

IBM WebSphere Commerce



WebAuction Sample Store: Online Help Files

Version 5.4

IBM WebSphere Commerce



WebAuction Sample Store: Online Help Files

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、113 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
WebAuction Sample Store: 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

第 1 章 Sample stores	1
サンプル・ストア	1
WebAuction サンプル・ストア	1
第 2 章 Creating a store using a sample	3
サンプルを使用したストアの作成	3
サンプル・ストア・アーカイブの作成	3
ストア・データベース資産の変更	4
Web 資産の変更	13
ストア・サービスからのサンプル・ストア・アーカイブの発行	13
コマンド行を使用したストア・アーカイブの発行 (Windows)	15
ストアのための Payment Manager のセットアップ	17
サンプル・ストア用のスケジュールされたジョブの作成	18
サンプル・ストアの E メール通知の構成	20
第 3 章 Sample store database assets	23
ストア・データベース資産	23
サンプル・ストアのアクセス制御データベース資産	24
第 4 章 Sample store pages	27
サンプル・ストア住所録ページ	27
サンプル・ストア・フッター	28
サンプル・ストア・ヘッダー	30
サンプル・ストア・ヘルプ・ページ	31
サンプル・ストア・ホーム・ページ	31
サンプル・ストア - 左側のナビゲーション・フレーム	32
サンプル・ストア・ログイン・ページ	33
サンプル・ストア新着商品ページ	36
サンプル・ストア・オーダー要約ページ	37
サンプル・ストア・プライバシー・ページ	38
サンプル・ストア商品ページ	38
サンプル・ストア登録ページ	40
サンプル・ストア・ショッピング・カート	43
サンプル・ストア請求先住所の選択ページ	44
サンプル・ストア「配送先住所の選択」ページ	46
サンプル・ストア配送方法ページ	47
WebFashion と NewFashion のサンプル・ストア・バンドル表示ページ	48
WebFashion と NewFashion のサンプル・ストア・パッケージ表示ページ	49
WebFashion サンプル・ストア - クイック・チェックアウト・プロフィールの作成ページ	50
サンプル・ストア・クイック・チェックアウト要約ページ	53
WebFashion および NewFashion サンプル・ストア購入希望商品リスト・ページ	54
WebFashion および NewFashion の E メール通知ページ	56
WebFashion および NewFashion サンプル・ストア「オーダーの表示」ページ	57
第 5 章 Auction pages	61
「オークション - すべての入札」ページ	61
「オークション - 自動入札の作成」ページ	62
「オークション - 自動入札フォーム」ページ	62
「オークション - 自動入札の変更」ページ	64


「オークション - 入札の変更」 ページ	64
「オークション」 ページ	64
「オークション - 入札の確認」 ページ	65
「オークションのイメージ・マップ」 ページ	65
「オークション - 入札の作成」 ページ	66
「オークション - 入札のフォーム」 ページ	67
「入札のオーバーライド件名」 ページ	68
「オークション - 共通」 ページ	68
「オーダーの完了」 ページ	69
「完了オーダーの件名」 ページ	70
「オークション - アイテムの表示」 ページ	70
「オークション - エラー入札送信のフレーム」 ページ	71
「オークション - ErrorDataBean 共通」 ページ	71
「オークション・ヘッダー」 ページ	72
「オークション・ホーム」 ページ	72
「オークション - ログイン・フォーム」 ページ	73
「すべてのオークション・リスト」 ページ	73
「Auction Message Forum Append (オークション - メッセージ・フォーラムの付加)」 ページ	75
「オークション - メッセージ共通」 ページ	76
「オークション - メール表示」 ページ	76
「Auction Message Forum Add (オークション - メッセージ・フォーラムの追加)」 ページ	79
「オークション・メイン・ホーム」 ページ	80
「オークション - メール・リスト」 ページ	80
「オークション・ナビゲーション」 ページ	81
「入札のオーバーライド」 ページ	82
「開始オークションの件名」 ページ	83
「オークションの開始」 ページ	84
「オークション - 最高入札の表示」 ページ	85
ユーザーのオークションの入札	85
「オークション・ギャラリー」 ページ	87
「Auction Message Forum List (オークション - メッセージ・フォーラム・リスト)」 ページ	89
「落札者」 ページ	90
「落札者件名」 ページ	91
第 6 章 Sample store use cases	93
サンプル・ストアの使用事例	93
WebFashion クイック・チェックアウト・プロフィール作成の使用事例	93
WebFashion クイック・チェックアウトの使用事例	95
新規住所追加の使用事例	96
WebFashion と NewFashion の購入希望商品リストへのアイテム追加の使用事例	97
商品カテゴリー・ビューの使用事例	98
ショッピング・カートチェックアウトの使用事例	99
住所編集の使用事例 (Business Edition)	101
ホーム・ページの使用事例	102
ログオンの使用事例	103
個人アカウント管理の使用事例	104
WebFashion および NewFashion オーダー表示の使用事例	105
商品ページ表示の使用事例	106
個人情報変更の使用事例	107
登録の使用事例	108
ショッピング・カート表示の使用事例	109
WebFashion と NewFashion の購入希望商品リスト表示の使用事例	110

