	ProductSKU	必填	ITEMSPC	PARTNUMBER	N/A
3	VersionName		ITEMVERSN	VERSIONNAME	保留供 IBM 内部使用。
4	StoreID	必填	RECEIPT	STORE_ID	N/A
5	FulfillmentCenterID	必填	RECEIPT	FFMCENTER_ID	N/A
6	Comment		RECEIPT	COMMENT1	N/A
7	QTYAdjusted	必填	INVADJUST	QUANTITY	可以为正值或负值
8	InvAdjCodeID	必填	INVADJUST	INVADJCODE_ID	N/A

Create_WCS_ShipmentConfirmation 消息

Create_WCS_ShipmentConfirmation 消息是一个入站消息，它包含有关发出商品的装运确认的信息。实现中心应用程序生成此请求并将其发送到 WebSphere Commerce 入站消息队列。在 WebSphere Commerce 处理此消息之后，调用 ReleaseShipConfirm 命令。

如果此命令成功完成，则它重定向到视图任务 ReleaseShipConfirmRedirectView。Response_WCS_CreateShipConfirm 响应消息由 CreateShipConfirmOK.jsp 生成。此命令更新必需的数据库，将商品的实现状态更改为确认装运。它通过密钥管理器从 MANIFEST 表获取新的 manifest_id，并用输入的数据传播 MANIFEST 表。如果此命令执行成功而且 UpdateManifestStatus 为 1，则缺省的 ReleaseShipNotify.jsp 生成一个通知电子邮件。

如果此命令遇到错误，则它重定向到视图任务 ReleaseShipConfirmErrorView。此 MQSeries 错误视图任务由 CreateShipConfirmError.jsp 实现。

注意：如果入站消息不包含有效的 ActualShipDate，则 GenericApplicationError viewname 用于错误消息编辑。响应消息由 GenericApplicationErrorXML.jsp 生成。

Create_WCS_ShipmentConfirmation 消息使用 XML 消息格式并遵循 Create_WCS_ShipmentConfirmation_10.dtd。

下表描述了 Create_WCS_ShipmentConfirmation 消息的格式。关于数据库列的描述，请随着链接查看与其关联的表。除非另外注明，所有的字段都是可选的。用于类型 INT、BIGINT 和 DOUBLE 的数据库字段的 XML 消息中的标记值长度应当分别为 CHAR(10)、CHAR(19) 和 CHAR(16)。

级别	字段名	注释	表名	列名	注解
1	UpdateManifestStatus		MANIFEST	UPDATEMANIFESTSTATUS	缺省值 0，父元素 ShipmentConfirmation 的一个属性
2	ShipModeID	必填	MANIFEST	SHIPMODE_ID	N/A
3	OrderNumber	必填	MANIFEST	ORDERS_ID	N/A
4	OrderReleaseNum	必填	MANIFEST	ORDERRELEASENUM	N/A
5	PackageID		MANIFEST	PACKAGEID	N/A
6	TrackingID		MANIFEST	TRACKINGID	N/A
7	PickUpRecordID		MANIFEST	PICKUPRECORDID	N/A
8	ActualShipDate	必填	MANIFEST	DATESHIPPED	ISO 8601 日期格式
9	ShippingCosts	必填	MANIFEST	SHIPPINGCOSTS	N/A
10	Weight	必填	MANIFEST	WEIGHT	N/A
11	measure	必填	MANIFEST	WEIGHTMEASURE	Weight 的一个属性
12	currency	必填	MANIFEST	SETCCURR	ShippingCosts 的一个属性

UpdateManifestStatus 标志的缺省值为 0。如果它设置成 1，则调用一个任务命令更新载货单状态，同时发送一封电子邮件通知顾客产品装运完成。

生成该响应消息的 JSP 文件是 Response_WCS_CreateShipConfirm.jsp。

ReleaseShipNotify.jsp 是缺省的 JSP 文件，它生成电子邮件（如果 UpdateManifestStatus 标志设置为 1 的话）。请参阅 Message composition templates 获取有关 JSP 文件位置的信息。

Create_WCS_Customer 消息

Create_WCS_Customer 消息是包含购物者的顾客信息的入站消息。一个后端应用程序生成此消息并将其发送到 WebSphere Commerce 入站消息队列，WebSphere Commerce 系统在此队列中接收消息。WebSphere Commerce 通过调用 UserRegistrationAdd 命令为新的购物者添加信息。

Create_WCS_OrderStatus 消息使用 XML 消息格式并遵循 Create_WCS_OrderStatus_20.dtd。

映射到此消息的数据库字段的字段格式类似于用于 Update_WCS_Customer 消息的格式。

Update_WCS_Customer 消息

Update_WCS_Customer 消息是包含购物者的顾客信息的入站消息。一个后端应用程序生成此消息并将其发送到 WebSphere Commerce 入站消息队列，WebSphere Commerce 系统在此队列中接收消息。WebSphere Commerce 通过调用 UserRegistrationUpdate 命令来更新顾客的信息。

下表描述 Update_NC_Customer 消息的格式。关于数据库列的描述，请随着链接查看与其关联的表。除非另外注明，所有的字段都是可选的。用于类型 INT、BIGINT、和 DOUBLE 的数据库字段的 XML 消息中的标记值长度应当分别为 CHAR(10)、CHAR(19) 和 CHAR(16)。

级别	字段名	注释	表名	列名
1	Registration	必填	N/A	N/A
1.1	LogonInfo	必填	N/A	N/A
1.1.1	LogonID	必填	USERREG	LOGONID
1.1.2	Password		USERREG	LOGONPASSWORD
1.1.3	VerifyPassword		N/A	N/A
1.2	StatusInfo		N/A	N/A
1.2.1	CustomerStatus		USERREG	STATUS
1.2.2	PasswordExpired		N/A	N/A
1.2.2.A1	value	属性	USERREG	PASSWORDEXPIRED
1.3	Challenge		N/A	N/A
1.3.1	Question		USERREG	CHALLENGEQUESTION
1.3.2	Answer		USERREG	CHALLENGEANSWER
2	AddressInfo		N/A	N/A
2.1	AddressID		ADDRESS	ADDRESS_ID
2.2	AddressNickName		ADDRESS	NICKNAME
2.3	AddressType		ADDRESS	ADDRESSTYPE
2.4	PersonName		N/A	N/A
2.4.1	Title		ADDRESS	PERSONTITLE
2.4.2	LastName		ADDRESS	LASTNAME
2.4.3	FirstName		ADDRESS	FIRSTNAME
2.4.4	MiddleName		ADDRESS	MIDDLENAME
2.5	Address		N/A	N/A
2.5.A1	primary	属性	ADDRESS	ISPRIMARY
2.5.A2	self	属性	ADDRESS	SELFADDRESS
2.5.1	AddressLine	首先出现	ADDRESS	ADDRESS1
2.5.1	AddressLine	第二出现	ADDRESS	ADDRESS2
2.5.1	AddressLine	第三出现	ADDRESS	ADDRESS3
2.5.2	City		ADDRESS	CITY
2.5.3	State		ADDRESS	STATE
2.5.4	ZipCode		ADDRESS	ZIPCODE
2.5.5	Country		ADDRESS	COUNTRY

级别	字段名	注释	表名	列名
2.6	ContactInfo		N/A	N/A
2.6.1	Telephone	首先出现	ADDRESS	PHONE1
2.6.1	Telephone	第二出现	ADDRESS	PHONE2
2.6.1.A1	type	属性	ADDRESS	PHONE1TYPE/ PHONE2TYPE
2.6.1.A2	publish	属性	ADDRESS	PUBLISH PHONE1 / PUBLISH PHONE2
2.6.2	BestCallingTime		ADDRESS	BESTCALLINGTIME
2.6.3	Fax	首先出现	ADDRESS	FAX1
2.6.3	Fax	第二出现	ADDRESS	FAX2
2.6.4	Email	首先出现	ADDRESS	EMAIL1
2.6.4	Email	第二出现	ADDRESS	EMAIL2
2.7	Billing		N/A	N/A
2.7.1	Code		ADDRESS	BILLINGCODE
2.7.2	CodeType		ADDRESS	BILLINGCODETYPE
2.8	PackageSuppression		ADDRESS	PACKAGESUPPRESSION
2.9	AddressField	首先出现	ADDRESS	FIELD1
2.9	AddressField	第二出现	ADDRESS	FIELD2
2.9	AddressField	第三出现	ADDRESS	FIELD3
3	Profile		N/A	N/A
3.A1	type	属性	USERS	PROFILETYPE
3.1	Personal		N/A	N/A
3.1.1	DistinguishedName		USERS	DN
3.1.2	PreferredCurrency		USERS	SETCURR
3.1.3	PreferredLanguage		USERS	LANGUAGE_ID
3.1.4	UserField	首先出现	USERS	FIELD1
3.1.4	UserField	第二出现	USERS	FIELD2
3.1.4	UserField	第三出现	USERS	FIELD3
3.1.5	DisplayName		USERPROF	DISPLAYNAME
3.1.6	Photo		USERPROF	PHOTO
3.1.7	PreferredMeasure		USERPROF	PREFERREDMEASURE
3.1.8	PreferredCommunication		USERPROF	PREFERREDCOMM
3.1.9	PreferredDelivery		USERPROF	PREFERREDDELIVERY
3.1.10	Description		USERPROF	DESCRIPTION
3.1.14	UserProfileField	首先出现	USERPROF	FIELD1
3.1.14	UserProfileField	第二出现	USERPROF	FIELD2
3.2	Business		N/A	N/A
3.2.1	BusinessTitle		ADDRESS	BUSINESSTITLE
3.2.2	Organization		N/A	N/A
3.2.2.1	OrganizationID		BUSPROF	ORG_ID
3.2.2.2	OrganizationName		ADDRESS	ORGNAME
3.2.2.3	OrganizationUnitId		BUSPROF	ORGUNIT_ID

级别	字段名	注释	表名	列名
3.2.2.4	OrganizationUnitName		ADDRESS	ORGUNITNAME
3.2.3	Employee		N/A	N/A
3.2.3.1	EmployeeID		BUSPROF	EMPLOYEEID
3.2.3.2	AlternateID		BUSPROF	ALTERNATEID
3.2.3.1	EmployeeType		BUSPROF	EMPLOYEEETYPE
3.2.4	OfficeAddress		ADDRESS	OFFICEADDRESS
3.2.6	DepartmentNumber		BUSPROF	DEPARTMENTNUM
3.2.7	Manager		BUSPROF	MANAGER
3.2.8	Secretary		BUSPROF	SECRETARY
3.3	Demographics		N/A	N/A
3.3.1	Age		USERDEMO	AGE
3.3.2	Gender		USERDEMO	GENDER
3.3.3	Income		USERDEMO	INCOME
3.3.3.A1	Currency	属性	USERDEMO	INCOMECURRENCY
3.3.4	MaritalStatus		USERDEMO	MARITALSTATUS
3.3.5	Children		USERDEMO	CHILDREN
3.3.6	Household		USERDEMO	HOUSEHOLD
3.3.7	CompanyName		USERDEMO	COMPANYNAME
3.3.8	Hobbies		USERDEMO	HOBBIES
3.3.9	OrderBefore		USERDEMO	ORDERBEFORE
3.3.10	TimeZone		USERDEMO	TIMEZONE
3.3.11	DemographicField	首先出现	USERDEMO	FIELD1
3.3.11	DemographicField	第二出现	USERDEMO	FIELD2
3.3.11	DemographicField	第三出现	USERDEMO	FIELD3
3.3.11	DemographicField	第四出现	USERDEMO	FIELD4
3.3.11	DemographicField	第五出现	USERDEMO	FIELD5
3.3.11	DemographicField	第六出现	USERDEMO	FIELD6
3.3.11	DemographicField	第七出现	USERDEMO	FIELD7
4	UserData		N/A	N/A
4.1	UserDataField		N/A	N/A

Update_WCS_ProductInventory 消息

Update_WCS_ProductInventory 消息是包含产品的库存信息的入站消息。一个后端应用程序生成此消息并将其发送到 WebSphere Commerce 入站消息队列，WebSphere Commerce 系统在此队列中接收消息。WebSphere Commerce 用新的库存信息更新 INVENTORY 表。

Update_WCS_ProductInventory 消息使用 XML 消息格式并遵循 Update_WCS_ProductInventory_20.dtd。

下表中描述了 XML 元素值的格式和源。关于数据库列的描述，请随着链接查看与其关联的表。除非另外注明，所有的字段都是可选的。用于类型 INT、BIGINT 和 DOUBLE 的数据库字段的 XML 消息中的标记值长度应当分别为 CHAR(10)、CHAR(19) 和 CHAR(16)。

级别	XML 元素	注释	表名	列名	注解
1	ProductNumber ByMerchant	必需 (请 参阅 注 解)	INVENTORY	CATENTRY_ID	仅在未使用 ProductSKU 时必需。使用 ProductSKU 时不得使用此属性。
2	MerchantID	必填	INVENTORY	STORE_ID	
3	Quantity	必填	INVENTORY	QUANTITY	
4	UserData		N/A	N/A	
5	FulfillmentCenterID		INVENTORY	FFMCENTER_ID	
6	ProductSKU	必需 (请 参阅 注 解)	CATENTRY	PARTNUMBER	仅在未使用 ProductNumberByMerchant 时必需。使用 ProductNumberByMerchant 时不得使用此属性。

行为:

- WebSphere Commerce 产品引用号 (引用 INVENTORY 表中的 CATENTRY_ID) 和商家引用号 (STORE_ID) 用于更新 INVENTORY 表中的行。
- 如果 INVENTORY 表中的该行不存在，将发生异常。

“产品价格更新”消息

“产品价格更新”消息是包含产品的价格信息的入站消息。后端应用程序生成此消息并将其发送到 WebSphere Commerce 入站消息队列。当 WebSphere Commerce 系统接收此消息时，它就运行 ProductOfferPriceUpdate 命令。

“产品价格更新”消息支持两种应用程序数据集合：“产品价格更新”版本 01 和“产品价格更新”版本 02。版本 02 包括版本 01 中数据的超集。

“产品价格更新”消息使用 WebSphere Commerce 消息格式，它由一组在缓冲区中按顺序排列的记录组成。下列数据描述了“产品价格更新”消息:

```
<ECEDOC>
<PROLOG>PRODUCT_PRICE_UPDATE_PROLOG_DATA</PROLOG>
<HEADER>
  <HDR010>PRODUCT_PRICE_UPDATE_HDR010_DATA</HDR010>
</HEADER>
</ECEDOC>
```

注意:

- 所有记录在缓冲区中按顺序排列。此处使用缩进是为了可读性；缓存区中并不会出现缩进。
- 在缓冲区中，数据段中的所有字段是左对齐的，右边用空格填充。

“产品价格更新”的数据段

- **PRODUCT_PRICE_UPDATE_PROLOG_DATA**
指定应用程序数据定义的消息类型。在此情况下，消息为“产品价格更新”。
- **PRODUCT_PRICE_UPDATE_HDR010_DATA**
指定“产品价格更新”消息中的商品或产品定价信息。

