

IBM WebSphere Commerce



**接続性と通知: オンライン・ヘルプ・ファイル**

*Version 5.4*



IBM WebSphere Commerce



接続性と通知: オンライン・ヘルプ・ファイル

*Version 5.4*

**ご注意**

本書の情報およびそれによってサポートされる製品をご使用になる前に、161 ページの『特記事項』に記載されている一般情報をお読みください。

本マニュアルに関するご意見やご感想は、次の URL からお送りください。今後の参考にさせていただきます。

<http://www.ibm.com/jp/manuals/main/mail.html>

なお、日本 IBM 発行のマニュアルはインターネット経由でもご購入いただけます。詳しくは

<http://www.ibm.com/jp/manuals/> の「ご注文について」をご覧ください。

(URL は、変更になる場合があります)

原 典： IBM WebSphere Commerce  
Connectivity and Notification: Online Help Files  
Version 5.4

発 行： 日本アイ・ピー・エム株式会社

担 当： ナショナル・ランゲージ・サポート

第3刷 2002.6

この文書では、平成明朝体™W3、平成明朝体™W9、平成角ゴシック体™W3、平成角ゴシック体™W5、および平成角ゴシック体™W7を使用しています。この(書体\*)は、(財)日本規格協会と使用契約を締結し使用しているものです。フォントとして無断複製することは禁止されています。

注\* 平成明朝体™W3、平成明朝体™W9、平成角ゴシック体™W3、  
平成角ゴシック体™W5、平成角ゴシック体™W7

© Copyright International Business Machines Corporation 1996, 2002. All rights reserved.

© Copyright IBM Japan 2002

# 目次

<b>第 1 章 プログラム・アダプター</b>	1
CommandProperty オブジェクト	1
デバイス形式のアルゴリズム	2
HTTP 上での XML	3
ミドルウェアとしての MQSeries	3
WebSphere CommerceMQSeries アダプター	4
MQSeries アダプターの並列対順次メッセージ処理	4
<b>第 2 章 プログラム・アダプターの構成</b>	5
HTTP を使用した XML 要求用にプログラム・アダプターを使用可能にする	5
アダプターの追加	5
MQSeries MA88 製品の拡張パックのダウンロードおよびインストール	6
MQSeries アダプターを使用可能にする	7
JMS for MQSeries の構成	8
WebSphere Application Server のクラスパス変数の更新	9
JMSAdmin を使用した JMS の構成	9
MQSeries アダプターを使用した応答処理	12
<b>第 3 章 MQSeries のプログラム・アダプター・セキュリティー</b>	13
HTTP 要求のプログラム・アダプター・セキュリティー	14
メッセージ構成サービス	14
メッセージ・システム	14
一般アプリケーション・エラー XML メッセージと一般システム・エラー XML メッセージ	15
アウトバウンド・メッセージ構成のセットアップ	15
メッセージ・システム構成サービスの使用例	16
メッセージ・システム構成サービスにおけるエラー処理	17
メッセージ・システム構成メソッドの呼び出し	18
<b>第 4 章 OrderItemStatus コマンド</b>	21
GetPickPackListDetail コマンド	23
BroadcastMessage コマンド	23
OrderInvoiceStatus コマンド	25
OrderShippingStatus コマンド	25
OrderStatus コマンド	25
ProductOfferPriceUpdate コマンド	27
ProductInventoryUpdate コマンド	28
SendWCSOrder タスク・コマンド	29
SendXMLOrder コマンド	30
NewInboundMessage コマンド	30
OrderConfirmStatus コマンド	31
ProductListPriceUpdate コマンド	31
<b>第 5 章 アウトバウンド・メッセージ・システムの処理方法</b>	33
アウトバウンド・メッセージ・システム	34
アウトバウンド・メッセージ・システム管理	34
アウトバウンド・バックエンド組み込みメッセージ	35
アウトバウンド配送組み込みメッセージ	35
アウトバウンド・メッセージ・システム・ストア管理	36
アウトバウンド・メッセージ・システム・サイト管理	37

トランスポート・アダプターへの新規メッセージの追加	37
<b>第 6 章 アウトバウンド・メッセージ送信/受信の送信サービスを使用可能にする</b>	39
メッセージ・システム・トランスポート・アダプターを使用可能にする	39
エラー状況メッセージ・タイプのトランスポートへの割り当て	40
ストアのトランスポート・メソッドのアクティブ化	40
ストアへのトランスポート・メソッドの追加	40
ストアのトランスポート・メソッドの構成	41
ストアのトランスポート・メソッドの非アクティブ化	41
ストアのメッセージ・タイプへのトランスポート・メソッドの割り当て	41
サイトのトランスポート・メソッドのアクティブ化	42
サイトへのトランスポート・メソッドの追加	42
サイトのメッセージ・タイプへのトランスポート・メソッドの割り当て	42
サイトのトランスポート・メソッドの構成	43
サイトのトランスポート・メソッドの非アクティブ化	43
E メール・トランスポート・メソッドのシステム設定値の検査	43
通知のアクティブ化	44
エラー通知を使用可能にする	44
配送通知 E メールを使用可能にする	45
ブロードキャスト・メッセージを使用可能にする	45
オーダー状況通知を使用可能にする	46
管理コンソールからメッセージを送信できるようにする	46
アウトバウンド OrderCreate メッセージを使用可能にする	47
<b>第 7 章 メッセージ・タイプ</b>	49
アウトバウンド・メッセージ・システム・インターフェース	51
アウトバウンド・メッセージ拡張	52
新規アウトバウンド・メッセージ・サポート	53
アウトバウンド・メッセージの UserData エレメント	53
アウトバウンド・メッセージ・システム・インターフェースのプログラミング例	54
メッセージ構成テンプレート	56
初期化サービス	56
メッセージの内容設定サービス	56
E メール部品または接続サービスの追加	57
構成可能メッセージ・データ・サービス	58
送信サービス	58
その他のサービス	58
<b>第 8 章 インバウンド・バックエンド組み込みメッセージ</b>	61
インバウンド配送組み込みメッセージ	62
<b>第 9 章 新規インバウンド XML メッセージの追加</b>	65
システムへの新規 DTD ファイルの追加	65
インバウンド・メッセージ DTD ファイルのリストへの追加	65
インバウンド・メッセージ拡張	66
インバウンド・メッセージの UserData エレメント	66
<b>第 10 章 メッセージ・マッパー</b>	67
XML メッセージ・マッパー	67
レガシー・メッセージ・マッパー	67
インバウンド・メッセージ・テンプレート定義ファイル	68
メッセージ・マッパーの除去	69
メッセージ・マッパーの追加	69

新規インバウンド・メッセージ・サポート	69
<b>第 11 章 NewInboundMessage コマンドのカスタマイズ</b>	<b>71</b>
<b>第 12 章 メッセージ・マッパー構成</b>	<b>73</b>
テンプレート定義ファイルを使用した XML 構文解析	73
sys_template.xml ファイル	74
user_template.xml ファイル	74
ec_template.dtd ファイル	75
テンプレート定義ファイルの TemplateDocument エlement	76
テンプレート定義ファイルの TemplateTag エlement	77
<b>第 13 章 メッセージ・システム・バックエンド組み込みメッセージ</b>	<b>79</b>
<b>第 14 章 配送組み込みメッセージ</b>	<b>81</b>
<b>第 15 章 組み込みメッセージ DTD ファイル</b>	<b>83</b>
バックエンド組み込みレガシー・メッセージ	84
バックエンド組み込み XML メッセージ	86
配送組み込みメッセージを使用したサンプル・シナリオ	86
ReleaseShipNotify メッセージ	87
Response_WCS_ExpectedInvRecord メッセージ	88
Response_WCS_PickBatch メッセージ	89
Report_WCS_PickPackListDetail メッセージ	91
Response_WCS_CreateInvReceipt メッセージ	93
Response_WCS_UpdateInvReceipt メッセージ	94
Response_WCS_CreateShipConfirm メッセージ	95
Update_WCS_OrderStatus メッセージ	96
オーダー状況の更新メッセージ	99
Update_WCS_ProductPrice メッセージ	102
Create_WCS_ExpectedInventoryRecord メッセージ	103
Create_WCS_PickBatch メッセージ	104
Inquire_WCS_PickPackListDetail メッセージ	104
Create_WCS_InventoryReceipt メッセージ	105
Update_WCS_InventoryReceipt メッセージ	106
Create_WCS_ShipmentConfirmation メッセージ	106
Create_WCS_Customer メッセージ	107
Update_WCS_Customer メッセージ	108
Update_WCS_ProductInventory メッセージ	110
商品価格の更新メッセージ	111
商品数量の更新メッセージ	112
オーダー作成メッセージ	112
新規顧客へのメッセージ	114
更新顧客へのメッセージ	114
Update_NC_Customer メッセージ	115
Update_NC_OrderStatus メッセージ	117
Create_NC_Customer メッセージ	121
Update_NC_ProductInventory メッセージ	121
Update_NC_ProductPrice メッセージ	122
Report_NC_PurchaseOrder メッセージ	123
CUSTOMER_NEW_HDR010_DATA	131
CUSTOMER_UPDATE_HDR010_DATA	135
ORDER_CREATE_HDR010_DATA	140

ORDER_STATUS_UPDATE_HDR010_DATA . . . . .	141
PRODUCT_PRICE_UPDATE_HDR010_DATA . . . . .	144
PRODUCT_QUANTITY_UPDATE_HDR010_DATA . . . . .	145
ORDER_CREATE_HDR020_DATA . . . . .	146
ORDER_CREATE_HDR030_DATA . . . . .	147
ORDER_CREATE_HDR040_DATA . . . . .	148
ORDER_CREATE_ITM010_DATA . . . . .	149
ORDER_STATUS_UPDATE_ITM010_DATA. . . . .	153
ORDER_CREATE_PROLOG_DATA. . . . .	154
ORDER_STATUS_UPDATE_PROLOG_DATA . . . . .	154
PRODUCT_PRICE_UPDATE_PROLOG_DATA . . . . .	155
PRODUCT_QUANTITY_UPDATE_PROLOG_DATA . . . . .	155
CUSTOMER_NEW_PROLOG_DATA . . . . .	155
CUSTOMER_UPDATE_PROLOG_DATA . . . . .	156
アウトバウンド・メッセージの DATUSR_DATA . . . . .	156
インバウンド・メッセージの DATUSR_DATA . . . . .	156
NCCustomer_10.mod ファイル. . . . .	157
NCCCommon.mod - ソース・ファイル . . . . .	159
NCCustomer_10.mod - ソース・ファイル. . . . .	159
<b>特記事項 . . . . .</b>	<b>161</b>



---

## 第 1 章 プログラム・アダプター

プログラム・アダプターを使用すると、外部システムは Program プロトコルに対して XML 要求を渡すことにより、WebSphere Commerce と通信できるようになります。プログラム・アダプターは調達システムなどの外部システムに、HTTP を介して WebSphere Commerce と通信するための共通の方法を提供し、WebSphere Commerce は、これらのシステムに対してバイヤー / サプライヤー・トランザクションでのサプライヤーとして機能することができます。プログラム・アダプターは以下のアクションを実行することによって、インバウンド XML 要求を処理します。

- 要求を認識し、それが XML 要求であるかどうかを検査する。要求の以下の 3 つの属性がサポートされている場合、XML 要求と認めることができます。
  - content-type
  - method
  - character encoding

サポートされる要求属性は、アダプター構成で指定されます。

- 要求の入カストリームを抽出する。
- メッセージ・マッパーを呼び出し、入カストリームの内容を渡す。
- メッセージ・マッパーによって戻される WebSphere Commerce コマンドを表す CommandProperty オブジェクトを受け取る。
- 応答の生成で使用する適切なデバイス形式を決定する。
- コマンドを実行する。
- XML 応答メッセージを送信する。これは、JSP によって作成され、実行したコマンドで指定されたビュー名と受信した要求のデバイス形式に基づきます。

各要求は別個のセッションとして扱われます。メッセージの認証情報は、メッセージの制御域で指定されます。デフォルトでは、プログラム・アダプターはユーザー ID とパスワードのパラメーターを調べ、要求の認証性を判別します。プログラム・アダプターは、レガシー・メッセージをサポートしていません。レガシー・メッセージは、認証情報の仕様をサポートしていないからです。

プログラム・アダプターのライフ・サイクルは、WebSphere Commerce インスタンス全体に存在します。

その構成パラメーターが除去されたりアダプターが使用不能でない限り、インスタンスの開始時にこれは初期化され、これはインスタンスが実行されている間存在します。

WebSphere Commerce がデバイスからの受信要求を処理する方法に関するアーキテクチャー情報については、*WebSphere Commerce プログラマーズ・ガイド* を参照してください。

---

### CommandProperty オブジェクト

CommandProperty オブジェクトは、コントローラー・コマンドを代表するものです。オブジェクトには、実行されるコマンド名、コマンドを実行するときのコマンド・プロパティ、およびコマンドのパラメーターが含まれます。メッセージ・マッパーの目的は、インバウンド要求メッセージを、アダプターによって実行されるコントローラー・コマンドに変換することです。WebSphere Commerce のすべてのコンポーネントは、メッセージ・マッパーを使用してデータを拡張 TypedProperty オブジェクトにマップすることができますが、メッセージ・マッパーの主な目的は、XML オブジェクトを、コントローラー・コマンドを表す共通 Java オブジェクトに変換することです。

CommandProperty データ・タイプは、以下の 3 つの部分で構成されます。

- `commandName`: スtring形式の、実行されるコマンド名。
- `requestProperties`: TypedProperty オブジェクト形式の、コマンドを実行するときのコマンド・プロパティ。
- `executionProperties`: TypedProperty オブジェクト形式の、コマンドを実行するための制御データ。

---

## デバイス形式のアルゴリズム

デバイス形式のアルゴリズムは、特定の要求に対する応答として使用するのに適切な JSP を決定するために使用されます。要求を受け取ってから、メッセージ・マップパーを使用して要求を CommandProperty オブジェクトに変換するとき、要求を処理するために使用されるメッセージ・マップパーおよびアダプターは、応答の内容を生成するための適切なデバイス形式を決定します。メッセージ・マップパーの ID はアダプターのデバイス形式 ID に追加され、応答の全体のデバイス形式 ID を決定します。この全体のデバイス形式 ID と VIEWNAME を使用して、VIEWREG テーブルから適切な JSP を取得します。これが、応答の内容を生成します。

要求を受け入れて、メッセージ・マップパーを使用する各アダプターには、デバイス形式 ID が与えられます。この ID は、アダプターの `instance_name.xml` 構成ファイルで定義されます。この ID は、`instance_name.xml` 構成ファイルにあるアダプターの構成で定義されます。メッセージ・マップパーを使用してアダプターを受け取る、各デフォルト・デバイス形式 ID のインターバルは、-10000 です。プログラム・アダプターのデバイス形式 ID は -10000 で、MQSeries アダプターのデバイス形式 ID は -20000 です。レガシー・メッセージをサポートするアダプターは、デバイス形式 ID -30000 を使用します。適切な JSP と JSP を呼び出すための表示コマンドを決定するとき、メッセージ・マップパー ID はアダプターのデバイス形式 ID に追加され、応答ビュー用のデバイス形式 ID を決定します。ただし、計算済みデバイス形式 ID のビューが存在しない場合、アダプターのインターバルのデフォルト・ビューが使用されます。HTTP 上の XML 要求の場合、デフォルト・デバイス形式 ID は -10000 で、MQSeries アダプター要求の場合、デフォルト・デバイス形式 ID は -20000 になります。応答ビューを区別することは必要です。というのは、HTTP 応答は `HttpForwardViewCommand` インターフェースを使用し、MQSeries アダプターからの応答は `MessagingViewCommand` インターフェースを使用するからです。

このアルゴリズムにより、1 つのアダプターを使用して、最大で 9999 の使用可能メッセージ・マップパーを定義することができます。各メッセージ・マップパーの構成ファイルには、インターバルが 1 の ID 番号が含まれています。メッセージ・マップパーを使用する追加のアダプターを構成するには、インターバル 10000 のデバイス形式 ID を指定しなければなりません。たとえば、新規要求機構には、デバイス形式 ID インターバル 40000 を使用できるかもしれません。

**注 1:** 計算済みデバイス形式 ID が見つからない場合、アダプターのデフォルト・デバイス形式 ID を使用して結果を入手します。この番号は、HTTP アダプター構成で構成可能です。

**注 2:** インターバルの大きさは重要ではありません。たとえば、メッセージ・マップパーが 2 つしか存在しない場合、インターバルのサイズは 3 になるかもしれません。インターバル 10000 がデフォルトで使用されます。

**例 1:** インバウンド・メッセージは、ID -1 を持つメッセージ・マップパーと、デバイス形式 ID -10000 を持つプログラムによって処理されます。デバイス形式のアルゴリズムに従って、そのインバウンド・メッセージへの応答に対して生成されるデバイス形式 ID は -10001 になります。VIEWREG テーブルの VIEWNAME で定義された JSP ファイルと、デバイス形式 ID -10001 を使用して、応答が作成されます。

**例 2:** インバウンド・メッセージは、ID -1 を持つ同じメッセージ・マップパーによって処理されます。しかし、デバイス形式 ID -20000 を持つ MQSeries アダプターによって処理されるという点で、例 1 とは異なります。そのメッセージへの応答に対して生成されるデバイス形式 ID は -20001 になります。こうして、VIEWREG テーブルの VIEWNAME で定義された JSP ファイルと、デバイス形式 ID -20001 を使用して、応答が作成されます。要求に応答するために同じ JSP を使用することができても、JSP を呼び出すために使用されるクラスは異なる場合があることにご注意ください。

---

## HTTP 上での XML

WebSphere Commerce はプログラム・アダプターを使用して、HTTP 上でインバウンド XML メッセージを受け取ることができます。以下のステップは、HTTP 要求上の XML のフロー全体を示しています。

1. 外部システムは HTTP 上で、WebSphere Commerce に XML メッセージを送信します。
2. 要求はプログラム・アダプターにマップされます。
3. プログラム・アダプターは、適切なメッセージ・マップパーに XML 要求を渡します。
4. メッセージ・マップパーは XML 要求を CommandProperty オブジェクトに変換した後、プログラム・アダプターに再び渡します。
5. プログラム・アダプターは、実行用のコマンドを準備し、それを WebController 渡して実行します。
6. プログラム・アダプターは適切な XML 応答を生成し、要求を行った外部システムに XML 応答を戻します。

プログラム・アダプターは、XML 要求を受け取るときに要求を送信した外部システムの認証情報を検査する必要があります。、すべての XML 要求を処理できるわけではありません。XML 要求を WebSphere commerce コマンドにマップすることができても、要求が処理されたことを確認するための検査を行う必要があります。詳しくは、HTTP 要求のプログラム・アダプター・セキュリティーを参照してください。

---

## ミドルウェアとしての MQSeries

MQSeries アダプターを使った場合、ミドルウェアとして MQSeries を使用して、バックエンド・システムおよび外部システムに WebSphere Commerce を組み込むことができます。MQSeries アダプターは、バックエンド・システムおよび外部システムからのメッセージを WebSphere Commerce で受信するようにできます。サポートされるソフトウェアは MQSeries バージョン 5.2 以上の MA88 製品拡張機能付きです。

以下の 2 つのモードいずれかで MQ Java を使用して、MQSeries をミドルウェアとしてセットアップすることができます。

### • バインディング・モード

MQSeries サーバーと同じマシンに WebSphere Commerce をインストールし、それを Java ネイティブ・インターフェース (JNI) を使用する MQSeries Java を介して MQSeries サーバーに接続します。通信は、ネットワーク経由ではなく、キュー・マネージャー API への直接の JNI 呼び出しで行われるので、ネットワーク接続を使用して実行されるクライアント・モードよりバインディング・モードのほうがパフォーマンスは良くなります。

### • クライアント・モード

WebSphere Commerce を任意のマシンにインストールし、MQSeries サーバーをバックエンド・システムにインストールします。

MQSeries 接続、キュー、およびチャンネルを検査するには、テスト・プログラムを実行してメッセージを書き込んでそのメッセージを MQSeries キューから読み取ってください。詳しくは、お手持ちの MQSeries 文書を参照してください。

---

## WebSphere CommerceMQSeries アダプター

WebSphere Commerce MQSeries アダプター、または単に MQSeries アダプターは、MQSeries 経由でインバウンド・メッセージを処理し、それによってバックエンド・システムへの組み込みを実現する WebSphere Commerce のコンポーネントです。MQSeries アダプターは、MQ メッセージを取り出す JMS-MQ CCF コネクタと、これらのメッセージを実行するために呼び出されるプログラム・アダプターの組み合わせで成ります。

MQSeries アダプターは、一連の事前定義メッセージを持ち、WebSphere Commerce のビジネス処理をバックエンドまたは外部システムの処理に組み込めるようにします。各着信メッセージによって WebSphere Commerce 内の処理がアクティブにされると、データベース・テーブルが更新され、その他の操作が実行されます。送られるメッセージの詳細は、バックエンド組み込みおよび配送組み込みメッセージを参照してください。既存の事前定義メッセージのほかに、アダプターはメッセージ拡張と新規メッセージをサポートします。

---

## MQSeries アダプターの並列対順次メッセージ処理

MQSeries アダプターは、順次と並列の 2 つの方法で、インバウンド・メッセージを処理することができます。順次処理とは、各メッセージが順に並べられ、キューに入って、順番に処理されることを意味します。この方法では、各メッセージは、直前のメッセージの処理が完了するまで待機する必要があります。一方、並列処理とは、多くのメッセージを同時に処理できることを意味します。各メッセージが直前のものの完了を待つ必要はなく、多くのメッセージを同時に実行することができます。

一般的に、並列処理によってスループットはより高速化しますが、すべてのタイプの要求に適しているわけではありません。トランザクションの順次性が維持されていなければならない場合もあります。たとえば、新規の顧客がストアで登録してから、住所情報を修正し、次に購入オーダーを作成した場合、処理に備えて、それらのトランザクションの順序を保存しておいたほうがよいと思われます。アカウントがすでに作成済みでない限り、住所変更または購入オーダーを実行することはできません。同様に、正しい配送情報を添付されていない購入オーダーは、実行しないほうがよいと思われます。

一般的に、可能な場合には並列処理を使用するのが望ましいですが、この方式を使用してよいかどうかを個々のデータ別に判断する必要があります。

---

## 第 2 章 プログラム・アダプターの構成

プログラム・アダプターは、*instance\_name.xml* 構成ファイルを使用して構成されます。プログラム・アダプター・エントリーは、アダプターを定義するために使用されます。ProgramAdapter ノード間の情報がプログラム・アダプターの構成を定義します。使用するメッセージ・マッパー、サポートされているコンテンツ・タイプ、および要求を HTTP 上の XML の要求として識別する他の要求の属性を定義します。*instance\_name.xml* 構成で、プログラム・アダプターのエントリーが以下のようにになっている必要があります。

```
<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">
      </SessionContext>
      <SessionContextConfig />
      </SessionContext>
      <Configuration supportedMethods="POST, M-POST"
        supportedContentTypes="text/xml, text/xml-SOAP"
        supportedMessageMappers="WCS.INTEGRATION"
        supportedCharacterEncoding="IS08859-1, UTF-8" />
      </ProgramAdapter>
    </HttpAdapter>
  </HttpAdapters>
```

---

### HTTP を使用した XML 要求用にプログラム・アダプターを使用可能にする

インスタンスが作成されると、プログラム・アダプターは使用不可になります。HTTP を使用した XML をサポートするため、*instance\_name.xml* 構成ファイルを使用してプログラム・アダプターを使用可能にする必要があります。HTTP を使用した XML 要求をサポートするためにプログラム・アダプターを使用可能にするには、以下のようにします。

1. *instance\_name.xml* 構成ファイルをオープンする。
2. HttpAdapters セクションを見つけ、enabled パラメーターを “true” に設定する。

---

### アダプターの追加

新規のアダプターを追加するには、*instance\_name.xml* 構成ファイル内のアダプターのグループに手動で追加する必要があります。新規のアダプターを追加するには、以下のようにします。

1. *instance\_name.xml* 構成ファイルをオープンする。
2. HttpAdaptersXML ノードを見つけ、アダプターを定義するために以下のような XML 構文を追加する。

```
<HttpAdapter
  enabled="true/false"
  deviceFormatType="--デバイス装置--"
  deviceFormatId="#"
  name="--名前--"
  factoryClassname="-- HttpAdapterFactory をインプリメントするクラス --">
```

< --- アダプター構成情報を入れるための XML の自由範囲形式 --->

注: 上記の構文は、プログラム・アダプターを追加するのに使用されます。異なるタイプのアダプターを追加するには、クラス・インプリメンテーションをそれに応じて変更してください。上記の例では、HttpAdapterFactory がクラス・インプリメンテーションです。

## MQSeries MA88 製品の拡張パックのダウンロードおよびインストール

MQSeries MA88 製品の拡張パックをインストールするには、以下のようにします。

注: この情報は、MQSeries インストール手順が更新されると変更されます。



1. ユーザーのオペレーティング・システムに該当する ma88 製品の拡張パックを以下の URL からダウンロードする。  
`http://www.ibm.com/software/ts/mqseries/txppacs/ma88.html`. このファイルは圧縮 (ZIP) 形式になっています。
  - 同じ URL から文書 *MQSeries Using Java* を PDF 形式でダウンロードする。
2. SupportPac を解凍してからインストールする。
3. セットアップ手順に従って、製品の拡張機能を `MQ_install_path¥java` ディレクトリーにインストールする。ここで `MQ_install_path` は MQSeries がインストールされているパスです。
4. 以下のディレクトリーにある `admin.config` ファイルを更新する。

 ドライブ:¥Program Files¥WebSphere¥CommerceServer¥bin

 ドライブ:¥WebSphere¥CommerceServer¥bin

 /usr/WebSphere/CommerceServer/bin

  /opt/WebSphere/CommerceServer/bin

 /QIBM/Proddata/WebCommerce/bin

5. `MQ_install_path¥java¥lib` ディレクトリーを `com.ibm.ejs.sm.util.process.Nanny.path` 変数に追加する。iSeries クラス・ライブラリーの変更については、WebSphere Application Server のクラスパス変数の更新 を参照してください。











1. ユーザーのオペレーティング・システムに該当する ma88 製品の拡張パックを以下の URL からダウンロードする。  
`http://www.ibm.com/software/ts/mqseries/txppacs/ma88.html` このファイルは圧縮 (ZIP) 形式になっています。 `ma88_iSeries.zip`. という名前のファイルを取得したことを確認します。
  - 同じ URL から文書 *MQSeries Using Java* を PDF 形式でダウンロードする。
2. InfoZip's Unzip を使用して圧縮解除する。これにより、ファイル `ma88_400.sav` が作成されます。
3. iSeries 400 上の適切なライブラリー (たとえば QGPL) に保存ファイル MA88 を作成する。  
`CRTSAVF FILE(QGPL/MA88)`
4. この保存ファイルに、バイナリー・イメージとして `ma88_iSeries.sav` を転送する。これを実行するのに FTP を使用する場合には、入力されるコマンドは、以下のようになります。  
`PUT C:¥TEMP¥MA88_iSeries.SAV QGPL/MA88`
5. RSTLICPGM を使用して、Java の MQSeries クラス、プロダクト ID 5648C60 をインストールする。  
`RSTLICPGM LICPGM(5648C60) DEV(*SAVF) SAVF(QGPL/MA88)`

- ステップ 2 で作成された保存ファイルを削除する。  
DLTF FILE(QGPL/MA88)

---

## MQSeries アダプターを使用可能にする

MQSeries アダプター・メッセージを使用可能にするには、以下のチェックリストを使用します。

- MQSeries バージョン 5.2 をインストールする。MQSeries バインディング・モード構成または MQSeries クライアント・モード構成のいずれかをセットアップする方法については、*MQSeries Using Java* を参照してください。e-Integrator Version 3.0 では、バインディング・モードを使用します。
  -    ユーザー・ログオン ID に、キュー・マネージャーおよび定義されているキューに対する読み取りと書き込みの権限が必要です。e-Integrator Version 3.0 では、バインディング・モードを使用します。
  -  インスタンス・ユーザー・プロファイルには、キュー・マネージャーおよび定義されているキューに対する読み取りと書き込みの権限が必要です。この権限を定義するには、GRTMQMAUT コマンドを使用します。
- 以下の MQSeries オブジェクトが定義されていることを確認します。
  - キュー・マネージャー
  - インバウンド・メッセージ・キュー
  - アウトバウンド・メッセージ・キュー
  - 伝送キュー
  - エラー・キュー
  - パラレル・キュー
  - シリアル・キュー
-     MQSeries キュー・マネージャー・コード化文字セット ID を 1208 に設定する (UTF8)。以下の MQSeries コマンドをコマンド行から実行します。  






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

ここで、*YourQueueManagerName* は MQSeries キュー・マネージャーの名前です。
- MQSeries クライアント・モードを使用している場合には、必要なチャンネルをすべて定義し、MQSeries クライアントが MQSeries サーバーとの通信に使用するチャンネル名を識別しなければなりません。
- MQSeries MA88 製品の拡張パックをダウンロードして、インストールする。  
この製品の拡張機能には、MQSeries アダプターが MQSeries との通信に使用できる Java メッセージ・サービス (JMS) API が含まれています。
- JMS for MQSeries の構成。  
JMS QueueConnectionFactory、および対応する MQSeries オブジェクトにマップされる JMS キューを作成する必要があります。これによって、MQSeries アダプターは JMS を通じて MQSeries エンティティにアクセスできるようになります。
- JMSAdmin を使用して JMS を構成する。
- メッセージ・システム・トランスポート・アダプターを使用可能にする。
- WebSphere Application Server のクラスパス変数を更新する。

注: MQSeries アダプターを使用するには、WebSphere Commerce サーバーとインスタンスを開始する前に、キュー・マネージャーが開始されていることを確認してください。

## JMS for MQSeries の構成

JMS (Java メッセージ・サービス) を使って作業を行うようにメッセージ・システムを構成するには、以下のようにします。

1.  iSeries QShell は、Unix コマンドがご使用の WebSphere Commerce マシンで実行可能なエミュレーターを提供します。残りのコマンドの実行には、Java 1.3 を使用することが必要です。Java バージョン 1.3 を使用するためにユーザー・プロファイルを設定するには、ユーザー SystemDefault.properties ファイルに `java.version=1.3` という行を追加してください。詳細については、iSeries Java 資料を参照してください。QShell を開始する前に、以下のようにする必要があります。
2.  CL プロンプトに STRQSH と入力して iSeries QShell を始動する。
3.     WebSphere Commerce マシンから、`classpath`変数を更新します。
  - a. 以下のコマンドをすべて 1 行に入力する。



```
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:このステートメントは長いので、コマンド・プロンプトに 1 行のステートメントとして追加することはできません。2 つのセグメントに追加しなければなりません。

ここで

`WAS_install_path`は WebSphere Application Server をインストールしたパスです

ここで、`MQ_install_path`は、MQSeries をインストールしたパスです。

- b. 以下のコマンドを入力することによって、`MQ_JAVA_INSTALL_PATH`という名前の新規環境変数を追加する。



```
set MQ_JAVA_INSTALL_PATH=MQ_install_path%java
```







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

ここで、`MQ_install_path`は、MQSeries をインストールしたパスです。

次のコマンドを入力することによって、WebSphere Application Server に添付の JDK を使用するように環境を更新する。



```
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. JMSAdmin を使用して JMS を構成する。

---

## WebSphere Application Server のクラスパス変数の更新

インスタンス用の WebSphere Application Server classpath変数を更新するには、以下のようにします。

1. WebSphere Application Server 拡張管理コンソールをオープンする。
2. WebSphere Commerce インスタンスを実行しているホストを選択する。
3. **WebSphere** 管理可能ドメインを選択する。
4. 「**Nodes (ノード)**」 をクリックする。
5. ご使用のホスト名を選択する。
6. 「**Application Servers**」 を選択する。
7. WebSphere Commerce Server *instance\_name* を選択する。  
ここで、*instance\_name* は WebSphere Commerce インスタンスの名前です。 iSeries の場合、  
「*instance\_name* - WebSphere Commerce Server」 を選択する。
8. インスタンスの JVM 設定テーブルに移動する。
9. 「**追加**」 を選択して新しいシステム・プロパティを追加する。
10. 以下のシステム・プロパティを入力する。  
name = ws.ext.dirs value = MQJAVA/lib iSeries の場合: value=/QIBM/ProdData/mqm/java/lib
11. 「**適用**」 をクリックして、変更内容を適用する。
12. MQSeries が使用されるすべての WebSphere Commerce インスタンスについてステップ 2~7 を繰り返す。
13. WebSphere 拡張管理コンソールをクローズする。

### 400

iSeries の場合、上記のステップ 1~11 を実行してから以下の手順に従います。

1. 「**JVM**」 をクリックする。
2. 「システム・プロパティ」ボックスの右側で、「**追加**」 をクリックする。新しいシステム・プロパティがリストに表示されます。
3. 「名前」フィールドに `java.library.path` と入力します。
4. 「値」フィールドに `/QSYS.LIB/QMQMJAVA.LIB` と入力します。
5. 「**適用**」 をクリックして、変更内容を適用する。
6. MQSeries が使用されるすべての WebSphere Commerce インスタンスについてステップ 2~7 を繰り返す。
7. WebSphere 拡張管理コンソールをクローズする。

---

## JMSAdmin を使用した JMS の構成

キュー・マネージャーおよび作成されたキューを WebSphere Commerce ネームスペース内にマップするには、Qshell で以下のようにします。

1. WebSphere Application Server が実行中であって、しかも環境変数およびクラスパスが設定済みであることを確認する。
2. 次のディレクトリに変更する。

WIN `MQ_install_path¥java¥bin`  
AIX `MQ_install_path/java/bin`  
SOLARIS LINUX `MQ_install_path/java/bin`  
400 `/QIBM/ProdData/WebCommerce/bin`

ここで `MQ_install_path`は、MQSeries がインストールされているパスです。

- JMSAdmin.configファイルをテキスト・エディターでオープンする。

以下の 3 つの変数が示されている値に設定されていることを確認する。

```
INITIAL_CONTEXT_FACTORY=com.ibm.ejs.ns.jndi.CNInitialContextFactory
PROVIDER_URL=iiop://host_name:was_port
SECURITY_AUTHENTICATION=none
```

ここで

*host\_name*

インスタンス・ホスト名

*was\_port*

インスタンスを構成するのに使用される WebSphere Application Server

- コマンド行から、JMSAdmin プログラムを実行する。

WIN `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`

管理コマンド行インターフェースがロードされ、Initctx>プロンプトが表示されるのを待ちます。

- キュー・マネージャーへのキュー接続ファクトリーを WebSphere Application Server ネームスペースに登録する:

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

ここで

*JMSQueueConnectionFactory*

これは MQSeries 用の JMS 構成にある QueueConnectionFactory ConnectionSpec 属性で定義されます。これは構成マネージャーの「ConnectionSpec-JMS Interface CCF 接続」ページに見出すことができます。

*YourQueueManagerName*

MQSeries キュー・マネージャーの名前。

- WIN AIX SOLARIS LINUX コード化文字セット ID を 1208 に設定する (UTF8):

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

ここで

*JMSQueueConnectionFactory*

MQQueueConnectionFactory JMS オブジェクトの名前。

- 以下の JMS キューを定義する。キュー・タイプを定義するには、適切なコマンドをすべて 1 行に入力します。 **JMSSerialInboundQueue** — シリアル・インバウンド・キュー

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

例:

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

- **JMSParallelInboundQueue** — パラレル・インバウンド・キュー

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

- **JMSInboundQueue** — インバウンド・キュー

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

- **JMSOutboundQueue** — アウトバウンド・キュー

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

- **JMSErrorQueue** --エラー・キュー

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

*YourQueueManagerName*

MQSeries キュー・マネージャーの名前。

*YourSerialInboundQueueName*

シリアル・インバウンド・キュー用に作成される MQSeries キューの名前。

*YourParallelInboundQueueName*

パラレル・インバウンド・キューに作成される MQSeries キューの名前。

*YourInboundQueueName*

インバウンド・メッセージ・キューに作成される MQSeries キューの名前。

*YourOutboundQueueName*

アウトバウンド・キューに作成される MQSeries キューの名前。

*YourErrorQueueName*

エラー・キューに作成される MQSeries キューの名前。

キューのデフォルト名は、構成マネージャーで `ConnectionSpec-JMS` インターフェース CCF 接続値として使用される名前と同じものです。デフォルト名を変更する場合は、構成マネージャーでの名前も変更する必要があります。なおアウトバウンド・メッセージの場合の名前の変更は、管理コンソールで行えます。

8. アウトバウンド・キューおよびエラー・キューでは、宛先クライアントを、JMS がネイティブ MQSeries アプリケーションと対話していることを示すように設定する必要があります。以下のコマンドを実行する。

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

9.  WebSphere Commerce サーバーと同じマシンに MQSeries クライアントがセットアップされている MQSeries クライアント / サーバーを使用している場合、以下のコマンドを実行します。

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

ここで、*YourMQServerHostName*は、MQSeries サーバーの名前です。

このコマンドはリモート MQSeries サーバーに接続し、クライアントを構成します。

10. `end` と入力し、管理コマンド行インターフェースを終了します。

---

## MQSeries アダプターを使用した応答処理

MQSeries アダプターは WebSphere Commerce システムおよび機能の代替インターフェースとして使用することができます。MQSeries アダプターは、インバウンド要求を処理するだけでなく、要求への応答を提供することもできます。応答を作成するコマンドを作成し、MQSeries アダプターから応答を提供することができます。

インバウンド・メッセージが MQSeries アダプターによって受信されると、コマンドにマップされ、そのコマンドはインバウンド・メッセージで指定されているパラメーター付きで呼び出されます。応答がこのインバウンド要求から生成される場合、VIEWREG テーブルのエントリーが必要です。VIEWREG テーブルの使用法についての詳細は、*WebSphere Commerce プログラマーズ・ガイド* を参照してください。応答がコマンドによって生成される場合、その応答はインバウンド要求を受信するコネクタによって、並列またはシリアル・コネクタのアウトバウンド・キューに置かれます。

インバウンド要求でエラーが発生すると、エラーを生成するメッセージがエラー・キューに置かれます。アウトバウンド・キューへのエラー応答を生成するには、正しいデバイス形式を使用してエラー応答をマップするために、VIEWREG テーブルに別のエントリーが必要です。

インバウンド・メッセージに対して応答を提供する MQSeries アダプター・コンポーネントをセットアップするには、以下のようにします。

1. インバウンドの並列コネクタのための以下と同じような構文が含まれることを確認し、*instance\_name.xml* ファイルに、インバウンドの並列および直列コネクタのためのアウトバウンド・キューが含まれることを検査してください。

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

2. JMSAdmin を使用して、直列および並列アウトバウンド・キューに対して JMS 名を定義する。
3. VIEWREG テーブルのエントリーを追加して、インバウンド要求に必要な応答を定義する。このステップは、既存の WebSphere Commerce コマンドと新規のコマンドの両方に必要です。以下の値を使用してください。

### VIEWNAME

通常の HTTP 応答と同じ値

### INTERFACENAME

com.ibm.commerce.messaging.viewcommands.MessagingViewCommand

### CLASSNAME

com.ibm.commerce.messaging.viewcommands.MessagingViewCommandImpl

### PROPERTIES

docname=*JSP filename*

### DEVICEFMT\_ID

- -20000 (XML インバウンド・メッセージの場合)
- -30000 (既存のインバウンド・メッセージの場合)
- -5 を追加 (新規インバウンド・メッセージの場合)

---

## 第 3 章 MQSeries のプログラム・アダプター・セキュリティー

セキュリティーのレベルを変更するには、MQSeries アダプター構成の SessionContext ノードにあるクラス属性の値を変更しなければなりません。

WebController は要求を実行する前に、要求を実行する認証情報を判別します。これは、指定されるクラスによって判別されます。つまり、CommandProperty オブジェクトを使用して、要求の認証情報を判別します。

使用できるセキュリティーのレベルは 2 つあります。

- レベル 1: 限定セキュリティー
- レベル 2: すべての要求にユーザー ID およびパスワードが必要

また、自分用にカスタマイズしたセキュリティー・クラスも作成できます。制限は、ProgramAdapterSessionContext インターフェースをインプリメントしなければならないことです。

### セキュリティー・レベル 1: 限定セキュリティー

このレベルのセキュリティーは、デフォルトで、インストール時に使用可能になります。すべての要求は、明示的に指定されない限り、デフォルト・ユーザー ID を使用して処理されるものとします。他のユーザー ID により要求が処理される場合には、このユーザー ID は、要求で指定されます。指定されるユーザー ID のパスワードは必要ではなく、この要求は、指定されるユーザーの認証情報を使用して WebSphere Commerce により処理されます。

このセキュリティー・レベルを組み込むクラスは、

```
messaging.programadapter.security.DefaultCredentialsProgramAdapterSessionContextImpl
```

です。

### セキュリティー・レベル 2: すべての要求に対するログオンおよびパスワード

このレベルのセキュリティーでは、WebSphere Commerce に送信される要求にユーザー ID およびパスワードが含まれている必要があります。指定されるユーザー ID にパスワードが適合しない場合、セキュリティー例外が出され、要求は拒否されます。ユーザー ID およびパスワードが要求に指定されていない場合、要求はゲスト・ユーザーとして処理されます。つまり、実行前に許可の必要なコマンドには、要求で認証情報がなければなりません。

このセキュリティー・レベルを組み込むクラスは、

```
messaging.programadapter.security.CredentialsSpecifiedProgramAdapterSessionContextImpl
```

です。

認証情報の妥当性検査では、顧客がブラウザーを介して WebSphere Commerce と対話する際に使用されるのと同じ方法を使用します。したがって、ユーザー登録を WebSphere Commerce で処理するか、あるいはサード・パーティー・ソフトウェアで処理するかを考慮します。

認証情報の指定は必須ではないものの、WebSphere Commerce でサポートされる XML 組み込みメッセージは、このレベルのセキュリティーをサポートします。

---

## HTTP 要求のプログラム・アダプター・セキュリティー

このレベルのセキュリティーでは、WebSphere Commerce に送信される要求にユーザー ID およびパスワードが含まれている必要があります。指定されるユーザー ID にパスワードが適合しない場合、セキュリティー例外が出され、要求は拒否されます。ユーザー ID およびパスワードが要求に指定されていない場合、要求はゲスト・ユーザーとして処理されます。つまり、実行前に許可の必要なコマンドには、要求で認証情報がなければなりません。このセキュリティーのフォームは、ブラウザー・クライアントからの HTTP 要求と同じ方法で処理します。

このセキュリティー・レベルを組み込むクラスは、

```
messaging.programadapter.security.CredentialsSpecifiedProgramAdapterSessionContextImpl
```

です。

認証情報の妥当性検査では、顧客がブラウザーを介して WebSphere Commerce と対話する際に使用されるのと同じ方法を使用します。したがって、ユーザー登録を WebSphere Commerce で処理するか、あるいはサード・パーティー・ソフトウェアで処理するかを考慮します。

認証情報の指定は必須ではないものの、WebSphere Commerce でサポートされる XML 組み込みメッセージは、このレベルのセキュリティーをサポートします。このセキュリティーのインプリメントでは、レガシー・メッセージはサポートされていません。

---

## メッセージ構成サービス

一部の WebSphere Commerce メッセージの中には、メッセージ構成サービスを使用するものがあります。

構成サービスは、JSP テンプレートを使ってアウトバウンド・メッセージを生成してから、トランスポートで送信します。構成サービスがメッセージで使用される場合、JSP を実行してオーダー番号やストア番号などの情報を渡します。テンプレートが実行されると JSP は data bean を使用して、メッセージに必要な追加情報をデータベースから取り出すことができます。次に出力が生成され、形式化されたメッセージがトランスポートを通じて送信されます。他の JavaServer Page と同様、メッセージ・テンプレートも変更することができます。

変更できる生成メッセージのフィーチャーには、次のようなものがあります。

- メッセージのレイアウト。
- データベースから取り出されてページ上に表示されるストア、オーダー、または顧客についての情報。
- 「ご注文ありがとうございます。」などの顧客向けメッセージ・テキスト。
- 生成された出力の形式。たとえば、メッセージは、HTML 形式、非暗号化テキスト形式、または XML 形式で送信する場合があります。

---

## メッセージ・システム

WebSphere Commerce メッセージ・システムは、WebSphere Commerce が外部環境と通信する機能を提供します。そのような通信には、バックエンド・システムまたは外部システムとのメッセージの送受信と、WebSphere Commerce 内部でイベントが発生したことを顧客や管理者に知らせる通知送信などがあります。これは、インバウンド・システムとアウトバウンド・メッセージ・システムの 2 つのサブシステムで実現されます。インバウンド・システムは、バックエンド・システムおよび外部システムから来るインバウンド・メッセージを管理するのに対して、アウトバウンド・メッセージ・システムは、ユーザーへの通知の送信と、バックエンド・システムと外部システムへのアウトバウンド・メッセージの送信を担います。

たとえば、オーダーが配送されたことを顧客に通知する E メール・メッセージを送信するように、メッセージ・システムをセットアップすることができます。メッセージ・システムには、WebSphere Commerce をバックエンド・システムに組み込むためのメカニズムが備えられています。ストアでオーダーが作成されるたびに、バックエンド・システムにアウトバウンド・メッセージを送信するように、WebSphere Commerce を構成することも可能です。このオーダー情報は、必要なオーダー実行処理にバックエンド・システムで使用できます。バックエンド・システムはこの後、オーダー状況メッセージを WebSphere Commerce に返信し、オーダー・デリバリーの発生やオーダーの送り状の発行を知らせることができます。また、顧客を更新する E メールを送信することも可能です。

---

## 一般アプリケーション・エラー XML メッセージと一般システム・エラー XML メッセージ

WebSphere Commerce では、一般アプリケーション・エラーと一般システム・エラーが起きることがあります。これらのメッセージは XML 形式で現れ、アウトバウンド・キューに送信されます。