特記事項 113

第 1 章 Sample stores

サンプル・ストア

WebSphere Commerce では、いくつかのサンプル・ストアが提供されています。これらのストアをベースにして、独自のストアを作成することができます。各ストアは、最近の大手小売サイトで見られる機能が表示されるように設計されています。WebSphere Commerce で提供されているサンプル・ストアは以下のとおりです。

- InFashion。企業顧客間のオンライン・ファッション・ストア。
- WebFashion。さらに機能が強化された、企業顧客間のオンライン・ファッション・ストア。
- WebAuction。企業顧客間のオンライン・オークション・ストア。
- NewFashion。新機能を持つ、企業顧客間のオンライン・ファッション・ストア。
-  Business ToolTech。企業間のオンライン・ハードウェア提供ストア。

それぞれのサンプル・ストアはストア・アーカイブとして WebSphere Commerce にパッケージ化されているため、個別にインストールする必要はありません。各サンプル・ストアを表示するのに必要なのは、ストア・サービス・ツールを使用してそのサンプル・ストアに基づく新規ストア・アーカイブを作成してから、そのアーカイブを WebSphere Commerce サーバーに発行するだけです。詳しくは、ストア・サービスを使用したストア・アーカイブの作成を参照してください。

WebSphere Commerce でのストアの作成では、サンプルのストア・アーカイブを選択してから修正することが基本になっているため、すべてのサンプル・ストアは、どのようなストアのベースとしても機能するように設計されています。サンプル・ストアは、典型的なショッピング・フローに基づいています。すべてのサンプル・ストア・ページはカスタマイズできます。

各サンプルについて詳しくは、次のリンクの中にあるそのサンプル・ストアのファイルを参照してください。

WebAuction サンプル・ストア

WebAuction は、WebSphere Commerce に備えられているオンライン・ファッション・ストアのうちの 1 つです。このサンプル・ストアには、最近の大手の小売サイトで一般に使用されている機能の多くがインプリメントされています。WebAuction サンプル・ストアに組み込まれているフィーチャーのいくつかを以下に示します。

- 3 つのオークション・タイプをサポート: オープン・クライ・オークション、シールド・ビッド・オークション、およびダッチ・オークション
- オークション・ギャラリー
- 入札フォームおよび自動入札フォーム
- オークション通知を含むテキスト・メッセージ
- 商品カタログおよびアイテムの説明
- オークションごとの詳細情報
- 多文化サポート
- 購入希望商品リスト
- マーケティング・キャンペーンの作成

- クイック・チェックアウト
- パッケージおよびバンドル
- 割引

WebAuction ストアは、機能的なオンライン・ストアに必要なページおよびフィーチャーをすべて備えています。WebAuction は、ストア・アーカイブとして WebSphere Commerce にパッケージ化されているため、個別にインストールする必要はありません。サンプル・ストアを表示するために必要なことは、ストア・サービス・ツールを使用して WebAuction ベースの新規ストア・アーカイブを作成し、次にそれを WebSphere Commerce Server に発行するだけです。詳しくは、ストア・サービスを使用したストア・アーカイブの作成を参照してください。


WebSphere Commerce でのストアの作成は、サンプルのストア・アーカイブを選択してから変更することが基本になっているため、WebAuction は、実際のストアのベースとして機能するように設計されています。これは、シンプルでありながら、実証済みのショッピング・フローに基づいており、すべてのサンプル・ストア・ページは、簡単にカスタマイズすることができます。