“产品数量更新”消息

“产品数量更新”消息是包含产品库存信息的入站消息。一个后端应用程序生成此消息并将其发送到 WebSphere Commerce 入站消息队列，WebSphere Commerce 系统在此队列中接收消息。

“产品数量更新”消息使用 WebSphere Commerce 消息格式，它由一组在数据缓冲区中按顺序排列的记录组成。下列数据描述了“产品数量更新”消息：

```
<ECEDOC>
<PROLOG>PRODUCT_QUANTITY_UPDATE_PROLOG_DATA</PROLOG>
<HEADER>
  <HDR010>PRODUCT_QUANTITY_UPDATE_HDR010_DATA</HDR010>
</HEADER>
</ECEDOC>
```

注意：

- 所有记录在缓冲区中按顺序排列。此处使用缩进是为了可读性；缓存区中并不会出现缩进。
- 在缓冲区中，数据段中的所有字段是左对齐的，右边用空格填充。

产品数量更新的数据段

- **PRODUCT_QUANTITY_UPDATE_PROLOG_DATA**
指定应用程序数据定义的消息类型。在此情况下，消息为“产品数量更新”。
- **PRODUCT_QUANTITY_UPDATE_HDR010_DATA**
指定“产品数量更新”消息中的产品信息。

“订单创建”消息

“订单创建”消息是包含已完成订单的订单详细信息的出站消息。WebSphere Commerce 系统生成此消息并将其数据发送至出站消息队列，后端系统在该队列接收消息。此应用程序获得该消息后继续进行完成订单的指定任务所需的任何后端商务过程。使用“订单创建”实现 WebSphere Commerce 系统开始所用订单的订单处理，但还需要另一独立系统完成某些附加或后端工作。

“订单创建”消息使用旧的消息格式，由一组在缓冲区中按顺序排列的记录组成。该消息包含订单、购物者、开票、商家和装运的信息。以下数据描述了“订单创建”消息：

```

<ECEDOC>
<PROLOG>ORDER_CREATE_PROLOG_DATA</PROLOG>
<HEADER>
  <HDR010>ORDER_CREATE_HDR010_DATA</HDR010>
  <HDR020>ORDER_CREATE_HDR020_DATA</HDR020>
  <HDR030>ORDER_CREATE_HDR030_DATA</HDR030>
  <HDR040>ORDER_CREATE_HDR040_DATA</HDR040>
  <USRLST>
    <DATUSR>DATUSR_DATA</DATUSR>
    .
    .DATUSR repeated loop
    .
    <DATUSR>DATUSR_DATA</DATUSR>
  </USRLST>
</HEADER>
<ITMLST>
  <ITMDAT>
    <ITM010>ORDER_CREATE_ITM010_DATA</ITM010>
    <USRLST>
      <DATUSR>DATUSR_DATA</DATUSR>
      .
      .DATUSR repeated loop
      .
      <DATUSR>DATUSR_DATA</DATUSR>
    </USRLST>
  </ITMDAT>
  .
  .ITEM repeated loop
  .
  <ITMDAT>
    <ITM010>ORDER_CREATE_ITM010_DATA</ITM010>
    <USRLST>
      <DATUSR>DATUSR_DATA</DATUSR>
      .
      .DATUSR repeated loop
      .
      <DATUSR>DATUSR_DATA</DATUSR>
    </USRLST>    </ITMDAT>
</ITMLST>
</ECEDOC>

```

注意:

- 所有记录在缓冲区中按顺序排列。此处使用缩进是为了可读性；缓存区中并不会出现缩进。
- 在缓冲区中，数据段中的所有字段是左对齐的，右边用空格填充。

订单创建的数据段

- ORDER_CREATE_PROLOG_DATA
指定应用程序数据定义的消息类型。在此情况下，消息为“订单创建”。
- ORDER_CREATE_HDR010_DATA
指定“订单创建”消息中的订单信息。
- ORDER_CREATE_HDR020_DATA
指定“订单创建”消息中的购物者信息。
- ORDER_CREATE_HDR030_DATA
指定“订单创建”消息中的开票信息。
- ORDER_CREATE_HDR040_DATA
指定“订单创建”消息中的商家信息。
- DATUSR_DATA
指定要添加到“订单创建”消息的可选信息。DATUSR_DATA 在此消息的 <HDR> 和 <ITM> 部分出现。
- ORDER_CREATE_ITM010_DATA
指定“订单创建”消息中的商品或产品装运信息。

顾客新消息

“顾客新建”消息是一种旧的格式的人站消息，它包含购物者的顾客信息。一个后端应用程序生成此消息并将其发送到 WebSphere Commerce 人站消息队列，WebSphere Commerce 系统在此队列中接收消息。WebSphere Commerce 通过调用 UserRegistrationAdd 控制器命令更新新购物者的信息。

以下数据描述“顾客新建”消息：

```
<ECEDOC>
<PROLOG>CUSTOMER_UPDATE_PROLOG_DATA</PROLOG>
<HEADER>
  <HDR010>CUSTOMER_UPDATE_HDR010_DATA</HDR010>
  <USRLST>
    <DATUSR>DATUSR_DATA</DATUSR>
    .
    .DATUSR repeated loop
    .
    <DATUSR>DATUSR_DATA</DATUSR>
  </USRLST>
</HEADER>
</ECEDOC>
```

注意：

- 所有记录在缓冲区中按顺序排列。此处使用缩进是为了可读性；缓存区中并不会出现缩进。
- 在缓冲区中，数据段中的所有字段是左对齐的，右边用空格填充。

顾客新建的数据段

- **CUSTOMER_NEW_PROLOG_DATA**
指定应用程序数据定义的消息类型。在这种情况下，消息为“顾客新建”。
- **CUSTOMER_NEW_HDR010_DATA**
在“顾客新建”消息中指定购物者信息。
- **DATUSR_DATA**
指定要添加至“顾客新建”消息的可选信息。DATUSR_DATA 在此消息的 <HDR> 部分中出现。

顾客更新消息

“顾客更新”消息是包含购物者的顾客信息的入站消息。一个后端应用程序生成此消息并将其发送到 WebSphere Commerce 入站消息队列，WebSphere Commerce 系统在此队列中接收消息。WebSphere Commerce 通过调用 UserRegistrationUpdate 命令来更新注册购物者的信息。

“顾客更新”消息使用 WebSphere Commerce 消息格式，它由一组在缓冲区中彼此连续跟随的记录组成。以下数据描述“顾客更新”消息：

```
<ECEDOC>
<PROLOG>CUSTOMER_UPDATE_PROLOG_DATA</PROLOG>
<HEADER>
  <HDR010>CUSTOMER_UPDATE_HDR010_DATA</HDR010>
  <USRLST>
    <DATUSR>DATUSR_DATA</DATUSR>
    .
    .DATUSR repeated loop
    .
    <DATUSR>DATUSR_DATA</DATUSR>
  </USRLST>
</HEADER>
</ECEDOC>
```

注意:

- 所有记录在缓冲区中按顺序排列。此处使用缩进是为了可读性；缓存区中并不会出现缩进。
- 在缓冲区中，数据段中的所有字段是左对齐的，右边用空格填充。

顾客更新的数据段

- **CUSTOMER_UPDATE_PROLOG_DATA**
指定应用程序数据定义的消息类型。在这种情况下，消息是“顾客更新”。
- **CUSTOMER_UPDATE_HDR010_DATA**
在“顾客更新”消息中指订购者信息。
- **DATUSR_DATA**
指定要添加至“顾客更新”消息的可选信息。DATUSR_DATA 在此消息的 <HDR> 部分中出现。

Update_NC_Customer 消息

Update_NC_Customer 信息是包含购物者的顾客信息的入站消息。一个后端应用程序生成此消息并将其发送到 WebSphere Commerce 入站消息队列，WebSphere Commerce 系统在此队列中接收消息。WebSphere Commerce 通过调用 UserRegistrationUpdate 命令来更新注册购物者的信息。由于此消息调用相同的 WebSphere Commerce 控制器命令，它与 Create_NC_Customer 消息是相同的。

Update_NC_Customer 消息使用 XML 消息格式并遵循 Update_NC_Customer_10.dtd 文件。

下表描述 Update_NC_Customer 消息的格式。关于数据库列的描述，请随着链接查看与其关联的表。除非另外注明，所有的字段都是可选的。用于类型 INT、BIGINT 和 DOUBLE 的数据库字段的 XML 消息中的标记值长度应当分别为 CHAR(10)、CHAR(19) 和 CHAR(16)。

级别	字段名	注释	表名	列名
1	LoginInfo	必填	N/A	N/A
1.1	LoginID	必填	USERREG	LOGONID
1.2	Password		USERREG	LOGONPASSWORD
1.3	VerifyPassword		USERREG	LOGONPASSWORD
2	MerchantID		N/A	N/A
2.A1	type	属性	N/A	N/A
3	MethodOfCommunication		USERPROF	PREFERREDCOMM
4	ChallengeQuestion		USERREG	CHALLENGEQUESTION
5	ChallengeAnswer		USERREG	CHALLENGEANSWER
6	ShopperField	首先出现	USERS	FIELD1
6	ShopperField	第二出现	USERS	FIELD2
6	ShopperField	第三出现	USERS	FIELD3
7	ContactPersonName		N/A	N/A
7.1	Title		ADDRESS	PERSONTITLE
7.2	FullName		N/A	N/A
7.3	LastName		ADDRESS	LASTNAME
7.4	FirstName		ADDRESS	FIRSTNAME
7.5	MiddleName		ADDRESS	MIDDLENAME
7.6	AlternateName		ADDRESS	NICKNAME
8	RepCompany		ADDRESS	ORGNAME
9	Address		N/A	N/A
9.1	AddressLine	首先出现	ADDRESS	ADDRESS1
9.1	AddressLine	第二出现	ADDRESS	ADDRESS2
9.1	AddressLine	第三出现	ADDRESS	ADDRESS3
9.2	City		ADDRESS	CITY
9.3	State		ADDRESS	STATE
9.4	Zip		ADDRESS	ZIPCODE
9.5	Country		ADDRESS	COUNTRY

级别	字段名	注释	表名	列名
10	ContactInfo		N/A	N/A
10.1	Telephone	首先出现	ADDRESS	PHONE1
10.1	Telephone	第二出现	ADDRESS	PHONE2
10.2	Email	首先出现	ADDRESS	EMAIL1
10.2	Email	第二出现	ADDRESS	EMAIL2
10.3	Fax		ADDRESS	FAX1
11	DayPhoneInfo		N/A	N/A
11.1	PhoneInfo		N/A	N/A
11.1.A1	type	属性	ADDRESS	PHONE1TYPE
11.1.A2	isListed	属性	ADDRESS	PUBLISHPHONE1
12	EveningPhoneInfo		N/A	N/A
12.1	PhoneInfo		N/A	N/A
12.1.A1	type	属性	ADDRESS	PHONE2TYPE
12.1.A2	isListed	属性	ADDRESS	PUBLISHPHONE2
13	BestTimeToCall		ADDRESS	BESTCALLINGTIME
14	IncludePackageInsert		ADDRESS	PACKAGESUPPRESSION
15	AddressOptField	首先出现	ADDRESS	FIELD1
15	AddressOptField	第二出现	ADDRESS	FIELD2
15	AddressOptField	第三出现	ADDRESS	FIELD3
16	Gender		N/A	N/A
16.A1	value	属性	USERDEMO	GENDER
17	AgeGroup		USERDEMO	AGE
18	IncomeGroup		USERDEMO	INCOME
19	MaritalStatus		USERDEMO	MARITALSTATUS
20	NumberOfChildren		USERDEMO	CHILDREN
21	NumberInHouse		USERDEMO	HOUSEHOLD
22	WorkCompany		USERDEMO	COMPANYNAME
23	Interests		USERDEMO	HOBBIES
24	PreviousOrder		USERDEMO	ORDERBEFORE
25	Demographics	首先出现	USERDEMO	FIELD1
25	Demographics	第二出现	USERDEMO	FIELD2
25	Demographics	第三出现	USERDEMO	FIELD3
25	Demographics	第四出现	USERDEMO	FIELD4
25	Demographics	第五出现	USERDEMO	FIELD5
25	Demographics	第六出现	USERDEMO	FIELD6
25	Demographics	第七出现	USERDEMO	FIELD7
26	UserData		N/A	N/A
26.1	UserDataField		N/A	N/A

Update_NC_OrderStatus 消息

Update_NC_OrderStatus 消息是一个入站消息，它包含有关 WebSphere Commerce 订单的状态信息。一个后端应用程序生成此消息并将其发送到 WebSphere Commerce 入站消息队列，WebSphere Commerce 系统在此队列中接收消息。WebSphere Commerce 使用 OrderStatus 命令用新的订单状态信息更新表 ORDSTAT 和 ORDISTAT。

update_NC_OrderStatus 消息使用 XML 消息格式并遵循 update_NC_OrderStatus_10.dtd。

下表描述 Update_NC_OrderStatus 消息的格式。关于数据库列的描述，请随着链接查看与其关联的表。除非另外注明，所有的字段都是可选的。用于类型 INT、BIGINT 和 DOUBLE 的数据库字段的 XML 消息中的标记值长度应当分别为 CHAR(10)、CHAR(19) 和 CHAR(16)。

级别	字段名	注释	表名	列名	注解
1	OrderStatus Header		N/A	N/A	N/A
1.1	OrderNumber ByBackend		ORDSTAT	OSMORDER	由后端系统生成的订单引用号。
1.2	OrderNumber ByNC	必填	ORDSTAT	ORDERS_ID	WebSphere Commerce 订单引用号。这是引用 ORDERS 表中 ORDERS_ID 列的外键。
1.3	OrderNumber ByBuyer		N/A	N/A	N/A
1.4	TotalPriceInfo		N/A	N/A	N/A
1.4.A1	currency	属性	ORDSTAT	OSCPCUR	表示价格的货币。货币格式必须遵守 ISO 4217 标准。
1.4.1	TotalNetPrice		ORDSTAT	OSPRTOT	订单的产品价格总额。
1.4.2	TotalTaxPrice		ORDSTAT	OSTXTOT	订单总税款。
1.4.3	TotalShippingPrice		ORDSTAT	OSSHTOT	订单装运费用总额。
1.4.4	TotalTaxOn ShippingPrice		ORDSTAT	OSSHTXTOT	订单装运费用税款总额。
1.5	RequisitionerID		N/A	N/A	N/A
1.5.A1	type	属性	N/A	N/A	N/A
1.6	状态		ORDSTAT	OSSTATUS	订单状态
1.7	DateTime Reference		N/A	N/A	N/A
1.7.1	PlacedDate		ORDSTAT	OSPLTIME	下订单的时间戳记
1.7.2	PlacedTime		N/A	N/A	N/A
1.7.3	LastUpdateDate		N/A	N/A	N/A