エラーがユーザーと関連がある場合、一般アプリケーション・エラー・メッセージはアウトバウンド・キューに送信されます。ユーザーが XML メッセージに無効なパラメーターを入力すると、ECAApplicationException が出されます。メッセージはアウトバウンド・キューに送信され、例外がログ・ファイルに記録されます。

**注:**このタイプの例外が出されると、Web コントローラーは、再試行可能コマンドとして指定されている場合でもコマンドを再試行しません。

一般システム・エラー・メッセージは、ランタイム例外または WebSphere Commerce 構成エラー (たとえば、ヌル・ポインター例外や変換ロールバック例外など) が検出されると、アウトバウンド・キューに送信されます。

一般 XML メッセージの内容は、インバウンド XML の内容によって異なりますが、その形式は以下のようになります。

```
<?xml version="1.0" encoding="Shift_JIS"?>
<WCS_Error type="GenericApplicationError"> (or <WCS_Error type="GenericSystemError">)
<RequestAttributes>
...(エラー・パラメーターおよびインバウンド XML メッセージのデータ)
<excMsg>(エラー・メッセージ・キー。_ERR_REMOTE_EXCEPTION など)</excMsg>
...
</RequestAttributes>
</WCS_Error>
```

---

## アウトバウンド・メッセージ構成のセットアップ

アウトバウンド・メッセージ用の構成サービスをセットアップして使用するには、以下のようになります。

- これらのメソッドのいずれかを使用して、トランスポートを適切なメッセージ・タイプに割り当てる。
  - サイトのメッセージ・タイプへのトランスポート・メソッドの割り当て

- ストアのメッセージ・タイプへのトランスポート・メソッドの割り当て  
使用されるそれぞれのトランスポートごとに、DEVICEFMT テーブルに指定されているような有効なデバイス形式を指定する必要があります。
2. トピック アウトバウンド・メッセージ・システム・インターフェースの情報を参照し、SendMsgCmd タスク・コマンドを使用してメッセージ・システム・オブジェクトを作成する。 setMsgType() および setStoreId() 初期化サービスを使用します。
  3. メッセージ・システムの構成メソッドを呼び出す。

---

## メッセージ・システム構成サービスの使用例

以下はメッセージ・システム構成サービスの使用法の例です。 DemoStore という名前のストアがあり、2つのトランスポート (E メールおよびファイル) を OrderAuthorized メッセージ・タイプに割り当てたい場合には、以下のようにします。

1. JSP ファイルの VIEWREG テーブルにエントリーを追加して、このアウトバウンド・メッセージの作成に使用する。 VIEWREG テーブルのキーは、ビュー名、ストア ID、およびデバイス形式 ID です。 VIEWREG テーブルの使用法についての詳細は、 *WebSphere Commerce プログラマーズ・ガイド* を参照してください。

**重要:** メッセージ・システムの構成サービスで使用するために作成した各表示は、インターフェースおよびクラス名のフィールドにメッセージ表示コマンドを使用しなければなりません。 また、これには JSP ファイルの名前が docname フィールドに含める必要があります。 要約すると、次のとおりです。

### INTERFACENAME

com.ibm.commerce.messaging.viewcommands.MessagingViewCommand

### CLASSNAME

com.ibm.commerce.messaging.viewcommands.MessagingViewCommandImpl

### PROPERTIES

以下の形式を使用して、JSP ファイル 'docname=jsp\_file' を指示します。

### DEVICEFMT\_ID

デバイス形式を示し、ユーザー・アプリケーションにカスタム・デバイス形式を使用していないかぎり、値 -3 (標準デバイス形式) を使用する必要があります。 VIEWREG エントリーに指定された DEVICEFMT\_ID は、メッセージ・タイプをトランスポートに割り当てた時に選択したデバイス形式と一致していなければなりません。

たとえば、DemoStore のストア ID が 5 であり、viewname が OrderAuthorized の場合には、次の SQL ステートメントを使用してレコードを挿入できます:

```
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. 管理コンソールを使用して E メールおよびファイル・トランスポートを OrderAuthorized メッセージに割り当て、設定値を構成します。 これは、サイトまたはストアのいずれかのレベルの管理者権限を使用して実行できます。 サイト・レベルで設定を行うと、すべてのストアに対してそれをアクセスできるようになります。
3. コマンドのインプリメンテーションでは、 OrderAuthorized メッセージ・タイプのメッセージ ID および DemoStore のストア ID を使用して、 SendMsgCmd コマンドのインスタンスを生成し、メッセージ



ング・サービスを使用して、setMsgType() および setStoreID() メソッドを呼び出します。 サイト・レベル構成を使用する必要がある場合には、ストア ID として 0 を指定し、JSP 名の最後に "&storeId=no" を付加してください。 そうでない場合には、ユーザーのストア ID を使用してください。(ユーザーのストアの構成が存在しない場合には、デフォルトとして、メッセージ・システムはサイト・レベル構成を自動的に使用します。)

4. アウトバウンド・メッセージ・システム・インターフェースの compose メソッドを呼び出して、追加のパラメーターを TypedProperty オブジェクトの形式で渡します。 viewname を指定することによって、メッセージの構成時に使用されたメッセージ・タイプのデフォルトの viewname を指定変更しません。
5. メッセージを即時送信したい場合、あるいはトランザクションが正常にコミットされた後で送信したい場合には、 SendMsgCmd で sendImmediate または sendTransacted を呼び出してください。 各メソッドの使用の詳細については、そのメッセージ・システムの資料を参照してください。
6. SendMsgCmd の execute メソッドを呼び出して、送信を行います。

---

## メッセージ・システム構成サービスにおけるエラー処理

JavaServer Pages (JSP) の処理中にエラーが起こると、JavaServer Pages (JSP) の結果には、ランタイムによって生成された広範な情報が入るのが普通です。これが JavaServer Pages (JSP) 障害で所要の結果でないと、動作を予測可能にする可能性がある方法は 2 つあります。

最初の方法としては、JavaServer Pages (JSP) で、予期しないエラー (未キャッチの例外など) が検出されると実行される、エラー・ページを指定できます。この方法を使用するには、以下の行をメイン・ページに含めることが必要です:

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

また、以下の行を YourError.jsp に含めることも必要です:

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

**注:** エラーのイベントでは、メッセージ構成の結果が YourError.jsp の結果になります。詳細については、JSP 1.0 仕様を参照してください。

2 番目の方法としては、エラー・メッセージがアウトバウンド・メッセージとして処理されるようにする代わりに、例外として処理される出力を生成するエラー JavaServer Pages (JSP) を使用できます。これを行うには、エラーを処理する方式として ComposerError.jsp を使用してください。これは上述と同じ方法で行われますが、YourError.jsp の先頭は ComposerError.jsp の最初の 2 行で始める必要があります。これによって、構成ランタイムは、エラーが起こったことを検出し、例外を発生できます。ComposerError.jsp は、以下のディレクトリーにあります。

▶ 1000

```
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

LINUX

/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 の最初の 2 行は、次のとおりです。

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

このファイルを使用するために、`errorPage="ComposerError.jsp"` を設定することによってエラー・ページであることを指示できます。追加情報が必要な場合には、このファイルを別のファイル名 (`YourError.jsp` など) にコピーして、追加エラー情報を新規ファイル中のこれら 2 行の後に入れられます。その後で、メイン JSP ページ内に `errorPage="YourError.jsp"` を設定することになります。これらの 2 行の後に指定されているすべての追加出力が、構成サービスによってスローされる例外のテキストの一部になります。

---

## メッセージ・システム構成メソッドの呼び出し

アウトバウンド・メッセージ・システム・インターフェースの `compose()` メソッドを呼び出すに、以下のパラメーターを指定してください:

- `viewname`: VIEWREG テーブル中の既存レコードの VIEWNAME 列に指定されているとおりの、構成ビューの名前。VIEWREG テーブルの使用法についての詳細は、*WebSphere Commerce プログラマーズ・ガイド* を参照してください。

**重要:** VIEWREG テーブル中で参照されているレコードで、INTERFACENAME および CLASSNAME 列の値には、すべての WebSphere Commerce メッセージ・システム表示コマンドと関連付けられているインターフェースおよびクラスの名前が入っていなければなりません。インターフェースの名前は、`com.ibm.commerce.messaging.viewcommands.MessagingViewCommand` でなければなりません。クラスの名前は `com.ibm.commerce.messaging.viewcommands.MessagingViewCommandImpl` でなければなりません。

- `CommandContext`: `CommandContext` インターフェース、またはインターフェースをインプリメントする `CommandContextImpl` クラスの詳細については、*WebSphere Commerce プログラマーズ・ガイド* を参照してください。
- `TypedProperty`: 型付きプロパティの値はストリング、または `toString()` メソッドをインプリメントするオブジェクトでなければなりません。TypedProperty の詳細については、*WebSphere Commerce プログラマーズ・ガイド* を参照してください。

compose() メソッドは、使用可能になっていて現在のメッセージ・タイプに割り当てられているトランスポートごとに管理コンソールで表示コマンドを実行します。このメソッドは、以下の処理を実行します:

- 管理コンソールに定義されているとおりに、各トランスポートからの storeId およびデバイス形式の他に viewname パラメーターも使用します。これらの値は、VIEWREG テーブル中で表示コマンドを検索するために使用されます。
- TypedPropertyパラメーターに指定されている値を渡して、表示コマンドを実行します。このコマンドの実行時に、システムは viewname、storeId、およびデバイス形式 ID を使用して VIEWREG テーブル中の PROPERTIES 列で JSP テンプレートを検索します。JSP テンプレートが実行されて、TypedProperty パラメーター中の値が渡されます。
- JSP はメッセージを構成して、send メソッドがオブジェクト内で呼び出されると、そのメッセージを適切なトランスポート経由で送信します。送信は、構成が実行されたメッセージング・オブジェクトでランザクション化、即時、または要求応答を使用して実行される場合があります。



---

## 第 4 章 OrderItemStatus コマンド

OrderItemStatus コマンドは、Update\_WCS\_OrderStatus、Update\_NC\_OrderStatus、または Order Status Update メッセージをバックエンド・システムから受信すると、OrderStatus、OrderConfirmStatus、OrderInvoiceStatus、および OrderShippingStatus コマンドによって内部的に呼び出されます。このコマンドは、アイテムの既存のオーダーの状況に関する情報を更新します。

### パラメーター

このコマンドのパラメーターのリストは以下のとおりです。各パラメーターは、ORDISTAT テーブル中のフィールドに対応しています。

**versioning: String “TRUE” または “FALSE”。** オプション。デフォルトは **FALSE** です。

TRUE の場合には、versioning が使用可能になります。

**orderItemId: INTEGER**

テーブル ORDISTAT 中の ORDERITEMS\_ID に定義されているとおりの WebSphere Commerce オーダー・アイテム参照番号。これは、テーブル ORDERITEMS 中の列 ORDERITEMS\_ID を参照する外部キーです。

**merchantItemNumber: STRING**

OIMITEM 列に定義されているとおりにバックエンド・システムに生成されるオーダー・アイテム番号。

**PartNumber: String**

PARTNUMBER 列に定義されるアイテム商品番号/SKU です。

**UnitOfMeasure: String**

OIUOFM 列に定義されるアイテムの計測単位です。

**RequestQuantity: Integer**

OIQTREQUEST 列に定義されているとおりに要求されるアイテムの数量。

**ConfirmQuantity: Integer**

OIQTCONFIRM 列に定義されているとおりに確認されるアイテムの数量。

**ShipQuantity: Integer**

OIQTSHIP 列に定義されているとおりの配送済みアイテムの数量。

**Currency: String**

OICPCUR 列に定義される表示価格の ISO 4217 通貨タイプです。

**UnitPrice: BigDecimal(20,4)**

OIUNPRC 列に定義される商品の単価です。

**PriceTotal: BigDecimal(20,4)**

OIPRTOT 列に定義されるアイテムの商品価格合計です。

**TaxTotal: BigDecimal(20,4)**

OITXTOT 列に定義されるアイテムの税合計です。

**ShippingTotal: BigDecimal(20,4)**

OISHTOT 列に定義されるアイテムの配送料合計です。

**ShippingTaxTotal: BigDecimal(20,4)**

OISHTXTOT 列に定義されるアイテムの配送料金税合計です。

**Status: String**

OISTATUS 列に定義されるアイテムの状況です。

**PlaceDateTime: Timestamp**

OIPLTIME 列に定義されるアイテムが実際に発行される日付です。

**RequestShipDateTime: Timestamp**

OIRSTIME 列に定義されるアイテムの配送要求日です。

**ScheduleShipDateTime: Timestamp**

OISSTIME 列に定義されるアイテムの配送予定日です。

**ActualShipDateTime: Timestamp**

OIASTIME 列に定義されるアイテムの実配送日です。

**InvoiceDateTime: Timestamp**

OIINVTIME 列に定義されるアイテムの送り状送付日です。

**InvoiceValue: BigDecimal(20,4)**

OIINVVAL 列に定義されるオーダーの送り状送付の所要日です。

**itemShipCondition: STRING**

OISCOND 列に定義されるアイテムの分割配送を受諾するかどうかを指定するコードです。'SC': 一回の配送、'SP': 分割配送

**itemComment: STRING**

OICMNT 列に定義されるアイテム状況に関するコメントです。

**field1: Integer**

FIELD1 列に定義されるアイテム状況のカスタマイズ・フィールド 1 です。

**field2: BigDecimal(15,2)**

FIELD2 列に定義されるアイテム状況のカスタマイズ・フィールド 2 です。

**field3: String**

FIELD3 列に定義されるアイテム状況のカスタマイズ・フィールド 3 です。

**動作**

- オーダーの初回オーダー・アイテム状況には orderItemId および merchantItemNumber が指定されている必要があります。
- orderItemId に指定されているオーダー・アイテムが ORDERITEMS テーブル中に存在しているかどうか調べます。
- ORDISTAT テーブル中にすでにエントリーがある各後続のオーダー・アイテム状況では、orderItemId は、既存の明細アイテムのためにバックエンド・システムによって新規明細アイテムを作成できるようにする必要はありません。
- 指定されている情報を使用して ORDISTAT テーブル中の行を作成または更新します。
- バージョン管理が使用可能になっていて、これがオーダーの初回オーダー・アイテム状況であると、新規行が ORDISTAT テーブル中に作成されることとなります。そうでない場合には、最終オーダー・アイテム状況のコピーが作成されて、そのバージョン (OIVERNBR) が既存バージョンの最大数 +1 に設定されます。指定された情報のすべてが入っている新規行が、バージョン (OIVERNBR) が 0 に設定されて作成されます。

**例外条件**

- 指定されている orderItemId が ORDERITEMS テーブル中の有効なオーダー番号ではありません。

- orderItemId または merchantItemNumber のいずれかが、特定オーダー・アイテムの初回オーダー・アイテム状況に指定されていません。
- orderItemId および merchantItemNumber が、後続のオーダー・アイテム状況更新で特定オーダー・アイテム状況の ORDISTAT テーブル中のものと不一致です。

---

## GetPickPackListDetail コマンド

このコマンドは、Inquire\_WCS\_PickPackListDetail XML メッセージの、ピック・チケット情報およびパッキング・リスト情報を検索します。

### パラメーター

#### pickBatchId

PICKBATCH テーブルの PICKBATCH\_ID 列で定義されているピッキング・バッチの参照番号。

### 動作

- pickBatchId が PICKBATCH テーブルにあるかどうかを検査します。
- その pickBatchId に対する ORDRELEASE テーブルにレコードが少なくとも 1 つあることを検査します。
- PICKBATCH テーブルからピック・チケット XML を、そして指定された pickBatchId のパッキング・リスト XML を検索し、Report\_WCS\_PickPackListDetail メッセージを組み立てる表示タスクに転送します。
- 応答として、ピック・チケット情報とパッキング・リスト情報を含む Report\_WCS\_PickPackListDetail メッセージが送信されます。

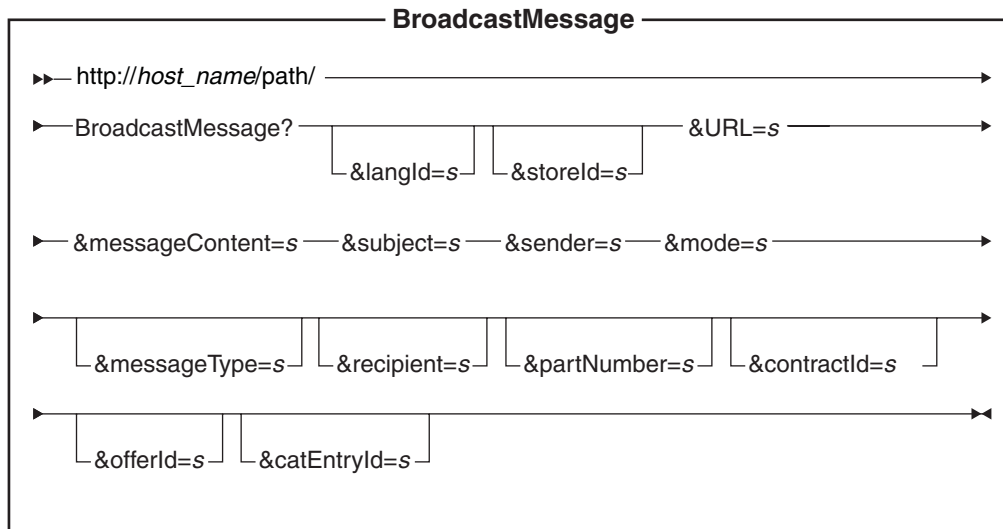
### 例外条件

- PICKBATCH または ORDRELEASE テーブルで、pickBatchId が無効、または、ありません。

---

## BroadcastMessage コマンド

このコマンドは、E メールを顧客にブロードキャストします。



### パラメーター値

**http://host\_name/path/**

WebSphere Commerce サーバーおよび構成パスの完全修飾名。

**langId** セッション期間中の優先言語を設定またはリセットします。ストアでサポートされている言語は、STORELANG テーブルにあります。

**storeId**

メッセージを送信するストアの参照番号

**URL** コマンドが正常に完了した時に呼び出される URL。

**messageContent**

送信しようとしているメッセージの内容。指定すると、この内容がメッセージ・テンプレートの内容に変更されます。 `messageType` パラメーターが指定されている場合は、このパラメーターを指定する必要があります。

**subject**

E メール「件名」行の内容。件名を指定しないと、メッセージ・タイプ構成に指定されているデフォルトの件名が使用されます。

**送信者** E メール「送信側」行の内容。 `sender` を指定しないと、メッセージ・タイプ構成に指定されているデフォルト `sender` が使用されます。

**mode** 可能な値:

1. 指定されたストアで商品を購入したことがあり、販売促進の E メールを受信する意思表示をした、すべてのショッパーにメッセージを送信します。
2. あるストアについて、指示された SKU をそのストアで購入し、販売促進の E メールを受信する意思表示をした、すべてのショッパーにメッセージを送信します。
3. 販売促進の E メールを受信する意思表示をした、すべてのショッパーにメッセージを送信します。

**messageType**

MSGTYPES テーブルの MSGTYPE\_ID 列で定義されている事前定義メッセージ・テンプレートへの参照。 `messageType` を指定しないと、ブロードキャスト・メッセージングのサイト・レベル・メッセージ・タイプが使用されます。デフォルトでは、サイト・レベル・メッセージ・タイプには関連したメッセージ・テンプレートがないため、ストアが独自のブロードキャスト・メッセージ・タイプをセットアップすることが推奨されます。新規メッセージ・タイプをセットアップする方法の詳細については、メッセージ・ログ を参照してください。

**宛先** E メール追加の宛先を、コンマ区切りのリストで指定します。

**partNumber**

モード 2 で E メールと関連付けられているアイテムの部品番号。

**contractId**

モード 2 でこの契約に基づいて購入済みの宛先メンバー。

**offerId**

モード 2 でこのオファーに基づいて購入済みの宛先メンバー。

**catEntryId**

モード 2 でこのカタログ・エントリを購入手続き済みの宛先メンバー。

**動作**

- `GetRecipientsCmd` タスク・コマンドを呼び出して、ブロードキャスト E メールメッセージ宛先のリストを作成する。



- 返信用 E メール・アドレス、メッセージ件名、メッセージ、および宛先のコンマ区切りリストのパラメーターを指定することによって別のタスク・コマンドを呼び出します。
- メッセージの本体には、指定されたメッセージ・タイプに対応するテンプレートを使用する。

#### 例外条件

宛先のリストを判別できなかつたり、メッセージを配信中にエラーがあった場合には、BroadcastMessageErrorView エラー・タスクが呼び出されます。

---

## OrderInvoiceStatus コマンド

OrderInvoiceStatus コマンドは、OrderInvoiceの OrderStatusType タグ値を持つ Update\_WCS\_OrderStatus XML メッセージを、バックエンド・システムから受信したときに使用されます。このコマンドは、既存オーダーの送り状状況に関する情報を更新します。パラメーター、動作、および例外条件は OrderStatus コマンドと同じです。2つのコマンド間の相違は、OrderInvoiceStatus コマンドは orderStatus パラメーターの値をデフォルト値 I (メッセージ中に指定されていないと、オーダーが送り状を発効済みであるということの意味する) に設定する点だけです。

OrderInvoiceStatus コマンドのパラメーター、動作、および例外の処理についての詳細は、OrderStatus コマンドを参照してください。

---

## OrderShippingStatus コマンド

OrderShippingStatus コマンドは、OrderShippingの OrderStatusType タグ値を持つ Update\_WCS\_OrderStatus XML メッセージを、バックエンド・システムから受信したときに使用されます。このコマンドは、既存オーダーの配送状況に関する情報を更新します。パラメーター、動作、および例外条件は OrderStatus コマンドと同じです。2つのコマンド間の相違は、OrderShippingStatus コマンドは orderStatus パラメーターの値をデフォルト値 S (メッセージ中に指定されていないと、オーダーが配送済みであるということの意味する) に設定する点だけです。

OrderShippingStatus コマンドのパラメーター、動作、および例外の処理についての詳細は、OrderStatus コマンドを参照してください。

---

## OrderStatus コマンド

OrderStatus コマンドは、Update\_WCS\_OrderStatus、Update\_NC\_OrderStatus、または Order Status Update メッセージが、バックエンド・システムから受信されると使用されます。このコマンドは、既存オーダーの状況に関する情報を更新します。

#### パラメーター

##### sequenceNumber

String。シリアルライゼーションのための WCS オーダー状況のシーケンス番号です。

##### lastUpdateTimestamp:

タイム・スタンプ形式の String。シリアルライゼーションのための、WebSphere Commerce オーダー状況の最終更新タイム・スタンプ。

**versioning: String “TRUE” または “FALSE”。** デフォルトは **FALSE** です。

TRUE の場合には、バージョン管理が使用可能になります。

下記パラメーターは ORDSTAT テーブルの列に対応しています。

**orderId: Integer。**

テーブル ORDSTAT 中の ORDERS\_ID 列に定義されているとおりの WebSphere Commerce オーダー参照番号。この参照番号は、ORDERS テーブル中の列 ORDERS\_ID を参照する外部キーです。

**merchantOrderNumber: STRING。必須。**

OSMORDER 列に定義されている通りにバックエンド・システム内で生成されているオーダー番号。

**currency: String**

OSPCUR 列に定義される表示価格の ISO 4217 通貨タイプです。

**priceTotal: BigDecimal(20,4)**

OSPRTOT 列に定義されるオーダーの商品価格合計です。

**taxTotal: BigDecimal(20,4)**

OSTXTOT 列への定義されるオーダーの税合計です。

**shippingTotal: BigDecimal(20,4)**

OSSHOT 列に定義されるオーダーの配送料合計です。

**shippingTaxTotal: BigDecimal(20,4)**

OSSHXTOT 列に定義されるオーダーの配送料金税合計です。

**orderStatus: String**

OSSTATUS 列に定義されるオーダーの状況です。

**placeDateTime: タイム・スタンプ**

OSPLTIME 列に定義されるオーダーが実際に発行される日付です。

**requestShipDateTime: タイム・スタンプ**

OSRSTIME 列に定義されるオーダーの配送要求日です。

**scheduleShipDateTime: タイム・スタンプ**

OSSSTIME 列に定義されるオーダーの配送予定日です。

**actualShipDateTime: タイム・スタンプ**

OSASTIME 列に定義されるオーダーの実配送日です。

**invoiceDateTime: タイム・スタンプ**

OSINVTIME 列に定義されるオーダーの送り状送付日です。

**invoiceValue: BigDecimal(20,4)**

OSINVVAL 列に定義されるオーダーの送り状送付の所要日です。

**shipCondition: String**

OSSCOND 列に定義されるオーダーの分割配送を受諾するかどうかを指定するコードです。'SC': 一回の配送、'SP': 分割配送

**shippingModeFlag: String**

OSSMFLAG 列に定義される、配送先住所および配送モードがオーダー・レベルか、またはオーダー・アイテム・レベルであることを指示するコード。O': オーダー・レベル、'I': アイテム・レベル

**comment: String**

OSCMNT 列に定義されるオーダー状況に関するコメントです。

**field1: Integer**

FIELD1 列に定義されるオーダー状況のカスタマイズ・フィールド 1 です。

**field2: BigDecimal(15,2)**

FIELD2 列に定義されるオーダー状況のカスタマイズ・フィールド 2 です。

**field3: String**

FIELD3 列に定義されるオーダー状況のカスタマイズ・フィールド 3 です。

**items: ハッシュ・テーブルのベクトル**

各ハッシュ・テーブルは 1 アイテムのパラメーターを表します。

**動作**

- オーダーの初回オーダー状況には orderId および merchantOrderNumber が指定されていなければなりません。
- orderId に指定されているオーダーが ORDERS テーブル中に存在しているかどうか調べます。
- sequenceNumber および lastUpdateTimestamp などのシリアルイゼーション情報が利用可能である場合には、値が ORDSTAT テーブル中の値より最新であるかどうかを調べ、最新でない場合には、更新は実行されません。
- 指定されたすべての情報を使用して ORDSTAT テーブル中の行を作成または更新します。
- バージョン管理が使用可能になっていて、これがオーダーの初回オーダー状況である場合には、新規行が ORDSTAT テーブル中に作成されることとなります。そうでない場合には、最終オーダー状況のコピーが作成されて、そのバージョン (OSVERNBR) が既存バージョンの最大数 +1 に設定されます。指定されたすべての情報が入っている新規行が、その OSVERNBR が 0 に設定されて作成されます。
- テーブル ORDERS の STATUS 列の値を 'G' に更新します。

**例外条件**

- 指定されている orderId が ORDERS テーブル中の有効なオーダー番号ではありません。
- orderId または merchantOrderNumber のいずれかが、特定オーダーの初回オーダー状況に指定されていません。
- orderId および merchantOrderNumber が後続のオーダー状況更新で特定オーダーのために ORDSTAT テーブル中にすでに存在しているものと不一致です。
- シリアルイゼーション情報が指定されていて、オーダー状況メッセージが順序通りではありません。

---

**ProductOfferPriceUpdate コマンド**

ProductOfferPriceUpdateコマンドは、Update\_WCS\_ProductPrice、Update\_NC\_ProductPrice、および商品価格更新メッセージの商品価格情報の更新に使用されます。

**パラメーター****offerId**

オファーを識別する参照番号。

**partNumber**

CATENTRY テーブルの PARTNUMBER 列に定義されている通りのカタログ・エントリーの部品番号。

**memberId**

カタログ・エントリーの所有者を識別する参照番号。

**catEntryId**

販売用にオファーされているカタログ・エントリー。

**currency**

価格を表示する ISO 4217 通貨タイプ。 この値は必須です。

**offerPrice**

この更新のために使用するオファー価格。

**precedence**

この更新のために使用する優先順位。

**tradingPositionContainerId**

オファーが一部である TradingPositionContainer。

**startDateTime**

オファーが有効である時刻範囲の開始。

**endDateTime**

オファーが有効である時刻範囲の終了。

**minimumQuantity**

このオファーのもとで単一オーダーで購入できる最小数量。

**maximumQuantity**

このオファーのもとで単一オーダーで購入できる最大数量。

**quantityUnit**

minimumQuantity および maximumQuantity の計測単位。

**動作**

- コマンドは OFFERPRICE テーブル中のレコードを更新します。
- 基本キーを判別する処理は、コマンドを実行するインバウンド・メッセージの形式によってわずかに異なります。 基本キーを判別する方法の説明については、特定のインバウンド・メッセージを参照してください。

**例外条件**

- currency パラメーターがエンプティータです。
- offerId パラメーター値が OFFERPRICE テーブル中に見つかりません。
- カタログ・エントリーが、partNumber と一緒にストアを所有している、memberId を使用して見つかりません。
- 一致している offerId の tradingPositionContainerId パラメーター値が OFFER テーブル中に見つかりません。
- precedence パラメーター値が最大値を超えています。 優先順位は  $10^{16}$  より小でなければなりません。

---

## ProductInventoryUpdate コマンド

ProductInventoryUpdateコマンドは、 Update\_NC\_ProductInventory メッセージ、 Update\_WCS\_ProductInventory メッセージ、または商品数量の更新メッセージの、商品在庫の更新に使用されます。

**パラメーター****catEntryId**

テーブル INVENTORY 中の CATENTRY\_ID 列に定義されている、WebSphere Commerce カタロ

グ・エントリー。カタログ・エントリーがエンプティである場合には、部品番号と、ストアおよびカタログの両方を所有しているメンバー ID の組み合わせが、カタログ・エントリーを得るために使用されることとなります。

#### **partNumber**

CATENTRY テーブルの PARTNUMBER 列に定義されている通りのカタログ・エントリーの部品番号。 catEntryId パラメーターがエンプティである場合には、カタログを所有しているメンバー ID と一緒に、テーブル CATENTRY 中のキー catEntryId を得るために使用されます。

#### **storeId**

テーブル INVENTORY 中の列 STORE\_ID を参照するストア ID。カタログ・エントリーおよびデフォルト配送センターと一緒に、これはテーブル INVENTORY 中の行へのキーです。

#### **inventoryQuantity**

テーブル INVENTORY 中の QUANTITY 列に定義されている通りの数量。

#### **fulfillmentCenterID**

テーブル INVENTORY 中の列 FFMCENTER\_ID を参照する配送センター ID。これは FFMCENTER データベース・テーブルに対するキーです。

#### **動作**

- コマンドは INVENTORY テーブル中のレコードを更新します。
- ストア ID (INVENTORY テーブル中の STORE\_ID を指す) は必須です。
- catalogEntryId および storeId が INVENTORY テーブル中の行を更新するために使用されます。
- catEntryId が存在していない場合には、storeId (STORE\_ID) がメンバー ID (CATENTRY テーブル中の MEMBER\_ID を指す) を得るために使用されます。メンバー ID は、カタログの所有者と同じものでなければなりません。メンバー ID (MEMBER\_ID) は、partNumber (CATENTRY テーブル中の PARTNUMBER を指す) と一緒に catEntryId を得るために使用されます。catEntryId は、そのカタログ・エントリーのストア ID およびデフォルト配送センター ID と一緒に、INVENTORY テーブル中の行を更新するために使用されます。
- INVENTORY テーブル中の行が存在していないと、エラーが起こります。

#### **例外条件**

このコマンドは、以下の例外が検出されると、エラー・ログ中にエントリーを生成します。

- storeId が存在していません。
- catEntryId が、partNumber と一緒にストアを所有している、メンバー ID を使用して見つかりません。
- catEntryId が、storeId およびデフォルト配送センター ID と一緒に、一致している行をテーブル INVENTORY 中で見つけられません。

---

## **SendWCSOrder タスク・コマンド**

SendWCSOrder コマンドは、Order Create レガシー・メッセージをバックエンド・システムに送信するために、WebSphere Commerce システムで使用されます。

#### **動作**

- タスク・コマンドは、そのコマンドを OrderProcess コマンド内の OrderMessagingCmd インターフェースに割り当てることによって使用可能にされます。使用可能になっていると、OrderProcess コマンドが処理を終了する前に呼び出されます。

- オーダー参照番号をその入力パラメーターとして使用すると、必要なオーダー情報のすべてが収集されます。
- 次にこのコマンドは、使用可能なオーダー情報を基にして、 `Order Create` レガシー・メッセージを `String` として作成して、それをアウトバウンド・メッセージ・システム内容設定サービスを使ってメッセージに保管します。
- メッセージ作成が成功すると、コマンドはアウトバウンド・メッセージ・システム送信サービスを使用してメッセージを送信しようとします。

#### 例外条件

このコマンドは、例外が検出されると、エラー・ログ中にエントリーを生成します。

---

## SendXMLOrder コマンド

`SendXMLOrder` コマンドは、`Report_NC_PurchaseOrder XML` メッセージをバックエンド・システムに送信するために、`WebSphere Commerce` アウトバウンド・メッセージ・システムによって使用されます。このコマンドがメッセージ構成テンプレートを使用して `XML` メッセージを生成し、そのメッセージをアウトバウンド・メッセージ・システムがバックエンド・システムに送信します。

#### 動作

- タスク・コマンドは、そのコマンドを `OrderProcess` コマンド内の `OrderMessagingCmd` インターフェースに割り当てることによって使用可能にされます。
- 使用可能になっていると、`OrderProcess` コマンドが処理を終了する前に呼び出されます。
- このタスク・コマンドは、`OrderCreateXML.jsp` 構成テンプレートを使用して必要なオーダー情報を収集して `Report_NC_PurchaseOrder` アウトバウンド `XML` メッセージを作成する、メッセージ・システム構成サービスを呼び出します。
- 構成が成功すると、コマンドはアウトバウンド・メッセージ・システム送信サービスを使用してメッセージを送信しようとします。

#### 例外条件

このコマンドは、例外が検出されると、エラー・ログ中にエントリーを生成します。

---

## NewInboundMessage コマンド

`NewInboundMessage` コマンドは、`user_template.xml` インバウンド・メッセージ・テンプレート定義ファイルを使用してインプリメントされていない、カスタマイズ済みのインバウンド・メッセージに対して使用されます。このコマンドは、アダプターがインバウンド・メッセージを認識しないときに実行されます。これは、`sys_template.xml` または `user_template.xml` インバウンド・メッセージ・テンプレート定義ファイルに定義されているのが、レガシー・メッセージでも `XML` メッセージでもないということです。最初に、`NewInboundMessage` コマンドにはプログラミング・ステートメントが何も入っていないので、コマンドを自分でカスタマイズしなければなりません。

#### 動作

- インバウンド・メッセージは、`getMessage()` メソッドを使用して検索できる `String` バッファに保管します。

#### 例外

performExecute() メソッドがインプリメントされていないと、デフォルトで例外が生成されます。

---

## OrderConfirmStatus コマンド

OrderConfirmStatus コマンドは、OrderConfirmの OrderStatusType タグ値を持つ Update\_WCS\_OrderStatus XML メッセージを、バックエンド・システムから受信したときに使用されます。このコマンドは、既存オーダーの確認状況に関する情報を更新します。パラメーター、動作、および例外条件は OrderStatus コマンドと同じです。2つのコマンド間の相違は、OrderConfirmStatus コマンドは orderStatus パラメーターの値をデフォルト値 C (メッセージ中に指定されていないと、オーダーが確認済みであるということを意味する) に設定する点だけです。

OrderConfirmStatus コマンドのパラメーター、動作、および例外の処理についての詳細は、OrderStatus コマンドを参照してください。

---

## ProductListPriceUpdate コマンド

ProductListPriceUpdate コマンドは、Update\_WCS\_ProductPrice XML メッセージの商品価格情報の更新に使用されます。このコマンドは、商品の表示価格に関する WebSphere Commerce データベース中の情報を更新します。

### パラメーター

#### partNumber

CATENTRY テーブルの PARTNUMBER 列に定義されている通りのカタログ・エントリーの部品番号。

#### memberId

カタログ・エントリーの所有者を識別する参照番号。

#### catEntryId

販売用にオファーされているカタログ・エントリー。

#### currency

価格を表示する ISO 4217 通貨タイプ。この値は必須です。

#### listPrice

LISTPRICE テーブルを更新するために使用される商品表示価格。

### 動作

- コマンドは LISTPRICE テーブル中のレコードを更新します。
- partNumber は memberId と一緒に、カタログ・エントリー・キー (テーブル CATENTRY 中の CATENTRY\_ID) を得るために使用されます。
- カタログ・エントリーが LISTPRICE テーブル中の既存のカタログ・エントリーと一致しているが、通貨タイプがそのカタログ・エントリーのレコードのいずれかの通貨タイプと一致していない場合には、新規レコードが LISTPRICE テーブル中に作成されることになります。これにより、価格を異なる通貨で同じカタログ・エントリーに指定できます。

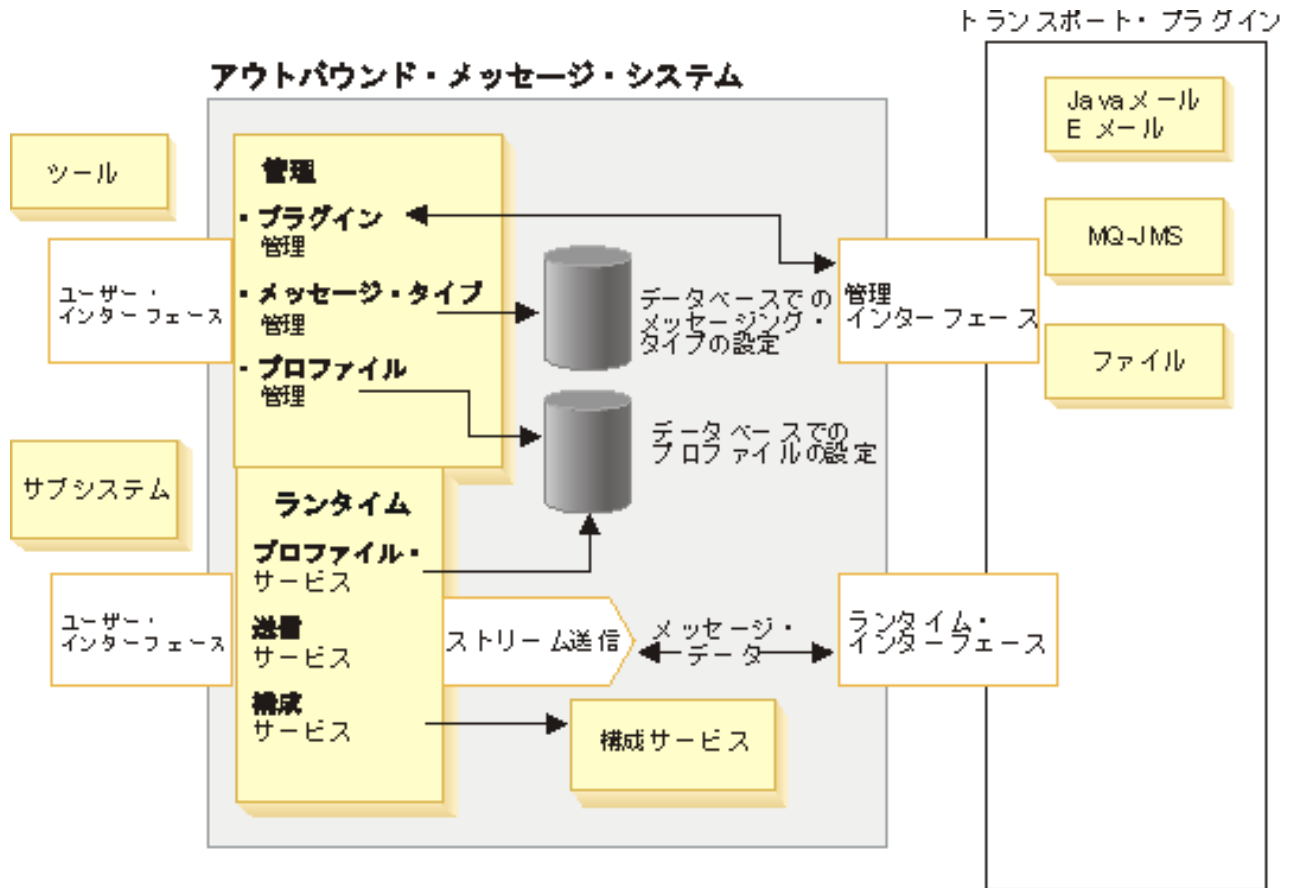
### 例外条件

- currency パラメーターがエンプティイです。
- カタログ・エントリーが、partNumber と一緒にストアを所有している、memberId を使用して見つかりません。





## 第 5 章 アウトバウンド・メッセージ・システムの処理方法



メッセージ・システムは、Common Connector Framework (CCF) をインプリメントするプラグイン・モデルを使用して、システムとさまざまなトランスポートとの間の共通インターフェースを提供します。システム管理として、サイトおよびストア管理者は、以下のタスクを実行できます。

- トランスポートを追加し、使用可能にし、構成する。管理者は、管理コンソールを使用して設定を作成します。メッセージ・システムとトランスポートとの間の通信は、1つの管理インターフェースを通じて行われます。
- トランスポートを個々のメッセージ・タイプに割り当て、設定がそれぞれに使用されるよう指示して、プロファイルを保守する。管理者は、管理コンソールを使用して、これを行います。

実行時、メッセージが WebSphere Commerce サブシステムによって生成されると、以下のイベントが発生します。

1. 該当するプロファイルが、メッセージ・タイプについて検索される。そのメッセージにストア・プロファイルが存在しない場合、サイト・プロファイルが使用されます。プロファイルは、どのトランスポート・メソッドや設定が使用されるかを決定するのに使用されます。
2. メッセージが構成サービスを使用する場合、テンプレートを使用してメッセージを生成する。
3. メッセージは、ランタイム・インターフェースを通じてトランスポートに送信され、トランスポートは通知を送付する。

外部トランスポートとの共通インターフェースを使用すると、トランスポートのインプリメンテーション詳細情報を、メッセージ・システムの運用と分離することができます。このアーキテクチャーのおかげで、CCF インターフェースに準拠するさらに別のトランスポートをプラグインすることができます。

---

## アウトバウンド・メッセージ・システム

WebSphere Commerce メッセージ・システムは、WebSphere Commerce 内で生成されたメッセージの定義と送信を全面的に管理する手段になります。このシステムを使うと、管理者、顧客、およびバックエンド・システムと配送センター・システムに対してどのようにカスタマー・オーダーやシステム・エラーなどのさまざまなイベントを通知するかを制御することができます。

アウトバウンド・メッセージ・システムを構成するには、管理コンソールを使用してください。メッセージ・システムは、SMTP を使用した E メールおよび UTF-8 エンコードを使用したファイルなどのトランスポートを使用してメッセージを送信できます。Eメールの場合にサポートされるアウトバウンド・プロトコルは SMTP であり、メッセージのエンコード方式は指定されている言語によって異なります。選択によっては、MQSeries を使用してバックエンド・システムまたは配送センター・システムにメッセージを送信するように、メッセージ・システムを構成することができます。

アウトバウンド・メッセージ・システムのランタイム環境は、きわめて自在にカスタマイズ可能なメッセージ環境として機能します。そのようなフィーチャーには以下のものがあります。

- 構成サービス  
事前定義されている JSP テンプレートを使用してメッセージをカスタマイズします。
- 複数メッセージ送信サポート  
複数のトランスポートによって単一メッセージを送信できます。
- 同一トランスポートによる複数通知メッセージ  
これは、複数の宛先にブロードキャスト E メールを送信するために役立ちます。
- 以下の 3 つの処理タイプのサポート:
  - 処理後  
現在のトランザクションの正常終了時に送信するメッセージの場合に使用します。
  - 即時  
イベントが WebSphere Commerce で発生すると送信されるメッセージの場合に使用します。このメッセージは、トランザクションがコミットされているかどうかかわらず送信されます。
  - 要求応答  
バックエンド・システムからの応答メッセージを必要とするメッセージの場合に使用します。

**注:** トランスポート属性 (たとえば、E メール・アドレスなど) およびファイル場所が有効であることを確認してください。メッセージ・システムは、属性を妥当性検査しません。属性が誤っていると、メッセージ送信で障害が起きることになります。

---

## アウトバウンド・メッセージ・システム管理

アウトバウンド・メッセージ・システムの管理は、サイト管理とストア管理の 2 つの主要カテゴリーに分けることができます。サイト管理者は、どのトランスポートが使用できるかなど、サイト内のすべてのストアで使用される基本的なフレームワークを提供します。たとえば、ストアがあらかじめ構成済みで、サイト管理者によって使用可能にされていない限り、そのストアは E メールをトランスポートとして使用できない場合があります。その場合、ストア管理者はサイト・レベルで行われた設定を受け入れるか、あるいはその設定をストアに合わせて変更することができます。

---

## アウトバウンド・バックエンド組み込みメッセージ

アウトバウンド・バックエンド・メッセージは、外部システムへ送信できる WebSphere Commerce 生成要求です。オーダーが発行されたことをバックエンド・システムに通信するための Report\_NC\_PurchaseOrder XML アウトバウンド・メッセージを生成するよう、WebSphere Commerce を構成することができます。XML メッセージは、Unicode UTF-8 形式でエンコードされ、アウトバウンド・メッセージ・システムで生成および送信されます。また、レガシー・オーダー作成メッセージを使用することもできます。これは、同様の機能を実行します。

アウトバウンド・メッセージには、WebSphere Commerce サーバーから外部システムに送信されるオーダー情報が入っています。外部システムでは、さらにオーダー実行処理が行われます。アウトバウンド・メッセージを使用可能にするには、まず Report\_NC\_PurchaseOrder XML メッセージまたはレガシー・オーダー作成メッセージのどちらを使用するかを選択する必要があります。この 2 つを両方同時に使用可能にすることはできません。

Report\_NC\_PurchaseOrder XML メッセージを使用する場合、以下の SQL ステートメントを使用して、CMDREG データベース・テーブルを更新してください。

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

これで、SendXMLOrderCmdImpl タスク・コマンドが割り当てられます。これは、メッセージを生成して、OrderProcess コマンドの OrderMessagingCmd インターフェースに送信します。