WebAuction のショッピング・フローについて詳しくは、サンプル・ストアのショッピング・フロー・チャートを参照してください。各ページがどのように機能するかなどの技術的な詳細については、そのページの対応する参照情報を参照してください。

第 2 章 Creating a store using a sample

サンプルを使用したストアの作成

サンプル・ストアの 1 つを使用してストアを作成するには、以下のようにします。

1. (オプション)  セラーとして機能する組織を作成する。
セラーとして機能する新規組織を作成するかどうかを決定するには、『共有されるデータ資産』を参照してください。
2. 次の役割を果たすようにユーザーを作成する。
 - サイト管理者 (デフォルトのサイト管理者を使用していない場合)
 - セラー管理者
 - ストア管理者
 - Store Developer (ストア・デベロッパー)
重要: ストア・アーカイブを発行するためには、すべてのストアについての、サイト管理者、ストア管理者、またはストア開発者のアクセス権が必要です。
ストア管理者権限をもつユーザーを作成する場合は、そのアクセス・グループがすべてのストアに適用されることを確認してください。
3. ストア・サービスを使用したストア・アーカイブの作成
4. ストア・データベース資産を変更する。
5. Web 資産を変更する。
6. ストア・アーカイブを発行する。
7. (オプション) ストアを構成する。
8. ストアのために Payment Manager をセットアップする。

重要:

1. サンプル・ストアで何らかのセットアップを行わなければ、すべての機能が正しく機能しない場合もあります。サンプル・ストアを基にしてストアを作成する場合は、セットアップを完了しなければならない場合があります。詳しくは、サンプル・ストアのセットアップを参照してください。
2. カタログや配送センターのような特定のストア・データ資産は、ストア間で共有されます。その結果、同じサンプル・ストアに基づいて複数のストアを発行する場合、カタログおよび配送センターは各ストアとも同じになります。同じサンプルで他のストアを発行した場合には、あるカタログで変更を行うとそれらの変更が上書きされます。詳細情報および、変更の上書きを避ける方法については、共有データ資産を参照してください。

サンプル・ストア・アーカイブの作成

ストア・サービスで、サンプルとして使用できるストア・アーカイブを作成するには、*IBM WebSphere* ストア開発者ガイドを参照してください。

ストア・データベース資産の変更

ストア・サービス中のツールを使用してストア・アーカイブを作成する場合、新規ストア・アーカイブには最初の時点ではその基礎としたサンプル・ストア・アーカイブ、たとえば `infashion.sar` と同じストア・データベース資産が入ります。ストア・アーカイブ中で、ストア・データベース資産は、XML ファイルの形を取ります。

ほとんどの場合、ストア・データベース資産を変更するには、XML ファイルを直接に編集しなければなりません。場合によっては、ストア・サービス中のツールを使用してデータベース資産を編集できます。

直接データベース資産を編集するオプションもあります。つまり、Commerce サーバーに対するストア・アーカイブを発行し終えてから、WebSphere Commerce アクセラレーターかローダー・パッケージを使用するか、または直接 SQL を挿入して、データベースを編集できます。ストア・アーカイブ中の資産ではなくデータベースを編集する方を選択した場合は、ストア・アーカイブを更新してデータベース中の変更内容に一致させるか、またはストア・アーカイブの使用を中止しなければなりません。

オプションは、以下の表にリストしてあります。

重要:

- 資産名を検索するストア・サービスのツールは、以下のテーブルにリストされています。つまり、ストア・サービスのツールを使用してストア・アーカイブを編集する場合は、ご使用のストア・アーカイブにある同じ資産名を使用する必要があります。
- ストア・アーカイブが WebSphere Commerce Server に発行されると、データベース情報は以下の資産列で指定された順でロードされます。そのため、資産の順は、`sarinfo.xml` ファイルに指定されたように、以下の資産の順序に一致している必要があります。
- 各 XML ファイルのデータベース情報の順序は、必ずしも以下のデータベース・テーブル列で指定された順序と一致している必要はありません。ただし、親テーブルの情報は子テーブルの情報の先に位置する必要があります。
- オプションとしてマークされた情報は、機能ストアを作成するために必要ではありません。
- データベース編集オプション列において、特に指定がなければ、SQL 挿入かローダー・パッケージを使用してすべてのデータベース資産を編集できることに注意してください。したがって、この列には、WebSphere Commerce アクセラレーターによって編集できる資産が明示されていることとなります。