级别	字段名	注释	表名	列名	注解
1.7.4	LastUpdateTime		ORDSTAT	OSUPDTIME	订单最后更新的时间戳记
1.8	ShipDate Reference		N/A	N/A	N/A
1.8.1	Requested ShipDate		ORDSTAT	OSRSTIME	请求的装运时间戳记
1.8.2	Scheduled ShipDate		ORDSTAT	OSSSTIME	已调度的装运时间戳记。
1.8.3	Actual ShipDate		ORDSTAT	OSASTIME	实际的装运时间戳记。
1.9	CustomerField	首先出现。	ORDSTAT	FIELD1	保留供定制使用。
1.9	CustomerField	第二出现。	ORDSTAT	FIELD2	保留供定制使用。
1.9	CustomerField	第三出现。	ORDSTAT	FIELD3	保留供定制使用。
1.10	UserData		N/A	N/A	N/A
2	OrderStatusItem		N/A	N/A	N/A
2.1	OrderNumber ByBackend		ORDISTAT	OSMORDER	后端系统订单号码。
2.2	ItemNumber ByBackend		ORDISTAT	OIMITEM	后端系统订购商品号。
2.3	OrderNumber ByNC	必填	ORDISTAT	ORDERS_ID	WebSphere Commerce 订单引用号。这是引用 ORDERS 表中 ORDERS_ID 列的外键。
2.4	ItemNumber ByNC		ORDISTAT	订单 ITMES_ID	WebSphere Commerce 商品引用号。这是外键, 该键引用 ORDERITEMS 表中的列 ORDERITEMS_ID。
2.5	Quantity		ORDISTAT	OIQTCONFIRM	确认的商品数量
2.6	ItemUnitPrice		ORDISTAT	OIUNPRC	单价
2.7	TotalPriceInfo		N/A	N/A	N/A
2.7.A1	currency	属性	ORDISTAT	OICPCUR	用于表示商品价格的货币。货币格式必须遵守 ISO 4217 标准。
2.7.1	TotalNetPrice		ORDISTAT	OIPRTOT	商品总价格。
2.7.2	TotalTaxPrice		ORDISTAT	OITXTOT	商品的总税款。
2.7.3	TotalShippingPrice		ORDISTAT	OISHTOT	商品的总装运费用。

级别	字段名	注释	表名	列名	注解
2.7.4	TotalTaxOnShippingPrice		ORDISTAT	OISHTXTOT	商品的总装运费用税款。
2.8	状态		ORDISTAT	OISTATUS	订购商品状态。
2.9	DateTimeReference		N/A	N/A	N/A
2.9.1	PlacedDate		ORDISTAT	OIPLTIME	订购商品的时间戳记
2.9.2	PlacedTime		N/A	N/A	N/A
2.9.3	LastUpdateDate		N/A	N/A	N/A
2.9.4	LastUpdateTime		N/A	N/A	N/A
2.10	ShipDateReference		N/A	N/A	N/A
2.10.1	RequestedShipDate		ORDISTAT	OIRSTIME	请求的装运时间戳记
2.10.2	ScheduledShipDate		ORDISTAT	OISSDATE	已调度的装运时间戳记。
2.10.3	ActualShipDate		ORDISTAT	OIASTIME	实际的装运时间戳记。
2.11	Instruction		ORDISTAT	OICMNT	购物者关于订购商品的注释。例如，购物者可以在订购的礼物上包含一条消息。
2.12	CustomerField	首先出现。	N/A	N/A	保留供定制使用。
2.12	CustomerField	第二出现。	N/A	N/A	保留供定制使用。
2.12	CustomerField	第三出现。	N/A	N/A	保留供定制使用。
2.13	UserData		N/A	N/A	N/A

Create_NC_Customer 消息

Create_NC_Customer 消息是包含购物者的顾客信息的入站消息。一个后端应用程序生成此消息并将其发送到 WebSphere Commerce 入站消息队列，WebSphere Commerce 系统在此队列中接收消息。WebSphere Commerce 通过调用 UserRegistrationAdd 命令为新的购物者更新信息。

Create_NC_OrderStatus 消息使用 XML 消息格式并遵循 Create_NC_OrderStatus_10.dtd。

映射到此消息的数据库字段的字段格式类似于用于 Update_NC_Customer 消息的格式。

Update_NC_ProductInventory 消息

Update_NC_ProductInventory 消息是包含产品的库存信息的入站消息。一个后端应用程序生成此消息并将其发送到 WebSphere Commerce 入站消息队列，WebSphere Commerce 系统在此队列中接收消息。WebSphere Commerce 用新的库存信息更新 INVENTORY 表。

Update_NC_ProductInventory 消息使用 XML 消息格式并遵循 Update_NC_ProductInventory_10.dtd。

注意：建议您使用 Update_WCS_ProductInventory 消息来代替 Update_NC_ProductInventory，因为它是一个改进版本。当使用 Update_WCS_ProductInventory 消息时，ProductNumberByMerchant 字段可以由 ProductSKU 替换（可选）。

下表中描述了 XML 元素值的格式和源。关于数据库列的描述，请随着链接查看与其关联的表。除非另外注明，所有的字段都是可选的。用于类型 INT、BIGINT 和 DOUBLE 的数据库字段的 XML 消息中的标记值长度应当分别为 CHAR(10)、CHAR(19) 和 CHAR(16)。

级别	XML 元素	注释	表名	列名
1	ProductNumber ByMerchant	必填	INVENTORY	CATENTRY_ID
2	MerchantID	必填	INVENTORY	STORE_ID
3	Quantity	必填	INVENTORY	QUANTITY
4	UserData		N/A	N/A

行为:

- WebSphere Commerce 产品引用号（引用 INVENTORY 表中的 CATENTRY_ID）和商家引用号（STORE_ID）用于更新 INVENTORY 表中的行。
- 如果 INVENTORY 表中的该行不存在，将发生异常。

Update_NC_ProductPrice 消息

Update_NC_ProductPrice 消息是包含产品价格信息的入站消息。一个后端应用程序生成此消息并将其发送到 WebSphere Commerce 入站消息队列，WebSphere Commerce 系统在此队列中接收消息。然后系统调用 ProductOfferPriceUpdate 命令，该命令将用新的价格信息更新 OFFERPRICE 表。

Update_NC_ProductPrice 消息使用 XML 消息格式并遵循 Update_NC_ProductPrice_10.dtd。

下表中描述了 XML 元素值的格式和源。关于数据库列的描述，请随着链接查看与其关联的表。除非另外注明，所有的字段都是可选的。对于 INT 类型的数据库字段，消息元素应当为 CHAR (12)。

级别	XML 元素	注释	表名	列名
1	ProductNumber ByMerchant	必填	CATENTRY	PARTNUMBER

级别	XML 元素	注释	表名	列名
2	MerchantID	必填	CATENTRY	MEMBER_ID
3	PriceGroupInfo		N/A	N/A
3.1	Precedence		OFFER	PRECEDENCE
3.2	Requisitioner GroupID		OFFER	TRADEPOSCN_ID
4	Currency	必填	OFFERPRICE	CURRENCY
5	ItemUnitPrice		OFFERPRICE	PRICE
6	Start Timestamp		OFFER	STARTDATE
7	End Timestamp		OFFER	ENDDATE
8	PriceCustom Field		N/A	N/A
9	UserData		N/A	N/A

Report_NC_PurchaseOrder 消息

Report_NC_PurchaseOrder 消息是一条包含已完成订单的订单详细信息的出站消息。WebSphere Commerce 出站消息传递系统使用 OrderCreateXML.jsp 组合模板生成此消息并将其数据发送至出站消息队列，后端系统在该队列接收该消息。此应用程序获得该消息后继续进行完成订单的指定任务所需的任何后端商务过程。使用 Report_NC_PurchaseOrder 完成 WebSphere Commerce 系统开始的订单的订单处理，但仍需要另一系统来完成某些附加或后端工作。

Report_NC_PurchaseOrder 消息使用 XML 消息格式并遵循 Report_NC_PO_10.dtd。

DTD 文件由一系列 XML 元素组成。该消息包含订单、购物者、开票、商家和装运的信息。以下数据描述了 Report_NC_PO_10.dtd，它用于 Report_NC_PurchaseOrder 消息：

下表中描述了 XML 元素值的格式和源。关于数据库列的描述，请随着链接查看与其关联的表。除非另外注明，所有的字段都是可选的。用于类型 INT、BIGINT 和 DOUBLE 的数据库字段的 XML 消息中的标记值长度应当分别为 CHAR(10)、CHAR(19) 和 CHAR(16)。

级别	XML 元素	注释	表名	列名
1	ReportPO Header	必填	N/A	N/A
1.1	OrderNumber ByBuyer		N/A	N/A
1.2	OrderNumber ByMerchant		ORDERS	ORMORDER
1.3	OrderNumber ByNC		ORDERS	ORDERS_ID

级别	XML 元素	注释	表名	列名
1.4	DateTime Reference		N/A	N/A
1.4.1	PlacedDate		ORDERS	TIMEPLACED
1.4.2	PlacedTime		ORDERS	TIMEPLACED
1.4.3	LastUpdate Date		N/A	N/A
1.4.4	LatestUpdate Time		N/A	N/A
1.5	TotalPriceInfo	必填	N/A	N/A
1.5.A1	currency	属性	ORDERS	CURRENCY
1.5.1	TotalNet Price	必填	ORDERS	TOTALPRODUCT
1.5.2	TaxInfo		N/A	N/A
1.5.2.1	Monetary Amount	必填	ORDERS	TOTALTAX
1.5.2.1.A1	currency	属性	ORDERS	CURRENCY
1.5.2.2	TaxType		N/A	N/A
1.5.2.3	Percentage		N/A	N/A
1.5.2.4	TaxExemption StatusType		N/A	N/A
1.5.2.5	TaxExemption Number		N/A	N/A
1.5.2.6	TaxJurisdiction Code		N/A	N/A
1.5.2.7	TaxJurisdiction CodeType		N/A	N/A
1.5.3	Total ShippingPrice	必填	ORDERS	TOTALSHIPPING
1.5.4	TotalTaxOn ShippingPrice	必填	ORDERS	TOTALTAXSHIPPING
1.6	Instruction		N/A	N/A
1.7	ShipStatus		ORDERS	STATUS
1.8	BillToInfo		N/A	N/A
1.8.1	OrgName		N/A	N/A
1.8.2	Address	必填	N/A	N/A
1.8.2.1	AddressLine	必需; 第一次重复出现	ADDRESS	ADDRESS1
1.8.2.1	AddressLine	第二次重复出现	ADDRESS	ADDRESS2
1.8.2.1	AddressLine	第三次重复出现	ADDRESS	ADDRESS3
1.8.2.2	City	必填	ADDRESS	CITY
1.8.2.3	State	必填	ADDRESS	STATE
1.8.2.4	Zip	必填	ADDRESS	ZIPCODE
1.8.2.5	Country	必填	ADDRESS	COUNTRY

级别	XML 元素	注释	表名	列名
1.8.3	ContactPerson 名称	必填	N/A	N/A
1.8.3.1	Title		N/A	N/A
1.8.3.2	FullName		N/A	N/A
1.8.3.3	LastName	必填	ADDRESS	LASTNAME
1.8.3.4	FirstName		ADDRESS	FIRSTNAME
1.8.3.5	MiddleName		ADDRESS	MIDDLENAME
1.8.3.6	AlternateName		ADDRESS	NICKNAME
1.8.4	ContactInfo	必填	N/A	N/A
1.8.4.1	Telephone	第一次重复出现	ADDRESS	PHONE1
1.8.4.1	Telephone	第二次重复出现	ADDRESS	PHONE2
1.8.4.2	Email	第一次重复出现	ADDRESS	EMAIL1
1.8.4.2	Email	第二次重复出现	ADDRESS	EMAIL2
1.8.4.3	Fax		ADDRESS	FAX1
1.9	MerchantInfo		N/A	N/A
1.9.1	OrgName	必填	STOREENTDS	DISPLAYNAME
1.9.2	OrgID	第一次重复出现	N/A	N/A
1.9.2.A1	type	属性	ORDERS	STOREENT_ID
1.9.3	Address	必填	N/A	N/A
1.9.3.1	AddressLine	必需; 第一次重复出现	STADDRESS	ADDRESS1
1.9.3.1	AddressLine	第二次重复出现	STADDRESS	ADDRESS2
1.9.3.1	AddressLine	第三次重复出现	STADDRESS	ADDRESS3
1.9.3.2	City	必填	STADDRESS	CITY
1.9.3.3	State	必填	STADDRESS	STATE
1.9.3.4	Zip	必填	STADDRESS	ZIPCODE
1.9.3.5	Country	必填	STADDRESS	COUNTRY
1.9.4	URL		N/A	N/A
1.9.5	Telephone		STADDRESS	PHONE1
1.9.6	ContactPerson 名称		N/A	N/A
1.9.6.1	Title		STADDRESS	PERSONTITLE
1.9.6.2	FullName		N/A	N/A
1.9.6.3	LastName	必填	STADDRESS	LASTNAME
1.9.6.4	FirstName		STADDRESS	FIRSTNAME
1.9.6.5	MiddleName		STADDRESS	MIDDLENAME
1.9.6.6	AlternateName		N/A	N/A
1.9.7	ContactInfo	必填	N/A	N/A
1.9.7.1	Telephone	第一次重复出现	STADDRESS	PHONE1
1.9.7.1	Telephone	第二次重复出现	STADDRESS	PHONE2
1.9.7.2	Email	第一次重复出现	STADDRESS	EMAIL1

级别	XML 元素	注释	表名	列名
1.9.7.3	Email	第二次重复出现	STADDRESS	EMAIL2
1.9.7.4	Fax		STADDRESS	FAX1
1.10	BuyOrgInfo		N/A	N/A
1.10.1	OrgName	必填	ADDRESS	ORGNAME
1.10.2	OrgID	第一次重复出现	N/A	N/A
1.10.2.A1	type	属性	N/A	N/A
1.10.3	Address		N/A	N/A
1.10.3.1	AddressLine	必需; 第一次重复出现	N/A	N/A
1.10.3.1	AddressLine	第二次重复出现	N/A	N/A
1.10.3.1	AddressLine	第三次重复出现	N/A	N/A
1.10.3.2	City	必填	N/A	N/A
1.10.3.3	State	必填	N/A	N/A
1.10.3.4	Zip	必填	N/A	N/A
1.10.3.5	Country	必填	N/A	N/A
1.10.4	URL		N/A	N/A
1.10.5	ContactInfo		N/A	N/A
1.10.5.1	Telephone	第一次重复出现	N/A	N/A
1.10.5.1	Telephone	第二次重复出现	N/A	N/A
1.10.5.2	Email	第一次重复出现	N/A	N/A
1.10.5.2	Email	第二次重复出现	N/A	N/A
1.10.5.3	Fax		N/A	N/A
1.11	ShipToInfo		N/A	N/A
1.11.1	ContactPerson 名称	必填	N/A	N/A
1.11.1.1	Title		N/A	N/A
1.11.1.2	FullName	必填	N/A	N/A
1.11.1.3	LastName	必填	N/A	N/A
1.11.1.4	FirstName		N/A	N/A
1.11.1.5	MiddleName		N/A	N/A
1.11.1.6	AlternateName		N/A	N/A
1.11.2	OfficeAddress Line		N/A	N/A
1.11.3	Address	必填	N/A	N/A
1.11.3.1	AddressLine	必需; 第一次重复出现	N/A	N/A
1.11.3.1	AddressLine	第二次重复出现	N/A	N/A
1.11.3.1	AddressLine	第三次重复出现	N/A	N/A
1.11.3.2	City	必填	N/A	N/A
1.11.3.3	State	必填	N/A	N/A
1.11.3.4	Zip	必填	N/A	N/A