レガシー・オーダー作成メッセージを使用する場合、以下の SQL ステートメントを使用して、データベースの CMDREG テーブルを更新してください。

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

これで、SendWCSOrderCmdImpl タスク・コマンドが割り当てられます。これは、メッセージを生成して、OrderProcess コマンドの OrderMessagingCmd インターフェースに送信します。変更を有効にするには、WebSphere Commerce インスタンスを再始動する必要があります。

新規のアウトバウンド・バックエンド組み込みメッセージを作成することもできます。

---

## アウトバウンド配送組み込みメッセージ

アウトバウンド配送組み込みメッセージは、配送センター・システムに送信できる WebSphere Commerce 生成の要求です。配送センター・システムに伝える情報を入れて、インバウンド・メッセージに対する応答としてアウトバウンド・メッセージを生成するよう WebSphere Commerce を構成することができます。

アウトバウンド XML メッセージは、Unicode UTF-8 形式でエンコードされ、アウトバウンド・メッセージ・システムで生成および送信されます。以下のアウトバウンド・メッセージを生成する JSP ファイルについての詳細は、メッセージ構成テンプレートを参照してください。

Response\_WCS\_ExpectedInvRecord アウトバウンド・メッセージを使うと、予定在庫レコードを求める要求の作成時に配送センター・システムに応答することができます。このメッセージには、生成された RA\_ID と RADETAIL\_ID が添付されます。アウトバウンド・メッセージは Create\_WCS\_ExpectedInventoryRecord メッセージによって呼び出され、RACreateResult.jsp ファイルによって生成されます。

Response\_WCS\_PickBatch アウトバウンド・メッセージを使うと、ピッキング・バッチが作成されたことを配送センター・システムに知らせることができます。 アウトバウンド・メッセージは、Create\_WCS\_PickBatch メッセージによって呼び出され、PickBatchResult.jsp ファイルによって生成されます。

Report\_WCS\_PickPackListDetail アウトバウンド・メッセージは、Inquire\_WCS\_PickPackListDetail メッセージによって呼び出されますが、特定の PICKBATCH\_ID の詳細情報を送信することで配送センターに回答するのに使われます。 このメッセージには、ORDRELEASE テーブルに入っている特定の PICKBATCH\_ID のピック・チケットおよびパッキング・リストが添付されます。

Response\_WCS\_CreateInvReceipt アウトバウンド・メッセージは、在庫領収書が作成されたことを配送センター・システムに知らせる手段になります。 アウトバウンド・メッセージは、Create\_WCS\_InventoryReceipt メッセージによって呼び出され、CreateInvReceiptOK.jsp ファイルによって生成されます。

Response\_WCS\_UpdateInvReceipt アウトバウンド・メッセージを使うと、在庫は受領後ただちに更新されたことを配送センター・システムに回答することができます。 アウトバウンド・メッセージは、Update\_WCS\_InventoryReceipt メッセージによって呼び出され、UpdateInvReceiptOK.jsp ファイルによって生成されます。

Response\_WCS\_CreateShipConfirm アウトバウンド・メッセージは、Create\_WCS\_ShipmentConfirmation メッセージによって呼び出され、配送確認が作成されたことを配送センター・システムに回答するのに使われます。

Release\_WCS\_ShipmentNotify アウトバウンド・メッセージは、オーダーの配送時に顧客に通知するかどうかのオプションを提供するアウトバウンド E メール・メッセージです。

---

## アウトバウンド・メッセージ・システム・ストア管理

ストア管理者は、ストアが使用するトランスポート・メソッドを使用可能にする責任を担います。 ストア管理者は、ストアのトランスポート・メソッドの追加、アクティブ化、非アクティブ化、および構成を行うことができ、トランスポート・メソッドをメッセージ・タイプに割り当てます。 ストア管理者には、サイト管理者が作成した設定を受け入れることも、それを変更することもできます。 以下は、ストア管理に関連するタスクのリストです。

- トランスポート・メソッドの追加。
- トランスポート・メソッドのアクティブに化または非アクティブ化。
- トランスポート・メソッドの構成。
- トランスポート・メソッドのメッセージ・タイプへの割り当て。

ストア管理者がサイト・レベルの設定を変更してしまうと、サイト管理者が特定の設定にどのような変更を加えても、そのストアは影響を受けません。ただし、それ以外の設定に加えられたサイト・レベルの変更は、ストア管理者によって修正されなければそのまま適用されます。たとえば、E メールが SMTP ホスト (smtp.host1.com) でサイトのために構成されているが、ストア A に smtp.host2.com が指定されていると、E メールに対するサイト・レベルのそれ以上のいかなる変更も、ストア A の Eメールのための設定に影響しません。

---

## アウトバウンド・メッセージ・システム・サイト管理

サイト管理者は、サイトがサポートするトランスポートを決定し、それらをサイト規模で構成します。サイト・レベルの管理では、ストア・レベル管理の設定に応じて変更可能なデフォルト設定が用意されています。

サイト管理者は、サイトのトランスポートおよびメッセージ・タイプをアクティブにして構成することができますが、ストア管理者によって独自の設定が指定されるようにすることもできます。以下に、サイト管理に関連したタスクをリストします。

- トランスポートの追加。
- トランスポートのアクティブ化または非アクティブ化。
- トランスポートの構成。これは、ストア管理者が変更できるデフォルト構成を提供します。
- メッセージ・タイプに対するトランスポートの割り当て。これらの割り当ては、ストア管理者が変更できます。

以下のタスクを実行できるのはサイト管理者だけです:

- 管理者に E メール・メッセージを送信するためのエラー通知を使用可能化
- バックエンド・システムにメッセージを送信するための MQSeries JMS トランスポートの使用可能化
- 既存オーダーの状況について更新するためのオーダー状況通知の使用可能化

---

## トランスポート・アダプターへの新規メッセージの追加

WebSphere Commerce を使うと、トランスポート・アダプターを拡張して追加メッセージを処理することができます。これにはメッセージの XML テンプレートの作成およびコントローラー・コマンドへのメッセージのマッピングが必要です。これを行うには、XML と WebSphere Commerce コントローラー・コマンドを熟知している必要があります。メッセージで新しいコントローラー・コマンドを作成する必要がある場合は、WebSphere Commerce データベース・スキーマと Java プログラミングも熟知している必要があります。



---

## 第 6 章 アウトバウンド・メッセージ送信/受信の送信サービスを使用可能にする

WebSphere Commerce は、アウトバウンド・メッセージの送信/受信の送信サービスを通して、ほかのシステムと対話することができます。これによって、WebSphere Commerce はほかのシステムにメッセージを送信したり、応答を待機したりすることができます。送信/受信の動作は、メッセージ要求を受信したシステムからの返答を待機する点を除いて、SendImmediate メソッドに類似します。

要求メッセージの送信後、WebSphere Commerce はインバウンド・キューを listen し、要求メッセージのメッセージ ID と等しい相関 ID 付きの返答メッセージがキューに置かれるまで待機します。

送信/受信に対しては、MQSeries アダプターに対して使用されるキューとは別のキューを使用してください。

WebSphere Commerce が送信/受信メッセージ・サービスを使用できるようにするには、以下のようになります。

1. 送信/受信に使用される新しいメッセージに対して、MSGTYPES テーブルでメッセージ・タイプを作成する。  
VIEWNAME 列の値は VIEWNAME で、送信するメッセージを生成する場合に使用されます。
2. sendReceiveImmediate送信モード、およびステップ 1 で作成されたメッセージ・タイプ ID を使用するコマンドを作成する。これは SendMsg インターフェースを呼び出すのに使用されます。
3. アウトバウンド・メッセージを生成するには、VIEWREG テーブルにエントリーを生成し、ステップ 1 で作成された VIEWNAME を JSP ファイルに関連付ける。エントリーには以下の値を使用してください。

### INTERFACENAME

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

### CLASSNAME

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

### PROPERTIES

`docname=JSP filename`

4. JSP ファイルを作成し、アウトバウンド・キューに送信するメッセージを生成する。
5. 管理コンソールを使用して、送信/受信に割り当てられているトランスポートがアクティブであることを確認する。
6. 管理コンソールを使用して、ステップ 1 で作成した各メッセージ・タイプを定義する。モードフィールドに 0 を指定します。これは送信/受信通信モードを示します。

---

## メッセージ・システム・トランスポート・アダプターを使用可能にする

メッセージ・システム・トランスポート・アダプターを使用可能にするには、以下のようになります。

1. 構成マネージャーを立ち上げる。
2. **ホスト名** → **インスタンス** を選択し、**コンポーネントフォルダー** をオープンする。
3. **TransportAdapter** を選択する。
4. **コンポーネント使用可能**の横のチェック・ボックスがアクティブになっていることを確認して、**適用**をクリックする。

5. 構成マネージャーを終了する。
6. WebSphere Application Server を再始動する。

---

## エラー状況メッセージ・タイプのトランスポートへの割り当て

エラー・メッセージが、個々のストアではなく、サイトに割り当てられていることを確認します。 トランスポート・メソッドにエラー状況メッセージ・タイプを割り当てるには、以下のようになります。

1. 管理コンソールをオープンし、サイト管理者としてログオンする。
2. 「構成」メニューから、「メッセージ・タイプ」を選択する。「メッセージ・タイプ構成」ページが表示されます。
3. 「新規」をクリックする。「メッセージのトランスポート割り当て」ページがオープンされます。
  - a. 「メッセージ・タイプ」ドロップダウン・リストから「WebSphere Commerce で発生するエラー条件の説明」を選択する。
  - b. 「メッセージ重大度」フィールドに 0 0 を指定する。
  - c. 「トランスポート」ドロップダウン・リストから「E メール」を選択する。
  - d. 「デバイス形式」ドロップダウン・リストから「標準デバイス形式」を選択する。
4. 「次へ」をクリックする。「メッセージ・トランスポート割り当てパラメーター」ページが表示されません。
  - a. 「ホスト」フィールドにメール・サーバーの完全修飾名を入力する。
  - b. 「プロトコル」フィールドに smtp と入力する。
  - c. 「宛先」フィールドに、エラー通知メッセージを受信する管理者を指定する。複数の宛先を指定する場合には、それらの宛先名をコンマで区切ってください。
  - d. 「送信側」フィールドにメッセージの送信側を指定する。  
このテキストは E メール・メッセージの「送信側」フィールドに現れます。
  - e. 「件名」フィールドにメッセージの件名を指定する。  
このテキストは E メール・メッセージの「件名」フィールドに現れます。
5. 「終了」をクリックする。

---

## ストアのトランスポート・メソッドのアクティブ化

ストアの既存のトランスポート・メソッドをアクティブにするには、以下のようになります。

1. 管理コンソールをオープンし、ストア管理者としてログオンする。
2. 「構成」メニューから、「トランスポート」を選択する。「トランスポートの構成」ページが表示されます。
3. アクティブにしたいトランスポートの横のチェック・ボックスをクリックする。
4. 「状況の変更」をクリックする。ページが再ロードされ、状況が変更されます。

---

## ストアへのトランスポート・メソッドの追加

ストアに新規トランスポート・メソッドを追加するには、以下のようになります。

1. 管理コンソールをオープンし、ストア管理者としてログオンする。
2. 「構成」メニューから、「トランスポート」を選択する。「トランスポートの構成」ページが表示されます。
3. 「追加」をクリックする。「トランスポートの追加」ダイアログがオープンされます。



4. ストアに追加したいトランスポートの横にあるチェック・ボックスを選択する。左上にあるチェック・ボックスを選択すれば、すべてのトランスポートを選択することができます。利用可能なトランスポートがなければ、サイト管理者によって利用可能にされたすべてのトランスポートがすでに追加されています。
5. 「追加」をクリックしてトランスポートを追加するか、または「キャンセル」をクリックして「トランスポート構成」ページに戻る。

---

## ストアのトランスポート・メソッドの構成

ストアにトランスポート・メソッドを構成するには、以下のようになります。

1. 管理コンソールをオープンし、ストア管理者としてログオンする。
2. 「構成」メニューから、「トランスポート」を選択する。「トランスポートの構成」ページが表示されます。
3. 構成したいトランスポートの横のチェック・ボックスを選択する。
4. 「構成」をクリックする。「トランスポート構成パラメーター」ページがオープンします。選択したトランスポートの名前が、パラメーター・テーブルの左上に表示されます。
5. 該当するフィールドにトランスポートの情報を指定する。
6. 「OK」をクリックして変更を受け入れるか、または「キャンセル」をクリックして、変更せずに「トランスポート構成」メニューに戻る。

注: 設定値を有効にしたい場合以外は、「OK」はクリックしないようにしてください。「OK」をクリックすると、このトランスポートの構成に対してサイト管理者によって行われた変更は、このストアには影響しなくなります。変更を行わなかった場合、あるいは設定値を有効にたくない場合には、「キャンセル」をクリックしてください。

---

## ストアのトランスポート・メソッドの非アクティブ化

ストアの既存のトランスポート・メソッドを非アクティブにするには、以下のようになります。

1. 管理コンソールをオープンし、ストア管理者としてログオンする。
2. 「構成」メニューから、「トランスポート」を選択する。「トランスポートの構成」ページが表示されます。
3. 非アクティブにしたいトランスポートのチェック・ボックスをクリックする。
4. 「状況の変更」をクリックする。ページが再ロードされ、トランスポート状況が非アクティブになったことを示します。

---

## ストアのメッセージ・タイプへのトランスポート・メソッドの割り当て

ストア管理者は、サイト管理者によって作成されたトランスポート・メソッドの割り当てを受け入れたり、あるいはそれらをストア用に変更することができます。ストアの特定メッセージ・タイプに使用されるトランスポート・メソッドを割り当てるには、以下のようになります。

1. 管理コンソールをオープンし、ストア管理者としてログオンする。
2. 「構成」メニューから、「メッセージ・タイプ」を選択する。「メッセージ・タイプ構成」ページが表示されます。
3. トランスポートを割り当てたいメッセージ・タイプの横のチェック・ボックスをクリックして、「変更」をクリックする。メッセージ・タイプがリストにない場合、「新規」をクリックする。「メッセージのトランスポート割り当て」ページがオープンされます。

4. これが新規のトランスポートの割り当てである場合、「メッセージ・タイプ」ドロップダウン・リストからトランスポートが割り当てられるメッセージ・タイプを選択する。
5. 適切なフィールドにトランスポートの構成値を入力する。一般に、「メッセージ重大度」として 0,0、そして標準デバイス形式が推奨されます。
6. 「次へ」をクリックして、指定したメッセージ・タイプ用のトランスポート・パラメーターを構成する。
7. このメッセージ・タイプに対して選択したトランスポート用の属性を入力する。
8. 変更を保存するには「終了」を、「メッセージ・タイプ構成」ページに戻るには「キャンセル」をクリックする。

**注:** 設定値を有効にしたい場合以外は、「終了」はクリックしないようにしてください。「終了」をクリックすると、このトランスポートの構成に対してサイト管理者によって行われた変更は、このストアには影響しなくなります。変更を行わなかった場合、あるいは設定値を有効にたくない場合には、「キャンセル」をクリックしてください。

---

## サイトのトランスポート・メソッドのアクティブ化

サイトの既存のトランスポート・メソッドをアクティブにするには、以下のようにします。

1. 管理コンソールをオープンし、サイト管理者としてログオンする。
2. 「構成」メニューから、「トランスポート」を選択する。「トランスポートの構成」ページが表示されます。
3. アクティブにしたいトランスポートの横のチェック・ボックスをクリックする。
4. 「状況の変更」をクリックする。ページが再ロードされ、トランスポートの状況がアクティブになったことを示します。

---

## サイトへのトランスポート・メソッドの追加

サイトに新規トランスポート・メソッドを追加するには、以下のようにします。

1. 管理コンソールをオープンし、サイト管理者としてログオンする。
2. 「構成」メニューから、「トランスポート」を選択する。「トランスポートの構成」ページが表示されます。
3. 「追加」をクリックして、「トランスポートの追加」ページをオープンする。
4. サイトに追加する意向があるトランスポートの横のチェック・ボックスを選択する。先頭のチェック・ボックスを選択して、トランスポートのすべてを選択できます。
5. 「追加」をクリックして変更を受け入れるか、または「キャンセル」をクリックして、「トランスポート構成」ページに戻る。  
サイトにトランスポート・メソッドを追加した場合には、それが自動的にアクティブにされます。

---

## サイトのメッセージ・タイプへのトランスポート・メソッドの割り当て

サイトの特定メッセージ・タイプに使用されるトランスポート・メソッドを割り当てるには、以下のようにします。

1. 管理コンソールをオープンし、サイト管理者としてログオンする。
2. 「構成」メニューから、「メッセージ・タイプ」を選択する。「メッセージ・タイプ構成」ページが表示されます。

3. トランスポートを割り当てたいメッセージ・タイプの横のチェック・ボックスをクリックして、**変更**をクリックする。メッセージ・タイプ がリストにない場合、「**新規**」をクリックする。「メッセージのトランスポート割り当て」ページがオープンされます。
4. これが新規のトランスポートの割り当てである場合、「**メッセージ・タイプ**」ドロップダウン・リストからトランスポートが割り当てられるメッセージ・タイプを選択する。
5. 適切なフィールドにトランスポートの構成値を入力する。一般に、「**メッセージ重大度**」として0,0、そして標準デバイス形式が推奨されます。
6. 「**次へ**」をクリックして、指定したメッセージ・タイプ用のトランスポート・パラメーターを構成する。
7. このメッセージ・タイプに対して選択したトランスポート用の属性を入力する。
8. 「**終了**」または「**キャンセル**」をクリックして、「**メッセージ・タイプ構成**」ページに戻る。

---

## サイトのトランスポート・メソッドの構成

サイトにトランスポート・メソッドを構成するには、以下のようにします。

1. 管理コンソールをオープンし、サイト管理者としてログオンする。
2. 「**構成**」メニューから、「**トランスポート**」を選択する。「トランスポートの構成」ページが表示されます。
3. 構成したいメソッドの横のチェック・ボックスを選択する。
4. 「**構成**」をクリックする。「トランスポート構成パラメーター」ダイアログがオープンされます。選択したトランスポート・メソッドの名前が、パラメーター・テーブルの左上に表示されます。
5. トランスポート・メソッドで使用される値を入力する。
6. 「**OK**」 をクリックして変更を受け入れるか、または「**キャンセル**」 をクリックして、「トランスポート構成」ページに戻る。

---

## サイトのトランスポート・メソッドの非アクティブ化

サイトの既存のトランスポート・メソッドを非アクティブにするには、以下のようにします。

1. 管理コンソールをオープンし、サイト管理者としてログオンする。
2. 「**構成**」メニューから、「**トランスポート**」を選択する。「トランスポートの構成」ページが表示されます。
3. 非アクティブにしたいトランスポートの横のチェック・ボックスをクリックする。
4. 「**状況の変更**」をクリックする。ページが再ロードされ、状況が変更されます。

---

## E メール・トランスポート・メソッドのシステム設定値の検査

E メール・トランスポートに対するデフォルトの設定値が設定されていることを、以下のようにして確認します。

1. 構成マネージャーを立ち上げる。
2. 「**インスタンス**」を選択して、次に、「**トランスポート**」→「**アウトバウンド**」→「**JavaMail**」→「**ConnectionSpec**」フォルダーをオープンする。
3. 右側のフレームの「**拡張**」タブをクリックする。
4. **host** の値を SMTP メール・サーバーに設定する。
5. **protocol** の値を smtp に設定する。

6. **適用**をクリックする。
7. WebSphere Commerce サーバーを再始動する。

**注:** これによって、E メール・トランスポートに対するシステム単位のデフォルトの設定値が提供されます。この設定値は管理コンソールで作成された設定値によって変更されます。

---

## 通知のアクティブ化

E メール・トランスポート方法のシステム・デフォルトを設定した後、以下のようにして通知をアクティブにします。

1. 以下のように設定して、`<instance_name>.xml`を編集する。

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

2. WebSphere Commerce サーバーを再始動する。

**注:** エラー通知用の Eメールの送信時に問題が生じる場合は、以下のアクションを取る必要があります。

- 構成値が正しいことを確認する。
- ストア・レベルのトランスポートを構成するすべてのエントリーを削除することによって、すべての構成データをクリーンアップする。以下の SQL ステートメントで、すべてのエントリーが削除されます。

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

- 管理コンソールをオープンして、すべてのメッセージ・タイプが正しく構成されていることを確認し、ストア・レベルではトランスポートを構成しない。

---

## エラー通知を使用可能にする

E メール・エラー通知を使用可能にするには、以下のようになります。

1. E メール・トランスポート・メソッドに対するシステム・デフォルトの設定値が設定されていることを確認する。
2. エラー・メッセージの通知をアクティブにする。
3. トランスポートにエラー状況メッセージ・タイプを割り当てる。
4. サイトのメッセージ・タイプにトランスポート・メソッドを割り当てる。

「メッセージ・トランスポート割り当て」ページで以下の値を入力します。

- a. 「**メッセージ・タイプ**」ドロップダウン・リストから「WebSphere Commerce で発生するエラー条件の説明」を選択する。
- b. 「**メッセージ重大度**」フィールドに 0 0 を指定する。
- c. 「**トランスポート**」ドロップダウン・リストから「E メール」を選択する。
- d. 「**デバイス形式**」ドロップダウン・リストから「標準デバイス形式」を選択する。

「メッセージ・トランスポート割り当てパラメーター」ページで以下の値を使用します。

- a. 「**ホスト**」フィールドにメール・サーバーの完全修飾名を入力する。

- b. 「**プロトコル**」フィールドに smtp と入力する。
- c. 「**宛先**」フィールドに、エラー通知メッセージを受信する管理者を指定する。複数の宛先を指定する場合には、それらの宛先名をコンマで区切ってください。
- d. 「**送信側**」フィールドにメッセージの送信側を指定する。  
このテキストは E メール・メッセージの「送信側」フィールドに現れます。
- e. 「**件名**」フィールドにメッセージの件名を指定する。  
このテキストは E メール・メッセージの「件名」フィールドに現れます。

5. WebSphere Commerce サーバーを再始動する。

**注:** エラー通知用の Eメールの送信時に問題が生じる場合は、以下のアクションを取る必要があります。

- 構成値が正しいことを確認する。
- ストア・レベルのトランスポートを構成するすべてのエントリーを削除することによって、すべての構成データをクリーンアップする。以下の SQL ステートメントで、すべてのエントリーが削除されます。

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

- 管理コンソールをオープンして、すべてのメッセージ・タイプが正しく構成されていることを確認し、ストア・レベルではトランスポートを構成しない。

---

## 配送通知 Eメールを使用可能にする

顧客のオーダーの配送時に、配送通知 Eメールが送信されるようにするには、以下のようになります。

- 以下の SQL ステートメントを使用して CMDREG データベース・テーブルを更新する。  

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

**注:** デフォルトでは、CMDREG データベース・テーブルで CLASSNAME として ReleaseShipNotifyDummyImpl を使用して、Release\_WCS\_ShipmentNotify メッセージが使用不可になっています。ReleaseShipNotifyDummyImpl は、何も実行しないダミー・インプリメンテーションです。

---

## ブロードキャスト・メッセージを使用可能にする

ブロードキャスト Eメール・メッセージを送信するには、以下のようになります。

1. BroadcastMessage.jsp という名前の JSP ファイルを作成する。
2. そのファイルをサイトまたはストア・ディレクトリーに入る。
3. 管理コンソールをオープンし、サイトまたはストア管理者としてログオン b b。
  - a. 「**構成**」メニューから、「**メッセージ・タイプ**」を選択する。「メッセージ・タイプ構成」ページが表示されます。
  - b. 「**新規**」をクリックする。「メッセージのトランスポート割り当て」ページがオープンされます。
    - 1) 「**メッセージ・タイプ**」ドロップダウン・リストから「ブロードキャスト・メッセージ」を選択する。
    - 2) 「**メッセージ重大度**」フィールドに 0 0 を指定する。
    - 3) 「**トランスポート**」ドロップダウン・リストから「Eメール」を選択する。
    - 4) 「**デバイス形式**」ドロップダウン・リストから「標準デバイス形式」を選択する。
  - c. 「**次へ**」をクリックする。「メッセージ・トランスポート割り当てパラメーター」ページが表示されます。

- 1) 「**ホスト**」フィールドにメール・サーバーの完全修飾名を入力する。
  - 2) 「**プロトコル**」フィールドに `smtp` と入力する。
  - 3) 「**宛先**」フィールドにデフォルト宛先を指定する。複数の宛先を指定する場合には、それらの宛先名をコンマで区切ってください。宛先名は実行時に顧客 E メール・アドレスで置き換えられます。
  - 4) 「**送信側**」フィールドにメッセージの送信側を指定する。このテキストは E メールに「送信側」行の内容として現れます。
  - 5) 「**件名**」フィールドにメッセージの件名を指定する。このテキストは E メールに「件名」行の内容として現れます。
- d. 「**終了**」をクリックする。
- e. メッセージを送信するには、`BroadcastMessage` コマンド 参照ファイルの指示に従う。  
たとえば、サイト内の任意のストアから部品番号 “sku1234” を購入したすべての顧客にメッセージを送信するには、ブラウザのアドレス行に次のように入力する:

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

---

## オーダー状況通知を使用可能にする

オーダー状況メッセージに関する通知を使用可能にするには、以下のようになります。

1. WebSphere Commerce マシン上でデータベース・コマンド・ウィンドウをオープンする。
2. 以下の SQL ステートメントを入力することによってオーダー状況通知コマンドを登録する。

```
update cmdreg set  
classname='com.ibm.commerce.messaging.commands.OrderStatusNotifySendCmdImpl'where  
interfacename='com.ibm.commerce.messaging.commands.OrderStatusNotifySendCmd'
```
3. WebSphere Commerce サーバーを停止する。
4. 必要であれば、サイトまたはストアにトランスポート・メソッドを追加する。
5. “オーダー状況の通知メッセージ” タイプをトランスポートに割り当てる。「**デバイス形式**」ドロップダウン・リストで「**HTTP ブラウザー**」を選択するようにします。

---

## 管理コンソールからメッセージを送信できるようにする

InFashion サンプル・ストアを基にしたストアの場合には、WebSphere Commerce 管理コンソールを使用して、顧客サービス担当者が顧客にメッセージを送信できるようにすることができます。WebSphere Commerce 管理コンソールからメッセージを送信できるようにするには、以下のようになります。

1. 管理コンソールをオープンし、サイト管理者またはストア管理者としてログオンする。
2. 「**構成**」メニューから、「**メッセージ・タイプ**」を選択する。「**メッセージ・タイプ構成**」ページが表示されます。
3. 「**新規**」をクリックする。「**メッセージのトランスポート割り当て**」ページがオープンされます。
  - a. **メッセージ・タイプ**ドロップダウン・リストから「顧客サービス担当者によって送信されるオーダー関連メッセージ」を選択する。
  - b. 「**メッセージ重大度**」フィールドに `0 0` を指定する。
  - c. 「**トランスポート**」ドロップダウン・リストから「**E メール**」を選択する。
  - d. 「**デバイス形式**」ドロップダウン・リストから「**標準デバイス形式**」を選択する。

4. 「次へ」をクリックする。「メッセージ・トランスポート割り当てパラメーター」ページが表示され  
ます。
  - a. 「ホスト」フィールドにメール・サーバーの完全修飾名を入力する。
  - b. 「プロトコル」フィールドに smtp と入力する。
  - c. 「宛先」フィールドにデフォルト宛先を指定する。複数の宛先を指定する場合には、それらの宛先名  
をコンマで区切ってください。宛先名は実行時に顧客 E メール・アドレスで置き換えられます。
  - d. 「送信側」フィールドにメッセージの送信側を指定する。  
このテキストは E メール・メッセージの「送信側」フィールドに現れます。(この値はストア・サー  
ビスで入力された E メール・アドレス値によってオーバーライドされます。)
  - e. 「件名」フィールドにメッセージの件名を指定する。  
このテキストは E メール・メッセージの「件名」フィールドに現れます。(この値はストア・サー  
ビスで入力された値によってオーバーライドされます。)
5. 「終了」をクリックする。

---

## アウトバウンド OrderCreate メッセージを使用可能にする

OrderCreate アウトバウンド・メッセージは、XML 形式かレガシー形式のいずれかで使用可能にすることが  
できます。この 2 つの形式は、以下のタスク・コマンドによって生成されます。

1. SendXMLOrder は Report\_NC\_PurchaseOrder XML メッセージを生成する。
2. SendWCSOrder は Order Create レガシー・メッセージを生成する。

WebSphere Commerce の以前のバージョンからマイグレーションして既存の形式を保持したい場合を除い  
て、XML 形式を使用することをお勧めします。両方を使用可能にすることはできません。

Report\_NC\_PurchaseOrder XML メッセージを使用可能にするには、以下の SQL ステートメントを使用し  
て、データベース CMDREG テーブルを更新してください。

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

Order Create レガシー・メッセージを使用可能にするには、以下の SQL ステートメントを使用してデー  
タベース CMDREG テーブルを更新してください。

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





## 第 7 章 メッセージ・タイプ

WebSphere Commerce アウトバウンド・メッセージ・システムは、種々のメッセージ・タイプを処理できます。各メッセージ・タイプは、WebSphere Commerce システム内で起こる特定のタイプのイベントに対する応答で、メッセージ・システムに送信されます。メッセージ・システムは、管理コンソールに指定されたメッセージ・タイプおよびメッセージ設定に従ってメッセージを処理します。以下の表は、アウトバウンド・メッセージ・システムによってサポートされるメッセージ・タイプを示しています。

MSGTYPE テーブル内の メッセージ・タイプ	管理コンソールの名前	使用法
ErrorMessage	WebSphere Commerce で発生したエラー状態の説明	WebSphere Commerce でエラーが発生したときに管理者が E メール・メッセージを受信できるようにするには、このメッセージ・タイプを構成してください。  このメッセージを使用可能にするには、エラー通知を使用可能にするを参照してください。
OrderCreateFixFormat	WebSphere Commerce のオーダー作成のためのアウトバウンド・メッセージ	オーダーが WebSphere Commerce で作成済みであることを示します。このメッセージは、アウトバウンド WebSphere Commerce オーダー作成メッセージを、バックエンド・システムに送信するために使用できます。  アウトバウンド OrderCreate メッセージを使用可能にするを参照してください。
OrderCreateXMLFormat	WebSphere Commerce の XML 作成のためのアウトバウンド・メッセージ	オーダーが WebSphere Commerce で作成済みであることを示します。このメッセージは、アウトバウンド WebSphere Commerce オーダー作成メッセージを、バックエンド・システムに送信するために使用できます。  アウトバウンド OrderCreate メッセージを使用可能にするを参照してください。
OrderStatusNotify	オーダー状況の通知メッセージ	オーダーの状況が変更済みであることを示します。  オーダー状況通知を使用可能にするを参照してください。

MSGTYPE テーブル内の メッセージ・タイプ	管理コンソールの名前	使用法
OrderAuthorized	与信済みオーダーのメッセージ	<p>オーダーが与信済みであることを示します。</p> <p>与信済みオーダー・メッセージを送信するには、 JSP テンプレートを作成する必要があります。</p> <p>OrderProcess コマンドを参照してください。</p>
OrderReceived	受信済みオーダーのメッセージ	<p>オーダーが受信済みであることを示します。</p> <p>受信済みオーダー・メッセージを送信するには、 JSP テンプレートを作成する必要があります。</p> <p>OrderProcess コマンドを参照してください。</p>
OrderRejected	拒否されたオーダーのメッセージ	<p>オーダーが拒否済みであることを示します。</p> <p>拒否済みオーダー・メッセージを送信するには、 JSP テンプレートを作成する必要があります。</p> <p>OrderProcess コマンドを参照してください。</p>
OrderCancel	キャンセルされたオーダーの通知メッセージ	<p>オーダーがキャンセル済みであることを示します。</p> <p>キャンセル済みオーダー・メッセージを送信するには、JSP テンプレートを作成する必要があります。</p> <p>OrderProcess コマンドを参照してください。</p>
PasswordNotify	パスワード・リセットの通知メッセージ	<p>顧客のパスワードがリセットされたことを示す E メール・メッセージを顧客に送信できるようにするには、このメッセージ・タイプを構成してください。</p>

MSGTYPE テーブル内の メッセージ・タイプ	管理コンソールの名前	使用法
BroadcastMessage	ブロードキャスト・メッセージ	顧客にブロードキャスト・メッセージを送信するには、このメッセージ・タイプを構成してください。  ブロードキャスト・メッセージを送信するには、ブロードキャスト・メッセージ JSP テンプレートを作成する必要があります。  BroadcastMessage コマンドを参照してください。
MerchantOrderNotify	マーチャントにオーダーを通知するためのメッセージ	OrderProcess コマンドの NotifyMerchant パラメーターと関連するもの。  通知メッセージを送信するには、JSP テンプレートを作成する必要があります。  OrderProcess コマンドを参照してください。
AdminOrderComment	顧客サービス担当者が送信するオーダー関連メッセージ	顧客サービス担当者が WebSphere Commerce アクセラレーターから顧客に E メール・メッセージを送信できるようにするには、このメッセージ・タイプを構成してください。
NotifyReleaseShip	オーダーが配送済みになったことを通知する、顧客へ送信されるメッセージ	顧客にオーダーの配送確認の E メール・メッセージを送信するには、このメッセージ・タイプを構成してください。  ReleaseShipConfirm コマンドを参照してください。

## アウトバウンド・メッセージ・システム・インターフェース

アウトバウンド・メッセージ・システムとの対話は SendMsgCmd タスク・コマンドによって実行できます。このタスク・コマンドは、メッセージ・システム・オブジェクトを構成して使用するのに必要なパラメーターを設定するために、メッセージ・システムによって提供されているメソッドに利用可能なすべてのインターフェースを外部化します。SendMsgCmd コマンド・インターフェースで利用可能なすべてのサービスのリストは、以下の通りです:

- 初期化サービス
- メッセージの内容設定サービス
- E メール部品または接続サービスの追加
- 構成可能メッセージ・データ・サービス
- 送信サービス
- その他のサービス

## アウトバウンド・メッセージ拡張

Report\_NC\_PurchaserOrder メッセージには、UserDataField を含む UserData XML が組み込まれています。SendXMLOrderMsg コマンドを拡張して、情報をヘッダー・レベルで追加したいか、あるいはアイテム・レベルで追加したいかに応じて、getHeaderExtensionRecords() メソッドまたは getItemExtensionRecords() メソッドのいずれかをインプリメントすることによって、UserData エレメントを Report\_NC\_PurchaserOrder アウトバウンド・メッセージ中に含まれていないエクストラ情報を渡すようにカスタマイズできます。デフォルトによって、getHeaderExtensionRecords() および getItemExtensionRecords() の両方のメソッドはヌル String 値を戻します。インプリメントされていると、両方のメソッドは、次のとおりに、一連の UserDataField エレメントが入っている String オブジェクトを戻します:

```
<UserDataField name="field_name">field_value</UserDataField>
```

・  
UserDataField 繰り返しループ

```
<UserDataField name="field_name">field_value</UserDataField>
```

コマンド拡張方法に関して詳しくは、*WebSphere Commerce プログラマーズ・ガイド* を参照してください。以下の表は、Report\_NC\_PurchaseOrder メッセージの XML エレメントのために呼び出されるメソッドのアウトラインです。

メッセージ	XML エレメント	呼び出されるメソッド
Report_NC_PurchaserOrder	Report_PO_Header	getHeaderExtensionRecords()
Report_NC_PurchaserOrder	Report_PO_Item	getItemExtensionRecords()

オーダー作成メッセージには USRLST レコードが組み込まれていて、このレコードには DATUSR\_DATA データ・セグメントが含まれています。DATUSR を使用して、メッセージにオプションのフィールドを追加することができます。SendWCSOrderMsg コマンドを拡張して、情報をヘッダー・レベルで追加したいか、あるいはアイテム・レベルで追加したいかに応じて、getHeaderExtensionRecords() メソッドまたは getItemExtensionRecords() メソッドのいずれかをインプリメントすることによって、DATUSR エレメントを Order Create アウトバウンド・メッセージ中に含まれていないエクストラ情報を渡すようにカスタマイズできます。デフォルトによって、getHeaderExtensionRecords() および getItemExtensionRecords() の両方のメソッドはヌル String 値を戻します。インプリメントされていると、両方のメソッドは、次のとおりに、一連の DATUSR レコードが入っている String オブジェクトを戻します:

```
<DATUSR>DATUSR_DATA</DATUSR>
```

:

: DATUSR 繰り返しループ

:

```
<DATUSR> DATUSR_DATA</DATUSR>
```

コマンドの拡張方法の詳細については、*WebSphere Commerce プログラマーズ・ガイド* を参照してください。以下のテーブルは、Order Create メッセージのセクションのために呼び出されるメソッドのアウトラインです:

メッセージ	USRLST セクション	呼び出されるメソッド
オーダーの作成	<HEADER>	getHeaderExtensionRecords()
オーダーの作成	<ITMDAT>	getItemExtensionRecords()

---

## 新規アウトバウンド・メッセージ・サポート

サポートされる XML および WebSphere Commerce アウトバウンド・メッセージに加えて、新しくアウトバウンド・メッセージのサポートを追加することができます。新規アウトバウンド・メッセージを追加するには、新規コントローラー・コマンドを作成して新規アウトバウンド・メッセージの内容を作成して、アウトバウンド・メッセージ・システムの送信サービスを使って、そのメッセージをバックエンド・システムに送信しなければなりません。

新規アウトバウンド・メッセージの内容を作成する方法は 2 つあります。最初の方法は、新規アウトバウンド・メッセージを自分の String バッファ内を作成して、メッセージの内容を直接設定するメッセージ内容設定サービスを使用することによってアウトバウンド・メッセージ・システムにそのメッセージを割り当てる方法です。この代替手段には、作成するコントローラー・コマンドでメッセージを作成する論理を組み込み必要があります。2 番目の方法は、新規アウトバウンド・メッセージをアウトバウンド・メッセージ・システム構成サービスを使用することによって作成する方法です。JSP テンプレートを使用することによって、構成サービスは、JSP テンプレート中に定義したメッセージのレイアウトおよび内容に従ってアウトバウンド・メッセージを生成します。

アウトバウンド・メッセージ・サービスを使用する方法については、「アウトバウンド・メッセージ・システム・インターフェース」の節を参照してください。コマンドの作成方法の詳細については、*WebSphere Commerce プログラマーズ・ガイド* を参照してください。

---

## アウトバウンド・メッセージの UserData エlement

すべてアウトバウンド・メッセージはオプションの Element として、UserData XML Element が組み込まれています。追加データを送信するには UserDataField レコードをこのメッセージに組み込んでください。メッセージに組み込まれていない特別なデータを渡すためのフィールドを、カスタマイズすることができます。

追加する新規フィールド名は UserDataField Element の名前属性にします。

以下の DTD は、UserData Element についての説明です:

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

以下は UserData Element の例です:

```
<UserData>
  <UserDataField name="field_name">field_value</UserDataField>
  .
  .UserDataField 繰り返しループ
  .
</UserData>
```

---

## アウトバウンド・メッセージ・システム・インターフェースのプログラミング例

以下の Java コード・セグメントは、アウトバウンド・メッセージ・システムとの対話をどのように実行できるかを示しています。例 1 は、新規 XML メッセージを作成して、そのメッセージをアウトバウンド・メッセージ・システム経由で送信する方法を示しています。例 2 は、E メール・メッセージを作成して、そのメッセージをアウトバウンド・メッセージ・システム経由で送信する方法を示しています。

### 例 1

```
try
{
    com.ibm.commerce.messaging.commands.SendMsgCmd api =
    (com.ibm.commerce.messaging.commands.SendMsgCmd)
    CommandFactory.createCommand(SendMsgCmd.NAME, getStoreId());
    // MSGTYPES テーブル中の msgType を 100 に設定しており、
    // ストア ID の 1 を使用しようとしているものとします。
    api.setMsgType(new Integer(100));
    api.setStoreID(new Integer(1));

    // msg の作成方法を選択しなければなりません。
    // 最初の選択: xml msg を String オブジェクトに作成してから、setContent() を使用します。
    String OrderCreateMsg = new String("<?xml version='1.0' encoding='UTF-8'?> ...");
    api.setContent(OrderCreateMsg);

    // あるいはテンプレート/ビュー名を渡すことによってメッセージ構成サービス (compose()) を使用します。
    // このビュー名は、JSP メッセージ・レイアウト・テンプレートを指している VIEWREG および MSGTYPES テーブルに
    // 登録されている必要があります。
    String viewName = new String("OrderCreateMsgView");
    TypedProperty tp = new TypedProperty();
    // orderRefNumber を読み取り tp に書き込みます
    tp.put("ORDER_REF_NUMBER", getOrderRn().toString());
    // languageId を読み取り tp に書き込みます
    tp.put("LANGUAGE_ID", getCommandContext().getLanguageId());
    // サービスを構成するために tp に保管されている viewName、コマンド・コンテキスト、およびパラメーターを渡します。
    // 正常終了時に、メッセージは、viewName によって指されている JSP メッセージ・レイアウト・テンプレートに
    // 定義されているメッセージ・レイアウトに従って作成されます。
    api.compose(viewName, getCommandContext(), tp);

    // メッセージを sendTransacted 送信サービスを使用して送出します。
    api.sendTransacted();
    // コントローラー・コマンドが得られたコマンド・コンテキストを設定します。
    api.setCommandContext(getCommandContext());
    // アウトバウンド・メッセージ・システム・サービスを実行します。
    api.execute();
}
catch (Exception ex )
{
    ex.printStackTrace(System.err);
}
```

### 例 2

```
try
{
    com.ibm.commerce.messaging.commands.SendMsgCmd api =
    (com.ibm.commerce.messaging.commands.SendMsgCmd)
    CommandFactory.createCommand(SendMsgCmd.NAME, getStoreId());
    // MSGTYPES テーブル中の msgType を 200 に設定しており、
    // ストア ID の 1 を使用しようとしているものとします。
    api.setMsgType(new Integer(200));
    api.setStoreID(new Integer(1));

    // msg の作成方法を選択しなければなりません。
    // 最初の選択: xml msg を String オブジェクトに作成してから、setContent() を使用します。
    String OrderNotifyMsg =
    new String("オーダーが受信されました。 お買い上げありがとうございます。");
    api.setContent(OrderNotifyMsg);
```

```

// あるいはテンプレート/ビュー名を渡すことによってメッセージ構成サービス (compose()) を使用します。
// このビュー名は、JSP メッセージ・レイアウト・テンプレートを指している VIEWREG および MSGTYPES テーブルに
// 登録されている必要があります。
String viewName = new String("OrderNotifyMsgView");
TypedProperty tp = null;
// サービスを構成するために tp に保管されている viewName、コマンド・コンテキスト、およびパラメーターを渡します。
// 正常終了時に、メッセージは、viewName によって指されている JSP メッセージ・レイアウト・テンプレートに
// 定義されているメッセージ・レイアウトに従って作成されます。
api.compose(viewName, getCommandContext(), tp);

// 件名、宛先、および送信側情報を構成可能メッセージ・データ・サービスを使用して設定します。
api.setConfigData("subject", "オーダーが受信されました");
api.setConfigData("recipient", getEmailAddress());
api.setConfigData("sender", "storeAdmin@storeABC.com");
// メッセージを sendImmediate 送信サービスを使用して送出します。
api.sendImmediate();
// コントローラー・コマンドが得られたコマンド・コンテキストを設定します。
api.setCommandContext(getCommandContext());
// アウトバウンド・メッセージ・システム・サービスを実行します。
api.execute();
}
catch (Exception ex )
{
ex.printStackTrace(System.err);
}

```

## メッセージ構成テンプレート

WebSphere Commerce アウトバウンド・メッセージ・システムには、幾つかのメッセージ・タイプの JavaServer Pages (JSP) 構成テンプレートが含まれています。これらのタイプのいずれかのメッセージが WebSphere Commerce 内に生成されるとき、メッセージ構成サービスは、対応するテンプレートを使用して、アウトバウンド・メッセージを作成します。作成されると、アウトバウンド・メッセージは、管理コンソールでメッセージ・タイプに割り当てられているトランスポートが何であろうと、そのトランスポートによって送信できます。メッセージ構成テンプレートを使用するメッセージ・タイプの例は、OrderCreateXMLFormat で、これは OrderCreateXML.jspテンプレートを使用します。この JavaServer Pages (JSP) テンプレートは以下のディレクトリーにあります。

**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**

**LINUX**

```
/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
```

以下のテーブルには、構成テンプレートを使用するメッセージ・タイプ、およびそれと関連した JSP ファイルが示されています。

メッセージ・タイプ	テンプレート
OrderCreateXMLFormat	OrderCreateXML.jsp
OrderStatusNotify	OrderStatusNotify.jsp
PasswordReset	PasswordResetNotification.jsp

これらの JSP ファイルはカスタマイズすることができます。

以下のテーブルに構成テンプレートを使用するその他のメッセージ・タイプを示します。これらのメッセージ・タイプを使用するには、ユーザー独自の JSP テンプレートをデフォルト名 (デフォルト名は VIEWREG テーブルに登録されている) で作成してください。

メッセージ・タイプ	デフォルトのテンプレート名
OrderAuthorized	OrderAuthorized.jsp
OrderReceived	OrderReceived.jsp
OrderRejected	OrderRejected.jsp
BroadcastMessage	BroadcastMessage.jsp
MerchantOrderNotify	MerchantOrderNotification.jsp
OrderCancel	OrderCanceledNotification.jsp

## 初期化サービス

以下のメソッドは、現在のメッセージに使用されているメッセージング・プロファイルを識別する初期パラメーターを設定します。これらのパラメーターは、管理コンソールで作成されて保守されている情報を検索します。