ストア・データベース資産	データベース・テーブル	ストア・アーカイブ編集オプション	データベース編集オプション
fulfillment	FFMCENTER (0...1)	<ul style="list-style-type: none">ストア・アーカイブ XML ファイルを編集する。	<ul style="list-style-type: none">WebSphere Commerce アクセラレーターの「新規の配送センター」と「配送センターの変更」のページ。配送センターの作成および配送センターの変更を参照。

ストア・データベース資産	データベース・テーブル	ストア・アーカイブ編集オプション	データベース編集オプション
store	STOREENT	<ul style="list-style-type: none"> ストア・アーカイブ XML ファイルを編集する。 「ストア・プロフィール」ノートブックを使用してストア情報を変更する。「ストア・プロフィール」ノートブックは以下のデータベース列を編集します。 <ul style="list-style-type: none"> IDENTIFIER DIRECTORY SETCURR 	
	STADDRESS	<ul style="list-style-type: none"> 「ストア・プロフィール」ノートブックを使用してストア情報を変更する。 	<ul style="list-style-type: none"> WebSphere Commerce アクセラレーターの「新規の配送センター」と「配送センターの変更」のページ。 配送センターの作成および配送センターの変更を参照。
	STOREENTDS	<ul style="list-style-type: none"> 「ストア・プロフィール」ノートブックを使用してストア情報を変更する。「ストア・プロフィール」ノートブックは以下のデータベース列を編集します。 <ul style="list-style-type: none"> DESCRIPTION DISPLAYNAME 	
	STORELANG	<ul style="list-style-type: none"> 「ストア・プロフィール」ノートブックを使用してストア情報を変更する。「ストア・プロフィール」ノートブックは以下のデータベース列を編集します。 <ul style="list-style-type: none"> LANGUAGE_ID 	
	STORELANGDS	<ul style="list-style-type: none"> ストア・アーカイブ XML ファイルを編集する。 	
	STORE	<ul style="list-style-type: none"> 「ストア・プロフィール」ノートブックを使用してストア情報を変更する。「ストア・プロフィール」ノートブックは以下のデータベース列を編集します。 <ul style="list-style-type: none"> PHONE1 CITY STORE_ID COUNTRY STATE EMAIL1 ADDRESS1 ADDRESS2 FAX ZIPCODE 	
	DISPENTREL (デフォルトの CATENTRY テンプレート。CATENTRY_ID = -1)	<ul style="list-style-type: none"> 「ストア・プロフィール」ノートブックを使用してストア情報を変更する。「ストア・プロフィール」ノートブックは PAGENAME データベース列を編集します。 	
	DISPCGPREL (デフォルトの CATEGORY テンプレート。CATGROUP_ID = -1)	<ul style="list-style-type: none"> 「ストア・プロフィール」ノートブックを使用してストア情報を変更する。「ストア・プロフィール」ノートブックは PAGENAME データベース列を編集します。 	
	VENDOR	<ul style="list-style-type: none"> ストア・アーカイブ XML ファイルを編集する。 	