级别	XML 元素	注释	表名	列名
1.11.3.5	Country	必填	N/A	N/A
1.11.4	ContactInfo		N/A	N/A
1.11.4.1	Telephone	第一次重复出现	N/A	N/A
1.11.4.1	Telephone	第二次重复出现	N/A	N/A
1.11.4.2	Email	第一次重复出现	N/A	N/A
1.11.4.2	Email	第二次重复出现	N/A	N/A
1.11.4.3	Fax		N/A	N/A
1.11.5	注释		N/A	N/A
1.12	Requisitioner Info		N/A	N/A
1.12.1	RequisitionerID	第一次重复出现	ORDERS	MEMBER_ID
1.12.1.A1	type	必填; 属性	N/A	N/A
1.12.1	RequisitionerID	第二次重复出现	USERREG	LOGONID
1.12.1.A2	type	必填; 属性	N/A	N/A
1.12.2	Requisitioner Group		N/A	N/A
1.12.3	Address		N/A	N/A
1.12.3.1	AddressLine	必需; 第一次重复出现	ADDRESS	ADDRESS1
1.12.3.1	AddressLine	第二次重复出现	ADDRESS	ADDRESS2
1.12.3.1	AddressLine	第三次重复出现	ADDRESS	ADDRESS3
1.12.3.2	City	必填	ADDRESS	CITY
1.12.3.3	State	必填	ADDRESS	STATE
1.12.3.4	Zip	必填	ADDRESS	ZIPCODE
1.12.3.5	Country	必填	ADDRESS	COUNTRY
1.12.4	ContactPerson 名称		N/A	N/A
1.12.4.1	Title		ADDRESS	PERSONTITLE
1.12.4.2	FullName	必填	N/A	N/A
1.12.4.3	LastName	必填	ADDRESS	LASTNAME
1.12.4.4	FirstName		ADDRESS	FIRSTNAME
1.12.4.5	MiddleName		ADDRESS	MIDDLENAME
1.12.4.6	AlternateName		N/A	N/A
1.12.5	ContactInfo	必填	N/A	N/A
1.12.5.1	Telephone	第一次重复出现	ADDRESS	PHONE1
1.12.5.1	Telephone	第二次重复出现	ADDRESS	PHONE2
1.12.5.2	Email	第一次重复出现	ADDRESS	EMAIL1
1.12.5.2	Email	第二次重复出现	ADDRESS	EMAIL2
1.12.5.3	Fax		ADDRESS	FAX1
1.13	ShipDate Reference		N/A	N/A

级别	XML 元素	注释	表名	列名
1.13.1	Requested ShipDate		N/A	N/A
1.13.2	Scheduled ShipDate		N/A	N/A
1.13.3	Actual ShipDate		N/A	N/A
1.14	PCardInfo		N/A	N/A
1.14.1	Monetary Amount		ORDPAYMTHD	MAXAMOUNT
1.14.1.A1	currency	属性	ORDERS	CURRENCY
1.14.2	CardType		ORDPAYMTHD	PAYMETHOD
1.14.3	CardNumber		ORDPAYMTHD	PAYDEVICE
1.14.4	Expiration Date		ORDPAYMTHD	ENDDATE
1.14.5	IssueDate		ORDPAYMTHD	STARTDATE
1.14.6	Credit Authorization Number		N/A	N/A
1.14.7	Customer Reference Number		N/A	N/A
1.15	Shipping CarrierInfo		N/A	N/A
1.15.1	Carrier		N/A	N/A
1.15.2	Method		N/A	N/A
1.16	BuyOrg Accounting Detail		N/A	N/A
1.16.1	Percentage		N/A	N/A
1.16.2	Monetary Amount		N/A	N/A
1.16.2.A1	currency	属性	N/A	N/A
1.16.3	BudgetCode	必填	N/A	N/A
1.16.4	Description		N/A	N/A
1.16.5	Calculation Code	必填	N/A	N/A
1.17.2	OrderCustomer Field	第一次重复出现	ORDERS	FIELD1
1.17.2	OrderCustomer Field	第一次重复出现	ORDERS	FIELD2
1.17.2	OrderCustomer Field	第二次重复出现	ORDERS	FIELD3
1.18	UserData		N/A	N/A

级别	XML 元素	注释	表名	列名
2	ReportPOItem	必需; 第一次重复出现	N/A	N/A
2.1	ItemLine Number		N/A	N/A
2.2	ItemNumber ByNC		ORDERITEMS	ORDERITEMS_ID
2.3	ProductNumber ByBuyer		N/A	N/A
2.4	ProductNumber ByMerchant	必填	CATENTRY	PARTNUMBER
2.5	Manufacturer 名称		N/A	N/A
2.6	Manufacturer URL		N/A	N/A
2.7	Manufacturer PartNumber		N/A	N/A
2.8	ItemUnitPrice	必填	ORDERITEMS	PRICE
2.8.A1	currency	属性	ORDERITEMS	CURRENCY
2.9	TaxInfo	第一次重复出现	N/A	N/A
2.9.1	Monetary Amount	第一次重复出现	N/A	N/A
2.9.1.A1	currency	属性	N/A	N/A
2.9.2	TaxType		N/A	N/A
2.9.3	Percentage		N/A	N/A
2.9.4	TaxExemption StatusType		N/A	N/A
2.9.5	TaxExemption Number		N/A	N/A
2.9.6	TaxJurisdiction Code		N/A	N/A
2.9.7	TaxJurisdiction CodeType		N/A	N/A
2.10	ItemProduct Quantity	必填	ORDERITEMS	QUANTITY
2.11	UnitOf Measure		N/A	N/A
2.12	Classification		N/A	N/A
2.13	ItemProduct ShortDescription		CATENTDESC	SHORTDESCRIPTION
2.14	Instruction		N/A	N/A
2.15	ShipToInfo		N/A	N/A
2.15.1	ContactPerson 名称	必填	N/A	N/A
2.15.1.1	Title		N/A	N/A

级别	XML 元素	注释	表名	列名
2.15.1.2	FullName	必需	N/A	N/A
2.15.1.3	LastName	必填	ADDRESS	LASTNAME
2.15.1.4	FirstName		ADDRESS	FIRSTNAME
2.15.1.5	MiddleName		ADDRESS	MIDDLENAME
2.15.1.6	AlternateName		ADDRESS	NICKNAME
2.15.2	Office AddressLine	此版本不支持	N/A	N/A
2.15.3	Address		N/A	N/A
2.15.3.1	AddressLine	必需; 第一次重复出现	ADDRESS	ADDRESS1
2.15.3.1	AddressLine	第二次重复出现	ADDRESS	ADDRESS2
2.15.3.1	AddressLine	第三次重复出现	ADDRESS	ADDRESS3
2.15.3.2	City	必填	ADDRESS	CITY
2.15.3.3	State	必填	ADDRESS	STATE
2.15.3.4	Zip	必填	ADDRESS	ZIPCODE
2.15.3.5	Country	必填	ADDRESS	COUNTRY
2.15.4	ContactInfo	必填	N/A	N/A
2.15.4.1	Telephone	第一次重复出现	ADDRESS	PHONE1
2.15.4.1	Telephone	第二次重复出现	ADDRESS	PHONE2
2.15.4.2	Email	第一次重复出现	ADDRESS	EMAIL1
2.15.4.2	Email	第二次重复出现	ADDRESS	EMAIL2
2.15.4.3	Fax		ADDRESS	FAX1
2.15.5	注释		ORDERITEMS	COMMENTS
2.16	Shipping CarrierInfo		N/A	N/A
2.16.1	Carrier	必填	SHIPMODE	CARRIER
2.16.2	Method	必填	SHIPMODE	CODE
2.17	ShipStatus		ORDERITEMS	STATUS
2.18	DateTime Reference	必填	N/A	N/A
2.18.1	PlacedDate	必填	ORDERITEMS	LASTCREATE
2.18.2	PlacedTime		ORDERITEMS	LASTCREATE
2.18.3	LastUpdate Date		ORDERITEMS	LASTUPDATE
2.18.4	LastUpdate Time		ORDERITEMS	LASTUPDATE
2.19	Product Measurement		N/A	N/A
2.20	BuyOrg Accounting Detail	第一次重复出现	N/A	N/A
2.20.1	Percentage		N/A	N/A

级别	XML 元素	注释	表名	列名
2.20.2	Monetary Amount		N/A	N/A
2.20.2.A1	currency	属性	N/A	N/A
2.20.3	BudgetCode	必填	N/A	N/A
2.20.4	Description		N/A	N/A
2.20.5	Calculation Code	必需	N/A	N/A
2.21	Service Allowance Charge	第一次重复出现	N/A	N/A
2.21.1	Allowance ChargeCode	必填	N/A	N/A
2.21.2	Percentage		N/A	N/A
2.21.2	Monetary Amount		N/A	N/A
2.21.2.A1	currency	属性	N/A	N/A
2.21.3	Description		N/A	N/A
2.21.4	Calculation Code	必填	N/A	N/A
2.22	ItemShipping Schedule		N/A	N/A
2.22.1	Quantity	必需	N/A	N/A
2.22.2	ShipDate Reference	必填	N/A	N/A
2.22.2.1	Requested ShipDate		N/A	N/A
2.22.2.1	Scheduled ShipDate		N/A	N/A
2.22.2.3	ActualShip Date		N/A	N/A
2.23	ItemCustomer Field	第一次重复出现	ORDERITEMS	FIELD1
2.23	ItemCustomer Field	第二次重复出现	ORDERITEMS	FIELD2
2.24	UserData		N/A	N/A

CUSTOMER_NEW_HDR010_DATA

“顾客新建”消息包括 CUSTOMER_NEW_HDR010_DATA 段。此数据段由“顾客新建”消息的新购物者的购物者信息组成。

下表描述了 CUSTOMER_NEW_HDR010_DATA 的字段的格式和源。请使用下表获得字段的长度。关于数据库列的描述，请按照链接查看与其关联的表。

字段名	字段类型	表名	列名	注解
NC_HDR010 Version Number	CHAR (2)	N/A	N/A	01 的固定值。
NC_LoginID	CHAR (31)	USERREG	LOGONID	注册购物者的唯一登录标识。
NC_Password	CHAR (12)	USERREG	LOGONPASSWORD	注册购物者的加密登录密码。
NC_Verify Password	CHAR (12)	N/A	N/A	注册购物者的加密登录密码的验证。要处理此消息，此字段的值应与 NC_Password 中的数据相同。
NC_Merchant RefNumber	CHAR (10)	N/A	N/A	N/A
NC_Method Comm	CHAR (2)	USERPROF	PREFERREDCOMM	购物者的首选通信方式： E1 — 电子邮件或 URL 地址 1 E2 — 电子邮件或 URL 地址 2 P1 — 电话号码 1 P2 — 电话号码 2
NC_Challenge Ques	CHAR (250)	USERREG	CHALLENGEQUESTION	用于口头确认购物者身份的密码提示问题。
NC_Challenge Ans	CHAR (250)	USERREG	CHALLENGEANSWER	对密码提示问题的应答。
NC_Shopper Field1	CHAR (254)	USERS	FIELD1	保留供商家定制使用。
NC_Shopper Field2	CHAR (254)	USERS	FIELD2	保留供商家定制使用。
NC_Title	CHAR (5)	ADDRESS	PERSONTITLE	购物者的称呼： 博士 先生 夫人 女士 N — 未提供（缺省值）
NC_Last Name	CHAR (30)	ADDRESS	LASTNAME	购物者的姓氏。
NC_First Name	CHAR (30)	ADDRESS	FIRSTNAME	购物者的名字。

字段名	字段类型	表名	列名	注解
NC_MiddleName	CHAR (30)	ADDRESS	MIDDLENAME	购物者的中间名。
NC_RepCompany	CHAR (80)	ADDRESS	ORGNAME	购物者代表的公司。
NC_Phone1	CHAR (30)	ADDRESS	PHONE1	购物者的主要联系电话。
NC_Phone2	CHAR (30)	ADDRESS	PHONE2	购物者的后备联系电话。
NC_Fax	CHAR (30)	ADDRESS	FAX1	购物者的传真号码。
NC_Addr1	CHAR (50)	ADDRESS	ADDRESS1	购物者地址第 1 行。
NC_Addr2	CHAR (50)	ADDRESS	ADDRESS2	购物者地址第 2 行。
NC_Addr3	CHAR (50)	ADDRESS	ADDRESS3	购物者地址第 3 行。
NC_City	CHAR (30)	ADDRESS	CITY	购物者所在市 / 县 / 区名。
NC_State	CHAR (20)	ADDRESS	STATE	购物者所在省、直辖市或同类地区的缩写。
NC_Country	CHAR (30)	ADDRESS	COUNTRY	购物者所在的国家 / 地区名。
NC_ZipCode	CHAR (20)	ADDRESS	ZIPCODE	购物者的邮政编码或同类信息。
NC_Email1	CHAR (254)	ADDRESS	EMAIL1	购物者的主要电子邮件或 URL 地址。
NC_Email2	CHAR (254)	ADDRESS	EMAIL2	购物者的备用电子邮件或 URL 地址。
NC_DayPhoneType	CHAR (3)	ADDRESS	PHONE1TYPE	日间电话类型，例如 TTY 代表电传打字机（适用于有听觉障碍者），或 PHN 代表标准电话。
NC_DayPhoneList	CHAR (1)	ADDRESS	PUBLISHPHONE1	1 — 列出日间电话号码。 0 — 不列出日间电话号码。