- `public void setMsgType(Integer msgType)` このメソッドは必須です。現在のメッセージのメッセージ・タイプを設定するために使用されます。
- `public void setStoreID(Integer storeId)` このメソッドは必須です。ストアのメッセージ・プロファイルを検索するために使用されます。サイト・レベル情報を検索するには、サイト・レベルのストア ID を使用できます。メッセージ・システムは、入力されたストアを基にしてプロファイルを検索しようとします。何も存在していない場合には、デフォルトのサイト ID を基にしてプロファイルを検索しようとします。
- `public void setPriority(Integer priority)` このメソッドは、オプションの初期化情報を提供します。指定されている優先順位の整数は検索するプロファイルを制限します。現在のメッセージが検索されるのは、優先順位範囲が含まれているプロファイルだけです。

## メッセージの内容設定サービス

構成サービスを使用するか、あるいはメッセージの内容を直接設定するかのいずれも可能です。メッセージ・システム構成サービスを使用するには、以下のサービスを使用してください:

- メッセージ・システム構成サービスを使用するには、以下のサービスを使用してください:
  - `public void compose( String viewName, CommandContext cmdContext, TypedProperty inParms)` このメソッドは、コンポーザー機能にアクセスします。ユーザーは、この機能によって JSP テンプレートを使用してメッセージの内容を設定できます。このトピックについて詳しくは、コンポーザーの資料を参照してください。 `cmdContext` パラメーターは、必要なコンテキスト情報をコンポーザーに指定します。ユーザーは、 `viewName` パラメーターによって、コンポーザーによってアクセスされる



JSPを決定できます。 `inParms` パラメーターは JSP に渡すデータを指定します。 どんな値を `TypedProperty` オブジェクトに入れられるかを左右しているルールがあります。 詳しくは、コンポーザーの資料を参照してください。

- メッセージの内容を直接設定するには、以下のサービスを使用してください:
  - `public void setContent(Integer transportId, Integer languageId, byte[] msgContent)` このメソッドによって、`msgContent` パラメーターを使用してメッセージの内容を設定できます。 その他の 2 つのパラメーターは存在していなければなりません。 `transportID` パラメーターによって、特定トランスポートに応じて内容を設定できます。 `languageID` パラメーターによって、特定言語に応じて内容を設定できます。 マルチリンガルのために内容の設定で柔軟性が最大になるように、JSP を実行するために構成メソッドを使用することをお勧めします。
  - `public void setContent(Integer transportId, Integer languageId, String msgContent)` この `setContent` メソッドは、内容をバイトの配列としてではなく `String` フォーマットで入力できる点を除き、前述と同じ機能を実行します。

---

## E メール部品または接続サービスの追加

一部の他のトランスポートの他に、標準 E メール・トランスポートによっても、添付をメッセージに追加できます。 メッセージ・システムのユーザーは、以下のメソッドによって、内容部分をまたは添付物をメッセージに添付できます。

- `public void addContentPart(byte[] msgAttachment)` このメソッドは、内容部分をメッセージに追加する単純な方法を提供します。 E メール・トランスポートの場合は、「内容部分」とは添付を指します。 `msgAttachment` パラメーターは、メッセージに追加する内容を表します。 これは、バイト形式に変換される、部分の実際の内容でなければならないことに注意してください。
- `public void addContentPart(byte[] msgAttachment, String partName, String partType)` このメソッドで、内容部分をメッセージに追加する場合の柔軟性がまします。 `msgAttachment` パラメーターは、メッセージに追加する内容を表します。 `partName` パラメーターは、内容部分に使用する名前を表します。 `partType` パラメーターは、送信しようとする部分の MIME タイプを表します。 たとえば、Mime Eメールの `partType` は `'text/plain'` とすることができます。

---

## 構成可能メッセージ・データ・サービス

以下の汎用メソッドは、メッセージに使用されるトランスポート・サービスを構成するために使用してください。

- `public void setConfigData(String key, String value)` ユーザーは、この汎用メソッドによってメッセージに使用されるトランスポート・サービスを構成できます。 `key` パラメーターは、変更する属性を識別するために使用される管理名を指しています。 `value` パラメーターは、割り当てる値です。 このメソッドを呼び出すと、ここに指定する値が、管理コンソールで割り当てた値を変更します。 メッセージ・システムに利用可能なデフォルト・トランスポートおよびそれぞれに設定される属性が入っている下のテーブルを参照してください。 Eメールの宛先を設定するための別の方法については、後述の `addMember` メソッドを参照してください。

メッセージ・システムに利用可能なデフォルト・トランスポートには、以下の属性が含まれています:

トランスポート	属性キー	説明
E メール	subject	Eメールの件名。
	recipient	宛先の E メール・アドレス。
	sender	送信者の E メール・アドレス。
	host	メッセージを送信するために使用するメール・ホスト。
	protocol	メール・ホストに接続するのに使用されるプロトコル。
ファイル	location	ファイルを書き込むロケーション。
	FileName	ファイルを書き込む名前。
	mode	実行する書き込むのタイプ。 0 - 追加、またはファイルが存在していない場合には作成 1 - 上書き

---

## 送信サービス

以下のメソッドは、アウトバウンド・メッセージ・システム送信サービスで提供されます。

- `public void sendImmediate()`  
このメソッドはメッセージを宛先に即時に送信します。発呼者は、メッセージの送信が完了するまでブロックされます。
- `public void sendTransacted()`  
このメソッドは、メッセージを `MSGSTORE` データベース・テーブルに保管します。事前判別時に、WebSphere Commerce スケジューラーが、バッチ・モードで保管されているメッセージをすべて送信するジョブを呼び出します。このメソッドを使用すると、発呼者が正常にコミットまたは終了した後で、送信が確実に行われます。`sendImmediate()` メソッドを使用して呼び出しのブロックが許容できない場合には、このメソッドを使用する必要があります。
- `sendReceiveImmediate()`  
このメソッドは要求応答送信を実行するために使用されます。このタイプの送信は、バックエンド組み込みメッセージのために `MQ-JMS` トランスポートで使用されます。応答の内容は内部的に保管され、`getReply()` メソッドによってアクセスできます。  
**ヒント:** `MQ-JMS` トランスポートを使用して送受信を実行するには、管理コンソール、または構成可能メッセージ・データ・サービスで `setConfigData()` メソッドのいずれかを使用して、モード属性を適切に設定済みであることを確認しなければなりません。
- `public byte[] getReply()`  
このメソッドは、`sendReceiveImmediate()` メソッドの結果を検索するために使用します。結果を得るには、コマンドを実行する `performExecute()` メソッドの後で呼び出す必要があります。トランスポートからの応答がバイトの配列として戻されます。

---

## その他のサービス

以下メソッドは、アウトバウンド・メッセージ・システムによって提供されているアウトラインです。

- `public void addMember(Long aMember)` この方法は、宛先データをメッセージ・システムに入力する 2 番目の方法 (1 番目は `setConfigData`) です。パラメーターは有効なメンバーを表しています。ユーザーはこのメソッドを繰り返し呼び出すことができ、呼び出すたびに追加のメンバーが宛先のリストに追加されます。内部的には、メッセージ・システムが適切なアドレスをメンバーから抽出します。  
**注 1:**妥当性検査はメンバー・アドレスに対して確実に実行されていることが重要です。  
**注 2:**宛先は、`addMember` または `setConfigData` のいずれか一方 (しかし、両方ではない) を使用して設

定できます。両方とも入力すると、setConfigData エントリーが addMember エントリーによって変更されます。また、メッセージ・システムでは、ユーザーが内容を設定する前に宛先を入力することが必要であることにも注意してください。これは、メンバー情報がメッセージ中で使用されている内容を変更することがあるためです。

- `public void remTransport(Integer aRemoveTransportID)` このメソッドは、メソッドを送付するためにどのトランスポートを使用するかについて柔軟性を提供します。パラメーターは、このメッセージ・タイプのために有効なトランスポート ID です。このメソッドを呼び出すと、メッセージ・システムは、使用される可能性のあるトランスポートのメッセージ・リストからトランスポートを除去します。すなわち、パラメーターとして渡されるトランスポートがこの特定の呼び出しには使用不可になります。
- `public void setPartialSend(Boolean partialSend)` このメソッドが適用できるのは、addMember メソッドが宛先を設定するために使用されている場合だけです。ブール・パラメーターは、部分送信が許可されているかどうかを表します。メッセージ・システムでは、部分送信というのは、追加されるメンバーのいずれかに、そのプロファイルと関連付けられている優先 E メール・アドレスがある限り、メッセージを送信することを指します。以下は、partialSend パラメーターの説明です：
  - **true:** 部分送信を許可します。適切な場所にアドレスをセットアップしていないメンバーはスキップされます。
  - **false:** すべてまたはなし。メンバーのいずれかの E メールが適切に構成されていなくても、例外を戻します。
- **注:** メッセージ・システムによってアドレスが正しい形式になっているか確認するための検査は実行されません。部分送信は、適切な場所に値が存在していることを原則として作動します。



---

## 第 8 章 インバウンド・バックエンド組み込みメッセージ

インバウンド・メッセージとは、WebSphere Commerce が外部アプリケーションから受け取るメッセージのことです。各インバウンド・メッセージは、特定の機能を実行する WebSphere Commerce のコマンドをアクティブにします。インバウンド・メッセージの処理でエラーが起きた場合、そのメッセージはエラー・キューに入れられます。

WebSphere Commerce は、以下の 5 つの機能を実行するインバウンド・バックエンド・メッセージをサポートします。

- 顧客登録の作成
- 顧客登録の更新
- オーダーの状況の更新
- 商品の在庫の更新
- 商品価格の更新

上にリストされたどの機能も、XML 形式の要求メッセージでアクティブにすることができますが、一部の機能は、レガシー・メッセージ形式を使ってアクティブにすることもできます。一般的に、XML メッセージ形式のほうをお勧めします。XML メッセージは、UTF-8 形式でエンコードされます。

顧客登録を作成するには、`Create_WCS_Customer XML` メッセージを使用します。既存のバックエンド・システムにすでに顧客情報を記録している場合、この情報を最初から再作成するのではなく、このメッセージを使用して WebSphere Commerce データベースに顧客データを登録します。メッセージは、既存の顧客情報をバックエンド・サーバーから WebSphere Commerce サーバーに送信します。また、`Create_NC_Customer XML` メッセージ、および「顧客更新」のレガシー・メッセージを使用して、同様の機能を実行することもできます。ただし、`Create_WCS_Customer XML` メッセージには、より詳しい顧客情報を入力する機能も用意されています。

WebSphere Commerce データベースにすでに存在する顧客登録を更新するには、`Update_WCS_Customer XML` メッセージを使用します。このメッセージを使用すると、バックエンドの顧客管理システムは、顧客情報を更新してから、登録済みショッパーの情報を更新するためのメッセージを WebSphere Commerce に送信します。また、`Update_NC_Customer XML` メッセージおよび「顧客更新」のレガシー・メッセージを使用して、同様の機能を実行することもできます。ただし、`Update_WCS_Customer XML` メッセージには、さらに顧客情報を更新する機能も用意されています。

WebSphere Commerce データベースにすでに存在するオーダーの状況を更新するには、`Update_WCS_OrderStatus XML` メッセージを使用します。バックエンド・アプリケーションが処理するオーダーの状況に合わせて WebSphere Commerce データベースを更新するには、このメッセージを使用します。たとえば、このメッセージをバックエンド・システムから WebSphere Commerce に送信すれば、バックエンド実行システムからのオーダーの配送状況を WebSphere Commerce システム内で更新することができます。また、`Update_NC_OrderStatus XML` メッセージまたは「オーダー状況更新」のレガシー・メッセージを使用して、同様の機能を実行することもできます。ただし、`Update_WCS_OrderStatus XML` メッセージには、オーダー状況情報をさらに更新する機能も用意されています。

WebSphere Commerce データベースにすでに存在する商品の在庫を更新するには、`Update_WCS_ProductInventory XML` メッセージを使用します。外部またはバックエンド・システムの在庫システムが商品在庫を保守する場合にはこのメッセージが最適です。外部システムから WebSphere Commerce にこのメッセージを送って、WebSphere Commerce データベース内の在庫を更新することがで

きます。また、「商品数量更新」のレガシー・メッセージを使用して、同様の機能を実行することもできます。あるいは、Update\_NC\_ProductInventory XML メッセージでも、同じ機能を実行することができます。ただし、Update\_WCS\_ProductInventory XML メッセージには、在庫情報をさらに更新する機能も用意されています。

WebSphere Commerce データベースにすでに存在する商品の表示価格あるいはオファー価格を更新するには、Update\_WCS\_ProductPrice XML メッセージを使用します。外部またはバックエンド・システムの在庫システムが、価格を含む商品情報を保守する場合にはこのメッセージが最適です。外部システムは、このメッセージのうちの 1 つを WebSphere Commerce に送信して、WebSphere Commerce データベース内の商品の表示価格またはオファー価格を更新することができます。商品のオファー価格を更新する場合、Update\_NC\_ProductPrice XML メッセージまたは「商品価格更新」のレガシー・メッセージを使用して、同様の機能を実行することもできます。ただし、表示価格情報の更新には、Update\_WCS\_ProductPrice メッセージしか使用できません。

## インバウンド配送組み込みメッセージ

インバウンド配送組み込みメッセージとは、WebSphere Commerce が配送センター・システムから受信する要求のことです。各インバウンド・メッセージは、特定の機能を実行する WebSphere Commerce のコマンドをアクティブにします。インバウンド・メッセージの処理でエラーが起きた場合、失敗したメッセージはエラー・キューに入れられます。

WebSphere Commerce は、以下の機能を実行する配送組み込みメッセージをサポートします。

- 予定在庫レコードの作成
- ピッキング・バッチの作成
- ピック・チケットとパッキング・リストの要求
- 領収書の作成
- RECEIPT テーブルのもとで WebSphere Commerce データベースによって保守される商品の在庫受領レベルの調整
- 配送センターからの配送確認の発行

上にリストされたどの機能も、XML 形式のメッセージでアクティブにすることができます。XML メッセージは、UTF-8 形式でエンコードされます。

下の表は、使用されるインバウンド配送組み込みメッセージと、このメッセージによって呼び出されるコントローラー・コマンドの概要を示しています。

XML メッセージ	説明	コントローラー・コマンド
Create_WCS_ExpectedInventoryRecord	予定在庫レコードを作成します。	ExpectedInventoryRecordCreate
Create_WCS_PickBatch	ピッキング・バッチを生成します。	PickBatchGenerateCmd
Inquire_WCS_PickPackListDetail	CreatePickBatch で作成された詳細情報を要求します。	GetPickPackListDetailCmd
Create_WCS_InventoryReceipt	アイテムの在庫レコードを作成します。	ReceiptCreateCmd
Update_WCS_InventoryReceipt	アイテムの在庫を調整します。	InventoryAdjustCmd
Create_WCS_ShipmentConfirmation	アイテムの配送確認を発行します。	ReleaseShipConfirmCmd

### 予定在庫レコードの作成

WebSphere Commerce データベース内に予定在庫レコードを作成するには、`Create_WCS_ExpectedInventoryRecord` XML メッセージを使います。取引先の在庫およびオーダーに関する情報を配送センターが管理する場合のインスタンス用にこのメッセージが理想的です。今後の在庫の入手の可能性に関する情報が WebSphere Commerce に知らされ、また、在庫レベルの追跡記録をとることができます。商品の在庫レベルが低下した場合に、配送センターがこのメッセージのうちの 1 つを WebSphere Commerce に送信すれば、WebSphere Commerce データベース内に予定在庫レコードを作成することができます。このレコードは、バック・オーダー用に使用できます。

### ピッキング・バッチの作成

WebSphere Commerce データベース内にピッキング・バッチ・レコードを作成するには、`Create_WCS_PickBatch` XML メッセージを使います。ピッキング・バッチ機能を配送センターが管理する場合のインスタンスには、このメッセージが理想的です。配送センターは、このメッセージのうちの 1 つを WebSphere Commerce に送信すれば、WebSphere Commerce データベース内にピッキング・バッチを作成することができます。ピッキング・バッチとは、特定の配送センターおよびストアで発送準備は完了したけれども未処理になっているオーダーをすべてまとめたものです。ピッキング・バッチ内のすべての発送オーダーのピック・チケットは、作成されてから `PICKBATCH` テーブルに保管されます。各発送オーダーのピッキング・リストは、作成されてから `ORDRELEASE` テーブルに保管されます。

### 配送パッキング・リストの詳細情報の表示

ピック・チケットの詳細情報を問い合わせるには、`Inquire_WCS_PickPackListDetail` XML メッセージを使います。このメッセージは、`Create_WCS_PickBatch` メッセージと一緒に使用します。配送センターが `Create_WCS_PickBatch` メッセージを WebSphere Commerce に送信すると、ピッキング・バッチが作成され、新規のピッキング・バッチ ID が配送センターに戻されます。次に配送センターは、そのピッキング・バッチ ID をパラメーターとして指定して `Inquire_WCS_PickPackListDetail` メッセージを送信すると、ピック・チケットとパッキング・リストの詳細情報を取り出すことができます。

### 在庫領収書の作成

オーダーした商品の領収書を作成するには、`Create_WCS_InventoryReceipt` XML メッセージを使います。このメッセージを使うと、取引先からオーダーされた商品の領収書を作成することができますが、それは、手元にある在庫を WebSphere Commerce データベース内で更新する (`RECEIPT` データベース・テーブルのもとで) のに役立ちます。配送センターは、このメッセージのうちの 1 つを WebSphere Commerce に送信すれば、オーダーされた商品の追跡記録をとるのに使える領収書を作成することができます。

### 在庫レベルの更新

在庫の物理カウントと、WebSphere Commerce で保守されている在庫レベルとの相違を解決するには、`Update_WCS_InventoryReceipt` XML メッセージを使います。このメッセージを使うのは、外部の配送センター・システムが在庫配送を管理する場合です。配送センター・システムは、このメッセージを WebSphere Commerce に送信して、商品の在庫レベルを調整することができます。

### 配送確認の発行

WebSphere Commerce に対して配送確認を発行するには、`Create_WCS_ShipmentConfirmation` XML メッセージを使います。配送センターは、このメッセージのうちの 1 つを WebSphere Commerce に送信すれば、オーダーは配送済みであることを確認するための配送確認メッセージを作成することができます。またこのメッセージでは、オーダーの配送時に顧客に E メール通知を送信するかどうかのオプションも利用できます。顧客に対する E メール通知を使用可能にする方法の詳細は、配送通知 E メールを使用可能にするを参照してください。






---

## 第 9 章 新規インバウンド XML メッセージの追加

新規インバウンド・メッセージのサポートを追加するには、以下のステップが必要です。

1. 新規 XML メッセージの DTD を定義する。既存の XML ファイル用の DTD ファイルをガイドとして使用できます。デフォルトでは、これらのファイルは以下のディレクトリーにあります。

 ドライブ: %Program Files%WebSphere%CommerceServer\$xml%messaging

 ドライブ: %WebSphere%CommerceServer\$xml%messaging

 /usr/WebSphere/CommerceServer/xml/messaging

  /opt/WebSphere/CommerceServer/xml/messaging

 /QIBM/ProdData/WebCommerce/xml/messaging

2. システムへの新規 DTD ファイルを追加する。
3. 新規メッセージ用に user\_template.xml インバウンド・メッセージ・テンプレート定義ファイルを更新する。これを行うには、インバウンド・メッセージ・テンプレート定義ファイルにアウトラインが示されている構造上のガイドラインを参照してください。

---

### システムへの新規 DTD ファイルの追加

XML メッセージ・マッパーが新規インバウンド XML メッセージを認識して処理できるようにするには、以下のステップに従ってください。

1. 新規メッセージ用に作成した DTD ファイルを他の DTD ファイルと同じディレクトリーに入れる。デフォルトでは、ディレクトリーは以下のとおりです。

 ドライブ: %Program Files%WebSphere%CommerceServer\$xml%messaging

 ドライブ: %WebSphere%CommerceServer\$xml%messaging

 /usr/WebSphere/CommerceServer/xml/messaging

  /opt/WebSphere/CommerceServer/xml/messaging

 /QIBM/ProdData/WebCommerce/xml/messaging

2. インバウンド・メッセージ DTD ファイルのリストに新規 DTD ファイルの名前を追加する。

---

### インバウンド・メッセージ DTD ファイルのリストへの追加

新規 DTD ファイルをインバウンド・メッセージ用の DTD ファイルのリストに追加するには、以下のようになります。

1. 構成マネージャーを立ち上げる。
2. 「インスタンス・プロパティ」を選択し、次に、「**Messaging**」フォルダーをオープンする。
3. 「インバウンド・メッセージ DTD ファイル」フィールドで、新規 DTD ファイルの名前をリストの終わりに追加し、新規のファイル名の前にコンマを入れる。
4. 「適用」をクリックして、変更内容を保管する。
5. WebSphere Application Server 管理コンソールからインスタンスを停止してから、再始動する。

---

## インバウンド・メッセージ拡張

WebSphere Commerce では、各メッセージによって実行される WebSphere Commerce コントローラー・コマンドを変更して、すべてのインバウンド・メッセージの機能を変更または拡張することができます。使用される任意のインバウンド・メッセージ・コマンドに、追加の前処理ステートメントまたは後処理ステートメントを指定することができます。あるいは既存の処理全体をオーバーライドすることができます。これらを行うには Java プログラミングの知識が必要です。

インバウンド・メッセージをバックエンド・システムから受信すると、その情報がコマンド・パラメーター用に処理されて、指定したすべてのパラメーターとともに WebSphere Commerce コントローラー・コマンドが呼び出されます。コマンドが実行されると、次の 3 つのメソッドを以下の順序通りに順に呼び出す `performExecute()` メソッドが呼び出されます。

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

WebSphere Commerce を初めてインストールしたときは、プログラミング・ステートメントが入っているのは `doProcess()` メソッドだけです。このコマンドを拡張して、`doPreProcess()` メソッドをインプリメントすることによって前処理ステートメントを追加できるか、あるいは `doPostProcess()` メソッドをインプリメントすることによって後処理ステートメントを追加できます。あるいは、`doProcess()` または `performExecute()` のいずれかのメソッドをインプリメントして処理全体を上書きできます。コマンドの拡張方法の詳細については、*WebSphere Commerce プログラマーズ・ガイド* を参照してください。

---

## インバウンド・メッセージの UserData エlement

すべてインバウンド・メッセージはオプションの Element として、UserData XML Element が組み込まれています。追加データのトランスポートには、UserDataField Element を組み込んでください。メッセージに組み込まれていない特別なデータを渡すためのフィールドを、カスタマイズすることができます。

追加する新規フィールド名は UserDataField Element の名前属性にします。

以下の DTD は、UserData Element についての説明です:

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

以下は UserData Element の例です:

```
<ユーザー・データ>
  <UserDataField name="field_name">field_value</UserDataField>
  .
  .UserDataField 繰り返しループ
  .
</UserData>
```

---

## 第 10 章 メッセージ・マッパー

メッセージ・マッパーは、XML メッセージをとり、それを `CommandProperty` オブジェクトに変換する機構です。これはメッセージを `CommandProperty` オブジェクトに変換し、すべての WebSphere Commerce コンポーネントで使用できるようにするための、共通のインターフェースを提供します。

サポートされるアダプター (プログラム・アダプターなど) と WebSphere Commerce コンポーネントの両方がメッセージ・マッパーを呼び出すことができます。両方の場合とも、メッセージ・マッパーは、以下のタスクを実行します。

- XML メッセージを受け取る。
- メッセージを `CommandProperty` オブジェクトに変換する。
- XML メッセージを変換できない場合、ヌルを戻す。

インバウンド・メッセージがバックエンド・システムまたは外部システムによって WebSphere Commerce に送られ、ある種のアクションを要求します。WebSphere Commerce がそのアクションを実行するには、メッセージ・マッパーによって XML メッセージを処理し、要求されているアクションを判別する必要があります。WebSphere Commerce にはインストール時に、XML 組み込みメッセージを解析するための XML メッセージ・マッパーと、バックエンド組み込みレガシー・メッセージを解析するためのレガシー・メッセージ・マッパーの 2 つのメッセージ・マッパーが組み込まれます。

`CommandProperty` オブジェクトは、サポートされるデバイス・アダプターからの要求に適応するための WebSphere Commerce コマンドを表します。メッセージを `CommandProperty` オブジェクトに変換するためにメッセージ・マッパー機構を使用するコンポーネントは他にもあります。

メッセージ・マッパーのライフ・サイクルは、WebSphere Commerce 全体に存在します。これはインスタンスの開始時に初期化され、インスタンスが実行されている間存在します。

---

### XML メッセージ・マッパー

XML メッセージ・マッパーは、インバウンド XML メッセージからの XML データを `CommandProperty` オブジェクトに変換する責任があります。これは、ECSAX パーサーの拡張機能です。

`instance_name.xml` 構成ファイルにある XML メッセージ・マッパーの XML 構成ノードは、以下のように表示されます。

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

---

### レガシー・メッセージ・マッパー

レガシー・メッセージ・マッパーは、インバウンド・レガシー・メッセージからのデータを `CommandProperty` オブジェクトに変換する責任があります。`instance_name.xml` 構成ファイルにあるレガシー・メッセージ・マッパーの XML 構成ノードは、以下のように表示されます。

```

<MessageMapper messageId="-2"
classname="com.ibm.commerce.messaging.programadapter.messageadapter.nclegacy.NetCMessageMapper"
  enable="true">
  name="NC.LEGACY">
</configuration> (構成)
</MessageMapper>

```

## インバウンド・メッセージ・テンプレート定義ファイル

WebSphere Commerce には XML メッセージ・マッパーが備えられていますが、これを使用すれば、インバウンド XML メッセージ・テンプレート定義ファイルに基づいてインバウンド XML メッセージを WebSphere Commerce コマンド・インターフェースにマップすることができます。







インバウンド XML メッセージがメッセージ・マッパーに渡されるたびに、テンプレート定義ファイルにメッセージが定義されているかどうか調べられて確認されます。定義されていれば、そのメッセージに対する WebSphere Commerce コントローラー・コマンド名とパラメーター名が検索されて、着信メッセージを解析してパラメーターの値が取得されます。メッセージの解析が完了したら、そのコマンドのコマンド名とパラメーターの名前と値のペアの入ったオブジェクトがメッセージ・マッパーから戻されます。

メッセージ・テンプレート定義ファイルは、インバウンド XML メッセージ用の XML 構文解析情報を定義するのに使用されます。これらのファイルに定義されたそれぞれのメッセージには、以下の 2 つの基本エレメントがあります。

- **TemplateDocument:** メッセージが使用する DTD ファイル、メッセージの受信時に呼び出されるコマンド、使用されるタグ・マッピング、タグ・マッピングが開始される XML エレメントを定義します。
- **TemplateTag:** DTD ファイル内の XML エレメントの、WebSphere Commerce 内のコマンドのパラメーター名に対するマッピングを定義します。テンプレート・タグ・エレメントはパラメーター名を識別し、その値が着信メッセージ上のどこにあるかをメッセージ・マッパーに知らせます。

WebSphere Commerce に備えられたテンプレート定義ファイルは 2 種類あります。sys\_template.xml ファイルは、既存の WebSphere Commerce インバウンド XML メッセージをマップするのに使用されるテンプレート定義です。user\_template.xml は、追加のインバウンド XML メッセージの追加を可能にします。両方のファイルとも、ec\_template.dtd テンプレート定義 DTD ファイルに基づいており、XML 形式のものであります。

デフォルトでは、テンプレート定義、テンプレート定義 DTD、およびインバウンド XML メッセージの DTD ファイルはすべて、以下のディレクトリーに保管されます。

	drive:¥WebSphere¥CommerceServer¥xml¥messaging
	ドライブ:¥Program Files¥WebSphere¥CommerceServer¥xml¥messaging
	/usr/WebSphere/CommerceServer/xml/messaging
	 /opt/WebSphere/CommerceServer/xml/messaging
	/QIBM/Proddata/WebCommerce/xml/messaging

**重要:** セキュリティー上の理由により、必ず、sys\_template.xml および user\_template.xml メッセージ・テンプレート定義ファイルには、権限を与えられた人だけがアクセスおよび変更できるようにする必要があります。

ります。権限のない人にこのファイルへの書き込みアクセスを許可すると、WebSphere Commerce コマンドを起動できる新規インバウンド・メッセージを、サイト管理者であるかのように作成できるようになってしまいます。

---

## メッセージ・マッパーの除去

メッセージ・マッパーを除去するには、*instance\_name.xml* 構成ファイル内のメッセージ・マッパーのグループから手動で除去する必要があります。メッセージ・マッパーを除去するには、以下のようにします。

1. *instance\_name.xml* 構成ファイルをオープンする。
2. `MessageMapperGroup` という名前のコンポーネントを見つける。
3. 除去するメッセージ・マッパーの XML 構成ノードを見つける。以下のようにします。

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

4. `enable` パラメーターを `"false"`に変更する。これによりメッセージ・マッパーが使用禁止になり、使用ができなくなります。

---

## メッセージ・マッパーの追加

新規のメッセージ・マッパーを追加するには、*instance\_name.xml* 構成ファイル内のメッセージ・マッパーのグループに手動で追加する必要があります。新規のメッセージ・マッパーを追加するには、以下のようにします。

1. *instance\_name.xml* 構成ファイルをオープンする。
2. `MessageMapperGroup` という名前のコンポーネントを見つける。
3. そのコンポーネントのプロパティ・タグ間に以下の XML ノードを追加して、使用するメッセージ・マッパーを定義する。

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

これらのパラメーターの詳細については、メッセージ・マッパー構成を参照してください。

4. メッセージ・マッパーの構成ノードに、そのメッセージ・マッパーが必要とする追加の構成パラメーターを加えてください。これは `TypedProperty` オブジェクトに変換され、メッセージ・マッパーの `Init` メソッドに渡されます。加えられる可能性のある追加のパラメーターの例を、以下に示します。

```
<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"
/>
```

---

## 新規インバウンド・メッセージ・サポート

サポートされる XML および WebSphere Commerce メッセージに加えて、新しくインバウンド・メッセージのサポートを追加することができます。インバウンド・メッセージの新規追加には、2 つの基本メソッドがあります。

お勧めする方法は、`user_template.xml` インバウンド・メッセージ・テンプレート定義ファイルを使用することによって新規インバウンド XML メッセージを追加することです。このファイル中で、新規インバウンド・メッセージが呼び出すコントローラー・コマンドを指示して、各エレメントと対応しているコマンド・パラメーターを指示できます。メッセージを受信すると、XML メッセージ・マッパーは、実行するコマンドと使用するパラメーターを識別します。その後で、コマンドがサイト管理者権限を使用して呼び出されます。セキュリティ上の理由から、`user_template.xml` メッセージ・テンプレート定義ファイルにアクセスして変更できるのは必ず権限保持者のみにしなければなりません。そうでない場合には、権限のないユーザーが新規インバウンド・メッセージを書き込み、`WebSphere Commerce` コマンドをサイト管理者として呼び出せることとなります。

XML メッセージ・マッパーでインバウンド XML メッセージ・テンプレート定義ファイルを使用したくない場合には、`NewInboundMessage` コマンドをインプリメントして新規メッセージを追加することもできます。このコマンドは、メッセージ・マッパーがメッセージを既存のレガシー・メッセージ、またはインバウンド XML メッセージ・テンプレート定義ファイルで定義されている XML メッセージとして認識しないときに呼び出されます。`NewInboundMessage` コマンドは既成プログラムではありませんので、それが呼び出されて処理をさせることは、すべて作成者のコントロールに任されます。しかし、このメソッドのプログラミングでは、特に新規メッセージの数が多い場合に、かなりのプログラミング上の労力が必要です。

---

## 第 11 章 NewInboundMessage コマンドのカスタマイズ

ユーザーが作成したメッセージを処理するように NewInboundMessage コマンドをカスタマイズするには、以下のようにします。

1. NewInboundMessage コマンドを展開する。 これを行う方法については、 *WebSphere Commerce プログラマーズ・ガイド* を参照してください。
2. インバウンド・メッセージを受け取るには、メッセージをストリングとして戻す、コマンドの getMessage() メソッドを使用する。
3. コマンドの performExecute() メソッドをインプリメントする。 このメソッドの内側に、ユーザーのインバウンド・メッセージを処理するプログラミング・ステートメントを入れます。
4. 以下の SQL ステートメントを使用し、 WebSphere Commerce データベース内の CMDREG テーブルを更新することにより、新規拡張コマンドを登録する。

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





---

## 第 12 章 メッセージ・マップー構成

`instance_name.xml` 構成ファイルは、候補のメッセージ・マップーすべてをリストしており、名前、クラス、デバイス形式、これが使用可能か不可か、そして特定の構成パラメーターを含みます。特定のメッセージ・マップーのパラメーターを見つけるには、`MessageMapperGroup` という名前の `instance_name.xml` ファイルのコンポーネントを検索します。それぞれのメッセージ・マップーに異なる構成パラメーターがある可能性があるため、`MessageMapperGroup` コンポーネントには、それぞれに XML 構成情報のノードがあります。この構成パラメーター・ノードの形式の要件は、パラメーターが名前-値の組でなければならないことだけです。こうすると、メッセージ・マップー構成パラメーターを簡単に `TypeProperty` オブジェクトに変換できます。

以下に、個々のメッセージ・マップーごとにパラメーターをリストします。

- `name`: メッセージ・マップーの名前。
- `class`: メッセージ・マップー・インターフェースのインプリメンテーションを含むクラス。
- `messageMapperId`: メッセージ・マップーの ID。それぞれのメッセージ・マップー ID は固有でなくてはなりません。
- `enabled`: メッセージ・マップーを使用して初期化するかどうかを示します。この値が `"true"` に設定されている場合には、インスタンスの始動時にメッセージ・マップーが初期化され、使用可能になります。この値が `"false"` に設定されている場合には、メッセージ・マップーは使用不可です。

---

### テンプレート定義ファイルを使用した XML 構文解析

XML メッセージ・マップーがインバウンド XML メッセージを構文解析するとき、文書タイプ、バージョン (利用可能な場合)、およびエレメント名を、メッセージ中の XML 文書から 1 つずつ取得します。メッセージ・マップーは、テンプレート定義ファイル中の `ECTemplate` エレメントに定義されているテンプレート文書を検索します。メッセージ・マップーがテンプレート文書から取得する情報は、以下のとおりです:

- タグ・マッピングを開始するエレメント。
- 使用するタグ・テンプレート。
- インバウンド・メッセージによって呼び出すコマンド名。このコマンドは、既存の `WebSphere Commerce` コントローラー・コマンド、または作成済みの新規コントローラー・コマンドのいずれかになります。

タグ・マッピングが開始されると、メッセージ・マップーは、`ECTemplate` ファイルで定義されている `TemplateTag` を調べて、インバウンド XML メッセージから生成される XPath に基づくフィールド名とタイプを判別してから、そのフィールドの値を設定します。フィールドと値のペアは、フィールドの `FieldInfo` 属性中の値を基にして、`commandProperty` または `messageProperty` のいずれかの `TypedProperty` ハッシュ・テーブルに保管されます。全インバウンド・メッセージが正常に構文解析された後に、`commandProperty` および `messageProperty` オブジェクトの他に、コマンド名が入っている `PropertyCommand` オブジェクトも戻されます。

---







## sys\_template.xml ファイル

sys\_template.xml ファイルには、WebSphere Commerce によってサポートされているすべてのインバウンド XML メッセージのアウトラインが入っています。このファイルで、各メッセージのデータ・フィールド、該当する WebSphere Commerce コントローラー・コマンドへのメッセージのマッピング、およびそのコマンドの該当するパラメーターへのメッセージ内の各フィールドのマッピングを定義します。

sys\_template.xml の構造は、メッセージが従わなければならない形式が定義されている ec\_template.dtd ファイルに基づきます。

このファイルには新規のメッセージを追加しないでください。ユーザー独自のインバウンド・メッセージを追加するには、user\_template.dtd ファイルを使用します。

すべての XML ファイルは、以下のディレクトリーにあります:

	ドライブ:¥Program Files¥WebSphere¥CommerceServer¥xml¥messaging
	drive:¥WebSphere¥CommerceServer¥xml¥messaging
	/usr/WebSphere/CommerceServer/xml/messaging
	 /opt/WebSphere/CommerceServer/xml/messaging
	/QIBM/Proddata/WebCommerce/xml/messaging







---

## user\_template.xml ファイル

user\_template.xml は、ユーザー・システムによってサポートする新規インバウンド XML メッセージを追加できる、XML メッセージ・テンプレート定義ファイルです。このファイルには、サポートしたい個々の新規 XML メッセージごとにアウトラインを追加する必要があります。sys\_template.xml ファイルは、このテンプレート・ファイルを使用するか更新する方法のガイドとして使用できます。

アウトラインでは、使用するタグ・テンプレート、タグ・マッピングを開始するエレメント、呼び出す WebSphere Commerce コントローラー・コマンドの名前、および各 XML エレメントに対応する URL パラメーターが示されます。新規インバウンド XML メッセージを使用して新規 WebSphere Commerce コマンドを呼び出す場合の、新規 WebSphere Commerce コントローラー・コマンドを作成して登録する方法については、*WebSphere Commerce プログラマーズ・ガイド* を参照してください。

すべての XML ファイルは、以下のディレクトリーにあります:

	ドライブ:¥Program Files¥WebSphere¥CommerceServer¥xml¥messaging
	ドライブ:¥WebSphere¥CommerceServer¥xml¥messaging
	/usr/WebSphere/CommerceServer/xml/messaging
	 /opt/WebSphere/CommerceServer/xml/messaging
	/QIBM/Proddata/WebCommerce/xml/messaging

自分自身のメッセージを追加する前に、必ず以下の行を組み込むようにしてください:

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

テンプレート定義ファイル中で定義する各メッセージには、以下の 2 つの基本エレメントが含まれます:

- TemplateDocument
- TemplateTag







これらのエレメントのそれぞれの説明については、以下のリンクに従ってください。

---

## ec\_template.dtd ファイル

ec\_template.dtd ファイルは、sys\_template.xml ファイルにあるすべてのインバウンド・メッセージ用の、下書き用青写真をもっていますので、user\_template.xml ファイルへ新規メッセージを作成する際に使用します。

すべての DTD ファイルは、以下のディレクトリーにあります:

	ドライブ:%Program Files%WebSphere%CommerceServer\$xml%messaging	
	drive:%WebSphere%CommerceServer\$xml%messaging	
	/usr/WebSphere/CommerceServer/xml/messaging	
		/opt/WebSphere/CommerceServer/xml/messaging
	/QIBM/Proddata/WebCommerce/xml/messaging	

以下は ec\_template.dtd ファイルの説明です:

```
<!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 >
```

## テンプレート定義ファイルの TemplateDocument エlement

インバウンド XML メッセージごとに、メッセージ中に定義されているテンプレート文書が少なくとも 1 つはなければなりません。これは TemplateDocument Element で定義されます。TemplateDocument Element には、以下の 4 つのサブElementがあります:

- **DocumentType (オプションのバージョン属性付き):** このパーツは、XML 文書名またはルート・Element 名、およびルート・Element の “version” 属性 (存在している場合) を指定します。DocumentType Element の例は、以下のとおりです:  
`<DocumentType version='1.0'>Reset_Password</DocumentType>`
- **StartElement:** このパーツは、XML メッセージ・マッパーがタグ・マッピングを開始するElement を指定します。これは、メッセージの DTD 中のElement と対応していなければなりません。タグ・マッピングで、コマンド・パラメーターとして使用される名前と値のペアが生成されます。StartElement の使用例は、以下のとおりです:  
`<StartElement>PasswordInfo</StartElement>`  
この例では、このメッセージのために DTD ファイル中に定義されている PasswordInfo という名前のElement があるはずですが、たとえば、DTD ファイルには、PasswordInfo Element を定義する以下のような行が入ります:  
`<!ELEMENT PasswordInfo (Password, ConfirmPassword)>`
- **TemplateTagName:** このパーツは、メッセージに使用するタグ・マッピングを指定します。タグ・マッピングは、テンプレート定義ファイル中のどこか他の場所で TemplateTag Element の項に定義されます。これは、TemplateTagName Element の値が、テンプレート定義ファイル中のどこかに指定されている TemplateTag Element の name 属性と一致するはずであるという意味です。詳細については、TemplateTag Element の節を参照してください。TemplateTagName Element の使用例は、以下のとおりです:  
`<TemplateTagName>PasswordReset10Map</TemplateTagName>`  
この例では、以下の TemplateTag Element がテンプレート定義ファイル中のどこか他の場所に存在しているはずですが:  
`<TemplateTag name='PasswordReset10Map'> ... </TemplateTag>`  
開始タグと終了タグの間のElement はメッセージのマッピングによって異なります。
- **CommandMapping:** このElement は、メッセージによって呼び出されるコマンドを判別します。このElement には、Command と呼ばれるサブElement が 1 つ入っています。Command Element は、メッセージが正常にマップされると実行される WebSphere Commerce コントローラー・コマンドを指示するために使用されます。ここには CommandName と呼ばれる 1 つの必須属性が入っていて、コマンドの名前を指示するために使用されます。この属性の値は、URLREG テーブルに登録されている既存のコマンドと対応していなければなりません。たとえば、メッセージは、以下の構文を使用してパスワードをリセットするコマンドと関連付けることができます:  
`<CommandMapping> <Command CommandName='ResetPassword' /> </CommandMapping>`  
また、複数のコマンドと同じメッセージを関連付けることもできます。これを行うには、Command Element の以下の追加の属性を使用することが必要です。
  - **Condition 属性:** 条件の構文は、次のとおりです ([ ] はオプションであるという意味で、\* は繰り返してかまわないという意味です):

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

```
fieldName
```

これは、TemplateTag 定義中の Tag エレメントの Field 属性と一致しているはずですが、このタグの XPath 属性がインバウンド XML メッセージ中に存在していると、条件は true です。

#### **fieldName="fieldValue "**

XML メッセージ中の値は fieldName 中の値に設定されます。メッセージがマップされるときに、値が fieldValueと同じであると、条件は true です。

#### **fieldName1="fieldValue1" AND fieldName2="fieldValue2"**

メッセージの構文解析時に、XML メッセージ中の値がフィールド fieldName1 および fieldName2 に設定されます。この値が fieldValue1 および fieldValue2とそれぞれ同じである場合には、条件は true です。

- **TemplateTagName 属性:** この Command エレメントに TemplateTagName 属性を指定すると、条件が true になると常に、新規タグ・テンプレートが TemplateTagName 中に定義されている名前で、インバウンド XML メッセージの残りに使用されることとなります。
- **Constant エレメント:** そのコマンドのために TypedProperty に入れられる定数のリスト。
  - **Field:** TypedProperty に入れられることになる名前と値のペアのフィールド名。
  - **FieldInfo:** TemplateTag 中の定義を参照してください。  
値は <Constant> と </Constant> のタグの間に配置されます。

複数の Command エレメントを使用して複数のコマンドにマップする方法の例は、以下のとおりです:  
<CommandMapping> <Command CommandName='ResetPassword' Condition='Verb="Reset" AND Noun="Password"'/> <Command CommandName='AdminResetPassword' Condition='Verb="Reset" AND Noun="AdminPassword"'/> </CommandMapping>この例では、メッセージのために DTD ファイル中に定義されている Noun エレメントと Verb エレメントがあります。

---

## テンプレート定義ファイルの TemplateTag エレメント

TemplateTag エレメントは、TemplateDocument エレメントと一緒にテンプレート定義ファイル中で使用されます。幾つかの TemplateTag エレメントをインバウンド XML メッセージごとに定義できます。しかし、各 TemplateTag エレメントは name 属性を使用して TemplateDocument エレメントにリンクしてなければなりません。name と関連付けられている値は、テンプレート定義ファイルで定義されている TemplateDocument エレメントの TemplateTagName サブエレメント中の値と一致しているはずですが。

各 TemplateTag エレメントには、Tag サブエレメント中にタグ定義のリストが含まれています。Tag エレメントには、以下の 5 個の属性を入れることができます。

- **XPath:** TemplateDocument に指示されている StartElement に対して相対的な XML エレメントのパス。XPath エレメントは、名前と値のペアのフィールド名を検索するためのキーです。XPath の例は、以下のとおりです。
  - E1/E2: エレメント E1 の内側にネストしているエレメント E2。このタイプのパスの例は、以下のとおりです。  
XPath='Address/ZipCode'
  - E1/E2/E3: エレメント E2 (エレメント E1 の内側にネストしている) の内側にネストしているエレメント E3。このタイプのパスの例は、以下のとおりです。  
XPath='ContactInfo/Address/ZipCode'