ストア・データベース資産	データベース・テーブル	ストア・アーカイブ編集オプション	データベース編集オプション
catalog	CATGROUP	<ul style="list-style-type: none"> カタログ情報を変更する。 ストア・アーカイブ XML ファイルを編集する。 	<ul style="list-style-type: none"> WebSphere Commerce アクセラレーターの商品管理ツール
	CATGRPATTR	<ul style="list-style-type: none"> カタログ情報を変更する。 ストア・アーカイブ XML ファイルを編集する。 	
	CATGRPDESC	<ul style="list-style-type: none"> カタログ情報を変更する。 ストア・アーカイブ XML ファイルを編集する。 	<ul style="list-style-type: none"> WebSphere Commerce アクセラレーターの商品管理ツール
	CATALOG	<ul style="list-style-type: none"> カタログ情報を変更する。 ストア・アーカイブ XML ファイルを編集する。 	
	CATALOGDSC	<ul style="list-style-type: none"> カタログ情報を変更する。 ストア・アーカイブ XML ファイルを編集する。 	
	CATTOGRP	<ul style="list-style-type: none"> カタログ情報を変更する。 ストア・アーカイブ XML ファイルを編集する。 	
	CATGRPREL	<ul style="list-style-type: none"> カタログ情報を変更する。 ストア・アーカイブ XML ファイルを編集する。 	<ul style="list-style-type: none"> WebSphere Commerce アクセラレーターの商品管理ツール
	CATENTRY	<ul style="list-style-type: none"> カタログ情報を変更する。 ストア・アーカイブ XML ファイルを編集する。 	<ul style="list-style-type: none"> WebSphere Commerce アクセラレーターの商品管理ツール
	CATENTDESC	<ul style="list-style-type: none"> カタログ情報を変更する。 ストア・アーカイブ XML ファイルを編集する。 	<ul style="list-style-type: none"> WebSphere Commerce アクセラレーターの商品管理ツール
	ATTRIBUTE	<ul style="list-style-type: none"> カタログ情報を変更する。 ストア・アーカイブ XML ファイルを編集する。 	<ul style="list-style-type: none"> WebSphere Commerce アクセラレーターの商品管理ツール
	ATTRVALUE	<ul style="list-style-type: none"> カタログ情報を変更する。 ストア・アーカイブ XML ファイルを編集する。 	<ul style="list-style-type: none"> WebSphere Commerce アクセラレーターの商品管理ツール
	CATGPENREL	<ul style="list-style-type: none"> カタログ情報を変更する。 ストア・アーカイブ XML ファイルを編集する。 	<ul style="list-style-type: none"> WebSphere Commerce アクセラレーターの商品管理ツール
	CATENTREL	<ul style="list-style-type: none"> カタログ情報を変更する。 ストア・アーカイブ XML ファイルを編集する。 	
	BASEITEM	<ul style="list-style-type: none"> カタログ情報を変更する。 ストア・アーカイブ XML ファイルを編集する。 	<ul style="list-style-type: none"> WebSphere Commerce アクセラレーターの商品管理ツール
	ITEMSPC	<ul style="list-style-type: none"> カタログ情報を変更する。 ストア・アーカイブ XML ファイルを編集する。 	<ul style="list-style-type: none"> WebSphere Commerce アクセラレーターの商品管理ツール
	VERSIONSPC	<ul style="list-style-type: none"> カタログ情報を変更する。 ストア・アーカイブ XML ファイルを編集する。 	
	DISTARRANG	<ul style="list-style-type: none"> カタログ情報を変更する。 ストア・アーカイブ XML ファイルを編集する。 	
	RECEIPT	<ul style="list-style-type: none"> カタログ情報を変更する。 ストア・アーカイブ XML ファイルを編集する。 	
	RCPTAVAIL	<ul style="list-style-type: none"> カタログ情報を変更する。 ストア・アーカイブ XML ファイルを編集する。 	
	STOREITEM	<ul style="list-style-type: none"> カタログ情報を変更する。 ストア・アーカイブ XML ファイルを編集する。 	<ul style="list-style-type: none"> WebSphere Commerce アクセラレーターの商品管理ツール


ストア・データベース資産	データベース・テーブル	ストア・アーカイブ編集オプション	データベース編集オプション
tax	JURSTGROUP (税と配送で共通)	<ul style="list-style-type: none"> • ストア・アーカイブ XML ファイルを編集する。 • 「税」ノートブックを使用して、税設定を変更する。「税」ノートブックは以下のデータベース列を編集します。 <ul style="list-style-type: none"> - JURSTGROUP_ID - DESCRIPTION - SUBCLASS - STOREENT_ID - CODE 	
	JURST (税と配送で共通)	<ul style="list-style-type: none"> • 「税」ノートブックを使用して、税設定を変更する。「税」ノートブックは以下のデータベース列を編集します。 <ul style="list-style-type: none"> - JURST_ID - COUNTRY - STOREENT_ID - CODE - SUBCLASS - STATE 	
	JURSTGPREL (税と配送で共通)	<ul style="list-style-type: none"> • 「税」ノートブックを使用して、税設定を変更する。「税」ノートブックは以下のデータベース列を編集します。 <ul style="list-style-type: none"> - JURST_ID - JURSTGROUP_ID - SUBCLASS 	
	CALMETHOD (税と配送で共通)	<ul style="list-style-type: none"> • ストア・アーカイブ XML ファイルを編集する。 	
	TXCDCLASS (オプション、WebSphere Commerce アクセラレーターを使用してカテゴリ化している CALCODE の場合のみ。)	<ul style="list-style-type: none"> • ストア・アーカイブ XML ファイルを編集する。 	
	TAXCGRY	<ul style="list-style-type: none"> • 「税」ノートブックを使用して、税設定を変更する。「税」ノートブックは以下のデータベース列を編集します。 <ul style="list-style-type: none"> - TAXCGRY_ID - STOREENT_ID - NAME 	
	CALCODE (税と配送で共通)	<ul style="list-style-type: none"> • 「税」ノートブックを使用して、税設定を変更する。「税」ノートブックは以下のデータベース列を編集します。 <ul style="list-style-type: none"> - CALCODE_ID - CODE - CALUSAGE_ID - STOREENT_ID - GROUPBY - CALMETHOD_id - CALMETHOD_id_app - CALMETHOD_id_qfy 	