字段名	字段类型	表名	列名	注解
NC_Evening PhoneType	CHAR (3)	ADDRESS	PHONE2TYPE	夜间电话类型， 例如 TTY 代表 电传打字机（适 用于有听觉障碍 者），或 PHN 代表标准电话。
NC_Evening PhoneList	CHAR (1)	ADDRESS	PUBLISHPHONE2	1 — 列出夜间电 话号码。 0 — 不列出夜间 电话号码。
NC_BestTime ToCall	CHAR (1)	ADDRESS	BESTCALLINGTIME	最佳呼叫时间指 示符： D — 日间 E — 夜间
NC_Package Insert	CHAR (1)	ADDRESS	PACKAGESUPPRESSION	广告插页禁止标 志，指明购物者 是否要在装运的 订单中包含广告 插页。 1 — 包含 0 — 不包含
NC_Address OptField1	CHAR (3)	ADDRESS	FIELD1	保留供商家定制 使用。
NC_Address OptField2	CHAR (1)	ADDRESS	FIELD2	保留供商家定制 使用。
NC_Shopper Gender	CHAR (1)	USERDEMO	GENDER	购物者性别： F — 女 M — 男 N — 未提供（缺 省值）
NC_Shopper Age	CHAR (10)	USERDEMO	AGE	购物者年龄： 0 — 未提供（缺 省值） 1 — 0-9 岁 2 — 10-19 岁 3 — 20-29 岁 4 — 30-39 岁 5 — 40-49 岁 6 — 50-59 岁 7 — 60 岁或以 上

字段名	字段类型	表名	列名	注解
NC_Shopper Income	CHAR (10)	USERDEMO	INCOME	购物者年收入: 0 — 未提供 (缺省值) 1 - 0 - 109994.5 元 2 - 110000 - 219994.5 元 3 - 220000 - 329994.5 元 4 - 330000 元或 以上
NC_Marital Status	CHAR (1)	USERDEMO	MARITALSTATUS	购物者婚姻状 况: S — 单身 M — 已婚 C — 同居 P — 分居 D — 离婚 W — 丧偶 0 — 其它 N — 未提供
NC_Number OfChildren	CHAR (10)	USERDEMO	CHILDREN	子女数。缺省值 为 0。
NC_Number InHouse	CHAR (10)	USERDEMO	HOUSEHOLD	购物者的家庭成 员数。缺省值为 1。
NC_Shopper Company	CHAR (30)	USERDEMO	COMPANYNAME	购物者的工作单 位。
NC_Shopper Interest	CHAR (254)	USERDEMO	HOBBIES	购物者主要的兴 趣和业余爱好。
NC_Previous OrderFlag	CHAR (1)	USERDEMO	ORDERBEFORE	指示购物者先前 是否已下订单。
NC_Demog Field1	CHAR (1)	USERDEMO	FIELD1	保留供商家定制 使用。
NC_Demog Field2	CHAR (1)	USERDEMO	FIELD2	保留供商家定制 使用。
NC_Demog Field3	CHAR (1)	USERDEMO	FIELD3	保留供商家定制 使用。
NC_Demog Field4	CHAR (1)	USERDEMO	FIELD4	保留供商家定制 使用。
NC_Demog Field5	CHAR (254)	USERDEMO	FIELD5	保留供商家定制 使用。
NC_Demog Field6	CHAR (10)	USERDEMO	FIELD6	保留供商家定制 使用。

CUSTOMER_UPDATE_HDR010_DATA

“顾客更新”消息包括 CUSTOMER_UPDATE_HDR010_DATA 段。此数据段由“顾客更新”消息的注册购物者的购物者信息组成。

下表描述了 CUSTOMER_UPDATE_HDR010_DATA 的字段的格式和源。请使用下表获得字段的长度。关于数据库列的描述，请按照链接查看与其关联的表。

字段名	字段类型	表名	列名	注解
NC_HDR010 Version Number	CHAR (2)	N/A	N/A	01 的固定值。
NC_LoginID	CHAR (31)	USERREG	LOGONID	注册购物者的唯一登录标识。
NC_Password	CHAR (12)	USERREG	LOGONPASSWORD	注册购物者的加密登录密码。
NC_Verify Password	CHAR (12)	N/A	N/A	注册购物者的加密登录密码的验证。要处理此消息，此字段的值应与 NC_Password 中的数据相同。
NC_Merchant RefNumber	CHAR (10)	N/A	N/A	N/A
NC_Method Comm	CHAR (2)	USERPROF	PREFERREDCOMM	购物者的首选通信方式： E1 — 电子邮件或 URL 地址 1 E2 — 电子邮件或 URL 地址 2 P1 — 电话号码 1 P2 — 电话号码 2
NC_Challenge Ques	CHAR (250)	USERREG	CHALLENGEQUESTION	用于口头确认购物者身份的密码提示问题。
NC_Challenge Ans	CHAR (250)	USERREG	CHALLENGEANSWER	对密码提示问题的应答。
NC_Shopper Field1	CHAR (254)	USERS	FIELD1	保留供商家定制使用。
NC_Shopper Field2	CHAR (254)	USERS	FIELD2	保留供商家定制使用。

字段名	字段类型	表名	列名	注解
NC_Title	CHAR (5)	ADDRESS	PERSONTITLE	购物者的称呼： 博士 先生 夫人 女士 N — 未提供（缺省值）
NC_Last 名称	CHAR (30)	ADDRESS	LASTNAME	购物者的姓氏。
NC_First 名称	CHAR (30)	ADDRESS	FIRSTNAME	购物者的名字。
NC_Middle 名称	CHAR (30)	ADDRESS	MIDDLENAME	购物者的中间名。
NC_Rep Company	CHAR (80)	ADDRESS	ORGNAME	购物者代表的公司。
NC_Phone1	CHAR (30)	ADDRESS	PHONE1	购物者的主要联系电话。
NC_Phone2	CHAR (30)	ADDRESS	PHONE2	购物者的后备联系电话。
NC_Fax	CHAR (30)	ADDRESS	FAX1	购物者的传真号码。
NC_Addr1	CHAR (50)	ADDRESS	ADDRESS1	购物者地址第 1 行。
NC_Addr2	CHAR (50)	ADDRESS	ADDRESS2	购物者地址第 2 行。
NC_Addr3	CHAR (50)	ADDRESS	ADDRESS3	购物者地址第 3 行。
NC_City	CHAR (30)	ADDRESS	CITY	购物者所在市 / 县 / 区名。
NC_State	CHAR (20)	ADDRESS	STATE	购物者所在省、直辖市或同类地区的缩写。
NC_Country	CHAR (30)	ADDRESS	COUNTRY	购物者所在的国家 / 地区名。
NC_ZipCode	CHAR (20)	ADDRESS	ZIPCODE	购物者的邮政编码或同类信息。
NC_Email1	CHAR (254)	ADDRESS	EMAIL1	购物者的主要电子邮件或 URL 地址。
NC_Email2	CHAR (254)	ADDRESS	EMAIL2	购物者的备用电子邮件或 URL 地址。

字段名	字段类型	表名	列名	注解
NC_Day PhoneType	CHAR (3)	ADDRESS	PHONE1TYPE	日间电话类型， 例如 TTY 代表 电传打字机（适 用于有听觉障碍 者），或 PHN 代表标准电话。
NC_Day PhoneList	CHAR (1)	ADDRESS	PUBLISHPHONE1	1 — 列出日间电 话号码。 0 — 不列出日 间电话号码。
NC_Evening PhoneType	CHAR (3)	ADDRESS	PHONE2TYPE	夜间电话类型， 例如 TTY 代表 电传打字机（适 用于有听觉障碍 者），或 PHN 代表标准电话。
NC_Evening PhoneList	CHAR (1)	ADDRESS	PUBLISHPHONE2	1 — 列出夜间电 话号码。 0 — 不列出夜 间电话号码。
NC_BestTime ToCall	CHAR (1)	ADDRESS	BESTCALLINGTIME	最佳呼叫时间指 示符： D — 日间 E — 夜间
NC_Package Insert	CHAR (1)	ADDRESS	PACKAGESUPPRESSION	广告插页禁止标 志，指明购物者 是否要在装运的 订单中包含广告 插页。 1 — 包含 0 — 不包含
NC_Address OptField1	CHAR (3)	ADDRESS	FIELD1	保留供商家定制 使用。
NC_Address OptField2	CHAR (1)	ADDRESS	FIELD2	保留供商家定制 使用。
NC_Shopper Gender	CHAR (1)	USERDEMO	GENDER	购物者性别： F — 女 M — 男 N — 未提供（缺 省值）

字段名	字段类型	表名	列名	注解
NC_Shopper Age	CHAR (10)	USERDEMO	AGE	购物者年龄: 0 — 未提供 (缺省值) 1 — 0-9 岁 2 — 10-19 岁 3 — 20-29 岁 4 — 30-39 岁 5 — 40-49 岁 6 — 50-59 岁 7 — 60 岁或以上
NC_Shopper Income	CHAR (10)	USERDEMO	INCOME	购物者年收入: 0 — 未提供 (缺省值) 1 - 0 - 109994.5 元 2 - 110000 - 219994.5 元 3 - 220000 - 329994.5 元 4 - 330000 元或以上
NC_Marital 状态	CHAR (1)	USERDEMO	MARITALSTATUS	购物者婚姻状况: S — 单身 M — 已婚 C — 同居 P — 分居 D — 离婚 W — 丧偶 O — 其它 N — 未提供
NC_Number OfChildren	CHAR (10)	USERDEMO	CHILDREN	子女数。缺省值为 0。
NC_Number InHouse	CHAR (10)	USERDEMO	HOUSEHOLD	购物者的家庭成员数。缺省值为 1。
NC_Shopper Company	CHAR (30)	USERDEMO	COMPANYNAME	购物者的工作单位。
NC_Shopper Interest	CHAR (254)	USERDEMO	HOBBIES	购物者主要的兴趣和业余爱好。
NC_Previous OrderFlag	CHAR (1)	USERDEMO	ORDERBEFORE	指示购物者先前是否已下订单。
NC_Demog Field1	CHAR (1)	USERDEMO	FIELD1	保留供商家定制使用。
NC_Demog Field2	CHAR (1)	USERDEMO	FIELD2	保留供商家定制使用。

字段名	字段类型	表名	列名	注解
NC_Demog Field3	CHAR (1)	USERDEMO	FIELD3	保留供商家定制使用。
NC_Demog Field4	CHAR (1)	USERDEMO	FIELD4	保留供商家定制使用。
NC_Demog Field5	CHAR (254)	USERDEMO	FIELD5	保留供商家定制使用。
NC_Demog Field6	CHAR (10)	USERDEMO	FIELD6	保留供商家定制使用。

ORDER_CREATE_HDR010_DATA

“订单创建”消息包括 ORDER_CREATE_HDR010_DATA 段。此数据段由“订单创建”消息的订单规范组成。

在下面的表中描述了 ORDER_CREATE_HDR010_DATA 的字段的格式和源。关于数据库列的描述，请按照链接到与其关联的表。

字段名	字段类型	表名	列名	描述
NC_HDR010 Version Number	CHAR (2)	N/A	N/A	01 的固定值。
NC_Order RefNumber	CHAR (10)	ORDERS	ORDERS_ID	内部生成的唯一订单引用号。这是一个主键。
NC_Order Date	CHAR (8)	ORDERS	TIMEPLACED	下订单的日期，格式为 YYYYMMDD。
NC_Order Time	CHAR (6)	ORDERS	TIMEPLACED	订购的时间，格式为 HHMMSS。
NC_Currency Type	CHAR (10)	ORDERS	CURRENCY	表示价格的货币。货币格式必须遵守 ISO 4217 标准。
NC_Total Price	CHAR (16)	ORDERS	TOTALPRODUCT	订单的产品价格总额。
NC_Total TaxPrice	CHAR (16)	ORDERS	TOTALTAX	订单销售税总额。
NC_Total ShippingPrice	CHAR (16)	ORDERS	TOTALSHIPPING	订单装运费用总额。
NC_Total TaxShipping Price	CHAR (16)	ORDERS	TOTALTAXSHIPPING	订单装运费用税款总额。
NC_Shopper RefNumber	CHAR (10)	ORDERS	MEMBER_ID	购物者引用号。
NC_Merchant RefNumber	CHAR (10)	ORDERS	STOREENT_ID	商家引用号。
NC_Merchant OrderNumber	CHAR (30)	ORDERS	ORMORDER	由商家生成的唯一订单引用号。

字段名	字段类型	表名	列名	描述
NC_BillToRefNumber	CHAR (10)	ORDERS	ADDRESS_ID	开票地址引用号。
NC_OrderCustField1	CHAR (10)	ORDERS	FIELD1	保留供商家定制使用。
NC_OrderCustField2	CHAR (16)	ORDERS	FIELD2	保留供商家定制使用。
NC_OrderCustField3	CHAR (254)	ORDERS	FIELD3	保留供商家定制使用。

ORDER_STATUS_UPDATE_HDR010_DATA

“订单状态更新”消息包括 ORDER_STATUS_UPDATE_HDR010_DATA 段。此数据段由“订单状态更新”消息的订单规范组成。

在下面的表中描述了 ORDER_STATUS_UPDATE_HDR010_DATA 的字段的格式和源。请使用下表获得字段的长度。关于数据库列的描述，请按照链接查看与其关联的表。

字段名	字段类型	表名	列名	注解
NC_HDR010VersionNumber	CHAR (2)	N/A	N/A	01 的固定值。
NC_OrderRefNumber	CHAR (10)	ORDSTAT	ORDERS_ID	WebSphere Commerce 订单引用号。
NC_CustomerOrderNumber	N/A	N/A	N/A	N/A
NC_CurrencyType	CHAR (10)	ORDSTAT	OSPCUR	表示价格的货币。货币格式必须遵守 ISO 4217 标准。
NC_TotalPrice	CHAR (16)	ORDSTAT	OSPTOT	订单的产品价格总额。
NC_TotalTaxPrice	CHAR (16)	ORDSTAT	OSTXTOT	订单总税款。
NC_TotalShippingPrice	CHAR (16)	ORDSTAT	OSSHOT	订单装运费用总额。
NC_TotalTaxShippingPrice	CHAR (16)	ORDSTAT	OSSHXTOT	订单装运费用税款总额。
NC_ShopperLoginID	N/A	N/A	N/A	N/A
NC_MerchantOrderNumber	CHAR (30)	ORDSTAT	OSMORDER	由商家生成的订单引用号。

字段名	字段类型	表名	列名	注解
NC_Order 状态	CHAR (32)	ORDSTAT	OSSTATUS	订单状态: P — 未决状态 C — 完成状态 X — 已取消 I — 库存更新审批 (装运不再审批) M — 准备认证 (装运已经过库存更新)
NC_Schedule ShipDate	CHAR (8)	ORDSTAT	OSSSTIME	已调度的装运日期, 格式为 YYYYMMDD。
NC_Actual ShipDate	CHAR (8)	ORDSTAT	OSASTIME	实际的装运日期, 格式为 YYYYMMDD。
NC_PlaceDate	CHAR (8)	ORDSTAT	OSPLTIME	订购日期, 格式为 YYYYMMDD。