- E3@a1 : エレメント E3 中の属性 a1。このタイプのパスの例は、以下のとおりです。  
XPath='InvoiceInfo@InvoiceType'  
この場合には、InvoiceType 属性が、メッセージの適用先の送り状タイプの種類以外のものを指示するために使用される場合があります。
- E4[1] : エレメント E4 の初回インスタンス。E4 の複数インスタンスがある場合があります。たとえば、複数行の住所があるこの形式を使用できます。住所の各行には別個の Tag エレメントが入ることになります。
- E5[@a2="value 2"] : エレメント E5 のインスタンスの 1 つ。E5 の属性 a2 は “value 2” と等しくなります。たとえば、以下の形式を使用できます。  
XPath='InvoiceInfo[@InvoiceType="ShippingInvoice"]  
E5[@a1="value1"][@a2="value 2"] : エレメント E5 のインスタンス。E5 の属性 a1 は “value1” と等しくなり、E5 の属性 a2 は “value 2” と等しくなります。
- **XPathType:** XPath に指示されているエレメントのタイプ。この属性は、XML エレメントを XML パーサーによってどのように処理するかを指示します。サポートされるタイプは:
  - **PCDATA** : エレメントまたはエレメントの属性には、処理されて名前と値のペアに戻されるロー・インバウンド・データが入ります。これが XPathType のデフォルト値です。
  - **EMPTY:** エレメントがエンプティであるか、あるいは無視できるデータが入っています。このエレメントの名前と値のペアが戻されます。
  - **REPEAT** : エレメントは複数インスタンスにすることができます。各エレメントの PCDATA は名前と値のペアに戻されます。
  - **ATTRIBUTE:** エレメント・フィールド名が属性の値によって決定される場合には、この属性は ATTRIBUTE タイプになるはずですが。
  - **VECTOR:** 新規ハッシュ・テーブルがベクトルに追加されて、サブエレメント用に生成されている名前と値のペアはすべてが新規ハッシュ・テーブルに入れられることになります。
  - **USERDATA:** ユーザー定義エレメントを指示します。このエレメントは name と呼ばれる属性になり、値はフィールド名です。エレメントのデータは名前と値のペアに戻されます。
- **Field:** TypedPropertyに入れられることになる名前と値のペアのフィールド名。これは、呼び出されたコマンドで使用されるパラメーターの名前と一致するはずですが。
- **FieldType:** データ・フィールドのタイプ。フィールド・タイプは String または Date (ISO 8601 日付形式) とすることができます。String がデフォルトです。
- **FieldInfo:** 名前と値のペアが入れられる TypedProperty を指示します。Data がデフォルトです。名前と値のペアを複数の TypedPropertyに入れたい場合には、以下にリストされている複数の値をコンマで区切って指定しなければなりません:
  - **Data:** 名前と値のペアは、コマンドの引き数が入る commandProperty に入れられることになります。
  - **Control:** 名前と値のペアは、コマンドの制御情報 (USERID または PASSWORD など) が入る messageProperty に入れられることになります。
  - **Command:** 名前と値のペアは、どのコマンドを呼び出すかを決定するために使用されます。生成済みの名前と値のペアは、TemplateDocument エレメントの CommandMapping エレメントで使用されます。

エレメント XPath がタグ・テンプレート中に見つからないと、XML メッセージ中の XPath がフィールド名として使用され、XPathType は PCDATA となり、名前と値のペアは XPath をフィールド名として使用して生成されます。

TemplateTag エレメントがどのように使用されるかの例については、sys\_template.xml ファイルを参照してください。

---

## 第 13 章 メッセージ・システム・バックエンド組み込みメッセージ

WebSphere Commerce メッセージ・システムは、インバウンドおよびアウトバウンドのメッセージを使用して、バックエンド・システムに WebSphere Commerce を組み込むためのメカニズムとして機能します。インバウンド・メッセージは、バックエンド・システムから来たメッセージに基づいて WebSphere Commerce でコマンドを実行するのに使用します。アウトバウンド・メッセージは、アウトバウンド・メッセージ・システムで生成することができ、発生したイベント（新規カスタマー・オーダーなど）に合わせてバックエンド・システムを更新します。バックエンド組み込みメッセージを使用するには、アダプターがインストールされていて、しかも XML メッセージを受信するようにメッセージ・システムが構成されていなければなりません。

メッセージ・システムは、多くの XML 形式の事前定義メッセージを送受信できるようになっています。この形式により、非常に読みやすくなり、メッセージの変更および保守が容易になります。また、レガシー・メッセージ形式を使用することもできます。ただし、XML メッセージ形式のほうをお勧めします。各メッセージの説明は、インバウンドおよびアウトバウンドのバックエンド組み込みメッセージの項を参照してください。新規メッセージを追加することもできます。新規インバウンド・メッセージの場合、既存の WebSphere Commerce コマンドや、独自に作成したコマンドに関連付けることができます。





## 第 14 章 配送組み込みメッセージ

WebSphere Commerce には、インバウンドおよびアウトバウンドのメッセージを使って配送センター・システムへの組み込みを図るためのメカニズムが備えられています。インバウンド配送組み込みメッセージは、配送センター・システムから受信したインバウンド要求に基づいて WebSphere Commerce でコマンドを実行するのに使用します。アウトバウンド・メッセージは、アウトバウンド・メッセージ・システムで生成することができ、新規の在庫の納品、またはオーダーの配送といった発生イベントに合わせて配送センター・システムを更新します。配送組み込みメッセージを使用するには、アダプターがインストールされていて、しかも XML メッセージを受信するようにメッセージ・システムが構成されていなければなりません。

メッセージ・システムは、多くの XML 形式の事前定義メッセージを送受信できるようになっています。この形式により、非常に読みやすくなり、メッセージの変更および保守が容易になります。各メッセージの説明は、インバウンドおよびアウトバウンド配送組み込みメッセージの項を参照してください。新規メッセージを追加することもできます。新規インバウンド・メッセージの場合、既存の WebSphere Commerce コマンドや、独自に作成したコマンドに関連付けることができます。

XML メッセージの形式は、特定の DTD ファイルに定義される XML エレメントのセットで構成されます。各 DTD には、.mod ファイル拡張子によって識別される 1 つ以上の共通ファイルが含まれている場合があります。また、各インバウンド・メッセージは、sys\_template.xml メッセージ・テンプレート定義ファイルの中の、WebSphere Commerce コントローラー・コマンドと関連付けられます。すべての DTD、MOD、および XML ファイルは、以下のディレクトリーに入っています:

2000	ドライブ:%Program Files%WebSphere%CommerceServer\$xml%messaging
NT	drive:%WebSphere%CommerceServer\$xml%messaging
AIX	/usr/WebSphere/CommerceServer/xml/messaging
SOLARIS	LINUX /opt/WebSphere/CommerceServer/xml/messaging
400	QIBM/ProdData/WebCommerce/xml/messaging



## 第 15 章 組み込みメッセージ DTD ファイル

サポートされている WebSphere Commerce 組み込み XML メッセージはすべて、DTD ファイルにある情報で構成されています。いくつかの DTD ファイルでは、共通ファイル NCCCommon.mod または他の MOD ファイルの情報を使用します。DTD ファイルの XML エlement 値の形式とソースについては、これらの MOD ファイルで説明されています。

DTD ファイルと MOD ファイルはすべて、以下のディレクトリーに置かれています。

<b>2000</b>	ドライブ:¥Program Files¥WebSphere¥CommerceServer¥xml¥messaging
<b>NT</b>	ドライブ:¥WebSphere¥CommerceServer¥xml¥messaging
<b>AIX</b>	/usr/WebSphere/CommerceServer/xml/messaging
<b>SOLARIS</b>	<b>LINUX</b> /opt/WebSphere/CommerceServer/xml/messaging
<b>400</b>	/QIBM/Proddata/WebCommerce/xml/messaging

以下のテーブルは、サポートされている XML 組み込みメッセージと、そこで使用されるファイルをリストしたものです。

メッセージ	DTD および使用される MOD ファイル
Create_NC_Customer メッセージ	Create_NC_Customer_10.dtd NCCCommon.mod NCCustomer_10.mod
Update_NC_OrderStatus メッセージ	Update_NC_OrderStatus_10.dtd NCCCommon.mod
Update_NC_ProductInventory メッセージ	Update_NC_ProductInventory_10.dtd NCCCommon.mod
Update_NC_ProductPrice メッセージ	Update_NC_ProductPrice_10.dtd NCCCommon.mod
Report_NC_PurchaseOrder メッセージ	Report_NC_PO_10.dtd
Create_WCS_Customer メッセージ	Create_WCS_Customer_20.dtd NCCCommon.mod
Update_WCS_ProductPrice メッセージ	Update_WCS_ProductPrice_20.dtd NCCCommon.mod
Update_WCS_ProductInventory メッセージ	Update_WCS_ProductInventory_20.dtd NCCCommon.mod.
Update_WCS_Customer メッセージ	Update_WCS_Customer_10.dtd NCCCommon.mod NCCustomer_10.mod
Update_WCS_OrderStatus メッセージ	Update_WCS_OrderStatus_20.dtd NCCCommon.mod
Create_WCS_ExpectedInventoryCreate メッセージ	Create_WCS_ExpectedInventoryRecord_10.dtd
Create_WCS_PickBatch メッセージ	Create_WCS_PickBatch_10.dtd
Inquire_WCS_PickPackListDetail メッセージ	Inquire_WCS_PickPackListDetail_10.dtd

Create_WCS_InventoryReceipt メッセージ	Create_WCS_InventoryReceipt_10.dtd
Update_WCS_InventoryReceipt メッセージ	Update_WCS_InventoryReceipt_10.dtd
Create_WCS_ShipmentConfirmation メッセージ	Create_WCS_ShipmentConfirmation_10.dtd
Response_WCS_ExpectedInventoryRecord メッセージ	Response_WCS_ExpectedInventoryRecord_10.dtd
Response_WCS_PickBatch メッセージ	Response_WCS_PickBatch_10.dtd
Report_WCS_PickPackListDetail メッセージ	Report_WCS_PickPackListDetail_10.dtd
Response_WCS_CreateInvReceipt メッセージ	Response_WCS_CreateInvReceipt_10.dtd
Response_WCS_UpdateInvReceipt メッセージ	Response_WCS_UpdateInvReceipt_10.dtd
Response_WCS_CreateShipConfirm メッセージ	Response_WCS_CreateShipConfirm_10.dtd

## バックエンド組み込みレガシー・メッセージ

WebSphere Commerce は、レガシー・メッセージ形式を使用するメッセージをサポートしています。ただし XML メッセージは同じ機能を提供し、また XML メッセージの方が読みやすく保守しやすいので、以前のバージョンの WebSphere Commerce Suite からマイグレーションする場合を除き、代わりに XML メッセージを使用することをお勧めします。

WebSphere Commerce メッセージの形式は、メッセージ・ディスクリプターおよびアプリケーション・データの 2 つのセクションで構成されています。インバウンド・メッセージの場合、メッセージ記述子には、操作上必要な制御情報 (メッセージ ID およびタイプなど) が含まれています。アプリケーション・データには、処理対象の情報が含まれています。すべての WebSphere Commerce メッセージは、タグのセットと論理順次レコード、およびレコード内に定義されたデータ・セグメントで構成されています。サポートされるメッセージに含まれる <PROLOG>、<HDR>、<ITM>、および <DATUSR> レコードは、以下の形式になっています:

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

ここで、データ・セグメントは \_DATA サフィックスで識別されます。たとえば、オーダー作成メッセージのレコードは、以下のようになります:

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

各データ・セグメント (この例では、ORDER\_CREATE\_HDR010\_DATA) は、特定メッセージに対して特定フィールドおよびデータベースのテーブル情報で置き換えられます。

以下のテーブルは、6 個のサポートされるメッセージと、それぞれによって呼び出されるコントローラー・コマンドのアウトラインです。

メッセージ名	メッセージ・タイプ	データ・セグメント	コントローラー・コマンド
新規顧客	インバウンド	CUSTOMER_NEW_PROLOG_DATA CUSTOMER_NEW_HDR010_DATA インバウンド・メッセージの DATUSR_DATA	UserRegistration Add
更新顧客	インバウンド	CUSTOMER_UPDATE_PROLOG_DATA CUSTOMER_UPDATE_HDR010_DATA インバウンド・メッセージの DATUSR_DATA	UserRegistration Update

メッセージ名	メッセージ・タイプ	データ・セグメント	コントローラー・コマンド
オーダー作成	アウトバウンド	ORDER_CREATE_PROLOG_DATA ORDER_CREATE_HDR010_DATA ORDER_CREATE_HDR020_DATA ORDER_CREATE_HDR030_DATA ORDER_CREATE_HDR040_DATA アウトバウンド・メッセージの DATUSR_DATA ORDER_CREATE_ITM010_DATA	SendWCSOrder
オーダー状況の更新	インバウンド	ORDER_STATUS_UPDATE_PROLOG_DATA ORDER_STATUS_UPDATE_HDR010_DATA インバウンド・メッセージの DATUSR_DATA ORDER_STATUS_UPDATE_ITM010_DATA	OrderStatus
商品価格の更新	インバウンド	PRODUCT_PRICE_UPDATE_PROLOG_DATA PRODUCT_PRICE_UPDATE_HDR010_DATA	ProductOffer PriceUpdate
商品数量の更新	インバウンド	PRODUCT_QUANTITY_UPDATE_PROLOG_DATA PRODUCT_QUANTITY_UPDATE_HDR010_DATA	ProductInventory Update

## バックエンド組み込み XML メッセージ

WebSphere Commerce は、XML 形式を使用するインバウンドおよびアウトバウンド・メッセージのサポートを提供します。各インバウンド・メッセージは、コントローラー・コマンドを実行することによって、WebSphere Commerce サーバー内で特定の動作を呼び出します。各コントローラー・コマンドは、WebSphere Commerce データベースおよびサブシステムに対する操作を順に実行します。一部のコントローラー・コマンドは、複数の XML メッセージによって実行できます。さらに、メッセージの中には、メッセージの内容に応じて別のコマンドを呼び出すものもあります。

XML メッセージの形式は、特定の DTD ファイルに定義される XML エレメントのセットで構成されます。各 DTD には、.modファイル拡張子によって識別される 1 つ以上の共通ファイルが含まれている場合があります。また、各インバウンド・メッセージは、sys\_template.xmlメッセージ・テンプレート定義ファイルの中の、WebSphere Commerce コントローラー・コマンドと関連付けられます。すべての DTD、MOD、および XML ファイルは、以下のディレクトリーに入っています:

<b>2000</b>	ドライブ:¥Program Files¥WebSphere¥CommerceServer¥xml¥messaging
<b>NT</b>	drive:¥WebSphere¥CommerceServer¥xml¥messaging
<b>AIX</b>	/usr/WebSphere/CommerceServer/xml/messaging
<b>SOLARIS</b>	<b>LINUX</b> /opt/WebSphere/CommerceServer/xml/messaging
<b>400</b>	/QIBM/Proddata/WebCommerce/xml/messaging

以下のテーブルは、使用されるインバウンド・メッセージと、それらが呼び出すコントローラー・コマンドのアウトラインです。

XML メッセージ	説明	コントローラー・コマンド
Create_WCS_Customer、 Create_NC_Customer	新規ユーザーの登録レコードを作成するか、既存ユーザーのレコードを更新します。	UserRegistrationAdd

XML メッセージ	説明	コントローラー・コマンド
Update_NC_Customer、 Update_WCS_Customer	既存ユーザーの登録レコードを更新します。	UserRegistrationUpdate
Update_WCS_OrderStatus、 Update_NC_OrderStatus	オーダーの一般状況を更新します。	OrderStatus
Update_WCS_OrderStatus	オーダーの確認状況を更新します。	OrderConfirmStatus
Update_WCS_OrderStatus	オーダーの配送状況を更新します。	OrderShippingStatus
Update_WCS_OrderStatus	オーダーの送り状況を更新します。	OrderInvoiceStatus
Update_WCS_ProductPrice、 Update_NC_ProductPrice	商品のオファー価格情報を更新します。	商品オファー価格の更新
Update_WCS_ProductPrice	オーダーの表示価格情報を更新します。	ProductListPriceUpdate
Update_NC_ProductInventory Update_WCS_ProductInventory	商品在庫情報を更新します。	ProductInventoryUpdate

**注:**メッセージの中には名前に文字 NC が含まれているものがあり、その他のメッセージの名前には名前 WCS が含まれています。文字 NC を含む名前のメッセージは、以前のバージョンの WebSphere Commerce の XML メッセージです。名前に WCS が含まれているメッセージは、さらに柔軟性が増す更新済み形式を使用します。一般には、選択している WCS バージョンを使用することをお勧めします。

以下のテーブルは、生成されるコマンドの他に、アウトバウンド・メッセージ・システムによって使用されるバックエンド組み込みメッセージのアウトラインです。

XML メッセージ	説明	コントローラー・コマンド
Report_NC_PurchaseOrder	新規オーダーについての情報が含まれているメッセージをバックエンド・システムに送信します。	SendXMLOrder

## 配送組み込みメッセージを使用したサンプル・シナリオ

配送組み込みメッセージを使用して WebSphere Commerce で配送センター・システムと通信を行うことができます。これによりサイト管理者は、顧客に提供する商品の使用可能性についての情報を把握することができます。配送センター・システムと WebSphere Commerce システムが、配送組み込みメッセージを使って連絡する方法を以下のシナリオで説明します。

### シナリオ 1 - 予定在庫とバック・オーダー

配送センター・システムが、あるアイテムの在庫レベルが低いことを検出します。配送センター・システムは取引先にさらに在庫をオーダーし、Create\_WCS\_ExpectedInventoryRecord XML メッセージを使って、新しい在庫の予測受け取りを WebSphere Commerce へレポートします。

WebSphere Commerce は、在庫レベルが低い場合でも、在庫を増やす予測受け取りに基づいてバック・オーダーを可能にして、そのアイテムを販売提供し続けることができます。WebSphere Commerce は、WCSRaDetailIDパラメーターを含む Response\_WCS\_ExpectedInvRecord メッセージを応答送信します。

配送センターでは、この WCSRaDetailID パラメーターに注目します。配送センターでは、追加の出荷が到着すると、参照のための WCSRaDetailID パラメーターを含む Create\_WCS\_InventoryReceipt XML メッセージを送信します。追加の在庫に関連した WCSRaDetailIDがない場合は、その WCSRaDetailIDは省略することができます。

## シナリオ 2 - 在庫の更新

配送センターのスタッフが在庫数を実際に数えているとき、システムに記録されている在庫レベルと実際にある在庫が矛盾していることに気がつきます。配送センターでは Update\_WCS\_InventoryReceipt XML メッセージを使用して、WebSphere Commerce にその矛盾を知らせることができます。

WebSphere Commerce では、記録されている在庫レベルをそれに従って更新し、Response\_WCS\_UpdateInvReceipt メッセージで応答します。

## シナリオ 3 - オーダーの実行

配送センターではオーダーに対応するために、WebSphere Commerce に Create\_WCS\_PickBatch XML メッセージを送信して実行プロセスを開始します。

WebSphere Commerce では PickBatchID を含む応答メッセージを送信して、「配送準備完了」アイテムのリストをまとめます。

配送センターでは Inquire\_WCS\_PickPackListDetail XML メッセージを送信して、ピック・チケットとパッキング・リスト、そして他の配送情報を要求します。

WebSphere Commerce は、ピックしてパックするもののリストを応答します。

配送センターでは配送の準備をしてそれを顧客に送ります。このときオプションで、WebSphere Commerce に Create\_WCS\_ShipmentConfirmation XML メッセージで配送を通知します。配送センターは、顧客にも E メールで配送を通知することができます。

---

## ReleaseShipNotify メッセージ

ReleaseShipNotify メッセージは、オーダー・リリースが目録に記載されたときに顧客に通知送信される、アウトバウンド E メール・メッセージです。これは、ORDRELEASE テーブルの STATUS 列が MNF に更新されると発生します。E メール・メッセージは ReleaseShipNotify タスク・コマンドによって送信されますが、このときのメッセージのコンテンツを組み立てるために ReleaseShipNotify.jsp が使用されます。ReleaseShipNotify タスク・コマンドは、ReleaseManifest コントローラー・コマンドによって呼び出されません。

このメッセージは、配送センター・システムが内部のものか外部のものかにかかわらず使用されます。内部 WebSphere Commerce 配送センターを使用するとき、管理者は配送確認画面でリリース目録のオプションを選択して、このメッセージのトリガーをかけることができます。外部配送センター・システムを使用するときは、このメッセージは、UpdateManifestStatus 属性が 1 に設定されている配送確認メッセージによってトリガーがかけられます。

このメッセージは、ReleaseShipNotify タスク・コマンドをオーバーライドすることによって、ストア・レベルで使用可能または使用不可にすることができます。このメッセージでは、デフォルトでは、CMDREG テーブルの中のクラス名として、ReleaseShipNotifyDummyImpl を使用することはできません。

## Response\_WCS\_ExpectedInvRecord メッセージ

Response\_WCS\_ExpectedInvRecord メッセージは、WebSphere Commerce 予定在庫レコードについての情報を含むアウトバウンド・メッセージです。WebSphere Commerce は、インバウンド

Create\_WCS\_ExpectedInventoryRecord メッセージに対する応答として、このメッセージを生成します。

このインバウンド・メッセージに有効な StoreID または有効な ExpectedDateがある場合、このメッセージは ExpectedInventoryRecordCreate コマンドを呼び出し、正常に終了すると

ExpectedInventoryRecordCreateRedirectView 表示タスクに、失敗すると

ExpectedInventoryRecordCreateErrorView 表示タスクにリダイレクトします。

ExpectedInventoryRecordCreateRedirectView 表示タスクは、Response\_WCS\_ExpectedInvRecord 応答メッセージを組み立てるために、RACreateError.jspによってインプリメントされています。配送センターがその応答を、配送センターが送信した元のメッセージと正しく関連付けられるように、配送センターに返送される応答メッセージの中に BackendRaDetailIDパラメーターを含めることができます。

**注:** インバウンド・メッセージに有効な StoreIDまたは有効な ExpectedDateがない場合、エラー・メッセージの生成に GenericApplicationError ビュー名が使用されます。応答メッセージは、GenericApplicationErrorXML.jspにより生成されます。

Response\_WCS\_ExpectedInvRecord メッセージは、XML メッセージ形式を使用し、Response\_WCS\_ExpectedInvRecord\_10.dtd.に続きます。

次の表は、Response\_WCS\_ExpectedInvRecord メッセージの形式について説明しています。データベース列の説明については、それと関連したテーブルへのリンクに従ってください。特に注釈がない限り、すべてのフィールドはオプションです。タイプ INT、BIGINT、および DOUBLE のデータベース・フィールド用の XML メッセージ中のタグ値の長さは、それぞれ CHAR (10)、CHAR(19)、および CHAR(16) とする必要があります。

	フィールド名	コメント	テーブル名	列名	注
1	ResponseStatus	必須	N/A	N/A	コマンド失敗の際のエラー・コメント
2	status	必須	N/A	N/A	OK または ERROR (ResponseStatus の属性)
3	コード		N/A	N/A	エラー・コード (ResponseStatus の属性で、status="ERROR" の場合のみ)
4	BackendRaID		N/A	N/A	元の Create_WCS_ExpectedInventoryRecord メッセージによって参照される
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	参照として使用可能
10	ItemOwnerID		ITEMSPC	MEMBER_ID	
11	ProductSKU		ITEMSPC	PARTNUMBER	
12	WCSRaDetailID		RADETAIL	RADETAIL_ID	Create_WCS_InventoryReceipt メッセージで使用できる



## Response\_WCS\_PickBatch メッセージ

Response\_WCS\_PickBatch メッセージは、WebSphere Commerce ピッキング・バッチについての情報を含むアウトバウンド・メッセージです。WebSphere Commerce は、インバウンド Create\_WCS\_PickBatch メッセージに対する応答として、このメッセージを生成します。このインバウンド・メッセージに有効な StoreIDがある場合、このメッセージは PickBatchGenerate コマンドを呼び出し、正常に終了すると PickBatchGenerateRedirectView 表示タスクに、失敗すると PickBatchGenerateErrorView 表示タスクにリダイレクトします。PickBatchGenerateRedirectView は、応答処理のために PickBatchResult.jspによりインプリメントされています。Response\_WCS\_PickBatch XML メッセージでは、元の要求のバックエンド PickBatchID および新規に生成された PickBatchIDが、応答として返送されます。コマンドが失敗する場合には、PickBatchGenerateErrorView が使用されます。これは、PickBatchError.jsp によりインプリメントされています。

**注:** インバウンド・メッセージに有効な StoreIDがない場合、エラー・メッセージの生成に GenericApplicationError ビュー名が使用されます。応答メッセージは、GenericApplicationErrorXML.jspにより生成されます。

Response\_WCS\_PickBatch メッセージは XML メッセージ形式を使用し、Response\_WCS\_PickBatch\_10.dtd に続きます。

以下のテーブルは、Response\_WCS\_PickBatch メッセージ形式の説明です。データベース列の説明については、それと関連したテーブルへのリンクに従ってください。特に注釈がない限り、すべてのフィールドはオプションです。タイプ INT、BIGINT、および DOUBLE のデータベース・フィールド用の XML メッセージ中のタグ値の長さは、それぞれ CHAR (10)、CHAR(19)、および CHAR(16) とする必要があります。

レベル	フィールド名	コメント	テーブル名	列名	注
1	ResponseStatus	必須	N/A	N/A	Status="ERROR" の場合はエラー・コメント
2	状況	必須	N/A	N/A	OK または ERROR (ResponseStatus の属性)
3	code		N/A	N/A	エラー・コード (ResponseStatus の属性で、status="ERROR" の場合のみ)
4	MorePickBatch		N/A	N/A	YES または NO: さらに別の PickBatch に、別の Create_WCS_PickBatch メッセージを再び送信するかどうかを示す
5	BackendPickBatchID		N/A	N/A	元の PickBatch 要求の参照
6	WCSPickBatchID		PICKBATCH	PICKBATCH_ID	指定された StoreID および FulfillmentCenterID で PickBatch が使用できない場合には、“NULL”になる。 Inquire_WCS_PickPackListDetail メッセージで使用

## Report\_WCS\_PickPackListDetail メッセージ

Report\_WCS\_PickPackListDetail メッセージは、ピック・チケットとパッキング・リストの詳細情報をレポートするアウトバウンド・メッセージです。 WebSphere Commerce は、インバウンド

Inquire\_WCS\_PickPackListDetail メッセージに対する応答として、このメッセージを生成します。 インバウンド・メッセージは GetPickPackListDetail コマンドを呼び出します。これは、正常に終了すると

PickPackListRedirectView 表示タスクに、失敗すると PickPackListErrorView 表示タスクにリダイレクトします。 PickPackListRedirectView は、PickPackListResult.jsp によりインプリメントされ、応答を処理します。

PickPackListErrorView は、PickPackListError.jsp によりインプリメントされます。

Report\_WCS\_PickPackListDetail メッセージには、Create\_WCS\_PickPackListDetail\_10.dtd により生成される 2 つの固定 XML スリップが含まれます。 これらは、ピック・チケットおよびパッキング・リストです。 ピック・チケットとパッキング・リストの属性は、以下のテーブルで説明されます。

Report\_WCS\_PickPackListDetail メッセージは XML メッセージ形式を使用し、Report\_WCS\_PickPackListDetail\_10.dtd に続きます。

以下のテーブルは、Report\_PickPackListDetail メッセージ形式の説明です。 データベース列の説明については、それと関連したテーブルへのリンクに従ってください。 特に注釈がない限り、すべてのフィールドはオプションです。タイプ INT、BIGINT、および DOUBLE のデータベース・フィールド用の XML メッセージ中のタグ値の長さは、それぞれ CHAR (10)、CHAR(19)、および CHAR(16) とする必要があります。

レベル	XML エlement	コメント	テーブル名	列名	注
1	ResponseStatus	必須	N/A	N/A	status="ERROR" の場合はエラー・コメント
2	status	必須	N/A	N/A	OK または ERROR (ResponseStatus の属性)
3	code		N/A	N/A	エラー・コード (ResponseStatus の属性で、status="ERROR" の場合のみ)
4	PickPackListReport		N/A	N/A	status="OK" の場合にのみ存在。ピック・チケットおよびパッキング・リストを組み込みます。

ピック・チケット・エレメントの定義が PickTicket\_10.dtdに続きます。この DTD ファイルの内容は固定されており、変更できません。

レベル	XML エlement	コメント	テーブル名	列名	注
1	Pickticket	必須	N/A	N/A	Pickbatch_Information および Pickbatch_Items を組み込みます。1 つのメッセージに複数の Pickbatch_Items を含めることが可能です。
2	Pickbatch_Information	必須	N/A	N/A	Store_Name、Fulfillment_Center、および Pickbatch_Number の属性
3	Store_Name	必須	STOREENTDS	DISPLAYNAME	Pickbatch_Information の属性
4	Fulfillment_Center	必須	FFMCENTDS	DISPLAYNAME	Pickbatch_Information の属性
5	Pickbatch_Number	必須	PICKBATCH	PICKBATCH_ID	Pickbatch_Information の属性
6	Pickbatch_Items		N/A	N/A	SKU、Product_Name、Product_Description、および Quantity の属性
7	SKU	必須	BASEITEM	PARTNUMBER	Pickbatch_Items の属性
8	Product_Name	必須	BASEITMDSC	SHORTDESCRIPTION	Pickbatch_Items の属性
9	Product_Description		BASEITMDSC	LONGDESCRIPTION	Pickbatch_Items の属性
10	数量	必須	ORDERITEMS	QUANTITY	Pickbatch_Items の属性

パッキング・リストの定義が PackSlip\_10.dtd に続きます。この DTD ファイルの内容は固定されており、変更できません。

レベル	XML エlement	コメント	テーブル名	列名	注
1	Packslip	必須	N/A	N/A	Order_Information、Shipto、および Order_Items を含みます。1 つのメッセージに複数の Packslip を入れることができます。
2	Order_Information	必須	N/A	N/A	属性 Store_Name、Fulfillment_Center、Order_Number、Release_Number、PickBatch_Number、Order_Date、Catalog_Name、Shipping_Provider、Customer_Number、および Invoice_Method
3	Store_Name	必須	STOREENTDS	DISPLAYNAME	Order_Information の属性
4	Fulfillment_Center	必須	FFMCENTDS	DISPLAYNAME	Order_Information の属性
5	Order_Number	必須	ORDERITEMS	ORDERS_ID	Order_Information の属性
6	Release_Number	必須	ORDERITEMS	ORDERELEASENUM	Order_Information の属性
7	PickBatch_Number	必須	PICKBATCH	PICKBATCH_ID	Order_Information の属性
8	Order_Date	必須	ORDERS	TIMEPLACED	Order_Information の属性
9	Shipping_Provider	必須	SHPMODEDSC	DESCRIPTION	Order_Information の属性
10	Customer_Number	必須	ORDERS	MEMBER_ID	Order_Information の属性

レベル	XML エlement	コメント	テーブル名	列名	注
11	Invoice_Method		TERMDCOND	STRINGFIELD1	これはストリングです。 オプションは、both、e-Mail、printed の 3 つです。 デフォルトは NULL です。
12	Shipto	必須	N/A	N/A	属性 AddressID、 First_Name、 Last_Name、 Middle_Name、 Address_1、 Address_2、 Address_3、 City、 State、 Zip、 および Country
13	AddressID	必須	ADDRESS	ADDRESS_ID	Shipto の属性
14	First_Name		ADDRESS	FIRSTNAME	Shipto の属性
15	Last_Name	必須	ADDRESS	LASTNAME	Shipto の属性
16	Middle_Name		ADDRESS	MiddleName	Shipto の属性
17	Address_1	必須	ADDRESS	MiddleName	Shipto の属性
18	Address_2		ADDRESS	ADDRESS2	Shipto の属性
19	Address_3		ADDRESS	ADDRESS3	Shipto の属性
20	市区町村	必須	ADDRESS	City	Shipto の属性
21	State		ADDRESS	State	Shipto の属性
22	Zip		ADDRESS	ZIPCODE	Shipto の属性
23	Country	必須	ADDRESS	COUNTRY	Shipto の属性
24	Order_Items	必須	N/A	N/A	属性 SKU、 Product_Name、 Product_Description、 Quantity、 Catalog_Name、 Unit_Price、 Total_Price、 Currency、 および Customer_Comments に複数のコンポーネントを含めることができます。
25	SKU	必須	BASEITEM	PARTNUMBER	Order_Items の属性
26	Product_Name	必須	BASEITMDSC	SHORTDESCRIPTION	Order_Items の属性
27	Product_Description		BASEITMDSC	LONGDESCRIPTION	Order_Items の属性
28	数量	必須	ORDERITEMS	QUANTITY	Order_Items の属性
29	Catalog_Name	必須	CATENTDESC	NAME	Order_Items の属性
30	Unit_Price	必須	ORDERITEMS	PRICE	Order_Items の属性
31	Total_Price	必須	ORDERITEMS	TOTALPRODUCT	Order_Items の属性
32	Currency	必須	ORDERITEMS	CURRENCY	Order_Items の属性
33	Customer_Comments		ORDERITEMS	COMMENTS	Order_Items の属性
34	Component		N/A	N/A	SKU、 Product_Name、 Product_Description、 および Quantity の属性
35	SKU	必須	BASEITEM	PARTNUMBER	Component の属性
36	Product_Name	必須	BASEITMDSC	SHORTDESCRIPTION	Component の属性
37	Product_Description		BASEITMDSC	LONGDESCRIPTION	Component の属性
38	数量	必須	ORDERITEMS	QUANTITY	Component の属性

## Response\_WCS\_CreateInvReceipt メッセージ

Response\_WCS\_CreateInvReceipt メッセージは、WebSphere Commerce 在庫受け取りの作成のための情報を含むアウトバウンド・メッセージです。WebSphere Commerce は、インバウンド Create\_WCS\_InventoryReceipt メッセージに対する応答として、このメッセージを生成します。このインバウンド・メッセージに有効な storeIDおよび有効な ReceiptDateが組み込まれている場合、CreateInvReceiptOK.jsp を使用して応答メッセージを生成する表示タスク ReceiptCreateRedirectView を呼び出します。この応答メッセージには、RECEIPTと RCPTAVAIL の各テーブルに作成されたレコードの新しい receipt\_id が含まれます。コマンドでエラーが発生すると、表示タスク ReceiptCreateErrorView が使用されます。このエラー表示タスクは、CreateInvReceiptError.jsp によってインプリメントされています。予定在庫レコードに見合うだけの在庫がある場合、そのレコードはクローズされます。

**注:** インバウンド・メッセージに有効な StoreIDまたは有効な ReceiptDateがない場合、エラー・メッセージの生成に GenericApplicationError ビュー名が使用されます。応答メッセージは、GenericApplicationErrorXML.jspにより生成されます。

Response\_WCS\_CreateInvReceipt メッセージは XML メッセージ形式を使用し、Response\_WCS\_CreateInvReceipt\_10.dtdに続きます。

以下のテーブルは、Response\_WCS\_CreateInvReceipt メッセージ形式の説明です。データベース列の説明については、それと関連したテーブルへのリンクに従ってください。特に注釈がない限り、すべてのフィールドはオプションです。タイプ INT、BIGINT、および DOUBLE のデータベース・フィールド用の XML メッセージ中のタグ値の長さは、それぞれ CHAR (10)、CHAR(19)、および CHAR(16) とする必要があります。

レベル	フィールド名	コメント	テーブル名	列名	注
1	ResponseStatus	必須	N/A	N/A	エラー・コメント
2	status	必須	N/A	N/A	OK または ERROR (ResponseStatus の属性)
3	code		N/A	N/A	エラー・コード (ResponseStatus の属性で、status="ERROR" の場合のみ)
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 メッセージ

Response\_WCS\_UpdateInvReceipt メッセージは、アイテムの在庫を更新するための情報を含むアウトバウンド・メッセージです。WebSphere Commerce は、Update\_WCS\_InventoryReceipt メッセージへの応答としてこのメッセージを生成します。このインバウンド・メッセージに有効な StoreIDが含まれる場合、InventoryAdjust コマンドを呼び出し、このコマンドは表示タスク InventoryAdjustRedirectView にリダイレクトします。正常に終了すると、InventoryAdjustRedirectView は UpdateInvReceiptOK.jspを使用して、

応答メッセージを生成します。 コマンドでエラーが発生すると、表示タスク `InventoryAdjustErrorView` が使用されます。 このエラー表示タスクは、 `UpdateInvReceiptError.jsp` によってインプリメントされています。

メッセージで在庫調整が肯定されている場合、このコマンドによって `RECEIPT` および `RCPTAVAIL` データベース・テーブルに新しい行が作成されます。 在庫調整が否定されている場合は、適切なピック・メソッドを使って `RECEIPT` テーブルの `QTYONHAND` 列がマークされます。

**注:** インバウンド・メッセージに有効な `StoreID`がない場合、エラー・メッセージの生成に `GenericApplicationError` ビュー名が使用されます。 応答メッセージは、 `GenericApplicationErrorXML.jsp`により生成されます。

`Response_WCS_UpdateInvReceipt` メッセージは XML メッセージ形式を使用し、 `Response_WCS_UpdateInvReceipt_10.dtd`に続きます。

以下のテーブルは、 `Response_WCS_UpdateInvReceipt` メッセージ形式の説明です。 データベース列の説明については、それと関連したテーブルへのリンクに従ってください。 特に注釈がない限り、すべてのフィールドはオプションです。 タイプ `INT`、 `BIGINT`、 および `DOUBLE` のデータベース・フィールド用の XML メッセージ中のタグ値の長さは、それぞれ `CHAR(10)`、 `CHAR(19)`、 および `CHAR(16)` とする必要があります。

レベル	フィールド名	コメント	テーブル名	列名	注
1	<code>ResponseStatus</code>	必須	N/A	N/A	エラー・コメント
2	<code>status</code>	必須	N/A	N/A	OK または ERROR (ResponseStatus の属性)
3	<code>code</code>		N/A	N/A	エラー・コード (ResponseStatus の属性で、 <code>status="ERROR"</code> の場合のみ)
4	<code>ItemOwnerID</code>		ITEMSPC	MEMBER_ID	N/A
5	<code>ProductSKU</code>		ITEMSPC	PARTNUMBER	N/A
6	<code>StoreID</code>		RECEIPT	STORE_ID	N/A
7	<code>FulfillmentCenterID</code>		RECEIPT	FFMCENTER_ID	N/A
8	<code>QTYAdjusted</code>		INVADJUST	QUANTITY	N/A
9	<code>InvAdjCodeID</code>		INVADJUST	INVADJCODE_ID	N/A

## Response\_WCS\_CreateShipConfirm メッセージ

`Response_WCS_CreateShipConfirm` メッセージは、オーダーに対する配送確認の作成のための情報を含むアウトバウンド・メッセージです。 `WebSphere Commerce` は、インバウンド

`Create_WCS_ShipmentConfirmation` メッセージに対する応答として、このメッセージを生成します。 このインバウンド・メッセージに有効な `ActualShipDate`が含まれている場合、これは `ReleaseShipConfirm` コマンドを呼び出し、正常に終了すると、表示タスク `ReleaseShipConfirmRedirectView` にリダイレクトします。

`Response_WCS_CreateShipConfirm` 応答メッセージは、 `CreateShipConfirmOK.jsp` によって生成されます。 コマンドは必要なデータベースを更新し、アイテムの実行状況を変更して配送を確認します。 コマンドは

キー・マネージャーを介して MANIFEST テーブルから新しい manifest\_id を取得し、入力データとともに MANIFEST テーブルを伝搬します。 Release\_WCS\_ShipmentNotify メッセージが使用可能になっていると、コマンドが正常に実行されて UpdateManifestStatus が 1 の場合に、デフォルトの ReleaseShipNotify.jsp によって通知 E メールが生成されます。

コマンドでエラーが発生すると、コマンドによって表示タスク ReleaseShipConfirmErrorView にリダイレクトされます。このエラー表示タスクは、CreateShipConfirmError.jsp によってインプリメントされます。

**注:** インバウンド・メッセージに有効な ActualShipDateがない場合、エラー・メッセージの生成に GenericApplicationError ビュー名が使用されます。応答メッセージは、GenericApplicationErrorXML.jspにより生成されます。

Response\_WCS\_CreateShipConfirm メッセージは XML メッセージ形式を使用し、Response\_WCS\_CreateShipConfirm\_10.dtdに続きます。

以下のテーブルは、Response\_WCS\_CreateShipConfirm メッセージ形式の説明です。データベース列の説明については、それと関連したテーブルへのリンクに従ってください。特に注釈がない限り、すべてのフィールドはオプションです。タイプ INT、BIGINT、および DOUBLE のデータベース・フィールド用の XML メッセージ中のタグ値の長さは、それぞれ CHAR (10)、CHAR(19)、および CHAR(16) とする必要があります。

レベル	フィールド名	コメント	テーブル名	列名	注
1	ResponseStatus	必須	N/A	N/A	Status="ERROR" の場合はエラー・コメント
2	status	必須	N/A	N/A	OK または ERROR (ResponseStatus の属性)
3	code		N/A	N/A	エラー・コード (ResponseStatus の属性で、status="ERROR" の場合のみ)
4	ShipModeID		MANIFEST	SHIPMODE_ID	N/A
5	OrderNumber		MANIFEST	ORDERS_ID	N/A
6	OrderReleaseNum		MANIFEST	ORDERRELEASENUM	N/A

## Update\_WCS\_OrderStatus メッセージ

Update\_WCS\_OrderStatus メッセージは、WebSphere Commerce オーダーの状況情報を含むインバウンド・メッセージです。このメッセージには 4 つの形式が考えられます。メッセージの各形式は同じ XML エレメントを持っていますが、それぞれは異なるコマンドと関連しています。コマンドへのメッセージ形式のマッピングは次のようになります。

オーダー状況のメッセージ・タイプ	コマンド
OrderConfirm	OrderConfirmStatus
OrderShipping	OrderShippingStatus
OrderInvoice	OrderInvoiceStatus
OrderStatus	OrderStatus

バックエンド・システムはオーダーを受け取ると、オーダー実行状況情報を含むこのメッセージを生成して、これを WebSphere Commerce インバウンド・メッセージ・キューへ送信します。このメッセージはそこで WebSphere Commerce システムが受信します。 WebSphere Commerce は、該当するコントローラー・コマンドを呼び出し、 ORDSTAT および ORDISTAT テーブルを新しいオーダー状況情報で更新します。

Update\_WCS\_OrderStatus メッセージは XML メッセージ形式を使用し、 Update\_WCS\_OrderStatus\_20.dtd に従います。

以下のテーブルは、Update\_WCS\_OrderStatus メッセージ形式の説明です。4つの Order Status メッセージ・タイプのそれぞれは、特に注釈を付けた場合を除き、同じ形式に従います。以下のテーブルは、XML エlement値の形式およびソースの説明です。データベース列の説明については、それと関連したテーブルへのリンクに従ってください。特に注釈がない限り、すべてのフィールドはオプションです。タイプ INT、BIGINT、および DOUBLE のデータベース・フィールド用の XML メッセージ中のタグ値の長さは、それぞれ CHAR (10)、CHAR(19)、および CHAR(16) とする必要があります。

レベル	フィールド名	コメント	テーブル名	列名	注
1	<i>OrderStatusType</i>	必須	N/A	N/A	OrderConfirm、OrderShipping、OrderInvoice、または OrderStatus
1.1	SerializationInfo		N/A	N/A	指定されている場合には、メッセージのシリアライゼーションを調べるために使用されます。
1.1.1	SequenceNumber		ORDSTAT	OSSEQNUM	
1.1.2	LastUpdate Timestamp		ORDSTAT	OSUPDTIME	
1.2	OrderStatus Header	必須	N/A	N/A	
1.2.A1	Versioning	Attribute	N/A	N/A	'TRUE' または 'FALSE' の値。TRUE の場合には、バージョン管理が使用可能になります。



レベル	フィールド名	コメント	テーブル名	列名	注
1.2.1	OrderNumber		ORDSTAT	ORDERS_ID/ OSMORDER	type=ByWCS の場合は ORDERS_ID (WebSphere Commerce オーダー参照番号)、そうでない場合は OSMORDER (バックエンド・システムによって生成されるオーダー参照番号)。
	type	Attribute			<b>ByWCS</b> または <b>ByBackend</b>
1.2.2	TotalPriceInfo		N/A	N/A	
1.2.2.A1	currency	Attribute	ORDSTAT	OSPCUR	
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	OSSHXTOT	
1.2.3	Status		ORDSTAT	OSSTATUS	デフォルト値: 'C'= 確認済み - OrderConfirm について 'S'= 配送済み - OrderShipping について 'I'= 送り状発行済み - OrderInvoice について
1.2.4	PlacedDate	必須	ORDSTAT	OSPLTIME	
1.2.5	ShippingInfo		N/A	N/A	
1.2.5.A1	ShipCondition	Attribute	ORDSTAT	OSSCOND	出荷の部分出荷を許可するかどうかを指示するコード 'SC'= 一括配送 'SP'=分割配送
1.2.5.A2	ShipModeFlag	Attribute	ORDSTAT	OSSMFLAG	配送先住所および配送方式がオーダー・レベルか、オーダー・アイテム・レベルかを指示するコード 'O' = オーダー・レベル 'I' = オーダー・アイテム・レベル

レベル	フィールド名	コメント	テーブル名	列名	注
1.2.5.1	RequestShipDate		ORDSTAT	OSRSTIME	
1.2.5.2	ScheduledShip Date		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	コメント		ORDSTAT	OSCMNT	
1.2.8	CustomerField	初回出現	ORDSTAT	FIELD1	
1.2.8	CustomerField	2 番目の出現	ORDSTAT	FIELD2	
1.2.8	CustomerField	3 番目の出現	ORDSTAT	FIELD3	
1.2.9	UserData		N/A	N/A	
1.3	OrderStatusItem	繰り返し	N/A	N/A	Vector
1.3.A1	Versioning	Attribute	N/A	N/A	'TRUE' または 'FALSE' の値。TRUE の場合には、バージョン管理が使用可能になります。
1.3.1	ItemNumber		ORDISTAT	ORDER ITMES_ID/ OIMITEM	type = ByWCS の場合は ORDERITMES_ID、type = ByBackend の場合は OIMITEM
1.3.1.A1	type	Attribute	N/A	N/A	ByWCS または 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	Attribute	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	
1.3.6	Status		ORDISTAT	OISTATUS	
1.3.7	PlacedDate		ORDISTAT	OIPLTIME	

レベル	フィールド名	コメント	テーブル名	列名	注
1.3.8	ShippingInfo		N/A	N/A	
1.3.8A1	ShipCondition		ORDISTAT	OISCOND	明細アイテムの部分出荷を許可するかどうかを指示するコード。SC'=配達完了 'SP'=分割配送
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	コメント		ORDISTAT	OICMNT	
1.3.11	CustomerField1	初回出現	ORDISTAT	FIELD1	
1.3.11	CustomerField2	2 番目の出現	ORDISTAT	FIELD2	
1.3.11	CustomerField3	3 番目の出現	ORDISTAT	FIELD3	
1.3.12	UserData		N/A	N/A	

## オーダー状況の更新メッセージ

オーダー状況の更新メッセージは、WebSphere Commerce オーダーの状況情報を含むインバウンド・メッセージです。バックエンド・アプリケーションがこのメッセージを生成し、WebSphere Commerce インバウンド・メッセージ・キューへ送信します。このメッセージはそこで WebSphere Commerce システムが受信します。WebSphere Commerce は OrderStatus コマンドを使用して、ORDSTAT および ORDISTAT テーブルを新しいオーダー状況情報で更新します。