ストア・データベース資産	データベース・テーブル	ストア・アーカイブ編集オプション	データベース編集オプション
	CALCODEDSC (税と配送で共通。オプション)	<ul style="list-style-type: none"> • ストア・アーカイブ XML ファイルを編集する。 	
	CALRULE (税と配送で共通)	<ul style="list-style-type: none"> • 「税」ノートブックを使用して、税設定を変更する。「税」ノートブックは以下のデータベース列を編集します。 <ul style="list-style-type: none"> - CALRULE_ID - CALCODE_ID - TAXCGRY_ID - CALMETHOD_ID 	
	CALSCALE (税と配送で共通)	<ul style="list-style-type: none"> • 「税」ノートブックを使用して、税設定を変更する。「税」ノートブックは以下のデータベース列を編集します。 <ul style="list-style-type: none"> - CALSCALE_ID - CODE - STOREENT_ID - CALUSAGE_ID - SETCURR - CALMETHOD_ID 	
	CALSCALEDS (税と配送で共通)	<ul style="list-style-type: none"> • ストア・アーカイブ XML ファイルを編集する。 	
	CALRANGE (税と配送で共通)	<ul style="list-style-type: none"> • 「税」ノートブックを使用して、税設定を変更する。「税」ノートブックは以下のデータベース列を編集します。 <ul style="list-style-type: none"> - CALRANGE_ID - CALSCALE_ID - CALMETHOD_ID - RANGESTART 	
	CALRLOOKUP (税と配送で共通)	<ul style="list-style-type: none"> • 「税」ノートブックを使用して、税設定を変更する。「税」ノートブックは以下のデータベース列を編集します。 <ul style="list-style-type: none"> - CALSCALE_ID - CALRULE_ID 	
	CRULESCALE (税と配送で共通)	<ul style="list-style-type: none"> • 「税」ノートブックを使用して、税設定を変更する。「税」ノートブックは以下のデータベース列を編集します。 <ul style="list-style-type: none"> - CALSCALE_ID - CALRULE_ID 	
	CALRULEMGP (オプション)	<ul style="list-style-type: none"> • ストア・アーカイブ XML ファイルを編集する。 	
	CALCOTXEX (税免除、割引により使用)	<ul style="list-style-type: none"> • ストア・アーカイブ XML ファイルを編集する。 	
	STENCALUSG (共通、ストアのデフォルト CALCODE、2 エントリ: 販売税 1、配送税 1)	<ul style="list-style-type: none"> • 「税」ノートブックを使用して、税設定を変更する。 	
taxfulfillment	TAXJCRULE	<ul style="list-style-type: none"> • ストア・アーカイブ XML ファイルを編集する。 • 「税」ノートブックを使用して、税設定を変更する。「税」ノートブックは以下のデータベース列を編集します。 <ul style="list-style-type: none"> - CALRULE_ID - FFMCENTER_ID - JURSTGROUP_ID 	