PRODUCT_PRICE_UPDATE_HDR010_DATA

“产品价格更新”消息包括 PRODUCT_PRICE_UPDATE_HDR010_DATA 段。此数据段由“产品价格更新”消息的产品或商品定价信息组成。

“产品价格更新”消息支持两种应用程序数据集: “产品价格更新”版本 01 和“产品价格更新”版本 02。版本 02 包括版本 01 中数据的超集。特别地, 用于版本 01 的 PRODUCT_PRICE_UPDATE_HDR010_DATA 段包含用于字段 NC_HDR010VersionNumber 的值 01; 而在版本 02 中包含用于字段 NC_HDR010VersionNumber 的值 02。另外, 版本 02 还包含称为 NC_ProductNumber 的附加字段。

下表描述了版本 02 的 PRODUCT_PRICE_UPDATE_HDR010_DATA 字段的格式和源。请使用下表获得字段的长度。关于数据库列的描述, 请按照链接查看与其关联的表。

字段名	字段类型	表名	列名	描述
NC_HDR010 VersionNumber	CHAR (2)	N/A	N/A	(定值 02)
NC_Product RefNumber	CHAR (10)	OFFER	CATENTRY_ID	供销售用的 CatalogEntry。
NC_Product Price	CHAR (16)	OFFERPRICE	PRICE	产品或商品的价格。
NC_Currency Type	CHAR (10)	OFFERPRICE	CURRENCY	表示价格的货币。货币格式必须遵守 ISO 4217 标准。
NC_Shopper Group	CHAR (10)	OFFER	TRADEPOSCN_ID	TradingPositionContainer, Offer 是它的一部分。
NC_Precedence	CHAR (10)	OFFER	PRECEDENCE	此价格的优先级。
NC_Merchant RefNumber	CHAR (10)	CATENTRY	MEMBER_ID	标识“产品目录条目”所有者的引用号。
NC_ProductNumber	CHAR (64)	CATENTRY	PARTNUMBER	标识“产品目录条目”所有者的引用号。

字段名	字段类型	表名	列名	描述
NC_Product PriceRefNum	CHAR (10)	OFFER	OFFER_ID	产品或商品价格引用号。
NC_Start Timestamp	CHAR (26)	OFFER	STARTDATE	产品或商品价格生效的日期，格式为 YYYYMMDD hh:mm:ss.ssssss。缺省值是当前的日期和时间。
NC_End Timestamp	CHAR (26)	OFFER	ENDDATE	产品或商品价格的到期日，格式为 YYYY-MM-DD hh: mm: ss.ssssss。缺省值是 9999-12-31 23:59:59.999999。
NC_Price CustomField1	N/A	N/A	N/A	保留供商家定制使用。
NC_Price CustomField2	N/A	N/A	N/A	保留供商家定制使用。

版本 02 行为:

- 货币（它引用 OFFERPRICE 表中的 CURRENCY）是必需的，且必须以 ISO 4217 格式指定。
- 如果已指定产品价格引用号（引用 OFFER 表中的 OFFER_ID），该值将与货币类型一起作为键使用以更新 OFFERPRICE 表中的某行。
- 如果未指定产品价格引用号（OFFER_ID）未指定，那么产品引用号（引用 CATENTRY 表中的（CATENTRY_ID）与优先级（引用 OFFER 表中的 PRECEDENCE）或贸易状态容器（引用 OFFER 表中的 TRADEPOSCN_ID）的组合将用于获取产品价格引用号（OFFER_ID）。该值将与货币类型一起作为更新 OFFERPRICE 表中某行的键使用。
- 如果未指定产品价格引用号（CATENTRY_ID），那么产品引用号（引用 CATENTRY 表中的 PARTNUMBER）与商家引用号（引用 CATENTRY 表中的 MEMBER_ID）的组合将用于获取产品引用号（CATENTRY_ID）。该值将与优先级（PRECEDENCE）或贸易状态容器（TRADEPOSCN_ID）一起用于获取产品价格引用号（OFFER_ID）。该产品价格引用号与货币类型一起作为更新 OFFERPRICE 表中某行的键使用。
- 如果该产品价格引用号（OFFER_ID）与数据库中现有的产品价格引用号匹配，但货币类型与具有该产品价格引用号的任何记录中的货币类型不匹配，则将在 OFFERPRICE 表中创建一个新记录。这就允许您对相同报价以不同货币类型指定价格。
- 如果未指定优先级（PRECEDENCE），那么 ProductOfferPriceUpdate 命令将找出与此给定值（没有优先级）匹配的所有先前记录。取这些值中的最大值，并加 1。如果前一记录不存在，那么将优先级值设置为 1。以新的优先级值在 OFFERPRICE 表插入一行。此优先级的值必须低于 10^{16} 。如果到达最大的值，那么将拒绝新的更新。

下表描述了版本 01 的 PRODUCT_PRICE_UPDATE_HDR010_DATA 的字段的格式和源:

字段名	字段类型	表名	列名	描述
NC_HDR010 VersionNumber	CHAR (2)	N/A	N/A	01 的固定值。
NC_Product RefNumber	CHAR (10)	OFFER	CATENTRY_ID	供销售用的 CatalogEntry。
NC_Product Price	CHAR (16)	OFFERPRICE	PRICE	产品或商品的价格。
NC_Currency Type	CHAR (10)	OFFERPRICE	CURRENCY	表示价格的货币。货币 格式必须遵守 ISO 4217 标准。
NC_Shopper Group	CHAR (10)	OFFER	TRADEPOSCN_ID	TradingPositionContainer, Offer 是它的一部分。
NC_Precedence	CHAR (10)	OFFER	PRECEDENCE	此价格的优先级。
NC_Merchant RefNumber	CHAR (10)	CATENTRY	MEMBER_ID	标识“产品目录条目” 所有者的引用号。
NC_Product PriceRefNum	CHAR (10)	OFFER	OFFER_ID	产品或商品价格引用 号。
NC_Start Timestamp	CHAR (26)	OFFER	STARTDATE	产品或商品价格生效的 日期和时间，格式为 YYYY-MM-DD hh:mm:ss:sssss。缺省值 是当前的日期和时间。
NC_End Timestamp	CHAR (26)	OFFER	ENDDATE	产品或商品价格失效的 日期和时间，格式为 YYYY-MM-DD hh:mm:ss:sssss。缺省值 是 9999-12-31 23:59:59.999999。
NC_Price CustomField1	CHAR (30)	N/A	N/A	保留供商家定制使用。
NC_Price CustomField2	CHAR (1)	N/A	N/A	保留供商家定制使用。

版本 01 行为:

- 货币类型（引用 OFFERPRICE 表中的 CURRENCY）是必需的，且必须以 ISO 4217 格式指定。
- 如果已指定产品价格引用号（引用 OFFER 表中的 OFFER_ID），则该值将与货币类型一起作为更新 OFFERPRICE 表中某行的键使用。
- 如果未指定产品价格引用号（OFFER_ID），那么产品引用号（引用 CATENTRY 表中的 CATENTRY_ID）与优先级（引用 OFFER 表中的 PRECEDENCE）或贸易状态容器（引用 OFFER 表中的 TRADEPOSCN_ID）的组合将用于获取产品价格引用号（OFFER_ID）。该值将与货币类型一起作为更新 OFFERPRICE 表中某行的键使用。
- 如果该产品价格引用号（OFFER_ID）与数据库中现有的产品价格引用号匹配，但货币类型与具有该产品价格引用号的任何记录中的货币类型不匹配，则将在 OFFERPRICE 表中创建一个新记录。这就允许您对相同报价以不同货币类型指定价格。

- 如果未指定优先级（PRECEDENCE），那么 ProductOfferPriceUpdate 命令将找出与此给定值（没有优先级）匹配的所有先前记录。取这些值中的最大值，并加 1。如果前一记录不存在，那么将优先级值设置为 1。以新的优先级值在 OFFERPRICE 表插入一行。此优先级的值必须低于 10^{16} 。如果到达最大的值，那么将拒绝新的更新。

PRODUCT_QUANTITY_UPDATE_HDR010_DATA

“产品数量更新”消息包括 PRODUCT_QUANTITY_UPDATE_HDR010_DATA 段。此数据段包括“产品数量更新”消息的产品或商品库存信息。

下表描述了 PRODUCT_QUANTITY_UPDATE_HDR010_DATA 的字段的格式和源。请使用下表获得字段的长度。关于数据库列的描述，请按照链接查看与其关联的表。

字段名	字段类型	表名	列名	注意
NC_HDR010 VersionNumber	CHAR (2)	N/A	N/A	01 的固定值。
NC_Product RefNumber	CHAR (10)	INVENTORY	CATENTRY_ID	产品数量请求的 CatalogEntry。
NC_Product Quantity	CHAR (10)	INVENTORY	QUANTITY	产品的数量。
NC_Product Number	CHAR (64)	CATENTRY	PARTNUMBER	“产品目录条目”的“部件 号”。
NC_Merchant RefNumber	CHAR (10)	INVENTORY	STORE_ID	产品数量对其适用的商店。

行为:

- 商家引用号（引用 INVENTORY 表中的 STORE_ID）是必需的。
- WebSphere Commerce 产品引用号（引用 INVENTORY 表中的 CATENTRY_ID）和商家引用号（STORE_ID）用于更新 INVENTORY 表中的行。
- 如果没有表明产品引用号（CATENTRY_ID），那么商家引用号（STORE_ID）就用于获取成员号（引用 CATENTRY 表中的 MEMBER_ID）。成员号必须与产品目录条目的所有者相同。成员号（MEMBER_ID）与产品号（引用 CATENTRY 表中的 PARTNUMBER）一起用于获取产品引用号（CATENTRY_ID）。产品引用号与商家引用号（STORE_ID）一起用于更新 INVENTORY 表中的行。
- 如果表 INVENTORY 中的该行不存在，就会出现错误。

ORDER_CREATE_HDR020_DATA

“订单创建”消息包括 ORDER_CREATE_HDR020_DATA 段。此数据段由“订单创建”消息的购物者说明组成。

下表描述了 ORDER_CREATE_HDR020_DATA 的字段的格式和源。请使用下表获得字段的长度。关于数据库列的描述，请按照链接查看与其关联的表。

字段名	字段类型	表名	列名	注解
NC_HDR020 Version Number	CHAR (2)	N/A	N/A	01 的固定值。
NC_Shopper LoginID	CHAR (31)	USERREG	LOGONID	注册购物者的唯一登录标识。
NC_Purchaser LastName	CHAR (30)	ADDRESS	LASTNAME	购买方的姓氏。
NC_Purchaser MiddleName	CHAR (30)	ADDRESS	MIDDLENAME	购买方的中间名。
NC_Purchaser FirstName	CHAR (30)	ADDRESS	FIRSTNAME	购买方的名字。
NC_Purchaser Addr1	CHAR (50)	ADDRESS	ADDRESS1	购物者地址第 1 行。
NC_Purchaser Addr2	CHAR (50)	ADDRESS	ADDRESS2	购物者地址第 2 行。
NC_Purchaser Addr3	CHAR (50)	ADDRESS	ADDRESS3	购物者地址第 3 行。
NC_Purchaser City	CHAR (30)	ADDRESS	CITY	购物者所在市 / 县 / 区名。
NC_Purchaser State	CHAR (20)	ADDRESS	STATE	购物者所在省 / 直辖市或同类信息的缩写。
NC_Purchaser Country	CHAR (30)	ADDRESS	COUNTRY	购买方所在的国家 / 地区。
NC_Purchaser ZipCode	CHAR (20)	ADDRESS	ZIPCODE	购物者的邮政编码或同类信息。
NC_Purchaser Email1	CHAR (254)	ADDRESS	EMAIL1	购买方的主要电子邮件或 URL 地址。
NC_Purchaser Email2	CHAR (254)	ADDRESS	EMAIL2	购买方的后备电子邮件或 URL 地址。
NC_Purchaser Phone1	CHAR (30)	ADDRESS	PHONE1	购买方的主要联系电话。
NC_Purchaser Phone2	CHAR (30)	ADDRESS	PHONE2	购物者的后备联系电话。
NC_Purchaser Fax	CHAR (30)	ADDRESS	FAX1	购物者的传真号。
NC_Purchaser CompanyName	CHAR (80)	ADDRESS	ORGNAME	购买方代表的公司。
NC_Purchaser Shopper GroupName	CHAR (50)	MBRGRP	MBRGRPNAME	购买方所属的购物者组。

ORDER_CREATE_HDR030_DATA

“订单创建”消息包括 ORDER_CREATE_HDR030_DATA 段。此数据段由“订单创建”消息的开票详细信息组成。

下表描述了 ORDER_CREATE_HDR030_DATA 的字段的格式和源。请使用下表获得字段的长度。关于数据库列的描述，请按照链接查看与其关联的表。

字段名	字段类型	表名	列名	注解
NC_HDR030 VersionNumber	CHAR (2)	N/A	N/A	01 的固定值。
NC_BillTo LastName	CHAR (30)	ADDRESS	LASTNAME	收单人的姓氏。
NC_BillTo MiddleName	CHAR (30)	ADDRESS	MIDDLENAME	收单人的中间名。
NC_BillTo FirstName	CHAR (30)	ADDRESS	FIRSTNAME	收单人的名字。
NC_BillTo Addr1	CHAR (50)	ADDRESS	ADDRESS1	收单人地址第 1 行。
NC_BillTo Addr2	CHAR (50)	ADDRESS	ADDRESS2	收单人地址第 2 行。
NC_BillTo Addr3	CHAR (50)	ADDRESS	ADDRESS3	收单人地址第 3 行。
NC_BillTo City	CHAR (30)	ADDRESS	CITY	收单人所在市 / 县 / 区名。
NC_BillTo State	CHAR (20)	ADDRESS	STATE	收单人所在省 / 直辖市或同类信息的缩写。
NC_BillTo Country	CHAR (30)	ADDRESS	COUNTRY	收单人所在的国家 / 地区。
NC_BillTo ZipCode	CHAR (20)	ADDRESS	ZIPCODE	收单人邮政编码或同类信息。
NC_BillTo Email1	CHAR (254)	ADDRESS	EMAIL1	收单人主要电子邮件或 URL 地址。
NC_Billto Email2	CHAR (254)	ADDRESS	EMAIL2	收单人后备电子邮件或 URL 地址。
NC_BillTo Phone1	CHAR (30)	ADDRESS	PHONE1	收单人主要联系电话。
NC_BillTo Phone2	CHAR (30)	ADDRESS	PHONE2	收单人后备联系电话。
NC_BillTo Fax	CHAR (30)	ADDRESS	FAX1	收单人传真号。