オーダー状況の更新メッセージは、オーダー状況の更新 バージョン 01 およびオーダー状況の更新バージョン 02 のアプリケーション・データ 2 セットをサポートしています。バージョン 02 は、バージョン 01 のデータのスーパーセットです。

オーダーの状況の更新メッセージは WebSphere Commerce のメッセージ・フォーマットが使用され、バッファ内で相互に順次に並んでいる一連のレコードで構成されています。以下のデータは、オーダー状況の更新メッセージについての説明です:

```
<ECEDOC>
<PROLOG>ORDER_STATUS_UPDATE_PROLOG_DATA</PROLOG>
<HEADER>
  <HDR010>ORDER_STATUS_UPDATE_HDR010_DATA</HDR010>
  <USRLST>
    <DATUSR>DATUSR_DATA</DATUSR>
    .
    .DATUSR 繰り返しループ
    .
    <DATUSR>DATUSR_DATA</DATUSR>
  </USRLST>
```

```

</HEADER>
<ITMLST>
  <ITMDAT>
    <ITM010>ORDER_STATUS_UPDATE_ITM010_DATA</ITM010>
    <USRLST>
      <DATUSR>DATUSR_DATA</DATUSR>
      .DATUSR 繰り返しループ
      .
    <DATUSR>DATUSR_DATA</DATUSR>
    </USRLST>
  </ITMDAT>
  .
  .ITEM 繰り返しループ
  .
  <ITMDAT>
    <ITM010>ORDER_STATUS_UPDATE_ITM010_DATA</ITM010>
    <USRLST>
      <DATUSR>DATUSR_DATA</DATUSR>
      .DATUSR 繰り返しループ
      .
    <DATUSR>DATUSR_DATA</DATUSR>
    </USRLST>
  </ITMDAT>
</ITMLST>
</ECEDOC>

```

**注:**

- すべてのレコードはバッファー内で順次の並びです。読みやすくするために、バッファーにはありませんが、ここでは字下げが使用されています。
- データ・セグメントのすべてのフィールドは、バッファー内では左寄せ、右スペース埋め込みです。

**オーダー状況更新のデータ・セグメント**

- **ORDER\_STATUS\_UPDATE\_PROLOG\_DATA**  
アプリケーション・データが定義するメッセージ・タイプを指定します。このケースでは、メッセージはオーダー状況更新です。
- **ORDER\_STATUS\_UPDATE\_HDR010\_DATA**  
オーダー状況更新メッセージ内のオーダー情報を指定します。
- **DATUSR\_DATA**  
オーダー状況更新メッセージへ追加されるオプションの情報を指定します。DATUSR\_DATA は、このメッセージの <HDR> および <ITM>セクションにあります。
- **ORDER\_STATUS\_UPDATE\_ITM010\_DATA**  
オーダー状況更新メッセージ内のアイテムまたは商品配送情報を指定します。

---

## Update\_WCS\_ProductPrice メッセージ

Update\_WCS\_ProductPrice メッセージは、商品の価格情報を含むインバウンド・メッセージです。バックエンド・アプリケーションがこのメッセージを生成し、WebSphere Commerce インバウンド・メッセージ・キューへ送信します。WebSphere Commerce システムは、このメッセージを受信すると、コントローラー・コマンドを実行します。実行されるコントローラー・コマンドはメッセージのタイプによって異なります。以下のテーブルは、関連したコントローラー・コマンドと一緒に送信できる 2 つのタイプのメッセージを示しています。

商品価格のメッセージ・タイプ	コントローラー・コマンド
オファー価格の更新	商品オファー価格の更新
表示価格の更新	ProductListPriceUpdate

Update\_WCS\_ProductPrice メッセージは XML メッセージ形式を使用し、Update\_WCS\_ProductPrice\_20.dtd に従います。

### オファー価格の更新

以下のテーブルは、Update\_WCS\_ProductPrice メッセージの OfferPrice 更新変形の形式の説明です。データベース列の説明については、それと関連したテーブルへのリンクに従ってください。特に注釈がない限り、すべてのフィールドはオプションです。タイプ INT、BIGINT、および DOUBLE のデータベース・フィールド用の XML メッセージ中のタグ値の長さは、それぞれ CHAR (10)、CHAR(19)、および CHAR(16) とする必要があります。

レベル	XML エlement	コメント	テーブル名	列名
1	OfferPriceInfo		N/A	N/A
1.1	ProductNumberByMerchant	必須	CATENTRY	PARTNUMBER
1.2	MerchantID	必須	CATENTRY	MEMBER_ID
1.2.A1	type	Attribute	N/A	N/A
1.3	Precedence		OFFER	PRECEDENCE
1.4	TradingPositionContainerID		OFFER	TRADEPOSCN_ID
1.5	Currency	必須	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	初回出現	OFFER	FIELD1
1.13	PriceCustomField	2 番目の出現	OFFER	FIELD2
1.14	UserData		N/A	N/A

### OfferPrice 更新の動作:

- 通貨タイプ (OFFERPRICE テーブル中の CURRENCY を指す) は必須であり、ISO 4217 形式で指定されていなければなりません。
- 部品番号 (CATENTRY テーブル中の PARTNUMBER を指す) とメンバー ID (CATENTRY テーブル中の MEMBER\_ID を指す) の組み合わせが、カタログ・エントリー (CATENTRY\_ID) を得るために使用されることとなります。この値が、優先順位 (PRECEDENCE) またはトレード位置コンテナ (TRADEPOSCN\_ID) のいずれかと一緒に、商品価格参照番号 (OFFER\_ID) を得るために使用されることとなります。商品価格参照番号が、通貨タイプと一緒に、OFFERPRICE テーブル中の行を更新するためのキーとして使用されることとなります。

- 商品価格参照番号 (OFFER\_ID) がデータベース中に既存の商品価格参照番号と一致しているが、通貨タイプがその商品価格参照番号をもつレコードのいずれかの通貨タイプと一致していない場合には、新規レコードが OFFERPRICE テーブル中に作成されることとなります。これにより、価格を異なる通貨で同じオファーに指定できます。
- 優先順位 (PRECEDENCE) が指定されていないと、ProductOfferPriceUpdate コマンドが、優先順位なしで指定されている値と一致している以前のすべてのレコードを探し出します。これらの値のうち最大値に +1 されます。以前のレコードが存在していないと、優先順位値は 1 に設定されます。新規行が新規優先順位値でテーブル OFFERPRICE に挿入されます。優先順位値は  $10^{16}$  より小でなければなりません。最大値に達してしまうと、新規更新は拒否されることとなります。

## 表示価格の更新

以下のテーブルは、Update\_WCS\_ProductPrice メッセージの ListPrice 更新変形の形式の説明です。データベース列の説明については、それと関連したテーブルへのリンクに従ってください。特に注釈がない限り、すべてのフィールドはオプションです。タイプ INT、BIGINT、および DOUBLE のデータベース・フィールド用の XML メッセージ中のタグ値の長さは、それぞれ CHAR (10)、CHAR(19)、および CHAR(16) とする必要があります。

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

### 表示価格更新の動作:

- コマンドは LISTPRICE テーブル中のレコードを更新します。
- partNumber は memberId と一緒に、カタログ・エントリー・キー (テーブル CATENTRY 中の CATENTRY\_ID) を得るために使用されます。
- カタログ・エントリーが LISTPRICE テーブル中の既存のカタログ・エントリーと一致しているが、通貨タイプがそのカタログ・エントリーのレコードのいずれかの通貨タイプと一致していない場合には、新規レコードが LISTPRICE テーブル中に作成されることとなります。これにより、価格を異なる通貨で同じカタログ・エントリーに指定できます。

---

## Create\_WCS\_ExpectedInventoryRecord メッセージ

Create\_WCS\_ExpectedInventoryRecord メッセージは、WebSphere Commerce データベースに予定在庫レコードを作成するための情報を含むインバウンド・メッセージです。配送センター・アプリケーションがこの要求を生成し、WebSphere Commerce インバウンド・メッセージ・キューへ送信します。このメッセージはそこで WebSphere Commerce システムが受信します。WebSphere Commerce がこのメッセージを処理した後、ExpectedInventoryRecordCreate コマンドが呼び出されます。このコマンドによって RA テーブルにレコードが作成され、また、メッセージの中の RADETAIL コンポーネントごとに RADETAIL レコードが 1 つ作成されます。

注: インバウンド・メッセージに有効な StoreIDまたは有効な ExpectedDateがない場合、エラー・メッセージの生成に GenericApplicationError ビュー名が使用され、ExpectedInventoryRecordCreate コマンドは呼び出されません。 応答メッセージは、GenericApplicationErrorXML.jspにより生成されます。

Create\_WCS\_ExpectedInventoryRecord メッセージは XML メッセージ形式を使用し、Create\_WCS\_ExpectedInventoryRecord\_10.dtdに続きます。

以下の表は、Create\_WCS\_ExpectedInventoryRecord メッセージ形式の説明です。 データベース列の説明については、それと関連したテーブルへのリンクに従ってください。 特に注釈がない限り、すべてのフィールドはオプションです。タイプ INT、BIGINT、および DOUBLE のデータベース・フィールド用の XML メッセージ中のタグ値の長さは、それぞれ CHAR (10)、CHAR(19)、および CHAR(16) とする必要があります。

	フィールド名	Comment	テーブル名	列名	注
1	BackendRaID		N/A	N/A	応答メッセージで使用して元の要求と関連付けることができる
2	StoreID	必須	RA	STORE_ID	N/A
3	VendorID	必須	RA	VENDOR_ID	N/A
4	OrderDate	必須	RA	ORDERDATE	N/A
5	ExternalID		RA	EXTERNALID	N/A
6	FulfillmentCenterID	必須	RADETAIL	FFMCENTER_ID	N/A
7	ItemOwnerID	必須	ITEMSPC	MEMBER_ID	N/A
8	ProductSKU	必須	ITEMSPC	PARTNUMBER	ProductSKU は ItemOwnerID とともに使用してアイテムの仕様を識別する
9	ExpectedDate	必須	RADETAIL	EXPECTEDDATE	ISO 8601 日付形式
10	QuantityOrdered	必須	RADETAIL	QTYORDERED	N/A
11	Comment		RADETAIL	RADETAILCOMMENT	N/A
12	BackendRaDetailID		N/A	N/A	応答メッセージで使用して元の要求と関連付けることができる。

## Create\_WCS\_PickBatch メッセージ

Create\_WCS\_PickBatch メッセージは、WebSphere Commerce ピッキング・バッチの生成のための情報を含むインバウンド・メッセージです。 配送センター・アプリケーションがこの要求を生成し、WebSphere Commerce インバウンド・メッセージ・キューへ送信します。 WebSphere Commerce がこのメッセージを処理した後、PickBatchGenerate コントローラー・コマンドが呼び出されます。 このコマンドはキー・マネージャーを通して新しい PICKBATCH\_ID を取得し、ORDRELEASE データベース・テーブルの STATUS 列で SHIPという値を持つ行をすべて選択し、入力 StoreID と FulfillmentCenterIDのための XML のピック・チケットとパッキング・リストを生成して、それらを PICKBATCH テーブルと ORDRELEASE テーブルにそれぞれ保管します。 応答として Response\_WCS\_PickBatch メッセージが送信されます。

注: インバウンド・メッセージに有効な StoreIDがない場合、エラー・メッセージの生成に GenericApplicationError ビュー名が使用され、PickBatchGenerate コマンドは呼び出されません。 応答メッセージは、GenericApplicationErrorXML.jsp により生成されます。

Create\_WCS\_PickBatch メッセージは XML メッセージ形式を使用し、Create\_WCS\_PickBatch\_10.dtdに続きます。

以下の表は、Create\_WCS\_PickBatch メッセージ形式の説明です。 データベース列の説明については、それと関連したテーブルへのリンクに従ってください。 特に注釈がない限り、すべてのフィールドはオプションです。タイプ INT、BIGINT、および DOUBLE のデータベース・フィールド用の XML メッセージ中のタグ値の長さは、それぞれ CHAR (10)、CHAR(19)、および CHAR(16) とする必要があります。

レベル	フィールド名	コメント	テーブル名	列名	注
1	BackendPickBatchID		N/A	N/A	これは Response_WCS_PickBatch メッセージで使用可
2	StoreID	必須	ORDERITEMS	STORE_ID	N/A
3	FulfillmentCenterID	必須	ORDERITEMS	FFMCENTER_ID	N/A

## Inquire\_WCS\_PickPackListDetail メッセージ

Inquire\_WCS\_PickPackListDetail メッセージは、以前の Create\_WCS\_PickBatch メッセージで作成されたピック・チケット情報とパッキング・リスト情報を要求する、インバウンド・メッセージです。 配送センター・アプリケーションがこの要求を生成し、WebSphere Commerce インバウンド・メッセージ・キューへ送信します。 処理時には、インバウンド・メッセージは GetPickPackListDetail コマンドを呼び出します。これは、正常に終了すると PickPackListResult 表示タスクに、失敗すると PickPackListErrorView 表示タスクにリダイレクトします。 正常に完了すると、このコマンドは PICKBATCH テーブルからピック・チケット XML を、指定された PICKBATCH\_ID に対する ORDRELEASE テーブルからパッキング・リスト XML を検索します。 応答として PickPackListResult.jsp ファイルは、応答にピック・チケットおよびパッキング・スリップを含むアウトバウンドの Report\_WCS\_PickPackListDetail メッセージを生成します。

Inquire\_WCS\_PickPackListDetail メッセージは、XML メッセージ形式を使用し、Inquire\_WCS\_PickPackListDetail\_10.dtd に続きます。

以下の表は、Inquire\_PickPackListDetail メッセージ形式の説明です。 データベース列の説明については、それと関連したテーブルへのリンクに従ってください。 特に注釈がない限り、すべてのフィールドはオプションです。タイプ INT、BIGINT、および DOUBLE のデータベース・フィールド用の XML メッセージ中のタグ値の長さは、それぞれ CHAR (10)、CHAR(19)、および CHAR(16) とする必要があります。

レベル	XML エlement	コメント	テーブル名	列名	注
1	PickBatchID	必須	PICKBATCH	PICKBATCH_ID	N/A

## Create\_WCS\_InventoryReceipt メッセージ

Create\_WCS\_InventoryReceipt メッセージは、WebSphere Commerce 在庫レコードの作成のための情報を含むインバウンド・メッセージです。 配送センター・アプリケーションがこの要求を生成し、WebSphere Commerce インバウンド・メッセージ・キューへ送信します。このメッセージはそこで WebSphere Commerce システムが受信します。 WebSphere Commerce システムは、このメッセージを受信すると、ReceiptCreate コントローラー・コマンドを呼び出します。 このコマンドが正常に実行されると、応答として Response\_WCS\_CreateInvReceipt メッセージが送信されます。 予定在庫レコードに見合うだけの在庫がある場合、そのレコードはクローズされます。



**注:** インバウンド・メッセージに有効な StoreIDまたは有効な ReceiptDateがない場合、エラー・メッセージの生成に GenericApplicationError ビュー名が使用され、ReceiptCreate コマンドは呼び出されません。応答メッセージは、GenericApplicationErrorXML.jspにより生成されます。

Create\_WCS\_InventoryReceipt メッセージは XML メッセージ形式を使用し、Create\_WCS\_InventoryReceipt\_10.dtdに続きます。

以下の表は、Create\_WCS\_InventoryReceipt メッセージ形式の説明です。データベース列の説明については、それと関連したテーブルへのリンクに従ってください。特に注釈がない限り、すべてのフィールドはオプションです。タイプ INT、BIGINT、および DOUBLE のデータベース・フィールド用の XML メッセージ中のタグ値の長さは、それぞれ CHAR (10)、CHAR(19)、および CHAR(16) とする必要があります。

レベル	フィールド名	Comment	テーブル名	列名	注
1	ItemOwnerID	必須	ITEMSPC	MEMBER_ID	N/A
2	ProductNumberbyMerchant	必須	ITEMSPC	PARTNUMBER	N/A
3	VersionName		ITEMVERSN	VERSIONNAME	IBM 内部使用のために予約済み。
4	StoreID	必須	RECEIPT	STORE_ID	N/A
5	FulfillmentCenterID	必須	RECEIPT	FFMCENTER_ID	N/A
6	VendorID	必須	RECEIPT	VENDOR_ID	N/A
7	Cost	必須	RECEIPT	COST	N/A
8	Currency	必須	RECEIPT	SETCCUR	N/A
9	QTYReceived	必須	RECEIPT	QTYRECEIVED	N/A
10	ReceiptDate	必須	RECEIPT	RECEIPTDATE	ISO 8601 日付形式
11	WCSRaDetailID		RECEIPT	RADETAIL_ID	RA テーブルでの予定在庫レコード。これが存在しない場合、受取は ad-hoc タイプになります。
12	ReceiptComment		RECEIPT	COMMENT1	N/A
13	QualityComment		RECEIPT	COMMENT2	N/A

## Update\_WCS\_InventoryReceipt メッセージ

Update\_WCS\_InventoryReceipt メッセージは、アイテムの在庫を調整するための情報を含むインバウンド・メッセージです。配送センター・アプリケーションがこのメッセージを生成し、WebSphere Commerce インバウンド・メッセージ・キューへ送信します。このメッセージはそこで WebSphere Commerce システムが受信します。WebSphere Commerce がこのメッセージを受信した後、InventoryAdjust コマンドが呼び出されます。

**注:** インバウンド・メッセージに有効な StoreIDがない場合、エラー・メッセージの生成に GenericApplicationError ビュー名が使用され、InventoryAdjust コマンドは呼び出されません。応答メッセージは、GenericApplicationErrorXML.jspにより生成されます。

Update\_WCS\_InventoryReceipt メッセージは XML メッセージ形式を使用し、Update\_WCS\_InventoryReceipt\_10.dtdに続きます。

以下の表は、Update\_WCS\_InventoryReceipt メッセージ形式の説明です。データベース列の説明については、それと関連したテーブルへのリンクに従ってください。特に注釈がない限り、すべてのフィールドは

オプションです。タイプ INT、BIGINT、および DOUBLE のデータベース・フィールド用の XML メッセージ中のタグ値の長さは、それぞれ CHAR (10)、CHAR(19)、および CHAR(16) とする必要があります。

レベル	フィールド名	コメント	テーブル名	列名	注
1	ItemOwnerID	必須	ITEMSPC	MEMBER_ID	N/A
2	ProductSKU	必須	ITEMSPC	PARTNUMBER	N/A
3	VersionName		ITEMVERSN	VERSIONNAME	IBM 内部使用のために予約済み。
4	StoreID	必須	RECEIPT	STORE_ID	N/A
5	FulfillmentCenterID	必須	RECEIPT	FFMCENTER_ID	N/A
6	コメント		RECEIPT	COMMENT1	N/A
7	QTYAdjusted	必須	INVADJUST	QUANTITY	正の値あるいは負の値
8	InvAdjCodeID	必須	INVADJUST	INVADJCODE_ID	N/A

## Create\_WCS\_ShipmentConfirmation メッセージ

Create\_WCS\_ShipmentConfirmation メッセージは、アイテムの配送確認を発行するための情報を含むインバウンド・メッセージです。 配送センター・アプリケーションがこの要求を生成し、 WebSphere Commerce インバウンド・メッセージ・キューへ送信します。 WebSphere Commerce がこのメッセージを処理した後、ReleaseShipConfirm コマンドが呼び出されます。

このコマンドは正常に終了すると、表示タスク ReleaseShipConfirmRedirectView にリダイレクトします。 Response\_WCS\_CreateShipConfirm 応答メッセージは、 CreateShipConfirmOK.jspによって生成されます。 コマンドは必要なデータベースを更新し、アイテムの実行状況を変更して配送を確認します。コマンドはキー・マネージャーを介して MANIFEST テーブルから新しい manifest\_id を取得し、入力データとともに MANIFEST テーブルを伝搬します。コマンドが正常に実行されて UpdateManifestStatus が 1 の場合、デフォルトの ReleaseShipNotify.jsp によって通知 E メールが生成されます。

コマンドでエラーが発生すると、コマンドによって表示タスク ReleaseShipConfirmErrorView にリダイレクトされます。 MQSeries 用のこのエラー表示タスクは、 CreateShipConfirmError.jsp によってインプリメントされています。

**注:** インバウンド・メッセージに有効な ActualShipDateがない場合、エラー・メッセージの生成に GenericApplicationError ビュー名が使用されます。 応答メッセージは、GenericApplicationErrorXML.jspにより生成されます。

Create\_WCS\_ShipmentConfirmation メッセージは XML メッセージ形式を使用し、 Create\_WCS\_ShipmentConfirmation\_10.dtdに続きます。

以下のテーブルは、Create\_WCS\_ShipmentConfirmation メッセージ形式の説明です。 データベース列の説明については、それと関連したテーブルへのリンクに従ってください。 特に注釈がない限り、すべてのフィールドはオプションです。タイプ INT、BIGINT、および DOUBLE のデータベース・フィールド用の XML メッセージ中のタグ値の長さは、それぞれ CHAR (10)、CHAR(19)、および CHAR(16) とする必要があります。

レベル	フィールド名	コメント	テーブル名	列名	注
1	UpdateManifestStatus		MANIFEST	UPDATEMANIFESTSTATUS	デフォルト 0。親エレメント ShipmentConfirmation の属性
2	ShipModeID	必須	MANIFEST	SHIPMODE_ID	N/A

レベル	フィールド名	コメント	テーブル名	列名	注
3	OrderNumber	必須	MANIFEST	ORDERS_ID	N/A
4	OrderReleaseNum	必須	MANIFEST	ORDERRELEASENUM	N/A
5	PackageID		MANIFEST	PACKAGEID	N/A
6	TrackingID		MANIFEST	TRACKINGID	N/A
7	PickUpRecordID		MANIFEST	PICKUPRECORDID	N/A
8	ActualShipDate	必須	MANIFEST	DATESHIPPED	ISO 8601 日付形式
9	ShippingCosts	必須	MANIFEST	SHIPPINGCOSTS	N/A
10	Weight	必須	MANIFEST	WEIGHT	N/A
11	measure	必須	MANIFEST	WEIGHTMEASURE	Weight の属性
12	currency	必須	MANIFEST	SETCCURR	ShippingCosts の属性

UpdateManifestStatus フラグのデフォルト値は 0 です。これが 1 に設定されると、表示される情報を更新するためにタスク・コマンドが呼び出され、商品の発送が完了したことを顧客に通知する E メールが送信されます。

応答メッセージを生成する JSP ファイルは、Response\_WCS\_CreateShipConfirm.jsp です。

ReleaseShipNotify.jsp はデフォルト JSP ファイルであり、UpdateManifestStatus フラグが 1 に設定されている場合には E メールを生成します。JSP ファイルの位置についての詳細は、メッセージ構成テンプレートを参照してください。

---

## Create\_WCS\_Customer メッセージ

Create\_WCS\_Customer メッセージは、ショッパーの顧客情報を含むインバウンド・メッセージです。バックエンド・アプリケーションがこのメッセージを生成し、WebSphere Commerce インバウンド・メッセージ・キューへ送信します。このメッセージはそこで WebSphere Commerce システムが受信します。

WebSphere Commerce は、UserRegistrationAdd コマンドを呼び出して新規ショッパーの情報を追加します。

Create\_WCS\_Customer メッセージは、XML メッセージ形式を使用し、Create\_WCS\_Customer\_10.dtd に従います。

このメッセージのデータベース・フィールドへのフィールドのマッピング形式は、Update\_WCS\_Customer メッセージの形式に似ています。

---

## Update\_WCS\_Customer メッセージ

Update\_WCS\_Customer メッセージは、ショッパーの顧客情報が入っているインバウンド・メッセージです。バックエンド・アプリケーションがこのメッセージを生成し、WebSphere Commerce インバウンド・メッセージ・キューへ送信します。このメッセージはそこで WebSphere Commerce システムが受信します。WebSphere Commerce は、UserRegistrationUpdate を呼び出して顧客に関する情報を更新します。

以下のテーブルは、Update\_WCS\_Customer メッセージ形式の説明です。データベース列の説明については、それと関連したテーブルへのリンクに従ってください。特に注釈がない限り、すべてのフィールドはオプションです。タイプ INT、BIGINT、および DOUBLE のデータベース・フィールド用の XML メッセージ中のタグ値の長さは、それぞれ CHAR (10)、CHAR(19)、および CHAR(16) とする必要があります。

レベル	フィールド名	コメント	テーブル名	列名
1	登録	必須	N/A	N/A
1.1	LogonInfo	必須	N/A	N/A
1.1.1	LogonID	必須	USERREG	LOGONID
1.1.2	パスワード		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	Attribute	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	敬称		ADDRESS	PERSONTITLE
2.4.2	LastName		ADDRESS	LASTNAME
2.4.3	FirstName		ADDRESS	FIRSTNAME
2.4.4	MiddleName		ADDRESS	MiddleName
2.5	住所		N/A	N/A
2.5.A1	primary	Attribute	ADDRESS	ISPRIMARY
2.5.A2	self	Attribute	ADDRESS	SELFADDRESS
2.5.1	AddressLine	初回出現	ADDRESS	ADDRESS1
2.5.1	AddressLine	2 番目の出現	ADDRESS	ADDRESS2
2.5.1	AddressLine	3 番目の出現	ADDRESS	ADDRESS3
2.5.2	市区町村		ADDRESS	City
2.5.3	State		ADDRESS	State
2.5.4	ZipCode		ADDRESS	ZIPCODE
2.5.5	Country		ADDRESS	COUNTRY
2.6	ContactInfo		N/A	N/A
2.6.1	Telephone	初回出現	ADDRESS	PHONE1
2.6.1	Telephone	2 番目の出現	ADDRESS	PHONE2
2.6.1.A1	type	Attribute	ADDRESS	PHONE1TYPE/ PHONE2TYPE
2.6.1.A2	publish	Attribute	ADDRESS	PUBLISH PHONE1 / PUBLISH PHONE2
2.6.2	BestCallingTime		ADDRESS	BESTCALLINGTIME
2.6.3	Fax	初回出現	ADDRESS	FAX1
2.6.3	Fax	2 番目の出現	ADDRESS	FAX2
2.6.4	Email	初回出現	ADDRESS	EMAIL1
2.6.4	Email	2 番目の出現	ADDRESS	EMAIL2

レベル	フィールド名	コメント	テーブル名	列名
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	初回出現	ADDRESS	FIELD1
2.9	AddressField	2 番目の出現	ADDRESS	FIELD2
2.9	AddressField	3 番目の出現	ADDRESS	FIELD3
3	Profile		N/A	N/A
3.A1	type	Attribute	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	初回出現	USERS	FIELD1
3.1.4	UserField	2 番目の出現	USERS	FIELD2
3.1.4	UserField	3 番目の出現	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	説明		USERPROF	DESCRIPTION
3.1.14	UserProfileField	初回出現	USERPROF	FIELD1
3.1.14	UserProfileField	2 番目の出現	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
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

レベル	フィールド名	コメント	テーブル名	列名
3.3.2	Gender		USERDEMO	GENDER
3.3.3	Income		USERDEMO	INCOME
3.3.3.A1	Currency	Attribute	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	初回出現	USERDEMO	FIELD1
3.3.11	DemographicField	2 番目の出現	USERDEMO	FIELD2
3.3.11	DemographicField	3 番目の出現	USERDEMO	FIELD3
3.3.11	DemographicField	4 番目の出現	USERDEMO	FIELD4
3.3.11	DemographicField	5 番目の出現	USERDEMO	FIELD5
3.3.11	DemographicField	6 番目の出現	USERDEMO	FIELD6
3.3.11	DemographicField	7 番目の出現	USERDEMO	FIELD7
4	UserData		N/A	N/A
4.1	UserDataField		N/A	N/A

## Update\_WCS\_ProductInventory メッセージ

Update\_WCS\_ProductInventory メッセージは、商品の在庫情報を含むインバウンド・メッセージです。バックエンド・アプリケーションがこのメッセージを生成し、WebSphere Commerce インバウンド・メッセージ・キューへ送信します。このメッセージはそこで WebSphere Commerce システムが受信します。WebSphere Commerce は、INVENTORY テーブルを新規在庫情報で更新します。

Update\_WCS\_ProductInventory メッセージは XML メッセージ形式を使用し、Update\_WCS\_ProductInventory\_20.dtd に続きます。

以下のテーブルは、XML エlement 値の形式およびソースの説明です。データベース列の説明については、それと関連したテーブルへのリンクに従ってください。特に注釈がない限り、フィールドはオプションです。タイプ INT、BIGINT、および DOUBLE のデータベース・フィールド用の XML メッセージ中のタグ値の長さは、それぞれ CHAR (10)、CHAR(19)、および CHAR(16) とする必要があります。

レベル	XML エlement	コメント	テーブル名	列名	注
1	ProductNumber ByMerchant	必須 (注 を参照)	INVENTORY	CATENTRY_ID	ProductSKU が使用されない場合にはのみ必須。ProductSKU が使用されている場合には、この属性は使用してはなりません。
2	MerchantID	必須	INVENTORY	STORE_ID	
3	数量	必須	INVENTORY	QUANTITY	
4	UserData		N/A	N/A	

レベル	XML エlement	コメント	テーブル名	列名	注
5	FulfillmentCenterID		INVENTORY	FFMCENTER_ID	
6	ProductSKU	必須 (注を参照)	CATENTRY	PARTNUMBER	ProductNumberByMerchant が使用されない場合のみ必須。 ProductNumberByMerchant が使用されている場合には、この属性は使用してはなりません。

#### 動作:

- WebSphere Commerce 商品参照番号 (INVENTORY テーブル中の CATENTRY\_ID を指す) およびマーチャント参照番号 (STORE\_ID) が INVENTORY テーブル中の行を更新するために使用されます。
- INVENTORY テーブル中の行が存在していないと、例外が起こります。

## 商品価格の更新メッセージ

商品価格の更新メッセージは、商品の価格情報を含むインバウンド・メッセージです。バックエンド・アプリケーションがこのメッセージを生成し、WebSphere Commerce インバウンド・メッセージ・キューへ送信します。WebSphere Commerce システムは、このメッセージを受信すると、ProductOfferPriceUpdate コマンドを実行します。

商品価格の更新メッセージは、商品価格の更新 バージョン 01 および商品価格の更新 バージョン 02 のアプリケーション・データ 2 セットをサポートしています。バージョン 02 は、バージョン 01 のデータのスーパーセットです。

商品価格の更新メッセージは WebSphere Commerce のメッセージ・フォーマットが使用され、バッファ内で相互に順次に並んでいる一連のレコードで構成されています。以下のデータは、商品価格の更新メッセージについての説明です:

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

#### 注:

- すべてのレコードはバッファ内で順次の並びです。読みやすくするために、バッファにはありませんが、ここでは字下げが使用されています。
- データ・セグメントのすべてのフィールドは、バッファ内では左寄せ、右スペース埋め込みです。

#### 商品価格更新のデータ・セグメント

- PRODUCT\_PRICE\_UPDATE\_PROLOG\_DATA  
アプリケーション・データが定義するメッセージ・タイプを指定します。このケースでは、メッセージは商品価格更新です。
- PRODUCT\_PRICE\_UPDATE\_HDR010\_DATA  
商品価格の更新メッセージのアイテムまたは商品価格情報を指定します。

---

## 商品数量の更新メッセージ

商品数量の更新メッセージは、商品の価格情報を含むインバウンド・メッセージです。バックエンド・アプリケーションがこのメッセージを生成し、WebSphere Commerce インバウンド・メッセージ・キューへ送信します。このメッセージはそこで WebSphere Commerce システムが受信します。

商品数量の更新メッセージは WebSphere Commerce のメッセージ・フォーマットが使用され、バッファー内で相互に順次に並んでいる一連のレコードで構成されています。以下のデータは、商品数量の更新メッセージについての説明です：

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

### 注：

- すべてのレコードはバッファー内で順次の並びです。読みやすくするために、バッファーにはありませんが、ここでは字下げが使用されています。
- データ・セグメントのすべてのフィールドは、バッファー内では左寄せ、右スペース埋め込みです。

### 商品数量更新のデータ・セグメント

- PRODUCT\_QUANTITY\_UPDATE\_PROLOG\_DATA  
アプリケーション・データが定義するメッセージ・タイプを指定します。このケースでは、メッセージは商品数量更新です。
- PRODUCT\_QUANTITY\_UPDATE\_HDR010\_DATA  
商品数量の更新メッセージ内の商品情報を指定します。

---

## オーダー作成メッセージ

オーダー作成メッセージは完了オーダーのオーダー詳細情報を含むアウトバウンド・メッセージです。WebSphere Commerce システムがこのメッセージを生成し、そのデータをアウトバウンド・メッセージ・キューへ送信します。このメッセージはそこでバックエンド・システムが受信します。このアプリケーションはメッセージを取り込み、オーダーの指定タスクの完了に必要なバックエンド・ビジネス処理を続けます。Order Create を使用して、WebSphere Commerce システムから開始するオーダー処理を実行しますが、そのオーダー処理では別のシステムによる追加やバックエンド作業もいくらか必要になります。

Order Create メッセージは、レガシー・メッセージ・フォーマットを使用し、バッファー内で相互に順次に並んでいる一連のレコードで構成されています。そのメッセージはオーダー、ショッパー、請求先、マーチャントおよび配送情報を含んでいます。以下のデータは、オーダー作成メッセージについての説明です：

```
<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 繰り返しループ
    .
  <DATUSR>DATUSR_DATA</DATUSR>
```



```

    </USRLST>
</HEADER>
<ITMLST>
  <ITMDAT>
    <ITM010>ORDER_CREATE_ITM010_DATA</ITM010>
    <USRLST>
      <DATUSR>DATUSR_DATA</DATUSR>
      .DATUSR 繰り返しループ
      .
    <DATUSR>DATUSR_DATA</DATUSR>
    </USRLST>
  </ITMDAT>
  .
  <ITMDAT>
    <ITM010>ORDER_CREATE_ITM010_DATA</ITM010>
    <USRLST>
      <DATUSR>DATUSR_DATA</DATUSR>
      .DATUSR 繰り返しループ
      .
    <DATUSR>DATUSR_DATA</DATUSR>
    </USRLST>
  </ITMDAT>
</ITMLST>
</ECEDOC>

```

**注:**

- すべてのレコードはバッファ内で順次の並びです。読みやすくするために、バッファにはありませんが、ここでは字下げが使用されています。
- データ・セグメントのすべてのフィールドは、バッファ内では左寄せ、右スペース埋め込みです。

**オーダー作成のデータ・セグメント**

- **ORDER\_CREATE\_PROLOG\_DATA**  
アプリケーション・データが定義するメッセージ・タイプを指定します。このケースでは、オーダー作成に対する更新メッセージです。
- **ORDER\_CREATE\_HDR010\_DATA**  
オーダー作成メッセージ内のオーダー情報を指定します。
- **ORDER\_CREATE\_HDR020\_DATA**  
オーダー作成メッセージ内のショッパー情報を指定します。
- **ORDER\_CREATE\_HDR030\_DATA**  
オーダー作成メッセージ内の請求先情報を指定します。
- **ORDER\_CREATE\_HDR040\_DATA**  
オーダー作成メッセージ内のマーチャント情報を指定します。
- **DATUSR\_DATA**  
オーダー作成メッセージへ追加されるオプションの情報を指定します。DATUSR\_DATA は、このメッセージの <HDR> および <ITM>セクションにあります。
- **ORDER\_CREATE\_ITM010\_DATA**  
オーダー作成メッセージ内のアイテムまたは商品配送情報を指定します。

---

## 新規顧客へのメッセージ

新規顧客へのメッセージは、レガシー形式 (ショッパーの顧客情報を含むインバウンド・メッセージ) です。バックエンド・アプリケーションがこのメッセージを生成し、WebSphere Commerce インバウンド・メッセージ・キューへ送信します。このメッセージはそこで WebSphere Commerce システムが受信します。WebSphere Commerce は、UserRegistrationAdd コントローラー・コマンドを呼び出して新規ショッパーの情報を更新します。

以下のデータは新規顧客へのメッセージについて説明しています:

```
<ECEDOC>
<PROLOG>CUSTOMER_UPDATE_PROLOG_DATA</PROLOG>
<HEADER>
  <HDR010>CUSTOMER_UPDATE_HDR010_DATA</HDR010>
  <USRLST>
    <DATUSR>DATUSR_DATA</DATUSR>
      .
      .DATUSR 繰り返しループ
      .
    <DATUSR>DATUSR_DATA</DATUSR>
  </USRLST>
</HEADER>
</ECEDOC>
```

### 注:

- すべてのレコードはバッファー内で順次の並びです。読みやすくするために、バッファーにはありませんが、ここでは字下げが使用されています。
- データ・セグメントのすべてのフィールドは、バッファー内では左寄せ、右スペース埋め込みです。

### 新規顧客のデータ・セグメント

- CUSTOMER\_NEW\_PROLOG\_DATA  
アプリケーション・データが定義するメッセージ・タイプを指定します。このケースでは、新規顧客へのメッセージです。
- CUSTOMER\_NEW\_HDR010\_DATA  
新規顧客へのメッセージ内の情報を指定します。
- DATUSR\_DATA  
新規顧客へのメッセージへ追加されるオプションの情報を指定します。DATUSR\_DATA は、このメッセージの <HDR> セクションにあります。

---

## 更新顧客へのメッセージ

更新顧客へのメッセージは、ショッパーの顧客更新を含むインバウンド・メッセージです。バックエンド・アプリケーションがこのメッセージを生成し、WebSphere Commerce インバウンド・メッセージ・キューへ送信します。このメッセージはそこで WebSphere Commerce システムが受信します。WebSphere Commerce は、UserRegistrationUpdate コマンドを呼び出して登録済みショッパーに関する情報を更新します。

更新顧客へのメッセージは WebSphere Commerce のメッセージ・フォーマットが使用され、バッファー内で相互に順次に並んでいる一連のレコードで構成されています。以下のデータは更新顧客へのメッセージについて説明しています:

```
<ECEDOC>
<PROLOG>CUSTOMER_UPDATE_PROLOG_DATA</PROLOG>
<HEADER>
```

```

<HDR010>CUSTOMER_UPDATE_HDR010_DATA</HDR010>
<USRLST>
  <DATUSR>DATUSR_DATA</DATUSR>
    .DATUSR 繰り返しループ
  <DATUSR>DATUSR_DATA</DATUSR>
</USRLST>
</HEADER>
</ECEDOC>

```

**注:**

- すべてのレコードはバッファ内では順次の並びです。読みやすくするために、バッファにはありませんが、ここでは字下げが使用されています。
- データ・セグメントのすべてのフィールドは、バッファ内では左寄せ、右スペース埋め込みです。

**更新顧客のデータ・セグメント**

- **CUSTOMER\_UPDATE\_PROLOG\_DATA**  
アプリケーション・データが定義するメッセージ・タイプを指定します。このケースでは、顧客に対する更新メッセージです。
- **CUSTOMER\_UPDATE\_HDR010\_DATA**  
更新顧客へのメッセージ内の情報を指定します。
- **DATUSR\_DATA**  
更新顧客へのメッセージへ追加されるオプションの情報を指定します。DATUSR\_DATA は、このメッセージの <HDR> セクションにあります。

---

## Update\_NC\_Customer メッセージ

Update\_NC\_Customer メッセージは、ショッパーの顧客情報を含むインバウンド・メッセージです。バックエンド・アプリケーションがこのメッセージを生成し、WebSphere Commerce インバウンド・メッセージ・キューへ送信します。このメッセージはそこで WebSphere Commerce システムが受信します。WebSphere Commerce は、UserRegistrationUpdate コマンドを呼び出して登録済みショッパーに関する情報を更新します。このメッセージは同じ WebSphere Commerce コントローラー・コマンドを呼び出すため、このメッセージは Create\_NC\_Customer メッセージと同一のものであります。

Update\_NC\_Customer メッセージは XML メッセージ・フォーマットを使用し、Update\_NC\_Customer\_10.dtd ファイルに従います。

以下のテーブルは、Update\_NC\_Customer メッセージ形式の説明です。データベース列の説明については、それと関連したテーブルへのリンクに従ってください。特に注釈がない限り、すべてのフィールドはオプションです。タイプ INT、BIGINT、および DOUBLE のデータベース・フィールド用の XML メッセージ中のタグ値の長さは、それぞれ CHAR (10)、CHAR(19)、および CHAR(16) とする必要があります。

レベル	フィールド名	コメント	テーブル名	列名
1	LoginInfo	必須	N/A	N/A
1.1	LoginID	必須	USERREG	LOGONID
1.2	パスワード		USERREG	LOGONPASSWORD
1.3	VerifyPassword		USERREG	LOGONPASSWORD
2	MerchantID		N/A	N/A
2.A1	type	Attribute	N/A	N/A

レベル	フィールド名	コメント	テーブル名	列名
3	MethodOfCommunication		USERPROF	PREFERREDCOMM
4	ChallengeQuestion		USERREG	CHALLENGEQUESTION
5	ChallengeAnswer		USERREG	CHALLENGEANSWER
6	ShopperField	初回出現	USERS	FIELD1
6	ShopperField	2 番目の出現	USERS	FIELD2
6	ShopperField	3 番目の出現	USERS	FIELD3
7	ContactPersonName		N/A	N/A
7.1	敬称		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	住所		N/A	N/A
9.1	AddressLine	初回出現	ADDRESS	ADDRESS1
9.1	AddressLine	2 番目の出現	ADDRESS	ADDRESS2
9.1	AddressLine	3 番目の出現	ADDRESS	ADDRESS3
9.2	市区町村		ADDRESS	City
9.3	State		ADDRESS	State
9.4	Zip		ADDRESS	ZIPCODE
9.5	Country		ADDRESS	COUNTRY
10	ContactInfo		N/A	N/A
10.1	Telephone	初回出現	ADDRESS	PHONE1
10.1	Telephone	2 番目の出現	ADDRESS	PHONE2
10.2	Email	初回出現	ADDRESS	EMAIL1
10.2	Email	2 番目の出現	ADDRESS	EMAIL2
10.3	Fax		ADDRESS	FAX1
11	DayPhoneInfo		N/A	N/A
11.1	PhoneInfo		N/A	N/A
11.1.A1	type	Attribute	ADDRESS	PHONE1TYPE
11.1.A2	isListed	Attribute	ADDRESS	PUBLISHPHONE1
12	EveningPhoneInfo		N/A	N/A
12.1	PhoneInfo		N/A	N/A
12.1.A1	type	Attribute	ADDRESS	PHONE2TYPE
12.1.A2	isListed	Attribute	ADDRESS	PUBLISHPHONE2
13	BestTimeToCall		ADDRESS	BESTCALLINGTIME
14	IncludePackageInsert		ADDRESS	PACKAGESUPPRESSION
15	AddressOptField	初回出現	ADDRESS	FIELD1
15	AddressOptField	2 番目の出現	ADDRESS	FIELD2
15	AddressOptField	3 番目の出現	ADDRESS	FIELD3

レベル	フィールド名	コメント	テーブル名	列名
16	Gender		N/A	N/A
16.A1	value	Attribute	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	初回出現	USERDEMO	FIELD1
25	Demographics	2 番目の出現	USERDEMO	FIELD2
25	Demographics	3 番目の出現	USERDEMO	FIELD3
25	Demographics	4 番目の出現	USERDEMO	FIELD4
25	Demographics	5 番目の出現	USERDEMO	FIELD5
25	Demographics	6 番目の出現	USERDEMO	FIELD6
25	Demographics	7 番目の出現	USERDEMO	FIELD7
26	UserData		N/A	N/A
26.1	UserDataField		N/A	N/A

## Update\_NC\_OrderStatus メッセージ

Update\_NC\_OrderStatus メッセージは、 WebSphere Commerce オーダーの状況情報を含むインバウンド・メッセージです。 バックエンド・アプリケーションがこのメッセージを生成し、 WebSphere Commerce インバウンド・メッセージ・キューへ送信します。このメッセージはそこで WebSphere Commerce システムが受信します。 WebSphere Commerce の OrderStatus コマンドは、 ORDSTAT および ORDISTAT テーブルを新しいオーダー状況情報で更新します。

Update\_NC\_OrderStatus メッセージは、XML メッセージ形式を使用し、 Update\_NC\_OrderStatus\_10.dtd に従います。

以下のテーブルは、Update\_NC\_OrderStatus メッセージの形式の説明です。 データベース列の説明については、それと関連したテーブルへのリンクに従ってください。 特に注釈がない限り、すべてのフィールドはオプションです。タイプ INT、BIGINT、および DOUBLE のデータベース・フィールド用の XML メッセージ中のタグ値の長さは、それぞれ CHAR (10)、CHAR(19)、および CHAR(16) とする必要があります。

レベル	フィールド名	コメント	テーブル名	列名	注
1	OrderStatus Header		N/A	N/A	N/A
1.1	OrderNumber ByBackend		ORDSTAT	OSMORDER	バックエンド・システムによって生成されたオーダー参照番号。