ストア・データベース資産	データベース・テーブル	ストア・アーカイブ編集オプション	データベース編集オプション
store-catalog-tax	CATENCALCD (オプション)	<ul style="list-style-type: none"> • ストア・アーカイブ XML ファイルを編集する。 	<ul style="list-style-type: none"> • WebSphere Commerce アクセラレーターの商品管理ツール
	CATGPCALCD (オプション)	<ul style="list-style-type: none"> • ストア・アーカイブ XML ファイルを編集する。 	
配送	JURSTGROUP (税と配送で共通)	<ul style="list-style-type: none"> • ストア・アーカイブ XML ファイルを編集する。 • 「配送」ノートブックを使用して配送設定を変更する。「配送」ノートブックは以下のデータベース列を編集します。 <ul style="list-style-type: none"> - JURSTGROUP_ID - DESCRIPTION - SUBCLASS - STOREENT_ID - CODE 	
	JURST (税と配送で共通)	<ul style="list-style-type: none"> • 「配送」ノートブックを使用して配送設定を変更する。「配送」ノートブックは以下のデータベース列を編集します。 <ul style="list-style-type: none"> - JURST_ID - COUNTRY - STOREENT_ID - CODE - SUBCLASS - STATE 	
	JURSTGPREL (税と配送で共通)	<ul style="list-style-type: none"> • 「配送」ノートブックを使用して配送設定を変更する。「配送」ノートブックは以下のデータベース列を編集します。 <ul style="list-style-type: none"> - JURST_ID - JURSTGROUP_ID - SUBCLASS 	
	SHIPMODE	<ul style="list-style-type: none"> • 「配送」ノートブックを使用して配送設定を変更する。「配送」ノートブックは以下のデータベース列を編集します。 <ul style="list-style-type: none"> - CODE - CARRIER - SHIPMODE_ID 	
	SHPMODEDSC	<ul style="list-style-type: none"> • 「配送」ノートブックを使用して配送設定を変更する。「配送」ノートブックは以下のデータベース列を編集します。 <ul style="list-style-type: none"> - SHIPMODE_ID - LANGUAGE_ID 	
	CALMETHOD (税と配送で共通)	<ul style="list-style-type: none"> • ストア・アーカイブ XML ファイルを編集する。 	
	CALCODE (税と配送で共通)	<ul style="list-style-type: none"> • 「配送」ノートブックを使用して配送設定を変更する。「配送」ノートブックは以下のデータベース列を編集します。 <ul style="list-style-type: none"> - CALCODE_ID - CODE - CALUSAGE_ID - STOREENT_ID - GROUPBY - CALMETHOD_ID 	
	CALCODEDSC (オプション)	<ul style="list-style-type: none"> • 「配送」ノートブックを使用して配送設定を変更する。 	

ストア・データベース資産	データベース・テーブル	ストア・アーカイブ編集オプション	データベース編集オプション
	CALRULE (税と配送で共通)	<ul style="list-style-type: none"> 「配送」ノートブックを使用して配送設定を変更する。「配送」ノートブックは以下のデータベース列を編集します。 <ul style="list-style-type: none"> - CALRULE_ID - CALCODE_ID - TAXCGRY_ID - CALMETHOD_ID 	
	CALSCALE (税と配送で共通)	<ul style="list-style-type: none"> 「配送」ノートブックを使用して配送設定を変更する。「配送」ノートブックは以下のデータベース列を編集します。 <ul style="list-style-type: none"> - CALSCALE_ID - CODE - STOREENT_ID - CALUSAGE_ID - SETCURR - CALMETHOD_ID 	
	CALRULEMGP (オプション)	<ul style="list-style-type: none"> ストア・アーカイブ XML ファイルを編集する。 	
	CALSCALEDS (税と配送で共通)	<ul style="list-style-type: none"> ストア・アーカイブ XML ファイルを編集する。 	
	CALRANGE (税と配送で共通)	<ul style="list-style-type: none"> 「配送」ノートブックを使用して配送設定を変更する。「配送」ノートブックは以下のデータベース列を編集します。 <ul style="list-style-type: none"> - CALRANGE_ID - CALSCALE_ID - CALMETHOD_ID 	
	CALRLOOKUP (税と配送で共通)	<ul style="list-style-type: none"> 「配送」ノートブックを使用して配送設定を変更する。「配送」ノートブックは以下のデータベース列を編集します。 <ul style="list-style-type: none"> - CALRLOOKUP_ID - SETCURR - CALRANGE_ID - VALUE 	
	CRULESCALE (税と配送で共通)	<ul style="list-style-type: none"> 「配送」ノートブックを使用して配送設定を変更する。「配送」ノートブックは以下のデータベース列を編集します。 <ul style="list-style-type: none"> - CALSCALE_ID - CALRULE_ID 	
	STENCALUSG (共通、ストアのデフォルト calcode。配送に 1 エントリー caluage、1。)	<ul style="list-style-type: none"> 「配送」ノートブックを使用して配送設定を変更する。 	