ORDER_CREATE_HDR040_DATA

“订单创建”消息包括 ORDER_CREATE_HDR040_DATA 段。此数据段由“订单创建”消息的商家信息组成。

下表描述了 ORDER_CREATE_HDR040_DATA 的字段的格式和源。请使用下表获得字段的长度。关于数据库列的描述，请按照链接查看与其关联的表。

字段名	字段类型	表名	列名	注解
NC_HDR040 Version Number	CHAR (2)	N/A	N/A	01 的固定值。
NC_Supplier 名称	CHAR (80)	STOREENTDS	DISPLAYNAME	商家的公司名称。
NC_Supplier Addr1	CHAR (50)	STADDRESS	ADDRESS1	商家的公司地址第 1 行。
NC_Supplier Addr2	CHAR (50)	STADDRESS	ADDRESS2	商家的公司地址第 2 行。
NC_Supplier Addr3	CHAR (50)	STADDRESS	ADDRESS3	商家的公司地址第 3 行。
NC_Supplier City	CHAR (30)	STADDRESS	CITY	商家公司所在市 / 县 / 区的名称。
NC_Supplier State	CHAR (20)	STADDRESS	STATE	商家公司所在的省 / 直辖市或同类信息的缩写。
NC_Supplier Country	CHAR (30)	STADDRESS	COUNTRY	商家公司所在国家 / 地区。
NC_Supplier ZipCode	CHAR (20)	STADDRESS	ZIPCODE	商家的邮政编码或同类信息。
NC_Supplier Phone1	CHAR (30)	STADDRESS	PHONE1	商家的公司电话号码。
NC_Supplier Contact LastName	CHAR (30)	STADDRESS	LASTNAME	商家联系人的姓氏。
NC_Supplier Contact MiddleName	CHAR (30)	STADDRESS	MIDDLENAME	商家联系人的中间名。
NC_Supplier Contact FirstName	CHAR (30)	STADDRESS	FIRSTNAME	商家联系人的名字。
NC_Supplier ContactTitle	CHAR (30)	STADDRESS	PERSONTITLE	商家联系人的称呼。
NC_Supplier Contact Phone1	CHAR (30)	STADDRESS	PHONE1	商家联系人的主要电话号码。
NC_Supplier Contact Phone2	CHAR (30)	STADDRESS	PHONE2	商家联系人的后备联系电话。
NC_Supplier Contact Email1	CHAR (254)	STADDRESS	EMAIL1	商家联系人的主要电子邮件或 URL 地址。
NC_Supplier Contact Email2	CHAR (254)	STADDRESS	EMAIL2	商家联系人的后备电子邮件或者 URL 地址。

ORDER_CREATE_ITM010_DATA

“订单创建”消息包括 ORDER_CREATE_ITM010_DATA 段。此数据段由“订单创建”消息的商品或产品装运规范组成。

下表描述了 ORDER_CREATE_ITM010_DATA 的字段的格式和源。请使用下表获得字段的长度。关于数据库列的描述，请按照链接查看与其关联的表。

字段名	字段类型	表名	列名	注解
NC_ITM010 VersionNumber	CHAR (2)	N/A	N/A	01 的固定值。
NC_Item LineNumber	CHAR (3)	N/A	N/A (作为顺序号 而生成)	N/A
NC_Item RefNumber	CHAR (10)	ORDERITEMS	ORDERITEMS_ID	内部生成的唯一 装运地址引用 号。这是一个主 键。
NC_Item ProductRefNumber	CHAR (10)	ORDERITEMS	CATENTRY_ID	商品或产品引用 号。这不是一个 外键。
NC_Item ProductNumber	CHAR (64)	CATENTRY	PARTNUMBER	商品库存标识 (SKU) 或产品 号。
NC_Item Product ShortDescription	CHAR (254)	CATENTDESC	SHORTDESCRIPTION	商品或产品的简 短描述, 包括其 名称。
NC_Item UnitPrice	CHAR (16)	ORDERITEMS	PRICE	商品的单价。
NC_Item CurrencyType	CHAR (10)	ORDERITEMS	CURRENCY	表示价格的货 币。货币格式必 须遵守 ISO 4217 标准。
NC_Item ProductQuantity	CHAR (10)	ORDERITEMS	QUANTITY	订购的数量。
NC_Item ShipToAddr RefNum	CHAR (10)	ORDERITEMS	ADDRESS_ID	送货地址的地址 引用号。
NC_Item ShipMode RefNum	CHAR (10)	ORDERITEMS	SHIPMODE_ID	商家装运方式引 用号

字段名	字段类型	表名	列名	注解
NC_ItemState	CHAR (1)	ORDERITEMS	STATUS	订单状态: P — 未决状态 C — 完成状态 X — 已取消 I — 库存更新审批 (装运不再审批) M - 准备认证 (装运已经过库存更新)
NC_ItemShipToLastName	CHAR (30)	ADDRESS	LASTNAME	装运接收顾客的姓氏。
NC_ItemShipToMiddleName	CHAR (30)	ADDRESS	MIDDLENAME	装运接收顾客的中间名。
NC_ItemShipToFirstName	CHAR (30)	ADDRESS	FIRSTNAME	装运接收顾客的名字。
NC_ItemShipToAddr1	CHAR (50)	ADDRESS	ADDRESS1	装运接收顾客第一地址行。
NC_ItemShipToAddr2	CHAR (50)	ADDRESS	ADDRESS2	装运接收顾客第二地址行。
NC_ItemShipToAddr3	CHAR (50)	ADDRESS	ADDRESS3	装运接收顾客第三地址行。
NC_ItemShipToCity	CHAR (30)	ADDRESS	CITY	装运接收顾客的市/县/区名。
NC_ItemShipToState	CHAR (20)	ADDRESS	STATE	装运接收顾客所在省/直辖市或同类信息的缩写。
NC_ItemShipToCountry	CHAR (30)	ADDRESS	COUNTRY	装运接收顾客所在的国家或地区。
NC_ItemShipToZipCode	CHAR (20)	ADDRESS	ZIPCODE	装运接收顾客的邮政编码或同类信息。
NC_ItemShipToEmail1	CHAR (254)	ADDRESS	EMAIL1	装运接收顾客的主要电子邮件或 URL 地址。
NC_ItemShipToEmail2	CHAR (254)	ADDRESS	EMAIL2	装运接收顾客的后备电子邮件或 URL 地址。
NC_ItemShipToPhone1	CHAR (30)	ADDRESS	PHONE1	装运接收顾客的主要联系电话。
NC_ItemShipToPhone2	CHAR (30)	ADDRESS	PHONE2	装运接收顾客的后备联系电话。

字段名	字段类型	表名	列名	注解
NC_Item ShipToFax	CHAR (30)	ADDRESS	FAX1	装运接收顾客的 传真号。
NC_Item ShippingCarrier	CHAR (30)	SHIPMODE	CARRIER	递送者标识符， 例如 Federal Express。
NC_Item ShippingMethod	CHAR (30)	SHIPMODE	CODE	递送者服务装运 方式，例如 FedEx Express Overnight。
NC_Item ShipToComment	CHAR (254)	ORDERITEMS	COMMENTS	顾客的注释，例 如礼物的贺词。
NC_Item Creation Timestamp	CHAR (64)	ORDERITEMS	LASTCREATE	建立“送货地 址”条目的时间 和日期。
NC_Item Update Timestamp	CHAR (64)	ORDERITEMS	LASTUPDATE	上次更新送货地 址条目的日期和 时间。
NC_Item CustField1	CHAR (10)	ORDERITEMS	FIELD1	保留供商家定制 使用。
NC_Item CustField2	CHAR (254)	ORDERITEMS	FIELD2	保留供商家定制 使用。

ORDER_STATUS_UPDATE_ITM010_DATA

“订单状态更新”消息包括 ORDER_STATUS_UPDATE_ITM010_DATA 段。此数据段由“订单状态更新”消息的商品或产品装运规范组成。

“订单状态更新”消息支持两个应用程序数据的集合：“订单状态更新”版本 01 和“订单状态更新”版本 02。版本 02 包括版本 01 中数据的超集。特别地是，版本 01 的 ORDER_STATUS_UPDATE_ITM010_DATA 段包含字段 NC_HDR010VersionNumber 的值 01；而版本 02 包含 NC_HDR010VersionNumber 的值 02。另外，版本 02 也包含一些版本 01 中没有的附加字段。

下表描述了版本 02 的 ORDER_STATUS_UPDATE_ITM010_DATA 的字段的格式和源。请使用下表获得字段的长度。关于数据库列的描述，请按照链接查看与其关联的表。

字段名	字段类型	表名	列名	注解
NC_ITM010 VersionNumber	CHAR (2)	N/A	N/A (固定值 02)	N/A
NC_Order RefNumber	CHAR (10)	ORDISTAT	ORDERS_ID	WebSphere Commerce 订单 引用号。
NC_Merchant OrderNumber	CHAR (30)	ORDISTAT	OSMORDER	商家的订单引用 号。

字段名	字段类型	表名	列名	注解
NC_ItemRefNumber	CHAR (10)	ORDISTAT	ORDERITMES_ID	WebSphere Commerce 商品引用号。
NC_ItemMerchantRefNumber	CHAR (30)	ORDISTAT	OIMITEM	商家商品引用号。
NC_OrderItemStatus	CHAR (32)	ORDISTAT	OISTATUS	订购商品状态。
NC_ItemScheduleShipDate	CHAR (8)	ORDISTAT	OISSTIME	已调度的装运日期，格式为 YYYYMMDD。
NC_ItemActualShipDate	CHAR (8)	ORDISTAT	OIASTIME	实际的装运日期，格式为 YYYYMMDD。
NC_ItemPlaceDate	CHAR (8)	ORDISTAT	OIPLTIME	订购日期，格式为 YYYYMMDD。
NC_ItemQuantity	CHAR (10)	ORDISTAT	OIQTCONFIRM	订购商品的数量。
NC_ItemCurrencyType	CHAR (10)	ORDISTAT	OICPCUR	用于表示商品价格的货币。价格格式必须遵守 ISO 4217 标准。
NC_ItemUnitPrice	CHAR (16)	ORDISTAT	OIPRTOT	商品单价。
NC_ItemTotalPrice	CHAR (16)	ORDISTAT	OITOTPRC	商品总价格。
NC_ItemTotalTaxPrice	CHAR (16)	ORDISTAT	OITXTOT	商品销售总额。
NC_ItemTotalShippingPrice	CHAR (16)	ORDISTAT	OISHTOT	商品总装运价格。
NC_ItemTotalTaxShippingPrice	CHAR (16)	ORDISTAT	OISHTXTOT	商品装运价格总税款。
NC_ItemComment	CHAR (250)	ORDISTAT	OICMNT	购物者关于订购商品的注释。例如，购物者可以在订购的礼物中包括贺词。

下表中描述了版本 01 的 ORDER_STATUS_UPDATE_ITM010_DATA 的字段的格式和源:

字段名	字段类型	表名	列名	描述
NC_ITM010VersionNumber	CHAR (2)	N/A	N/A	01 的固定值。

NC_Order RefNumber	CHAR (10)	ORDISTAT	ORDERS_ID	WebSphere Commerce 订单 引用号。
NC_Item RefNumber	CHAR (10)	ORDISTAT	ORDERITMES_ID	WebSphere Commerce 商品 引用号。
NC_Order ItemStatus	CHAR (32)	ORDISTAT	OISTATUS	订购商品状态。
NC_Item Schedule ShipDate	CHAR (8)	ORDISTAT	OISSTIME	已调度的装运日 期，格式为 YYYYMMDD。
NC_Item ActualShipDate	CHAR (8)	ORDISTAT	OIASTIME	实际的装运日 期，格式为 YYYYMMDD。
NC_Item PlaceDate	CHAR (8)	ORDISTAT	OIPLTIME	订购日期，格式 为 YYYYMMDD。
NC_Item Quantity	CHAR (10)	ORDISTAT	OIQTCONFIRM	订购商品的数 量。
NC_Item Comment	CHAR (250)	ORDISTAT	OICMNT	购物者关于订购 商品的注释。例 如，购物者可以 在订购的礼物中 包括贺词。

ORDER_CREATE_PROLOG_DATA

“订单创建”消息包括 ORDER_CREATE_PROLOG_DATA 段。此数据段标识了正在定义的消息类型；即，该段指示消息是带有字段值为 ON（订单新建）的“订单创建”消息。

ORDER_CREATE_PROLOG_DATA 的字段的格式和源在下表中有所描述：

字段名	字段类型	字段值
NC_MsgType	CHAR (10)	ON
NC_MsgVersion	CHAR (2)	01
NC_RESERVED	CHAR (10)	保留供 IBM 使用。

ORDER_STATUS_UPDATE_PROLOG_DATA

“订单状态更新”消息包括 ORDER_STATUS_UPDATE_PROLOG_DATA 段。此数据段标识了正在定义的消息类型；即，该段指示消息是带有字段值为 OS（订单状态）的“订单状态更新”消息。

“订单状态更新”消息支持两个应用程序数据的集合：“订单状态更新”版本 01 和“订单状态更新”版本 02。版本 02 包括版本 01 中数据的超集。特别的是，版本 01 的

ORDER_STATUS_UPDATE_PROLOG_DATA 段包含值为 01 的 NC_MsgVersion 字段；而版本 02 包含值为 02 的 NC_MsgVersion。

下表描述了版本 02 的 ORDER_STATUS_UPDATE_PROLOG_DATA 的字段的格式和源：

字段名	字段类型	字段值
NC_MsgType	CHAR (10)	OS
NC_MsgVersion	CHAR (2)	02
NC_RESERVED	CHAR (10)	保留供 IBM 使用。

下表描述了版本 01 的 ORDER_STATUS_UPDATE_PROLOG_DATA 的字段的格式和源：

字段名	字段类型	字段值
NC_MsgType	CHAR (10)	OS
NC_MsgVersion	CHAR (2)	01
NC_RESERVED	CHAR (10)	保留供 IBM 使用。

PRODUCT_PRICE_UPDATE_PROLOG_DATA

“产品价格更新”消息包括 PRODUCT_PRICE_UPDATE_PROLOG_DATA 段。此数据段标识了正在定义的消息类型；即，该段指示消息是带有字段值为 PP（产品价格，Product Price）的“产品价格更新”消息。

“产品价格更新”消息支持两种应用程序数据集合：“产品价格更新”版本 01 和“产品价格更新”版本 02。版本 02 包括版本 01 中数据的超集。特别的是，版本 01 的 PRODUCT_PRICE_UPDATE_PROLOG_DATA 段包含 NC_MsgVersion 字段的值 01；而版本 02 包含了 NC_MsgVersion 的值 02。

下表描述了版本 02 的 PRODUCT_PRICE_UPDATE_PROLOG_DATA 的字段的格式和源：

字段名	字段类型	字段值
NC_MsgType	CHAR (10)	PP
NC_MsgVersion	CHAR (2)	02
NC_RESERVED	CHAR (10)	保留供 IBM 使用。

下表描述了版本 01 的 PRODUCT_PRICE_UPDATE_PROLOG_DATA 的字段的格式和源：

字段名	字段类型	字段值
NC_MsgType	CHAR (10)	PP
NC_MsgVersion	CHAR (2)	01
NC_RESERVED	CHAR (10)	保留供 IBM 使用。

PRODUCT_QUANTITY_UPDATE_PROLOG_DATA

“产品数量更新”消息包括 PRODUCT_QUANTITY_UPDATE_PROLOG_DATA 段。此数据段标识了正在定义的消息类型；即，该段指示消息是带有字段值为 PQ（产品数量）的“产品数量更新”消息。