レベル	フィールド名	コメント	テーブル名	列名	注
1.2	OrderNumber ByNC	必須	ORDSTAT	ORDERS_ID	WebSphere Commerce のオー ダー参照番号。こ れは、テーブル ORDERS 中の列 ORDERS_ID を参 照する外部キーで す。
1.3	OrderNumber ByBuyer		N/A	N/A	N/A
1.4	TotalPriceInfo		N/A	N/A	N/A
1.4.A1	currency	Attribute	ORDSTAT	OSPCUR	価格を表す通貨。 通貨の形式は、 ISO 4217 規格に 準拠する必要があります。
1.4.1	TotalNetPrice		ORDSTAT	OSPRTOT	オーダーの商品価 格合計
1.4.2	TotalTaxPrice		ORDSTAT	OSTXTOT	オーダーの消費税 合計
1.4.3	TotalShippingPrice		ORDSTAT	OSSHTOT	オーダーの配送料 合計
1.4.4	TotalTaxOn ShippingPrice		ORDSTAT	OSSHTXTOT	オーダーの配送料 金税合計
1.5	RequisitionerID		N/A	N/A	N/A
1.5.A1	type	Attribute	N/A	N/A	N/A
1.6	Status		ORDSTAT	OSSTATUS	オーダーの状況
1.7	DateTime Reference		N/A	N/A	N/A
1.7.1	PlacedDate		ORDSTAT	OSPLTIME	オーダー発行タイ ム・スタンプ
1.7.2	PlacedTime		N/A	N/A	N/A
1.7.3	LastUpdateDate		N/A	N/A	N/A
1.7.4	LastUpdateTime		ORDSTAT	OSUPDIME	オーダーの最終更 新タイム・スタ ンプ。
1.8	ShipDate Reference		N/A	N/A	N/A
1.8.1	Requested ShipDate		ORDSTAT	OSRSTIME	配送要求タイム・ スタンプ。
1.8.2	Scheduled ShipDate		ORDSTAT	OSSSTIME	配送スケジュー ル・タイム・スタ ンプ。
1.8.3	Actual ShipDate		ORDSTAT	OSASTIME	実際の配送タイ ム・スタンプ。

レベル	フィールド名	コメント	テーブル名	列名	注
1.9	CustomerField	初回出現。	ORDSTAT	FIELD1	カスタマイズ用に予約済み。
1.9	CustomerField	2 番目の出現。	ORDSTAT	FIELD2	カスタマイズ用に予約済み。
1.9	CustomerField	3 番目の出現。	ORDSTAT	FIELD3	カスタマイズ用に予約済み。
1.10	UserData		N/A	N/A	N/A
2	OrderStatusItem		N/A	N/A	N/A
2.1	OrderNumber ByBackend		ORDISTAT	OSMORDER	バックエンド・システム・オーダー番号
2.2	ItemNumber ByBackend		ORDISTAT	OIMITEM	バックエンドシステム・オーダー・アイテム番号
2.3	OrderNumber ByNC	必須	ORDISTAT	ORDERS_ID	WebSphere Commerce のオーダー参照番号。これは、テーブル ORDERS 中の列 ORDERS_ID を参照する外部キーです。
2.4	ItemNumber ByNC		ORDISTAT	ORDER ITMES_ID	WebSphere Commerce のアイテム参照番号。これは、テーブル ORDERITEMS 中の列 ORDERITEMS_ID を参照する外部キーです。
2.5	数量		ORDISTAT	OIQTCONFIRM	確認されたアイテムの数量。
2.6	ItemUnitPrice		ORDISTAT	OIUNPRC	単価 単価。
2.7	TotalPriceInfo		N/A	N/A	N/A
2.7.A1	currency	Attribute	ORDISTAT	OICPCUR	アイテムの価格を表す通貨。通貨の形式は、ISO 4217 規格に準拠する必要があります。
2.7.1	TotalNetPrice		ORDISTAT	OIPRTOT	アイテムの合計価格
2.7.2	TotalTaxPrice		ORDISTAT	OITXTOT	アイテムの税額合計。

レベル	フィールド名	コメント	テーブル名	列名	注
2.7.3	TotalShippingPrice		ORDISTAT	OISHTOT	アイテムの配送料合計。
2.7.4	TotalTaxOnShippingPrice		ORDISTAT	OISHTXTOT	アイテムの配送料に対する税額合計。
2.8	Status		ORDISTAT	OISTATUS	オーダーのアイテム状況
2.9	DateTimeReference		N/A	N/A	N/A
2.9.1	PlacedDate		ORDISTAT	OIPLTIME	オーダー・アイテム発行タイム・スタンプ。
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	配送要求タイム・スタンプ。
2.10.2	ScheduledShipDate		ORDISTAT	OISSDATE	配送スケジュール・タイム・スタンプ。
2.10.3	ActualShipDate		ORDISTAT	OIASTIME	実際の配送タイム・スタンプ。
2.11	Instruction		ORDISTAT	OICMNT	オーダーされたアイテムに関するショッパーからのコメント。たとえば、ショッパーはオーダーした贈答品と一緒にあいさつ文を付けることができます。
2.12	CustomerField	初回出現。	N/A	N/A	カスタマイズ用に予約済み。
2.12	CustomerField	2 番目の出現。	N/A	N/A	カスタマイズ用に予約済み。
2.12	CustomerField	3 番目の出現。	N/A	N/A	カスタマイズ用に予約済み。
2.13	UserData		N/A	N/A	N/A



---

## Create\_NC\_Customer メッセージ

Create\_NC\_Customer メッセージは、ショッパーの顧客情報を含むインバウンド・メッセージです。バックエンド・アプリケーションがこのメッセージを生成し、WebSphere Commerce インバウンド・メッセージ・キューへ送信します。このメッセージはそこで WebSphere Commerce システムが受信します。WebSphere Commerce は、UserRegistrationAdd コマンドを呼び出して新規ショッパーの情報を更新します。

Create\_NC\_Customer メッセージは、XML メッセージ形式を使用し、Create\_NC\_Customer\_10.dtd に従います。

このメッセージのデータベース・フィールドへのフィールドのマッピング形式は、Update\_NC\_Customer メッセージの形式に似ています。

---

## Update\_NC\_ProductInventory メッセージ

Update\_NC\_ProductInventory メッセージは、商品の在庫情報を含むインバウンド・メッセージです。バックエンド・アプリケーションがこのメッセージを生成し、WebSphere Commerce インバウンド・メッセージ・キューへ送信します。このメッセージはそこで WebSphere Commerce システムが受信します。WebSphere Commerce は、INVENTORY テーブルを新規在庫情報で更新します。

Update\_NC\_ProductInventory メッセージは、XML メッセージ形式を使用し、Update\_NC\_ProductInventory\_10.dtd に従います。

**注:** Update\_WCS\_ProductInventory メッセージは改良バージョンであるので、Update\_NC\_ProductInventory の代わりに使用することをお勧めします。Update\_WCS\_ProductInventory メッセージを使用する場合は、オプションで ProductNumberByMerchant フィールドを ProductSKU に置き換えることができます。

以下のテーブルは、XML エlement 値の形式およびソースの説明です。データベース列の説明については、それと関連したテーブルへのリンクに従ってください。特に注釈がない限り、フィールドはオプションです。タイプ INT、BIGINT、および DOUBLE のデータベース・フィールド用の XML メッセージ中のタグ値の長さは、それぞれ CHAR (10)、CHAR(19)、および CHAR(16) とする必要があります。

レベル	XML エlement	コメント	テーブル名	列名
1	ProductNumber ByMerchant	必須	INVENTORY	CATENTRY_ID
2	MerchantID	必須	INVENTORY	STORE_ID
3	数量	必須	INVENTORY	QUANTITY
4	UserData		N/A	N/A

### 動作:

- WebSphere Commerce 商品参照番号 (INVENTORY テーブル中の CATENTRY\_ID を指す) およびマーチャント参照番号 (STORE\_ID) が INVENTORY テーブル中の行を更新するために使用されます。
- INVENTORY テーブル中の行が存在していないと、例外が起こります。

---

## Update\_NC\_ProductPrice メッセージ

Update\_NC\_ProductPrice メッセージは、商品の価格情報を含むインバウンド・メッセージです。バックエンド・アプリケーションがこのメッセージを生成し、WebSphere Commerce インバウンド・メッセージ・キューへ送信します。このメッセージはそこで WebSphere Commerce システムが受信します。その後、システムは、OFFERPRICE テーブルを新規価格情報で更新する ProductOfferPriceUpdate コマンドを呼び出します。

Update\_NC\_ProductPrice メッセージは、XML メッセージ形式を使用し、Update\_NC\_ProductPrice\_10.dtd に従います。

以下のテーブルは、XML エlement値の形式およびソースの説明です。データベース列の説明については、それと関連したテーブルへのリンクに従ってください。特に注釈がない限り、すべてのフィールドはオプションです。タイプ INT のデータベース・フィールドの場合は、メッセージ・Elementは CHAR (12) にする必要があります。

レベル	XML Element	コメント	テーブル名	列名
1	ProductNumber ByMerchant	必須	CATENTRY	PARTNUMBER
2	MerchantID	必須	CATENTRY	MEMBER_ID
3	PriceGroupInfo		N/A	N/A
3.1	Precedence		OFFER	PRECEDENCE
3.2	Requisitioner GroupID		OFFER	TRADEPOSCN_ID
4	Currency	必須	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 メッセージ

Report\_NC\_PurchaseOrder メッセージは、完了オーダーの詳細情報を含む、アウトバウンド・メッセージです。WebSphere Commerce アウトバウンド・メッセージ・システムは、OrderCreateXML.jsp 構成テンプレートを使用してこのメッセージを生成して、そのデータをアウトバウンド・メッセージ・キューに送信します。このメッセージはそこでバックエンド・システムが受信します。このアプリケーションはメッセージを取り込み、オーダーの指定タスクの完了に必要なバックエンド・ビジネス処理を継続します。

Report\_NC\_PurchaseOrder を使用して、WebSphere Commerce システムから開始するオーダー処理を実行しますが、そのオーダー処理では別のシステムによる追加やバックエンド作業もいくつか必要になります。

Report\_NC\_PurchaseOrder メッセージは、XML メッセージ形式を使用し、Report\_NC\_PO\_10.dtdに従います。

DTD ファイルは、XML エLEMENTのセットで構成されています。そのメッセージはオーダー、ショッパ一、請求先、マーチャントおよび配送情報を含んでいます。以下のデータは、Report\_NC\_PurchaseOrder メッセージに使用される Report\_NC\_PO\_10.dtd の説明です:

以下のテーブルは、XML エLEMENT値の形式およびソースの説明です。データベース列の説明については、それと関連したテーブルへのリンクに従ってください。特に注釈がない限り、すべてのフィールドはオプションです。タイプ INT、BIGINT、および DOUBLE のデータベース・フィールド用の XML メッセージ中のタグ値の長さは、それぞれ CHAR (10)、CHAR(19)、および CHAR(16) とする必要があります。

レベル	XML エLEMENT	コメント	テーブル名	列名
1	ReportPO Header	必須	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
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	必須	N/A	N/A
1.5.A1	currency	Attribute	ORDERS	CURRENCY
1.5.1	TotalNet Price	必須	ORDERS	TOTALPRODUCT
1.5.2	TaxInfo		N/A	N/A
1.5.2.1	Monetary Amount	必須	ORDERS	TOTALTAX
1.5.2.1.A1	currency	Attribute	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	合計 ShippingPrice	必須	ORDERS	TOTALSHIPPING

レベル	XML エレメント	コメント	テーブル名	列名
1.5.4	TotalTaxOnShippingPrice	必須	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	住所	必須	N/A	N/A
1.8.2.1	AddressLine	必須、繰り返し (初回)	ADDRESS	ADDRESS1
1.8.2.1	AddressLine	2 番目に繰り返されるオカレンス	ADDRESS	ADDRESS2
1.8.2.1	AddressLine	3 番目に繰り返されるオカレンス	ADDRESS	ADDRESS3
1.8.2.2	市区町村	必須	ADDRESS	City
1.8.2.3	State	必須	ADDRESS	State
1.8.2.4	Zip	必須	ADDRESS	ZIPCODE
1.8.2.5	Country	必須	ADDRESS	COUNTRY
1.8.3	ContactPerson 名前	必須	N/A	N/A
1.8.3.1	敬称		N/A	N/A
1.8.3.2	FullName		N/A	N/A
1.8.3.3	LastName	必須	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	必須	N/A	N/A
1.8.4.1	Telephone	最初に繰り返されるオカレンス	ADDRESS	PHONE1
1.8.4.1	Telephone	2 番目に繰り返されるオカレンス	ADDRESS	PHONE2
1.8.4.2	Email	最初に繰り返されるオカレンス	ADDRESS	EMAIL1
1.8.4.2	Email	2 番目に繰り返されるオカレンス	ADDRESS	EMAIL2
1.8.4.3	Fax		ADDRESS	FAX1
1.9	MerchantInfo		N/A	N/A
1.9.1	OrgName	必須	STOREENTDS	DISPLAYNAME
1.9.2	OrgID	最初に繰り返されるオカレンス	N/A	N/A
1.9.2.A1	type	属性	ORDERS	STOREENT_ID
1.9.3	住所	必須	N/A	N/A
1.9.3.1	AddressLine	必須、繰り返し (初回)	STADDRESS	ADDRESS1
1.9.3.1	AddressLine	2 番目に繰り返されるオカレンス	STADDRESS	ADDRESS2
1.9.3.1	AddressLine	3 番目に繰り返されるオカレンス	STADDRESS	ADDRESS3
1.9.3.2	市区町村	必須	STADDRESS	City
1.9.3.3	State	必須	STADDRESS	State
1.9.3.4	Zip	必須	STADDRESS	ZIPCODE
1.9.3.5	Country	必須	STADDRESS	COUNTRY
1.9.4	URL		N/A	N/A

レベル	XML エlement	コメント	テーブル名	列名
1.9.5	Telephone		STADDRESS	PHONE1
1.9.6	ContactPerson 名前		N/A	N/A
1.9.6.1	敬称		STADDRESS	PERSONTITLE
1.9.6.2	FullName		N/A	N/A
1.9.6.3	LastName	必須	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	必須	N/A	N/A
1.9.7.1	Telephone	最初に繰り返されるオカレンス	STADDRESS	PHONE1
1.9.7.1	Telephone	2 番目に繰り返されるオカレンス	STADDRESS	PHONE2
1.9.7.2	Email	最初に繰り返されるオカレンス	STADDRESS	EMAIL1
1.9.7.3	Email	2 番目に繰り返されるオカレンス	STADDRESS	EMAIL2
1.9.7.4	Fax		STADDRESS	FAX1
1.10	BuyOrgInfo		N/A	N/A
1.10.1	OrgName	必須	ADDRESS	ORGNAME
1.10.2	OrgID	最初に繰り返されるオカレンス	N/A	N/A
1.10.2.A1	type	属性	N/A	N/A
1.10.3	住所		N/A	N/A
1.10.3.1	AddressLine	必須、繰り返し (初回)	N/A	N/A
1.10.3.1	AddressLine	2 番目に繰り返されるオカレンス	N/A	N/A
1.10.3.1	AddressLine	3 番目に繰り返されるオカレンス	N/A	N/A
1.10.3.2	市区町村	必須	N/A	N/A
1.10.3.3	State	必須	N/A	N/A
1.10.3.4	Zip	必須	N/A	N/A
1.10.3.5	Country	必須	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	最初に繰り返されるオカレンス	N/A	N/A
1.10.5.1	Telephone	2 番目に繰り返されるオカレンス	N/A	N/A
1.10.5.2	Email	最初に繰り返されるオカレンス	N/A	N/A
1.10.5.2	Email	2 番目に繰り返されるオカレンス	N/A	N/A
1.10.5.3	Fax		N/A	N/A
1.11	ShipToInfo		N/A	N/A
1.11.1	ContactPerson 名前	必須	N/A	N/A
1.11.1.1	敬称		N/A	N/A
1.11.1.2	FullName	必須	N/A	N/A
1.11.1.3	LastName	必須	N/A	N/A
1.11.1.4	FirstName		N/A	N/A

レベル	XML エlement	コメント	テーブル名	列名
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	住所	必須	N/A	N/A
1.11.3.1	AddressLine	必須、繰り返し (初回)	N/A	N/A
1.11.3.1	AddressLine	2 番目に繰り返されるオカレンス	N/A	N/A
1.11.3.1	AddressLine	3 番目に繰り返されるオカレンス	N/A	N/A
1.11.3.2	市区町村	必須	N/A	N/A
1.11.3.3	State	必須	N/A	N/A
1.11.3.4	Zip	必須	N/A	N/A
1.11.3.5	Country	必須	N/A	N/A
1.11.4	ContactInfo		N/A	N/A
1.11.4.1	Telephone	最初に繰り返されるオカレンス	N/A	N/A
1.11.4.1	Telephone	2 番目に繰り返されるオカレンス	N/A	N/A
1.11.4.2	Email	最初に繰り返されるオカレンス	N/A	N/A
1.11.4.2	Email	2 番目に繰り返されるオカレンス	N/A	N/A
1.11.4.3	Fax		N/A	N/A
1.11.5	コメント		N/A	N/A
1.12	Requisitioner Info		N/A	N/A
1.12.1	RequisitionerID	最初に繰り返されるオカレンス	ORDERS	MEMBER_ID
1.12.1.A1	type	必須、属性	N/A	N/A
1.12.1	RequisitionerID	2 番目に繰り返されるオカレンス	USERREG	LOGONID
1.12.1.A2	type	必須、属性	N/A	N/A
1.12.2	Requisitioner Group		N/A	N/A
1.12.3	住所		N/A	N/A
1.12.3.1	AddressLine	必須、繰り返し (初回)	ADDRESS	ADDRESS1
1.12.3.1	AddressLine	2 番目に繰り返されるオカレンス	ADDRESS	ADDRESS2
1.12.3.1	AddressLine	3 番目に繰り返されるオカレンス	ADDRESS	ADDRESS3
1.12.3.2	市区町村	必須	ADDRESS	City
1.12.3.3	State	必須	ADDRESS	State
1.12.3.4	Zip	必須	ADDRESS	ZIPCODE
1.12.3.5	Country	必須	ADDRESS	COUNTRY
1.12.4	ContactPerson 名前		N/A	N/A
1.12.4.1	敬称		ADDRESS	PERSONTITLE
1.12.4.2	FullName	必須	N/A	N/A
1.12.4.3	LastName	必須	ADDRESS	LASTNAME
1.12.4.4	FirstName		ADDRESS	FIRSTNAME

レベル	XML エlement	コメント	テーブル名	列名
1.12.4.5	MiddleName		ADDRESS	MiddleName
1.12.4.6	AlternateName		N/A	N/A
1.12.5	ContactInfo	必須	N/A	N/A
1.12.5.1	Telephone	最初に繰り返されるオカレンス	ADDRESS	PHONE1
1.12.5.1	Telephone	2 番目に繰り返されるオカレンス	ADDRESS	PHONE2
1.12.5.2	Email	最初に繰り返されるオカレンス	ADDRESS	EMAIL1
1.12.5.2	Email	2 番目に繰り返されるオカレンス	ADDRESS	EMAIL2
1.12.5.3	Fax		ADDRESS	FAX1
1.13	ShipDate Reference		N/A	N/A
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	Attribute	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	配送 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	Attribute	N/A	N/A
1.16.3	BudgetCode	必須	N/A	N/A

レベル	XML エlement	コメント	テーブル名	列名
1.16.4	説明		N/A	N/A
1.16.5	計算 Code	必須	N/A	N/A
1.17.2	OrderCustomer Field	最初に繰り返されるオカレンス	ORDERS	FIELD1
1.17.2	OrderCustomer Field	最初に繰り返されるオカレンス	ORDERS	FIELD2
1.17.2	OrderCustomer Field	2 番目に繰り返されるオカレンス	ORDERS	FIELD3
1.18	UserData		N/A	N/A
2	ReportPOItem	必須、繰り返し (初回)	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	必須	CATENTRY	PARTNUMBER
2.5	製造業者 名前		N/A	N/A
2.6	製造業者 URL		N/A	N/A
2.7	Manufacturer PartNumber		N/A	N/A
2.8	ItemUnitPrice	必須	ORDERITEMS	PRICE
2.8.A1	currency	Attribute	ORDERITEMS	CURRENCY
2.9	TaxInfo	最初に繰り返されるオカレンス	N/A	N/A
2.9.1	Monetary Amount	最初に繰り返されるオカレンス	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 数量	必須	ORDERITEMS	QUANTITY



レベル	XML エlement	コメント	テーブル名	列名
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 名前	必須	N/A	N/A
2.15.1.1	敬称		N/A	N/A
2.15.1.2	FullName	必須	N/A	N/A
2.15.1.3	LastName	必須	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	オフィス AddressLine	このバージョンでサポートされていません	N/A	N/A
2.15.3	住所		N/A	N/A
2.15.3.1	AddressLine	必須、繰り返し (初回)	ADDRESS	ADDRESS1
2.15.3.1	AddressLine	2 番目に繰り返されるオカレンス	ADDRESS	ADDRESS2
2.15.3.1	AddressLine	3 番目に繰り返されるオカレンス	ADDRESS	ADDRESS3
2.15.3.2	市区町村	必須	ADDRESS	City
2.15.3.3	State	必須	ADDRESS	State
2.15.3.4	Zip	必須	ADDRESS	ZIPCODE
2.15.3.5	Country	必須	ADDRESS	COUNTRY
2.15.4	ContactInfo	必須	N/A	N/A
2.15.4.1	Telephone	最初に繰り返されるオカレンス	ADDRESS	PHONE1
2.15.4.1	Telephone	2 番目に繰り返されるオカレンス	ADDRESS	PHONE2
2.15.4.2	Email	最初に繰り返されるオカレンス	ADDRESS	EMAIL1
2.15.4.2	Email	2 番目に繰り返されるオカレンス	ADDRESS	EMAIL2
2.15.4.3	Fax		ADDRESS	FAX1
2.15.5	コメント		ORDERITEMS	COMMENTS
2.16	配送 CarrierInfo		N/A	N/A
2.16.1	Carrier	必須	SHIPMODE	CARRIER
2.16.2	Method	必須	SHIPMODE	CODE
2.17	ShipStatus		ORDERITEMS	STATUS
2.18	DateTime Reference	必須	N/A	N/A
2.18.1	PlacedDate	必須	ORDERITEMS	LASTCREATE
2.18.2	PlacedTime		ORDERITEMS	LASTCREATE

レベル	XML エlement	コメント	テーブル名	列名
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	最初に繰り返されるオカレンス	N/A	N/A
2.20.1	Percentage		N/A	N/A
2.20.2	Monetary Amount		N/A	N/A
2.20.2.A1	currency	Attribute	N/A	N/A
2.20.3	BudgetCode	必須	N/A	N/A
2.20.4	説明		N/A	N/A
2.20.5	計算 Code	必須	N/A	N/A
2.21	Service Allowance Charge	最初に繰り返されるオカレンス	N/A	N/A
2.21.1	Allowance ChargeCode	必須	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	Attribute	N/A	N/A
2.21.3	説明		N/A	N/A
2.21.4	計算 Code	必須	N/A	N/A
2.22	ItemShipping Schedule		N/A	N/A
2.22.1	数量	必須	N/A	N/A
2.22.2	ShipDate Reference	必須	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	最初に繰り返されるオカレンス	ORDERITEMS	FIELD1
2.23	ItemCustomer Field	2 番目に繰り返されるオカレンス	ORDERITEMS	FIELD2

レベル	XML エlement	コメント	テーブル名	列名
2.24	UserData		N/A	N/A

## CUSTOMER\_NEW\_HDR010\_DATA

新規顧客へのメッセージには、CUSTOMER\_NEW\_HDR010\_DATA セグメントが組み込まれています。このデータ・セグメントは、新規ショッパーへのショッパー情報（新規顧客へのメッセージ）で構成されています。

以下のテーブルは、CUSTOMER\_NEW\_HDR010\_DATA のフィールドの形式およびソースの説明です。フィールド長については、以下のテーブルを使用してください。データベース列の説明については、それと関連したテーブルへのリンクに従ってください。

フィールド名	フィールド・タイプ	テーブル名	列名	注
NC_HDR010 バージョン Number	CHAR (15)	N/A	N/A	固定値 01
NC_LoginID	CHAR (31)	USERREG	LOGONID	登録ショッパーの固有のログオン ID
NC_Password	CHAR (12)	USERREG	LOGONPASSWORD	登録ショッパーの暗号化されたログオン・パスワード
NC_Verify Password	CHAR (12)	N/A	N/A	登録ショッパーの暗号化されたログオン・パスワードの検査です。メッセージを処理するには、このフィールドの値が NC_Password のデータと同じでなければなりません。
NC_Merchant 参照番号	CHAR (10)	N/A	N/A	N/A
NC_Method 連絡方法	CHAR (15)	USERPROF	PREFERREDCOMM	ショッパーの希望する連絡方法: E1 - 電子メールまたは URL アドレス 1 E2 - 電子メールまたは URL アドレス 2 P1 - 電話番号 (昼間) P2 - 電話番号 (夜間)

フィールド名	フィールド・タイプ	テーブル名	列名	注
NC_Challenge 質問	CHAR (250)	USERREG	CHALLENGEQUESTION	ショッパーが本人であることを、言葉で確認するためのユーザー確認のための質問
NC_Challenge 応答	CHAR (250)	USERREG	CHALLENGEANSWER	ユーザー確認のための質問への応答
NC_Shopper フィールド 1	CHAR (254)	USERS	FIELD1	マーチャント・カスタマイズのために予約済み
NC_Shopper フィールド 2	CHAR (254)	USERS	FIELD2	マーチャント・カスタマイズのために予約済み
NC_Title	CHAR (5)	ADDRESS	PERSONTITLE	ショッパーの敬称: Dr Mr Mrs Ms N - 指定なし (デフォルト)
NC_Last 名前	CHAR (30)	ADDRESS	LASTNAME	ショッパーの姓
NC_First 名前	CHAR (30)	ADDRESS	FIRSTNAME	ショッパーの名
NC_Middle 名前	CHAR (30)	ADDRESS	MiddleName	ショッパーのミドル・ネーム
NC_Rep 会社名	CHAR (80)	ADDRESS	ORGNAME	ショッパーが所属する会社
NC_Phone1	CHAR (30)	ADDRESS	PHONE1	ショッパーの電話番号 (昼間)
NC_Phone2	CHAR (30)	ADDRESS	PHONE2	ショッパーの電話番号 (夜間)
NC_Fax	CHAR (30)	ADDRESS	FAX1	ショッパーのファクシミリ電話番号
NC_Addr1	CHAR (50)	ADDRESS	ADDRESS1	ショッパーのアドレス行 1
NC_Addr2	CHAR (50)	ADDRESS	ADDRESS2	ショッパーのアドレス行 2
NC_Addr3	CHAR (50)	ADDRESS	ADDRESS3	ショッパーのアドレス行 3
NC_City	CHAR (30)	ADDRESS	City	ショッパーの市区町村名
NC_State	CHAR (20)	ADDRESS	State	ショッパーの都道府県名
NC_Country	CHAR (30)	ADDRESS	COUNTRY	ショッパーの国/地域名

フィールド名	フィールド・タイプ	テーブル名	列名	注
NC_ZipCode	CHAR (20)	ADDRESS	ZIPCODE	ショッパーの郵便番号
NC_Email1	CHAR (254)	ADDRESS	EMAIL1	ショッパーの E メール 1 または URL アドレス
NC_Email2	CHAR (254)	ADDRESS	EMAIL2	ショッパーの E メール 2 または URL アドレス
NC_Day 電話のタイプ	CHAR (3)	ADDRESS	PHONE1TYPE	昼間の連絡先のタイプ、たとえば聴覚障害の人の場合はテレタイプライター、標準の電話の場合は PHN。
NC_Day 電話番号の掲載	CHAR (1)	ADDRESS	PUBLISHPHONE1	1 - 昼間の電話番号を名簿に掲載する 0 - 昼間の電話番号を名簿に掲載しない
NC_Evening 電話のタイプ	CHAR (3)	ADDRESS	PHONE2TYPE	夜間の連絡先のタイプ、たとえば聴覚障害の人の場合はテレタイプライター、標準の電話の場合は PHN。
NC_Evening 電話番号の掲載	CHAR (1)	ADDRESS	PUBLISHPHONE2	1 - 夜間電話番号を名簿に掲載する 0 - 夜間の電話番号を名簿に掲載しない
NC_BestTime 連絡時間帯	CHAR (1)	ADDRESS	BESTCALLINGTIME	連絡可能な時間帯 D - 昼間 E - 夜間
NC_Package パッケージの同梱	CHAR (1)	ADDRESS	PACKAGESUPPRESSION	オーダーの配送に、パッケージを同梱するかどうかのショッパーの指示 1 - 組み込む 0 - 組み込まない
NC_Address オプションのフィールド 1	CHAR (3)	ADDRESS	FIELD1	マーチャント・カスタマイズのために予約済み
NC_Address オプションのフィールド 2	CHAR (1)	ADDRESS	FIELD2	マーチャント・カスタマイズのために予約済み

フィールド名	フィールド・タイプ	テーブル名	列名	注
NC_Shopper Gender	CHAR (1)	USERDEMO	GENDER	ショッパーの性別: F - 女性 M - 男性 N - 指定なし (デフォルト)
NC_Shopper 年齢	CHAR (10)	USERDEMO	AGE	ショッパーの年齢: 0 - 指定なし (デフォルト) 1 - 10 歳未満 2 - 10 歳以上 20 歳未満 3 - 20 歳以上 30 歳未満 4 - 30 歳以上 40 歳未満 5 - 40 歳以上 50 歳未満 6 - 50 歳以上 60 歳未満 7 - 60 歳以上
NC_Shopper 収入	CHAR (10)	USERDEMO	INCOME	ショッパーの収入: 0 - 指定なし (デフォルト) 1 - 300 万円未満 2 - 300 万円以上 600 万円未満 3 - 600 万円以上 900 万円未満 4 - 900 万円以上
NC_Marital 状況	CHAR (1)	USERDEMO	MARITALSTATUS	ショッパーの既婚 / 独身: S - 独身 M - 既婚 C - 内縁 P - 別居 D - 離婚 W - 未亡人 0 - その他 N - 指定なし
NC_Number 子供の数	CHAR (10)	USERDEMO	CHILDREN	子供の数。 デフォルトは 0
NC_Number 世帯	CHAR (10)	USERDEMO	HOUSEHOLD	ショッパーの世帯の人数。 デフォルトは 1
NC_Shopper 会社名	CHAR (30)	USERDEMO	COMPANYNAME	ショッパーの勤務先
NC_Shopper 趣味	CHAR (254)	USERDEMO	HOBBIES	ショッパーの趣味

フィールド名	フィールド・タイプ	テーブル名	列名	注
NC_Previous オーダー・フラグ	CHAR (1)	USERDEMO	ORDERBEFORE	ショッパーが以前に オーダーの発行をし たかどうかの標識
NC_Demog フィールド 1	CHAR (1)	USERDEMO	FIELD1	マーチャント・カス タマイズのために予 約済み
NC_Demog フィールド 2	CHAR (1)	USERDEMO	FIELD2	マーチャント・カス タマイズのために予 約済み
NC_Demog フィールド 3	CHAR (1)	USERDEMO	FIELD3	マーチャント・カス タマイズのために予 約済み
NC_Demog フィールド 4	CHAR (1)	USERDEMO	FIELD4	マーチャント・カス タマイズのために予 約済み
NC_Demog フィールド 5	CHAR (254)	USERDEMO	FIELD5	マーチャント・カス タマイズのために予 約済み
NC_Demog フィールド 6	CHAR (10)	USERDEMO	FIELD6	マーチャント・カス タマイズのために予 約済み

## CUSTOMER\_UPDATE\_HDR010\_DATA

更新顧客へのメッセージには、CUSTOMER\_UPDATE\_HDR010\_DATA セグメントが組み込まれていま  
す。このデータ・セグメントは、更新ショッパーへのショッパー情報 (更新顧客へのメッセージ) で構成さ  
れています。

以下のテーブルは、CUSTOMER\_UPDATE\_HDR010\_DATA のフィールドの形式およびソースの説明です。  
フィールド長については、以下のテーブルを使用してください。データベース列の説明については、それと  
関連したテーブルへのリンクに従ってください。

フィールド名	フィールド・タイプ	テーブル名	列名	注
NC_HDR010 バージョン Number	CHAR (15)	N/A	N/A	固定値 01
NC_LoginID	CHAR (31)	USERREG	LOGONID	登録ショッパーの固 有のログオン ID
NC_Password	CHAR (12)	USERREG	LOGONPASSWORD	登録ショッパーの暗 号化されたログオ ン・パスワード

フィールド名	フィールド・タイプ	テーブル名	列名	注
NC_Verify Password	CHAR (12)	N/A	N/A	登録ショッパーの暗号化されたログイン・パスワードの検査です。メッセージを処理するには、このフィールドの値が NC_Password のデータと同じでなければなりません。
NC_Merchant 参照番号	CHAR (10)	N/A	N/A	N/A
NC_Method 連絡方法	CHAR (15)	USERPROF	PREFERREDCOMM	ショッパーの希望する連絡方法: E1 - 電子メールまたは URL アドレス 1 E2 - 電子メールまたは URL アドレス 2 P1 - 電話番号 (昼間) P2 - 電話番号 (夜間)
NC_Challenge 質問	CHAR (250)	USERREG	CHALLENGEQUESTION	ショッパーが本人であることを、言葉で確認するためのユーザー確認のための質問
NC_Challenge 応答	CHAR (250)	USERREG	CHALLENGEANSWER	ユーザー確認のための質問への応答
NC_Shopper フィールド 1	CHAR (254)	USERS	FIELD1	マーチャント・カスタマイズのために予約済み
NC_Shopper フィールド 2	CHAR (254)	USERS	FIELD2	マーチャント・カスタマイズのために予約済み
NC_Title	CHAR (5)	ADDRESS	PERSONTITLE	ショッパーの敬称: Dr Mr Mrs Ms N - 指定なし (デフォルト)
NC_Last 名前	CHAR (30)	ADDRESS	LASTNAME	ショッパーの姓
NC_First 名前	CHAR (30)	ADDRESS	FIRSTNAME	ショッパーの名



フィールド名	フィールド・タイプ	テーブル名	列名	注
NC_Middle 名前	CHAR (30)	ADDRESS	MiddleName	ショッパーのミドル・ネーム
NC_Rep 会社名	CHAR (80)	ADDRESS	ORGNAME	ショッパーが所属する会社
NC_Phone1	CHAR (30)	ADDRESS	PHONE1	ショッパーの電話番号 (昼間)
NC_Phone2	CHAR (30)	ADDRESS	PHONE2	ショッパーの電話番号 (夜間)
NC_Fax	CHAR (30)	ADDRESS	FAX1	ショッパーのファクシミリ電話番号
NC_Addr1	CHAR (50)	ADDRESS	ADDRESS1	ショッパーのアドレス行 1
NC_Addr2	CHAR (50)	ADDRESS	ADDRESS2	ショッパーのアドレス行 2
NC_Addr3	CHAR (50)	ADDRESS	ADDRESS3	ショッパーのアドレス行 3
NC_City	CHAR (30)	ADDRESS	City	ショッパーの市区町村名
NC_State	CHAR (20)	ADDRESS	State	ショッパーの都道府県名
NC_Country	CHAR (30)	ADDRESS	COUNTRY	ショッパーの国/地域名
NC_ZipCode	CHAR (20)	ADDRESS	ZIPCODE	ショッパーの郵便番号
NC_Email1	CHAR (254)	ADDRESS	EMAIL1	ショッパーの E メール 1 または URL アドレス
NC_Email2	CHAR (254)	ADDRESS	EMAIL2	ショッパーの E メール 2 または URL アドレス
NC_Day 電話のタイプ	CHAR (3)	ADDRESS	PHONE1TYPE	昼間の連絡先のタイプ、たとえば聴覚障害の人の場合はテレタイプライター、標準の電話の場合は PHN。
NC_Day 電話番号の掲載	CHAR (1)	ADDRESS	PUBLISHPHONE1	1 - 昼間の電話番号を名簿に掲載する 0 - 昼間の電話番号を名簿に掲載しない

フィールド名	フィールド・タイプ	テーブル名	列名	注
NC_Evening 電話のタイプ	CHAR (3)	ADDRESS	PHONE2TYPE	夜間の連絡先のタイプ、たとえば聴覚障害の人の場合はテレタイプライター、標準の電話の場合はPHN。
NC_Evening 電話番号の掲載	CHAR (1)	ADDRESS	PUBLISHPHONE2	1 - 夜間電話番号を名簿に掲載する 0 - 夜間の電話番号を名簿に掲載しない
NC_BestTime 連絡時間帯	CHAR (1)	ADDRESS	BESTCALLINGTIME	連絡可能な時間帯 D - 昼間 E - 夜間
NC_Package パッケージの同梱	CHAR (1)	ADDRESS	PACKAGESUPPRESSION	オーダーの配送に、パッケージを同梱するかどうかのショッパーの指示 1 - 組み込む 0 - 組み込まない
NC_Address オプションのフィールド 1	CHAR (3)	ADDRESS	FIELD1	マーチャント・カスタマイズのために予約済み
NC_Address オプションのフィールド 2	CHAR (1)	ADDRESS	FIELD2	マーチャント・カスタマイズのために予約済み
NC_Shopper Gender	CHAR (1)	USERDEMO	GENDER	ショッパーの性別: F - 女性 M - 男性 N - 指定なし (デフォルト)
NC_Shopper 年齢	CHAR (10)	USERDEMO	AGE	ショッパーの年齢: 0 - 指定なし (デフォルト) 1 - 10 歳未満 2 - 10 歳以上 20 歳未満 3 - 20 歳以上 30 歳未満 4 - 30 歳以上 40 歳未満 5 - 40 歳以上 50 歳未満 6 - 50 歳以上 60 歳未満 7 - 60 歳以上

フィールド名	フィールド・タイプ	テーブル名	列名	注
NC_Shopper 収入	CHAR (10)	USERDEMO	INCOME	ショッパーの収入: 0 - 指定なし (デフォルト) 1 - 300 万円未満 2 - 300 万円以上 600 万円未満 3 - 600 万円以上 900 万円未満 4 - 900 万円以上
NC_Marital 状況	CHAR (1)	USERDEMO	MARITALSTATUS	ショッパーの既婚 / 独身: S - 独身 M - 既婚 C - 内縁 P - 別居 D - 離婚 W - 未亡人 0 - その他 N - 指定なし
NC_Number 子供の数	CHAR (10)	USERDEMO	CHILDREN	子供の数。 デフォルトは 0
NC_Number 世帯	CHAR (10)	USERDEMO	HOUSEHOLD	ショッパーの世帯の人数。 デフォルトは 1
NC_Shopper 会社名	CHAR (30)	USERDEMO	COMPANYNAME	ショッパーの勤務先
NC_Shopper 趣味	CHAR (254)	USERDEMO	HOBBIES	ショッパーの趣味
NC_Previous オーダー・フラグ	CHAR (1)	USERDEMO	ORDERBEFORE	ショッパーが以前にオーダーの発行をしたかどうかの標識
NC_Demog フィールド 1	CHAR (1)	USERDEMO	FIELD1	マーチャント・カスタマイズのために予約済み
NC_Demog フィールド 2	CHAR (1)	USERDEMO	FIELD2	マーチャント・カスタマイズのために予約済み
NC_Demog フィールド 3	CHAR (1)	USERDEMO	FIELD3	マーチャント・カスタマイズのために予約済み
NC_Demog フィールド 4	CHAR (1)	USERDEMO	FIELD4	マーチャント・カスタマイズのために予約済み
NC_Demog フィールド 5	CHAR (254)	USERDEMO	FIELD5	マーチャント・カスタマイズのために予約済み

フィールド名	フィールド・タイプ	テーブル名	列名	注
NC_Demog フィールド 6	CHAR (10)	USERDEMO	FIELD6	マーチャント・カスタマイズのために予約済み

## ORDER\_CREATE\_HDR010\_DATA

オーダー作成のメッセージには、ORDER\_CREATE\_HDR010\_DATA セグメントが組み込まれています。このデータ・セグメントは、オーダー作成メッセージのためのオーダー仕様で構成されます。

以下のテーブルは、ORDER\_CREATE\_HDR010\_DATA のフィールドの形式およびソースの説明です。データベース列の説明については、それと関連したテーブルへのリンクに従ってください。

フィールド名	フィールド・タイプ	テーブル名	列名	説明
NC_HDR010 バージョン Number	CHAR (15)	N/A	N/A	固定値 01
NC_Order 参照番号	CHAR (10)	ORDERS	ORDERS_ID	固有なオーダー参照番号 (内部生成)。これは 1 次キーです。
NC_Order Date	CHAR (8)	ORDERS	TIMEPLACED	オーダーが発行される日付 (YYYYMMDD)
NC_Order Time	CHAR (6)	ORDERS	TIMEPLACED	オーダーが発行される時刻 (HHMMSS)
NC_Currency タイプ	CHAR (10)	ORDERS	CURRENCY	価格を表す通貨。通貨の形式は、ISO 4217 規格に準拠する必要があります。
NC_Total Price	CHAR (16)	ORDERS	TOTALPRODUCT	オーダーの商品価格合計
NC_Total 税額	CHAR (16)	ORDERS	TOTALTAX	オーダーの消費税合計
NC_Total ShippingPrice	CHAR (16)	ORDERS	TOTALSHIPPING	オーダーの配送料合計
NC_Total 配送料金 Price	CHAR (16)	ORDERS	TOTALTAXSHIPPING	オーダーの配送料金税合計
NC_Shopper 参照番号	CHAR (10)	ORDERS	MEMBER_ID	ショッパー参照番号
NC_Merchant 参照番号	CHAR (10)	ORDERS	STOREENT_ID	マーチャント参照番号
NC_Merchant オーダー番号	CHAR (30)	ORDERS	ORMORDER	マーチャントによって生成された固有のオーダー参照番号

フィールド名	フィールド・タイプ	テーブル名	列名	説明
NC_BillTo 参照番号	CHAR (10)	ORDERS	ADDRESS_ID	請求先住所の参照番号
NC_Order カスタム・フィールド 1	CHAR (10)	ORDERS	FIELD1	マーチャント・カスタマイズのために予約済み
NC_Order カスタム・フィールド 2	CHAR (16)	ORDERS	FIELD2	マーチャント・カスタマイズのために予約済み
NC_Order カスタム・フィールド 3	CHAR (254)	ORDERS	FIELD3	マーチャント・カスタマイズのために予約済み

## ORDER\_STATUS\_UPDATE\_HDR010\_DATA

オーダー状況更新のメッセージには、ORDER\_STATUS\_UPDATE\_HDR010\_DATA セグメントが組み込まれています。このデータ・セグメントは、オーダー状況更新メッセージのためのオーダー仕様で構成されません。

以下のテーブルは、ORDER\_STATUS\_UPDATE\_HDR010\_DATA のフィールドの形式およびソースの説明です。フィールド長については、以下のテーブルを使用してください。データベース列の説明については、それと関連したテーブルへのリンクに従ってください。

フィールド名	フィールド・タイプ	テーブル名	列名	注
NC_HDR010 バージョン Number	CHAR (15)	N/A	N/A	固定値 01
NC_Order 参照番号	CHAR (10)	ORDSTAT	ORDERS_ID	WebSphere Commerce のオーダー参照番号。
NC_Customer オーダー番号	N/A	N/A	N/A	N/A
NC_Currency タイプ	CHAR (10)	ORDSTAT	OSPCUR	価格を表す通貨。通貨の形式は、ISO 4217 規格に準拠する必要があります。
NC_TotalPrice	CHAR (16)	ORDSTAT	OSPRTOT	オーダーの商品価格合計
NC_Total 税額	CHAR (16)	ORDSTAT	OSTXTOT	オーダーの消費税合計
NC_Total ShippingPrice	CHAR (16)	ORDSTAT	OSSHTOT	オーダーの配送料合計
NC_Total 配送料金 Price	CHAR (16)	ORDSTAT	OSSHTXTOT	オーダーの配送料金税合計
NC_Shopper LoginID	N/A	N/A	N/A	N/A

フィールド名	フィールド・タイプ	テーブル名	列名	注
NC_Merchant オーダー番号	CHAR (30)	ORDSTAT	OSMORDER	マーチャントによって生成されたオーダー参照番号
NC_Order 状況	CHAR (32)	ORDSTAT	OSSTATUS	オーダー状況: P - 保留状態 C - 経過状態 X - キャンセル I - 在庫更新保留 (SHIPTO はもう保留状態ではない) M - 与信待ち (SHIPTO は在庫更新を完了)
NC_Schedule ShipDate	CHAR (8)	ORDSTAT	OSSTIME	スケジュールされた配達日 (YYYYMMDD)
NC_Actual ShipDate	CHAR (8)	ORDSTAT	OSASTIME	実配達日 (YYYYMMDD)
NC_PlaceDate	CHAR (8)	ORDSTAT	OSPLTIME	オーダーが発行される日付 (YYYYMMDD)

## PRODUCT\_PRICE\_UPDATE\_HDR010\_DATA

商品価格更新のメッセージには、PRODUCT\_PRICE\_UPDATE\_HDR010\_DATA セグメントが組み込まれています。このデータ・セグメントは商品価格の更新メッセージのための、商品またはアイテム価格で構成されています。

商品価格の更新メッセージは、商品価格の更新 バージョン 01 および商品価格の更新 バージョン 02 のアプリケーション・データ 2 セットをサポートしています。バージョン 02 は、バージョン 01 のデータのスーパーセットです。特に、バージョン 01 の PRODUCT\_PRICE\_UPDATE\_HDR010\_DATA セグメントは、NC\_HDR010VersionNumber フィールドが 01 の値であるのに対して、バージョン 02 の NC\_HDR010VersionNumber は 02 の値です。さらに、バージョン 02には、NC\_ProductNumber フィールドが追加されています。

バージョン 02 の PRODUCT\_PRICE\_UPDATE\_HDR010\_DATA のフィールドの形式とソースを以下のテーブルで説明します。フィールド長については、以下のテーブルを使用してください。データベース列の説明については、それと関連したテーブルへのリンクに従ってください。

フィールド名	フィールド・タイプ	テーブル名	列名	説明
NC_HDR010 バージョン番号	CHAR (15)	N/A	N/A	(固定値 02)
NC_Product 参照番号	CHAR (10)	OFFER	CATENTRY_ID	販売用に提供される CatalogEntry。
NC_Product Price	CHAR (16)	OFFERPRICE	PRICE	商品またはアイテムの価格
NC_Currency タイプ	CHAR (10)	OFFERPRICE	CURRENCY	価格を表す通貨。通貨の形式は、ISO 4217 規格に準拠する必要があります。