PRODUCT_QUANTITY_UPDATE_PROLOG_DATA 的字段的格式和源在下表中有所描述：

字段名	字段类型	字段值
NC_MsgType	CHAR (10)	PQ
NC_MsgVersion	CHAR (2)	01
NC_RESERVED	CHAR (10)	保留供 IBM 使用。

CUSTOMER_NEW_PROLOG_DATA

“顾客新建”消息包括 CUSTOMER_NEW_PROLOG_DATA 段。此数据段标识了正在定义的消息类型；即，该段指示消息是带有字段值为 NC（新顾客）的“顾客更新”消息。

CUSTOMER_NEW_PROLOG_DATA 的字段的格式和源在下表中有所描述：

字段名	字段类型	字段值
NC_MsgType	CHAR (10)	NC
NC_MsgVersion	CHAR (2)	01
NC_RESERVED	CHAR (10)	保留供 IBM 使用。

CUSTOMER_UPDATE_PROLOG_DATA

“顾客更新”消息包括 CUSTOMER_UPDATE_PROLOG_DATA 段。此数据段标识了正在定义的消息类型；即，该段指示消息是带有字段值为 UC（更新顾客）的“顾客更新”消息。

CUSTOMER_UPDATE_PROLOG_DATA 的字段的格式和源在下表中有所描述：

字段名	字段类型	字段值
NC_MsgType	CHAR (10)	UC
NC_MsgVersion	CHAR (2)	01
NC_RESERVED	CHAR (10)	保留供 IBM 使用。

出站消息的 DATUSR_DATA

出站消息“订单创建”包括 DATUSR_DATA 数据段。在此消息中包括 DATUSR 记录以发送附加数据。您可以定制字段，传送未包括在这些消息中的额外数据。

下表描述了 DATUSR_DATA 字段的格式和源:

字段名	字段类型	描述
NC_FieldName	CHAR (8)	要添加的新字段的名称。
NC_FieldLength	CHAR (10)	NC_FieldValue 字段的长度。
NC_FieldValue	CHAR (variable)	新字段的值的文本字符串。

进站消息的 DATUSR_DATA

“订单状态更新”、“顾客新建”和“顾客更新”进站消息包含 DATUSR_DATA 数据段。在这些消息中包括要接收附加数据的 DATUSR 记录。可以定制字段，传送未包括在这些消息中的额外数据。DATUSR_DATA 可以重复多次。

下表描述了 DATUSR_DATA 字段的格式和源:

字段名	字段类型	描述
NC_FieldName	CHAR (8)	要添加的新字段的名称。
NC_FieldLength	CHAR (10)	NC_FieldValue 字段的长度。
NC_FieldValue	CHAR (variable)	新字段的值的文本字符串。

NCCustomer_10.mod 文件

NCCustomer_10.mod 顾客公用文件由新购物者的购物者信息构成。它用于 Create_NC_Customer 和 Update_NC_Customer 消息。

所有 MOD 文件都在以下目录中:

- ▶ 2000 *drive:\Program Files\WebSphere\CommerceServer\xml\messaging*
- ▶ NT *drive:\WebSphere\CommerceServer\xml\messaging*
- ▶ AIX */usr/WebSphere/CommerceServer/xml/messaging*
- ▶ Solaris */opt/WebSphere/CommerceServer/xml/messaging*
- ▶ 400 */QIBM/Proddata/WebCommerce/xml/messaging*

NCCustomer_10.mod 的 XML 元素值的格式和源在下表中描述。关于数据库列的描述，请随着链接查看与其关联的表。除非另外注明，所有的字段都是可选的。

级别	XML 元素	注解	表名	列名
1	LoginInfo	必填	N/A	N/A
1.1	LoginID	必填	USERREG	LOGONID
1.2	Password	必填	USERREG	LOGONPASSWORD
1.3	VerifyPassword	必填	N/A	N/A
2	MerchantID		N/A	N/A
3	MethodOf Communication		USERPROF	PREFERREDCOMM

级别	XML 元素	注解	表名	列名
4	Challenge Question		USERREG	CHALLENGEQUESTION
5	Challenge Answer		USERREG	CHALLENGEANSWER
6	ShopperField	第一次重复出现	USERS	FIELD1
6	ShopperField	第二次重复出现	USERS	FIELD2
7	ContactPerson Name	必填	N/A	N/A
7.1	Title		ADDRESS	PERSONTITLE
7.2	FullName	此版本不支持	N/A	N/A
7.3	LastName	必填	ADDRESS	LASTNAME
7.4	FirstName		ADDRESS	FIRSTNAME
7.5	MiddleName		ADDRESS	MIDDLENAME
7.6	AlternateName	此版本不支持	N/A	N/A
8	RepCompany		ADDRESS	ORGNAME
9	Address	必填	N/A	N/A
9.1	AddressLine	必需; 第一次重复出现	ADDRESS	ADDRESS1
9.1	AddressLine	第二次重复出现	ADDRESS	ADDRESS2
9.1	AddressLine	第三次重复出现	ADDRESS	ADDRESS3
9.2	City	必填	ADDRESS	CITY
9.3	State	必填	ADDRESS	STATE
9.4	Zip	必填	ADDRESS	ZIPCODE
9.5	Country	必填	ADDRESS	COUNTRY
10	ContactInfo	必填	N/A	N/A
10.1	Telephone	第一次重复出现	ADDRESS	PHONE1
10.1	Telephone	第二次重复出现	ADDRESS	PHONE2
10.2	Email	第一次重复出现	ADDRESS	EMAIL1
10.2	Email	第二次重复出现	ADDRESS	EMAIL2
10.3	Fax		ADDRESS	FAX1
11	DayPhoneInfo		N/A	N/A
11.1	PhoneInfo		N/A	N/A
11.1.A1	type	属性	ADDRESS	PHONE1TYPE
11.1.A2	isListed	属性	ADDRESS	PUBLISHPHONE1
12	EveningPhone Info		N/A	N/A
12.1	PhoneInfo		N/A	N/A
12.1.A1	type	属性	ADDRESS	PHONE2TYPE
12.1.A2	isListed	属性	ADDRESS	PUBLISHPHONE2
13	BestTimeToCall		ADDRESS	BESTCALLINGTIME
14	Include PackageInsert		ADDRESS	PACKAGESUPPRESSION

级别	XML 元素	注解	表名	列名
15	Address OptField	第一次重复出现	ADDRESS	FIELD1
15	Address OptField	第一次重复出现	ADDRESS	FIELD2
16	Gender		N/A	N/A
16.A1	value	必需; 属性	USERDEMO	GENDER
17	AgeGroup		USERDEMO	AGE
18	IncomeGroup		USERDEMO	INCOME
19	MaritalStatus		N/A	N/A
19.A1	value	必需; 属性	USERDEMO	MARITALSTATUS
20	NumberOf Children		USERDEMO	CHILDREN
21	NumberIn House		USERDEMO	HOUSEHOLD
22	WorkCompany		USERDEMO	COMPANYNAME
23	Interests		USERDEMO	HOBBIES
24	PreviousOrder		USERDEMO	ORDERBEFORE
25	Demographics	第一次重复出现	USERDEMO	FIELD1
25	Demographics	第二次重复出现	USERDEMO	FIELD2
25	Demographics	第三次重复出现	USERDEMO	FIELD3
25	Demographics	第四次重复出现	USERDEMO	FIELD4
25	Demographics	第五次重复出现	USERDEMO	FIELD5
25	Demographics	第六次重复出现	USERDEMO	FIELD6
26	UserData		N/A	N/A

NCCommon.mod — 源文件

WebSphere Commerce 定义了所有基于 DTD 文件的入站 XML 消息。每个 DTD 文件都由几个 XML 元素和 NCCommon.mod 文件组成。另外，Create_NC_Customer_10.dtd 和 Update_NC_Customer_10.dtd 文件还包含来自 NCCustomer10.mod 公用文件的信息。

所有 MOD 和 DTD 文件都在以下目录中:

- ▶ 2000 `drive:\Program Files\WebSphere\CommerceServer\xml\messaging`
- ▶ NT `drive:\WebSphere\CommerceServer\xml\messaging`
- ▶ AIX `/usr/WebSphere/CommerceServer/xml/messaging`
- ▶ Solaris `/opt/WebSphere/CommerceServer/xml/messaging`
- ▶ 400 `/QIBM/Proddata/WebCommerce/xml/messaging`

NCCustomer_10.mod — 源文件

WebSphere Commerce 定义了所有基于 DTD 文件的入站 XML 消息。每个 DTD 文件都由几个 XML 元素和 NCCCommon.mod 文件组成。另外，Create_NC_Customer_10.dtd 和 Update_NC_Customer_10.dtd 文件还包含来自 NCCustomer10.mod 顾客公用文件的信息。

所有 MOD 和 DTD 文件都在以下目录中:

▶ 2000	drive:\Program Files\WebSphere\CommerceServer\xml\messaging
▶ NT	drive:\WebSphere\CommerceServer\xml\messaging
▶ AIX	/usr/WebSphere/CommerceServer/xml/messaging
▶ Solaris	/opt/WebSphere/CommerceServer/xml/messaging
▶ 400	/QIBM/Proddata/WebCommerce/xml/messaging

调用消息传递系统编辑方法

要调用出站消息传递系统接口的 `compose()` 方法，请指定以下参数:

- `viewname`: 要使用的编辑视图名称，如 VIEWREG 表中现有记录的 VIEWNAME 列中指定的那样。关于如何使用 VIEWREG 表的更多信息，请参阅《*WebSphere Commerce 程序员指南*》。

重要信息: 在 VIEWREG 表参阅的记录中，INTERFACENAME 和 CLASSNAME 列的值必须包含与所有 WebSphere Commerce 消息传递系统是命令关联的接口和类的名称。接口名称必须是

`com.ibm.commerce.messaging.viewcommands.MessagingViewCommand`。类的名称必须是 `com.ibm.commerce.messaging.viewcommands.MessagingViewCommandImpl`。

- `CommandContext`: 关于 `CommandContext` 接口或实现该接口的 `CommandContextImpl` 类的信息，请参阅《*WebSphere Commerce 程序员指南*》。
- `TypedProperty`: 已输入属性中的值必须是字符串或实现 `toString()` 方法的对象。关于 `TypedProperty` 的更多信息，请参阅《*WebSphere Commerce 程序员指南*》。

`compose()` 方法为每一个已经启用并已指定为管理控制台中当前消息类型的传送运行一个视图命令。此方法执行以下过程:

- 它使用 `viewname` 参数和每个传送的 `storeId` 和设备格式，这已在管理控制台中定义。这些值用于查找 VIEWREG 表中的视图命令。
- 它运行视图命令，将在 `TypedProperty` 参数中指定的值传送给该命令。运行命令时，系统使用 `viewname`、`storeId` 和设备格式标识查找 VIEWREG 表 PROPERTIES 列中的 JSP 模板。运行 JSP 模板并传送 `TypedProperty` 参数中的值。
- JSP 编辑消息，并在对象中调用发送方法时通过相应的传送发送它。可以使用在消息传递对象（在该对象上运行编辑）上处理的、立即的或回复进行发送。

SendXMLOrder 命令

SendXMLOrder 命令由 WebSphere Commerce 出站消息传递系统使用，将 Report_NC_PurchaseOrder XML 消息发送至后端系统。此命令使用消息编辑模板生成 XML 消息，然后出站消息传递系统将其发送至后端系统。

行为

- 通过将此任务命令分配到 OrderProcess 命令中的 OrderMessagingCmd 接口可启用该命令。
- 一旦启用，就在调用它之后完成 OrderProcess 命令的处理。
- 此任务命令调用了消息传递系统编辑服务，该服务使用 OrderCreateXML.jsp 编辑模板收集必需的订单信息，并构建 Report_NC_PurchaseOrder 出站 XML 消息。
- 如果编辑成功，该命令将试图使用出站消息传递系统发送服务发送此消息。

异常情况

如果遇到异常，此命令将在错误日志中生成一个条目。

第 17 章 实现集成消息

WebSphere Commerce 提供一个使用入站和出站消息以和实现中心系统集成的机制。入站实现集成消息用于在 WebSphere Commerce 中根据从实现中心系统接收到的入站请求运行命令。出站消息可以由出站系统生成，以便用发生的事件更新实现中心系统，例如接收新存货或订单装运。要使用实现集成消息，您必须已安装了适配器并且已配置了消息传递系统以接收 XML 消息。

消息传递系统可以发送和接收许多 XML 格式的预定义的消息。这种格式提供了高度的可读性，使得修改和维护变得简单易行。关于每种消息的说明，请参阅关于入站和出站实现集成消息的部分。您也可以添加新消息。对于新的入站消息，您既可以将它们与现有 WebSphere Commerce 命令相关联，也可与您创建的命令相关联。

XML 消息的格式由在特定的 DTD 文件内定义的 XML 元素集合构成。每个 DTD 可能包含一个或多个公用文件，由文件扩展名 .mod 标识。另外，每条入站消息与 sys_template.xml 消息模板定义文件中的一条 WebSphere Commerce 控制器命令关联。所有 DTD、MOD 和 XML 文件都在以下目录中：

▶ 2000	<i>drive:\Program Files\WebSphere\CommerceServer\xml\messaging</i>
▶ NT	<i>drive:\WebSphere\CommerceServer\xml\messaging</i>
▶ AIX	<i>/usr/WebSphere/CommerceServer/xml/messaging</i>
▶ Solaris	<i>/opt/WebSphere/CommerceServer/xml/messaging</i>
▶ 400	<i>QIBM/ProdData/WebCommerce/xml/messaging</i>

声明

任何对本文档中的 IBM 许可程序引用并不明示或暗示只能使用 IBM 的许可程序。只要不侵犯 IBM 的知识产权，任何同等功能的产品、程序或服务，都可以代替 IBM 产品、程序或服务。在与其它产品结合使用时，除了那些由 IBM 明确指定的产品之外，其验证由用户自行负责。

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

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

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

Lab Director
IBM Canada Ltd. Laboratory
8200 Warden Avenue
Markham, Ontario
L6G 1C7
Canada

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

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

本文档可能包含有关其它公司产品的信息，包括对这些公司因特网站点的引用。IBM 对这些信息的精确性、完整性或使用不负任何责任。

本产品是基于 SET 协议的。

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

商标

以下术语是国际商业机器公司在美国和 / 或其它国家或地区的商标或注册商标：

AIX

CICS

DB2

DB2 Extenders
IBM
SecureWay
400

Encina
iSeries
VisualAge

HotMedia
MQSeries
WebSphere

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

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

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

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

Oracle 是 Oracle Corporation 的注册商标。

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

UNIX 是 The Open Group 在美国和其它国家或地区的注册商标。

其它公司、产品或服务名称可能是其它公司的商标或服务标记。



中国印刷