フィールド名	フィールド・タイプ	テーブル名	列名	説明
NC_Shopper Group	CHAR (10)	OFFER	TRADEPOSCN_ID	オファーの一部である TradingPositionContainer。
NC_Precedence	CHAR (10)	OFFER	PRECEDENCE	この価格の優先順位。
NC_Merchant 参照番号	CHAR (10)	CATENTRY	MEMBER_ID	カタログ・エントリーの所有者を識別する参照番号。
NC_ProductNumber	CHAR (64)	CATENTRY	PARTNUMBER	カタログ・エントリーの所有者を識別する参照番号
NC_Product 価格参照番号	CHAR (10)	OFFER	OFFER_ID	商品またはアイテム価格の参照番号
NC_Start Timestamp	CHAR (26)	OFFER	STARTDATE	商品またはアイテムの価格が有効になる日付 (YYYYMMDD hh:mm:ss.ssssss の形式)。デフォルトは現在の日時
NC_End Timestamp	CHAR (26)	OFFER	ENDDATE	商品またはアイテムの価格の有効期限が切れる日付 (YYYY-MM-DD hh:mm:ss.ssssss の形式)。デフォルトは 9999-12-31 23:59:59.999999
NC_Price カスタム・フィールド 1	N/A	N/A	N/A	マーチャント・カスタマイズのために予約済み
NC_Price カスタム・フィールド 2	N/A	N/A	N/A	マーチャント・カスタマイズのために予約済み

#### バージョン 02 の動作:

- 通貨タイプ ( OFFERPRICE テーブル注の CURRENCY を指す) は必須であり、 ISO 4217 形式で指定されていなければなりません。
- 商品価格参照番号 ( OFFER テーブル中のOFFER\_ID を指す) が指定されていると、この値が通貨タイプと一緒に OFFERPRICE テーブル中の行を更新するためにキーとして使用されることとなります。
- 商品価格参照番号 (OFFER\_ID) が指定されていないと、商品参照番号 ( CATENTRY テーブル中の CATENTRY\_ID を指す) と、優先順位 (OFFER テーブル中の PRECEDENCE を指す) またはトレード位置コンテナ (OFFER テーブル中の TRADEPOSCN\_ID を指す) の組み合わせが商品価格参照番号 (OFFER\_ID) を得るために使用されることとなります。 この値が通貨タイプと一緒に OFFERPRICE テーブル中の行を更新するためにキーとして使用されることとなります。
- 商品参照番号 (CATENTRY\_ID) が指定されていないと、商品番号 (CATENTRY テーブル中の PARTNUMBER を指す) とマーチャント参照番号 (CATENTRY テーブル中の MEMBER\_ID を指す) の組み合わせが、商品参照番号 (CATENTRY\_ID) を得るために使用されることとなります。 この値が、優先順位 (PRECEDENCE) またはトレード位置コンテナ (TRADEPOSCN\_ID) のいずれかと一緒に、商品価格参照番号 (OFFER\_ID) を得るために使用されることとなります。 商品価格参照番号が、通貨タイプと一緒に、OFFERPRICE テーブル中の行を更新するためのキーとして使用されることとなります。

- 商品価格参照番号 (OFFER\_ID) がデータベース中に既存の商品価格参照番号と一致しているが、通貨タイプがその商品価格参照番号をもつレコードのいずれかの通貨タイプと一致していない場合には、新規レコードが OFFERPRICE テーブル中に作成されることとなります。これにより、価格を異なる通貨で同じオファーに指定できます。
- 優先順位 (PRECEDENCE) が指定されていないと、ProductOfferPriceUpdate コマンドが、優先順位なしで指定されている値と一致している以前のすべてのレコードを探し出します。これらの値のうち最大値に +1 されます。以前のレコードが存在していないと、優先順位値は 1 に設定されます。新規行が新規優先順位値でテーブル OFFERPRICE に挿入されます。優先順位値は  $10^{16}$  より小でなければなりません。最大値に達してしまうと、新規更新は拒否されることとなります。

以下のテーブルは、バージョン 01 の PRODUCT\_PRICE\_UPDATE\_HDR010\_DATA に対するフィールドの形式およびソースの説明です:

フィールド名	フィールド・タイプ	テーブル名	列名	説明
NC_HDR010 バージョン番号	CHAR (15)	N/A	N/A	固定値 01
NC_Product 参照番号	CHAR (10)	OFFER	CATENTRY_ID	販売用に提供される CatalogEntry。
NC_Product Price	CHAR (16)	OFFERPRICE	PRICE	商品またはアイテムの価格
NC_Currency タイプ	CHAR (10)	OFFERPRICE	CURRENCY	価格を表す通貨。通貨の形式は、ISO 4217 規格に準拠する必要があります。
NC_Shopper Group	CHAR (10)	OFFER	TRADEPOSCN_ID	オファーが一部である TradingPositionContainer。
NC_Precedence	CHAR (10)	OFFER	PRECEDENCE	この価格の優先順位。
NC_Merchant 参照番号	CHAR (10)	CATENTRY	MEMBER_ID	カタログ・エントリーの所有者を識別する参照番号。
NC_Product 価格参照番号	CHAR (10)	OFFER	OFFER_ID	商品またはアイテム価格の参照番号
NC_Start Timestamp	CHAR (26)	OFFER	STARTDATE	商品またはアイテムの価格の有効期限が有効になる日付 (YYYY-MM-DD hh:mm:ss:sssss の形式)。 デフォルトは現在の日時
NC_End Timestamp	CHAR (26)	OFFER	ENDDATE	商品またはアイテムの価格の有効期限が切れる日時 (YYYY-MM-DD hh:mm:ss:sssss の形式)。 デフォルトは 9999-12-31 23:59:59.999999
NC_Price カスタム・フィールド 1	CHAR (30)	N/A	N/A	マーチャント・カスタマイズのために予約済み
NC_Price カスタム・フィールド 2	CHAR (1)	N/A	N/A	マーチャント・カスタマイズのために予約済み



### バージョン 01 の動作:

- 通貨タイプ (OFFERPRICE テーブル中の CURRENCY を指す) は必須であり、ISO 4217 形式で指定されていなければなりません。
- 商品価格参照番号 (OFFER テーブル中の OFFER\_ID を指す) が指定されていると、この値が通貨タイプと一緒に OFFERPRICE テーブル中の行を更新するためにキーとして使用されることとなります。
- 商品価格参照番号 (OFFER\_ID) が指定されていないと、商品参照番号 (CATENTRY テーブル中の CATENTRY\_ID を指す) と、優先順位 (OFFER テーブル中の PRECEDENCE を指す) またはトレード位置コンテナ (OFFER テーブル中の TRADEPOSCN\_ID を指す) の組み合わせが商品価格参照番号 (OFFER\_ID) を得るために使用されることとなります。この値が通貨タイプと一緒に OFFERPRICE テーブル中の行を更新するためにキーとして使用されることとなります。
- 商品価格参照番号 (OFFER\_ID) がデータベース中に既存の商品価格参照番号と一致しているが、通貨タイプがその商品価格参照番号をもつレコードのいずれかの通貨タイプと一致していない場合には、新規レコードが OFFERPRICE テーブル中に作成されることとなります。これにより、価格を異なる通貨で同じオファーに指定できます。
- 優先順位 (PRECEDENCE) が指定されていないと、ProductOfferPriceUpdate コマンドが、優先順位なしで指定されている値と一致している以前のすべてのレコードを探し出します。これらの値のうちの最大値に +1 されます。以前のレコードが存在していないと、優先順位値は 1 に設定されます。新規行が新規優先順位値でテーブル OFFERPRICE に挿入されます。優先順位値は  $10^{16}$  より小でなければなりません。最大値に達してしまうと、新規更新は拒否されることとなります。

---

## PRODUCT\_QUANTITY\_UPDATE\_HDR010\_DATA

商品価格更新のメッセージには、PRODUCT\_QUANTITY\_UPDATE\_HDR010\_DATA セグメントが組み込まれています。このデータ・セグメントは、商品数量の更新メッセージの商品、またはアイテム在庫情報が組み込まれています。

以下のテーブルは、PRODUCT\_QUANTITY\_UPDATE\_HDR010\_DATA のフィールドの形式およびソースの説明です。フィールド長については、以下のテーブルを使用してください。データベース列の説明については、それと関連したテーブルへのリンクに従ってください。

フィールド名	フィールド・タイプ	テーブル名	列名	注
NC_HDR010 バージョン番号	CHAR (15)	N/A	N/A	固定値 01
NC_Product 参照番号	CHAR (10)	INVENTORY	CATENTRY_ID	商品数量を設定する CatalogEntry。
NC_Product Quantity	CHAR (10)	INVENTORY	QUANTITY	商品数量。
NC_Product Number	CHAR (64)	CATENTRY	PARTNUMBER	カタログ・エントリー の部品番号。
NC_Merchant 参照番号	CHAR (10)	INVENTORY	STORE_ID	商品数量を設定するス トア。

### 動作:

- マーチャント参照番号 (INVENTORY テーブル中の STORE\_ID を指す) は必須です。
- WebSphere Commerce 商品参照番号 (INVENTORY テーブル中の CATENTRY\_ID を指す) およびマーチャント参照番号 (STORE\_ID) が INVENTORY テーブル中の行を更新するために使用されます。

- 商品参照番号 (CATENTRY\_ID) が存在していないと、マーチャント参照番号 (STORE\_ID) がメンバー番号 (CATENTRY 所有者中の MEMBER\_ID を指す) を得るために使用されます。メンバー番号は、カタログ・エントリーの所有者と同じものでなければなりません。メンバー番号 (MEMBER\_ID) は、商品番号 (CATENTRY テーブル中の PARTNUMBER を指す) と一緒に、商品参照番号 (CATENTRY\_ID) を得るために使用されます。商品参照番号は、マーチャント参照番号 (STORE\_ID) と一緒に、INVENTORY テーブル中の行を更新するために使用されます。
- INVENTORY テーブル中の行が存在していないと、エラーが起こります。

## ORDER\_CREATE\_HDR020\_DATA

オーダー作成のメッセージには、ORDER\_CREATE\_HDR020\_DATA セグメントが組み込まれています。このデータ・セグメントは、オーダー作成メッセージのためのショッパー仕様で構成されます。

以下のテーブルは、ORDER\_CREATE\_HDR020\_DATA のフィールドの形式およびソースの説明です。フィールド長については、以下のテーブルを使用してください。データベース列の説明については、それと関連したテーブルへのリンクに従ってください。

フィールド名	フィールド・タイプ	テーブル名	列名	注
NC_HDR020 バージョン Number	CHAR (15)	N/A	N/A	固定値 01
NC_Shopper LoginID	CHAR (31)	USERREG	LOGONID	登録ショッパーの固有のログオン ID
NC_Purchaser 姓	CHAR (30)	ADDRESS	LASTNAME	購入者の姓
NC_Purchaser ミドル・ネーム	CHAR (30)	ADDRESS	MiddleName	購入者のミドル・ネーム
NC_Purchaser 名	CHAR (30)	ADDRESS	FIRSTNAME	購入者の名
NC_Purchaser 住所 1	CHAR (50)	ADDRESS	ADDRESS1	購入者の住所行 1
NC_Purchaser 住所 2	CHAR (50)	ADDRESS	ADDRESS2	購入者の住所行 2
NC_Purchaser 住所 3	CHAR (50)	ADDRESS	ADDRESS3	購入者の住所行 3
NC_Purchaser 市区町村	CHAR (30)	ADDRESS	City	購入者の市区町村
NC_Purchaser 都道府県	CHAR (20)	ADDRESS	State	購入者の都道府県名
NC_Purchaser 国	CHAR (30)	ADDRESS	COUNTRY	購入者の国/地域
NC_Purchaser 郵便番号	CHAR (20)	ADDRESS	ZIPCODE	購入者の郵便番号
NC_Purchaser E メール 1	CHAR (254)	ADDRESS	EMAIL1	購入者の E メールまたは URL アドレス
NC_Purchaser E メール 2	CHAR (254)	ADDRESS	EMAIL2	購入者の E メール 2 または URL アドレス

フィールド名	フィールド・タイプ	テーブル名	列名	注
NC_Purchaser 電話番号 (昼間)	CHAR (30)	ADDRESS	PHONE1	購入者の電話番号 (昼間)
NC_Purchaser 電話番号 (夜間)	CHAR (30)	ADDRESS	PHONE2	購入者の電話番号 (夜間)
NC_Purchaser ファクシミリ番号	CHAR (30)	ADDRESS	FAX1	購入者のファクシミリ電話番号
NC_Purchaser 会社名	CHAR (80)	ADDRESS	ORGNAME	購入者が所属する会社
NC_Purchaser ショッパー グループ名	CHAR (50)	MBRGRP	MBRGRPNAME	購入者が所属するショッパー・グループ

## ORDER\_CREATE\_HDR030\_DATA

オーダー作成のメッセージには、ORDER\_CREATE\_HDR030\_DATA セグメントが組み込まれています。このデータ・セグメントは、オーダー作成メッセージのための請求先詳細情報で構成されます。

以下のテーブルは、ORDER\_CREATE\_HDR030\_DATA のフィールドの形式およびソースの説明です。フィールド長については、以下のテーブルを使用してください。データベース列の説明については、それと関連したテーブルへのリンクに従ってください。

フィールド名	フィールド・タイプ	テーブル名	列名	注
NC_HDR030 バージョン番号	CHAR (15)	N/A	N/A	固定値 01
NC_BillTo LastName	CHAR (30)	ADDRESS	LASTNAME	請求先の 姓
NC_BillTo MiddleName	CHAR (30)	ADDRESS	MiddleName	請求先のミドル・ネーム
NC_BillTo FirstName	CHAR (30)	ADDRESS	FIRSTNAME	請求先の 名
NC_BillTo 住所 1	CHAR (50)	ADDRESS	ADDRESS1	請求先の住所行 1
NC_BillTo 住所 2	CHAR (50)	ADDRESS	ADDRESS2	請求先の住所行 2
NC_BillTo 住所 3	CHAR (50)	ADDRESS	ADDRESS3	請求先の住所行 3
NC_BillTo 市区町村	CHAR (30)	ADDRESS	City	請求先の市区町村名
NC_BillTo State	CHAR (20)	ADDRESS	State	請求先の都道府県名
NC_BillTo 国	CHAR (30)	ADDRESS	COUNTRY	請求先の国/地域
NC_BillTo 郵便番号	CHAR (20)	ADDRESS	ZIPCODE	請求先の郵便番号

フィールド名	フィールド・タイプ	テーブル名	列名	注
NC_BillTo E メール 1	CHAR (254)	ADDRESS	EMAIL1	請求先の E メールまたは URL アドレス 1
NC_Billto E メール 2	CHAR (254)	ADDRESS	EMAIL2	請求先の E メールまたは URL アドレス 2
NC_BillTo 電話番号 (昼間)	CHAR (30)	ADDRESS	PHONE1	請求先の電話番号 (昼間)
NC_BillTo 電話番号 (夜間)	CHAR (30)	ADDRESS	PHONE2	請求先の電話番号 (夜間)
NC_BillTo Fax	CHAR (30)	ADDRESS	FAX1	請求先のファクシミリ電話番号

## ORDER\_CREATE\_HDR040\_DATA

オーダー作成のメッセージには、ORDER\_CREATE\_HDR040\_DATA セグメントが組み込まれています。このデータ・セグメントは、オーダー作成メッセージのためのマーチャント情報で構成されます。

以下のテーブルは、ORDER\_CREATE\_HDR040\_DATA のフィールドの形式およびソースの説明です。フィールド長については、以下のテーブルを使用してください。データベース列の説明については、それと関連したテーブルへのリンクに従ってください。

フィールド名	フィールド・タイプ	テーブル名	列名	注
NC_HDR040 バージョン Number	CHAR (15)	N/A	N/A	固定値 01
NC_Supplier 名前	CHAR (80)	STOREENTDS	DISPLAYNAME	マーチャントの会社名
NC_Supplier 住所 1	CHAR (50)	STADDRESS	ADDRESS1	マーチャントの会社住所行 1
NC_Supplier 住所 2	CHAR (50)	STADDRESS	ADDRESS2	マーチャントの会社住所行 2
NC_Supplier 住所 3	CHAR (50)	STADDRESS	ADDRESS3	マーチャントの会社住所行 3
NC_Supplier 市区町村	CHAR (30)	STADDRESS	City	マーチャントの会社の市区町村名
NC_Supplier State	CHAR (20)	STADDRESS	State	マーチャントの会社の都道府県名
NC_Supplier 国	CHAR (30)	STADDRESS	COUNTRY	マーチャントの会社の国/地域
NC_Supplier 郵便番号	CHAR (20)	STADDRESS	ZIPCODE	マーチャントの会社の郵便番号
NC_Supplier 電話番号 (昼間)	CHAR (30)	STADDRESS	PHONE1	マーチャントの会社の電話番号
NC_Supplier 連絡先 LastName	CHAR (30)	STADDRESS	LASTNAME	マーチャント担当者の姓

フィールド名	フィールド・タイプ	テーブル名	列名	注
NC_Supplier 連絡先 MiddleName	CHAR (30)	STADDRESS	MiddleName	マーチャント担当者の ミドル・ネーム
NC_Supplier 連絡先 FirstName	CHAR (30)	STADDRESS	FIRSTNAME	マーチャント担当者の 名
NC_Supplier ContactTitle	CHAR (30)	STADDRESS	PERSONTITLE	マーチャント担当者の 敬称
NC_Supplier 連絡先 電話番号 (昼間)	CHAR (30)	STADDRESS	PHONE1	マーチャント担当者の 電話番号 (昼間)
NC_Supplier 連絡先 電話番号 (夜間)	CHAR (30)	STADDRESS	PHONE2	マーチャント担当者の 電話番号 (夜間)
NC_Supplier 連絡先 E メール 1	CHAR (254)	STADDRESS	EMAIL1	マーチャント担当者の 1 次 E メールまたは URL アドレス
NC_Supplier 連絡先 E メール 2	CHAR (254)	STADDRESS	EMAIL2	マーチャント担当者の 2 次 E メールまたは URL アドレス

## ORDER\_CREATE\_ITM010\_DATA

オーダー作成のメッセージには、ORDER\_CREATE\_ITM010\_DATA セグメントが組み込まれています。このデータ・セグメントは、オーダー作成メッセージのためのアイテム、または商品の配送仕様で構成されません。

以下のテーブルは、ORDER\_CREATE\_ITM010\_DATA のフィールドの形式およびソースの説明です。フィールド長については、以下のテーブルを使用してください。データベース列の説明については、それと関連したテーブルへのリンクに従ってください。

フィールド名	フィールド・タイプ	テーブル名	列名	注
NC_ITM010 バージョン番号	CHAR (15)	N/A	N/A	固定値 01
NC_Item 行番号	CHAR (3)	N/A	N/A (順次番号として 生成)	N/A
NC_Item 参照番号	CHAR (10)	ORDERITEMS	ORDERITEMS_ID	固有な配送先参照番号 (内部生成)。これは 1 次キーです。
NC_Item 商品参照番号	CHAR (10)	ORDERITEMS	CATENTRY_ID	アイテムまたは商品の 参照番号。これは外 部キーではありません。
NC_Item 商品番号	CHAR (64)	CATENTRY	PARTNUMBER	アイテム SKU または 商品番号

フィールド名	フィールド・タイプ	テーブル名	列名	注
NC_Item 商品 ShortDescription	CHAR (254)	CATENTDESC	SHORTDESCRIPTION	アイテムまたは商品の、名前を含む簡略説明
NC_Item 単価	CHAR (16)	ORDERITEMS	PRICE	アイテムの単価
NC_Item 通貨タイプ	CHAR (10)	ORDERITEMS	CURRENCY	価格を表す通貨。通貨の形式は、ISO 4217 規格に準拠する必要があります。
NC_Item 商品数量	CHAR (10)	ORDERITEMS	QUANTITY	オーダーされた数量
NC_Item 配送先 参照番号	CHAR (10)	ORDERITEMS	ADDRESS_ID	配送先住所の住所参照番号
NC_Item 配送モード 参照番号	CHAR (10)	ORDERITEMS	SHIPMODE_ID	マーチャントの配送モード参照番号
NC_ItemState	CHAR (1)	ORDERITEMS	STATUS	オーダー状況: P - 保留状態 C - 経過状態 X - キャンセル I - 在庫更新保留 (SHIPTO はもう保留状態ではない) M - 与信待ち (SHIPTO は在庫更新を完了)
NC_Item 配送先 LastName	CHAR (30)	ADDRESS	LASTNAME	配送先の顧客の姓
NC_Item 配送先 MiddleName	CHAR (30)	ADDRESS	MiddleName	配送先の顧客のミドル・ネーム
NC_Item 配送先 FirstName	CHAR (30)	ADDRESS	FIRSTNAME	配送先の顧客の名
NC_Item 配送先住所 1	CHAR (50)	ADDRESS	ADDRESS1	配送先の顧客の住所行 1
NC_Item 配送先住所行 2	CHAR (50)	ADDRESS	ADDRESS2	配送先の顧客の住所行 2
NC_Item 配送先住所行 3	CHAR (50)	ADDRESS	ADDRESS3	配送先の顧客の住所行 3
NC_Item 配送先市区町村名	CHAR (30)	ADDRESS	City	配送先の顧客の市区町村
NC_Item 配送先都道府県名	CHAR (20)	ADDRESS	State	配送先の顧客の都道府県名

フィールド名	フィールド・タイプ	テーブル名	列名	注
NC_Item 配送先国名	CHAR (30)	ADDRESS	COUNTRY	配送先の顧客の国/地域
NC_Item 配送先郵便番号	CHAR (20)	ADDRESS	ZIPCODE	配送先の顧客の郵便番号
NC_Item 配送先 E メール 1	CHAR (254)	ADDRESS	EMAIL1	配送先の顧客の E メールまたは URL アドレス 1
NC_Item 配送先 E メール 2	CHAR (254)	ADDRESS	EMAIL2	配送先の顧客の E メールまたは URL アドレス 2
NC_Item 配送先電話番号 (昼間)	CHAR (30)	ADDRESS	PHONE1	配送先の顧客の電話番号 (昼間)
NC_Item 配送先電話番号 (夜間)	CHAR (30)	ADDRESS	PHONE2	配送先の顧客の電話番号 (夜間)
NC_Item 配送先ファクシミリ番号	CHAR (30)	ADDRESS	FAX1	配送先の顧客のファクシミリ電話番号
NC_Item 配送の運送会社	CHAR (30)	SHIPMODE	CARRIER	配送業者 ID (たとえば、Federal Express)
NC_Item 配送方法	CHAR (30)	SHIPMODE	CODE	配送サービス配送モード (たとえば、FedEx Express 夜間運送)
NC_Item 配送先コメント	CHAR (254)	ORDERITEMS	COMMENTS	顧客からのコメント (ギフトに対するあいさつなど)
NC_Item 作成 Timestamp	CHAR (64)	ORDERITEMS	LASTCREATE	配送先エントリーが行われた日時
NC_Item 更新 Timestamp	CHAR (64)	ORDERITEMS	LASTUPDATE	配送先エントリーの最終更新日時
NC_Item カスタム・フィールド 1	CHAR (10)	ORDERITEMS	FIELD1	マーチャント・カスタマイズのために予約済み
NC_Item カスタム・フィールド 2	CHAR (254)	ORDERITEMS	FIELD2	マーチャント・カスタマイズのために予約済み

## ORDER\_STATUS\_UPDATE\_ITM010\_DATA

オーダー状況更新のメッセージには、ORDER\_STATUS\_UPDATE\_ITM010\_DATA セグメントが組み込まれています。このデータ・セグメントは、オーダー状況メッセージのためのアイテム、または商品の配送仕様で構成されます。

オーダー状況の更新メッセージは、オーダー状況の更新 バージョン 01 およびオーダー状況の更新バージョン 02 のアプリケーション・データ 2 セットをサポートしています。 バージョン 02 は、バージョン 01 のデータのスーパーセットです。 つまり、バージョン 01 の ORDER\_STATUS\_UPDATE\_ITM010\_DATA セグメントは、 NC\_HDR010VersionNumber フィールドが 01 の値であり、バージョン 02 の NC\_HDR010VersionNumber は 02 です。 さらに、バージョン 02 は、バージョン 01 にないフィールドをもっています。

以下のテーブルは、バージョン 02 の ORDER\_STATUS\_UPDATE\_ITM010\_DATA のフィールドの形式およびソースの説明です。 フィールド長については、以下のテーブルを使用してください。 データベース列の説明については、それと関連したテーブルへのリンクに従ってください。

フィールド名	フィールド・タイプ	テーブル名	列名	注
NC_ITM010 バージョン番号	CHAR (15)	N/A	N/A (固定値 02)	N/A
NC_Order 参照番号	CHAR (10)	ORDISTAT	ORDERS_ID	WebSphere Commerce のオーダー参照番号。
NC_Merchant オーダー番号	CHAR (30)	ORDISTAT	OSMORDER	マーチャントのオーダー 参照番号
NC_Item 参照番号	CHAR (10)	ORDISTAT	ORDERITMES_ID	WebSphere Commerce のアイテム参照番号。
NC_ItemMerchant 参照番号	CHAR (30)	ORDISTAT	OIMITEM	マーチャントのアイテム 参照番号
NC_Order アイテムの状況	CHAR (32)	ORDISTAT	OISTATUS	オーダーのアイテム状 況
NC_Item Schedule ShipDate	CHAR (8)	ORDISTAT	OISSTIME	スケジュールされた配 送日 (YYYYMMDD)
NC_Item 実配送日	CHAR (8)	ORDISTAT	OIASTIME	実配送日 (YYYYMMDD)
NC_Item 発行日	CHAR (8)	ORDISTAT	OIPLTIME	オーダーが発行される 日付 (YYYYMMDD)
NC_Item 数量	CHAR (10)	ORDISTAT	OIQTCONFIRM	オーダーされたアイテ ムの数量
NC_Item 通貨タイプ	CHAR (10)	ORDISTAT	OICPCUR	アイテムの価格を表す 通貨。 価格の形式 は、ISO 4217 規格に 準拠する必要があります。
NC_Item 単価	CHAR (16)	ORDISTAT	OIPRTOT	アイテムの単価
NC_Item 合計価格	CHAR (16)	ORDISTAT	OITOTPRC	アイテムの合計価格
NC_Item 合計税額	CHAR (16)	ORDISTAT	OITXTOT	アイテムの消費税合計
NC_ItemTotal ShippingPrice	CHAR (16)	ORDISTAT	OISHTOT	アイテムの配送料金



フィールド名	フィールド・タイプ	テーブル名	列名	注
NC_ItemTotal 配送料税額	CHAR (16)	ORDISTAT	OISHTXTOT	アイテムの配送料の税額合計
NC_Item コメント	CHAR (250)	ORDISTAT	OICMNT	オーダーされたアイテムに関するショッパーからのコメント。たとえば、ショッパーはオーダーした贈答品と一緒にあいさつ文を付けることができます。

以下のテーブルは、バージョン 01 の ORDER\_STATUS\_UPDATE\_ITM010\_DATA に対するフィールドの形式およびソースの説明です:

フィールド名	フィールド・タイプ	テーブル名	列名	説明
NC_ITM010 バージョン番号	CHAR (15)	N/A	N/A	固定値 01
NC_Order 参照番号	CHAR (10)	ORDISTAT	ORDERS_ID	WebSphere Commerce のオーダー参照番号。
NC_Item 参照番号	CHAR (10)	ORDISTAT	ORDERITMES_ID	WebSphere Commerce のアイテム参照番号。
NC_Order アイテムの状況	CHAR (32)	ORDISTAT	OISTATUS	オーダーのアイテム状況
NC_Item Schedule ShipDate	CHAR (8)	ORDISTAT	OISSTIME	スケジュールされた配送日 (YYYYMMDD)
NC_Item 実配送日	CHAR (8)	ORDISTAT	OIASTIME	実配送日 (YYYYMMDD)
NC_Item 発行日	CHAR (8)	ORDISTAT	OIPLTIME	オーダーが発行される日付 (YYYYMMDD)
NC_Item 数量	CHAR (10)	ORDISTAT	OIQTCONFIRM	オーダーされたアイテムの数量
NC_Item コメント	CHAR (250)	ORDISTAT	OICMNT	オーダーされたアイテムに関するショッパーからのコメント。たとえば、ショッパーはオーダーした贈答品と一緒にあいさつ文を付けることができます。

## ORDER\_CREATE\_PROLOG\_DATA

オーダー作成のメッセージには、ORDER\_CREATE\_PROLOG\_DATA セグメントが組み込まれています。このデータ・セグメントは、定義されているメッセージのタイプを識別します。そのセグメントは、ON (新規オーダー) のフィールド値をもつオーダー作成のメッセージであることを示すものです。

以下のテーブルは、ORDER\_CREATE\_PROLOG\_DATA のフィールドの形式およびソースの説明です:

フィールド名	フィールド・タイプ	フィールド値
NC_MsgType	CHAR (10)	ON
NC_MsgVersion	CHAR (15)	01
NC_RESERVED	CHAR (10)	IBM 用に予約済みです。

---

## ORDER\_STATUS\_UPDATE\_PROLOG\_DATA

オーダー状況更新のメッセージには、ORDER\_STATUS\_UPDATE\_PROLOG\_DATA セグメントが組み込まれています。このデータ・セグメントは、定義されているメッセージのタイプを識別します。そのセグメントは、OS (オーダーの状況) のフィールド値をもつオーダー状況の更新メッセージであることを示すものです。

オーダー状況の更新メッセージは、オーダー状況の更新 バージョン 01 およびオーダー状況の更新バージョン 02 のアプリケーション・データ 2 セットをサポートしています。バージョン 02 は、バージョン 01 のデータのスーパーセットです。特に、バージョン 01 の ORDER\_STATUS\_UPDATE\_PROLOG\_DATA セグメントは、NC\_MsgVersion フィールドが 01 の値であり、バージョン 02 の NC\_MsgVersion は 02 です。

以下のテーブルは、バージョン 02 の ORDER\_STATUS\_UPDATE\_PROLOG\_DATA に対するフィールドの形式およびソースの説明です:

フィールド名	フィールド・タイプ	フィールド値
NC_MsgType	CHAR (10)	OS
NC_MsgVersion	CHAR (15)	02
NC_RESERVED	CHAR (10)	IBM 用に予約済みです。

以下のテーブルは、バージョン 01 の ORDER\_STATUS\_UPDATE\_PROLOG\_DATA に対するフィールドの形式およびソースの説明です:

フィールド名	フィールド・タイプ	フィールド値
NC_MsgType	CHAR (10)	OS
NC_MsgVersion	CHAR (15)	01
NC_RESERVED	CHAR (10)	IBM 用に予約済みです。

---

## PRODUCT\_PRICE\_UPDATE\_PROLOG\_DATA

商品価格更新のメッセージには、PRODUCT\_PRICE\_UPDATE\_PROLOG\_DATA セグメントが組み込まれています。このデータ・セグメントは、定義されているメッセージのタイプを識別します。そのセグメントは、PP (商品価格) のフィールド値をもつ商品価格の更新メッセージであることを示すものです。

商品価格の更新メッセージは、商品価格の更新 バージョン 01 および商品価格の更新 バージョン 02 のアプリケーション・データ 2 セットをサポートしています。バージョン 02 は、バージョン 01 のデータのスーパーセットです。特に、バージョン 01 の PRODUCT\_PRICE\_UPDATE\_PROLOG\_DATA セグメントは、NC\_MsgVersion フィールドが 01 の値であり、バージョン 02 の NC\_MsgVersion は 02 です。

以下のテーブルは、バージョン 02 の PRODUCT\_PRICE\_UPDATE\_PROLOG\_DATA に対するフィールドの形式およびソースの説明です:

フィールド名	フィールド・タイプ	フィールド値
NC_MsgType	CHAR (10)	PP
NC_MsgVersion	CHAR (15)	02
NC_RESERVED	CHAR (10)	IBM 用に予約済みです。

以下のテーブルは、バージョン 01 の PRODUCT\_PRICE\_UPDATE\_PROLOG\_DATA に対するフィールドの形式およびソースの説明です:

フィールド名	フィールド・タイプ	フィールド値
NC_MsgType	CHAR (10)	PP
NC_MsgVersion	CHAR (15)	01
NC_RESERVED	CHAR (10)	IBM 用に予約済みです。

---

## PRODUCT\_QUANTITY\_UPDATE\_PROLOG\_DATA

商品数量更新のメッセージには、PRODUCT\_QUANTITY\_UPDATE\_PROLOG\_DATA セグメントが組み込まれています。このデータ・セグメントは、定義されているメッセージのタイプを識別します。そのセグメントは、PQ (商品指示) のフィールド値をもつ商品数量の更新メッセージであることを示すものです。

以下のテーブルは、PRODUCT\_QUANTITY\_UPDATE\_PROLOG\_DATA のフィールドの形式およびソースの説明です:

フィールド名	フィールド・タイプ	フィールド値
NC_MsgType	CHAR (10)	PQ
NC_MsgVersion	CHAR (15)	01
NC_RESERVED	CHAR (10)	IBM 用に予約済みです。

---

## CUSTOMER\_NEW\_PROLOG\_DATA

新規顧客へのメッセージには、CUSTOMER\_NEW\_PROLOG\_DATA セグメントが組み込まれます。このデータ・セグメントは、定義されているメッセージのタイプを識別します。そのセグメントは、NC (新規顧客) のフィールド値をもつ更新顧客へのメッセージであることを示すものです。

以下のテーブルは、CUSTOMER\_NEW\_PROLOG\_DATA のフィールドの形式およびソースの説明です:

フィールド名	フィールド・タイプ	フィールド値
NC_MsgType	CHAR (10)	NC
NC_MsgVersion	CHAR (15)	01
NC_RESERVED	CHAR (10)	IBM 用に予約済みです。

---

## CUSTOMER\_UPDATE\_PROLOG\_DATA

更新顧客へのメッセージには、CUSTOMER\_UPDATE\_PROLOG\_DATA セグメントが組み込まれています。このデータ・セグメントは、定義されているメッセージのタイプを識別します。そのセグメントは、UC (更新顧客) のフィールド値をもつ更新顧客へのメッセージであることを示すものです。

以下のテーブルは、CUSTOMER\_UPDATE\_PROLOG\_DATA のフィールドの形式およびソースの説明です:

フィールド名	フィールド・タイプ	フィールド値
NC_MsgType	CHAR (10)	UC
NC_MsgVersion	CHAR (15)	01
NC_RESERVED	CHAR (10)	IBM 用に予約済みです。

---

## アウトバウンド・メッセージの DATUSR\_DATA

アウトバウンド・メッセージのオーダー作成には、DATUSR\_DATA データ・セグメントが組み込まれています。追加データを送信するには DATUSR レコードをこのメッセージに組み込んでください。メッセージに組み込まれていない特別なデータを渡すためのフィールドを、カスタマイズすることができます。

以下のテーブルは、DATUSR\_DATA の各フィールドの形式およびソースの説明です:

フィールド名	フィールド・タイプ	説明
NC_FieldName	CHAR (8)	追加する新規フィールド名
NC_FieldLength	CHAR (10)	NC_FieldValue フィールド長さ
NC_FieldValue	CHAR (可変長)	新規フィールド値のテキスト・ストリング

---

## インバウンド・メッセージの DATUSR\_DATA

インバウンド・メッセージの、オーダーの状況更新メッセージ、顧客新規メッセージ、および顧客更新メッセージには、DATUSR\_DATA データ・セグメントが組み込まれています。追加データを受け取るには DATUSR レコードをこのメッセージに組み込んでください。メッセージに組み込まれていない特別なデータを渡すためのフィールドを、カスタマイズすることができます。DATUSR\_DATA は複数回繰り返すことができます。

以下のテーブルは、DATUSR\_DATA の各フィールドの形式およびソースの説明です:

フィールド名	フィールド・タイプ	説明
NC_FieldName	CHAR (8)	追加する新規フィールド名
NC_FieldLength	CHAR (10)	NC_FieldValue フィールド長さ
NC_FieldValue	CHAR (可変長)	新規フィールド値のテキスト・ストリング

## NCCustomer\_10.mod ファイル

NCCustomer\_10.mod 顧客共通ファイルは、新規ショッパーのショッパー情報から成っています。これは、Create\_NC\_Customer および Update\_NC\_Customer の両方のメッセージに使用されます。

すべての MOD ファイルは、以下のディレクトリーにあります:

**2000** ドライブ:¥Program Files¥WebSphere¥CommerceServer¥xml¥messaging

**NT** drive:¥WebSphere¥CommerceServer¥xml¥messaging

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

**SOLARIS** **LINUX** /opt/WebSphere/CommerceServer/xml/messaging

**400** /QIBM/Proddata/WebCommerce/xml/messaging

以下のテーブルは、NCCustomer\_10.mod の XML エlement値の形式およびソースに関する説明です。データベース列の説明については、それと関連したテーブルへのリンクに従ってください。特に注釈がない限り、すべてのフィールドはオプションです。

レベル	XML エlement	コメント	テーブル名	列名
1	LoginInfo	必須	N/A	N/A
1.1	LoginID	必須	USERREG	LOGONID
1.2	パスワード	必須	USERREG	LOGONPASSWORD
1.3	VerifyPassword	必須	N/A	N/A
2	MerchantID		N/A	N/A
3	MethodOf Communication		USERPROF	PREFERREDCOMM
4	Challenge Question		USERREG	CHALLENGEQUESTION
5	Challenge Answer		USERREG	CHALLENGEANSWER
6	ShopperField	最初に繰り返されるオカレンス	USERS	FIELD1
6	ShopperField	2 番目に繰り返されるオカレンス	USERS	FIELD2
7	ContactPerson 名前	必須	N/A	N/A
7.1	敬称		ADDRESS	PERSONTITLE
7.2	FullName	このバージョンでサポートされていません	N/A	N/A
7.3	LastName	必須	ADDRESS	LASTNAME
7.4	FirstName		ADDRESS	FIRSTNAME
7.5	MiddleName		ADDRESS	MiddleName
7.6	AlternateName	このバージョンでサポートされていません	N/A	N/A
8	RepCompany		ADDRESS	ORGNAME
9	住所	必須	N/A	N/A

レベル	XML エlement	コメント	テーブル名	列名
9.1	AddressLine	必須、繰り返し (初回)	ADDRESS	ADDRESS1
9.1	AddressLine	2 番目に繰り返されるオカレンス	ADDRESS	ADDRESS2
9.1	AddressLine	3 番目に繰り返されるオカレンス	ADDRESS	ADDRESS3
9.2	市区町村	必須	ADDRESS	City
9.3	State	必須	ADDRESS	State
9.4	Zip	必須	ADDRESS	ZIPCODE
9.5	Country	必須	ADDRESS	COUNTRY
10	ContactInfo	必須	N/A	N/A
10.1	Telephone	最初に繰り返されるオカレンス	ADDRESS	PHONE1
10.1	Telephone	2 番目に繰り返されるオカレンス	ADDRESS	PHONE2
10.2	Email	最初に繰り返されるオカレンス	ADDRESS	EMAIL1
10.2	Email	2 番目に繰り返されるオカレンス	ADDRESS	EMAIL2
10.3	Fax		ADDRESS	FAX1
11	DayPhoneInfo		N/A	N/A
11.1	PhoneInfo		N/A	N/A
11.1.A1	type	Attribute	ADDRESS	PHONE1TYPE
11.1.A2	isListed	Attribute	ADDRESS	PUBLISHPHONE1
12	EveningPhone Info		N/A	N/A
12.1	PhoneInfo		N/A	N/A
12.1.A1	type	Attribute	ADDRESS	PHONE2TYPE
12.1.A2	isListed	Attribute	ADDRESS	PUBLISHPHONE2
13	BestTimeToCall		ADDRESS	BESTCALLINGTIME
14	Include PackageInsert		ADDRESS	PACKAGESUPPRESSION
15	住所 OptField	最初に繰り返されるオカレンス	ADDRESS	FIELD1
15	住所 OptField	最初に繰り返されるオカレンス	ADDRESS	FIELD2
16	Gender		N/A	N/A
16.A1	value	必須、属性	USERDEMO	GENDER
17	AgeGroup		USERDEMO	AGE
18	IncomeGroup		USERDEMO	INCOME
19	MaritalStatus		N/A	N/A
19.A1	value	必須、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

レベル	XML エlement	コメント	テーブル名	列名
25	Demographics	最初に繰り返されるオカレンス	USERDEMO	FIELD1
25	Demographics	2 番目に繰り返されるオカレンス	USERDEMO	FIELD2
25	Demographics	3 番目に繰り返されるオカレンス	USERDEMO	FIELD3
25	Demographics	4 番目に繰り返されるオカレンス	USERDEMO	FIELD4
25	Demographics	5 番目に繰り返されるオカレンス	USERDEMO	FIELD5
25	Demographics	6 番目に繰り返されるオカレンス	USERDEMO	FIELD6
26	UserData		N/A	N/A

---

## NCCCommon.mod - ソース・ファイル

WebSphere Commerce では、DTD ファイルに基づいて、すべてのインバウンド XML メッセージを定義します。各 DTD ファイルは、数個の XML エlement と NCCCommon.mod ファイルで構成されています。さらに、Create\_NC\_Customer\_10.dtd および Update\_NC\_Customer\_10.dtd ファイルも NCCustomer10.mod コモン・ファイルから情報を組み込みます。

以下は、すべての MOD および DTD ファイルのあるディレクトリです:

<b>2000</b>	ドライブ:¥Program Files¥WebSphere¥CommerceServer¥xml¥messaging
<b>NT</b>	drive:¥WebSphere¥CommerceServer¥xml¥messaging
<b>AIX</b>	/usr/WebSphere/CommerceServer/xml/messaging
<b>SOLARIS</b>	<b>LINUX</b> /opt/WebSphere/CommerceServer/xml/messaging
<b>400</b>	/QIBM/Proddata/WebCommerce/xml/messaging

---

## NCCustomer\_10.mod - ソース・ファイル

WebSphere Commerce では、DTD ファイルに基づいて、すべてのインバウンド XML メッセージを定義します。各 DTD ファイルは、数個の XML エlement と NCCCommon.mod ファイルで構成されています。さらに、Create\_NC\_Customer\_10.dtd および Update\_NC\_Customer\_10.dtd ファイルも NCCustomer10.mod 顧客コモン・ファイルから情報を組み込みます。

以下は、すべての MOD および DTD ファイルのあるディレクトリです:

<b>2000</b>	ドライブ:¥Program Files¥WebSphere¥CommerceServer¥xml¥messaging
<b>NT</b>	drive:¥WebSphere¥CommerceServer¥xml¥messaging
<b>AIX</b>	/usr/WebSphere/CommerceServer/xml/messaging
<b>SOLARIS</b>	<b>LINUX</b> /opt/WebSphere/CommerceServer/xml/messaging
<b>400</b>	/QIBM/Proddata/WebCommerce/xml/messaging





## 特記事項

本書で IBM 製品、プログラム、またはサービスに言及していても、その IBM 製品、プログラム、またはサービスのみが使用可能であることを意味するものではありません。IBM 製品、プログラムまたはサービスに代えて、IBM の知的所有権を侵害することのない機能的に同等のプログラムまたは製品を使用することができます。ただし、IBM によって明示的に指定されたものを除き、他社の製品と組み合わせた場合の動作の評価と検証はお客様の責任で行っていただきます。

IBM は、本書に記載されている内容に関して特許権（特許出願中のものを含む。）を保有している場合があります。本書の提供は、お客様にこれらの特許権について実施権を許諾することを意味するものではありません。実施権の許諾については、下記の宛先に書面にてご照会ください。

〒106-0032 東京都港区六本木 3 丁目 2-31  
IBM World Trade Asia Corporation  
Intellectual Property Law & Licensing

本プログラムのライセンス保持者で、(i) 独自に作成したプログラムとその他のプログラム（本プログラムを含む）との間での情報交換、および (ii) 交換された情報の相互利用を可能にすることを目的として、本プログラムに関する情報を必要とする方は、下記に連絡してください。

Lab Director  
IBM Canada Ltd. Laboratory  
8200 Warden Avenue  
Markham, Ontario  
L6G 1C7  
Canada

本プログラムに関する上記の情報は、適切な使用条件の下で使用することができますが、有償の場合もあります。

本書には、日常の業務処理で用いられるデータや報告書の例が含まれています。より具体性を与えるために、それらの例には、個人、企業、ブランド、あるいは製品などの名前が含まれている場合があります。これらの名称はすべて架空のものであり、名称や住所が類似する企業が実在しているとしても、それは偶然にすぎません。

この文書には、他社のインターネット・サイトへの参照を含む、他社製品の情報が含まれている場合があります。IBM は、そのような情報の正確性、完全性、または使用については何ら責任を負いません。

この製品は、SET プロトコルに基づいています。

米国政府機関ユーザーの権利の制限 - IBM Corporation との間の GSA ADP Schedule Contract により、使用、複製、および開示が制限されます。

### 商標

以下は、IBM Corporation の商標です。

AIX	CICS	DB2
DB2 Extenders	Encina	HotMedia
IBM	iSeries	MQSeries

SecureWay  
400

VisualAge

WebSphere

Blaze Advisor は HNC Software, Inc. の米国およびその他の国における商標です。

Java およびすべての Java 関連の商標およびロゴは、Sun Microsystems, Inc. の米国およびその他の国における商標または登録商標です。

Lotus および Domino は、Lotus Development Corporation の商標です。

Microsoft、Windows、Windows NT および Windows ロゴは、Microsoft Corporation の米国およびその他の国における商標です。

Oracle は Oracle Corporation の登録商標です。

SET および SET ロゴは、SET Secure Electronic Transaction LLC の商標です。詳しくは、<http://www.setco.org/aboutmark.html> を参照してください。

UNIX は、The Open Group がライセンスしている米国およびその他の国における登録商標です。

他の会社名、製品名およびサービス名などはそれぞれ各社の商標または登録商標です。