ストア・データベース資産	データベース・テーブル	ストア・アーカイブ編集オプション	データベース編集オプション
ship fulfillment	SHPJCRULE (ストアに少なくとも 1 つのデフォルト・ルール)	<ul style="list-style-type: none"> 「配送」ノートブックを使用して配送設定を変更する。「配送」ノートブックは以下のデータベース列を編集します。 <ul style="list-style-type: none"> - CALRULE_ID - SHPARRANGE_ID - JURSTGROUP_ID 	
	SHPARRANGE	<ul style="list-style-type: none"> 「配送」ノートブックを使用して配送設定を変更する。「配送」ノートブックは以下のデータベース列を編集します。 <ul style="list-style-type: none"> - SHARRAND_ID - STORE_ID - FFMCENTER_ID - SHIPMODE_ID 	
store-catalog	STORECAT	<ul style="list-style-type: none"> ストア・アーカイブ XML ファイルを編集する。 	
	STORECENT	<ul style="list-style-type: none"> ストア・アーカイブ XML ファイルを編集する。 	
	STORECGRP	<ul style="list-style-type: none"> ストア・アーカイブ XML ファイルを編集する。 	
	DISPENTREL	<ul style="list-style-type: none"> ストア・アーカイブ XML ファイルを編集する。 	
	DISPCGPREL	<ul style="list-style-type: none"> ストア・アーカイブ XML ファイルを編集する。 	
store fulfillment	INVENTORY	<ul style="list-style-type: none"> ストア・アーカイブ XML ファイルを編集する。 	
offering	TRADEPOSCN (1)	<ul style="list-style-type: none"> ストア・アーカイブ XML ファイルを編集する。 	
	MGTRDPSCN (オプション、顧客グループ用)	<ul style="list-style-type: none"> ストア・アーカイブ XML ファイルを編集する。 	
	OFFER	<ul style="list-style-type: none"> ストア・アーカイブ XML ファイルを編集する。 	<ul style="list-style-type: none"> WebSphere Commerce アクセラレーターの商品管理ツール
	OFFERDESC	<ul style="list-style-type: none"> ストア・アーカイブ XML ファイルを編集する。 	
	OFFERPRICE	<ul style="list-style-type: none"> ストア・アーカイブ XML ファイルを編集する。 	<ul style="list-style-type: none"> WebSphere Commerce アクセラレーターの商品管理ツール
	LISTPRICE	<ul style="list-style-type: none"> ストア・アーカイブ XML ファイルを編集する。 	
command	URLREG	<ul style="list-style-type: none"> ストア・アーカイブ XML ファイルを編集する。 	
	CMDREG	<ul style="list-style-type: none"> ストア・アーカイブ XML ファイルを編集する。 	
	VIEWREG	<ul style="list-style-type: none"> ストア・アーカイブ XML ファイルを編集する。 	
currency	CURCONVERT	<ul style="list-style-type: none"> ストア・アーカイブ XML ファイルを編集する。 	
	CURFORMAT	<ul style="list-style-type: none"> ストア・アーカイブ XML ファイルを編集する。 	
	CURFMTDESC	<ul style="list-style-type: none"> ストア・アーカイブ XML ファイルを編集する。 	
	CURCVLIST	<ul style="list-style-type: none"> ストア・アーカイブ XML ファイルを編集する。 	
	CURLIST	<ul style="list-style-type: none"> 「ストア・プロフィール」ノートブックを使用してストア情報を変更する。「ストア・プロフィール」ノートブックは CURRSTR を編集します。 	
campaign	EMSPOT	<ul style="list-style-type: none"> ストア・アーカイブ XML ファイルを編集する。 	<ul style="list-style-type: none"> WebSphere Commerce アクセラレーターの e-マーケティング・スポット管理ツール
	CAMPAIGN	<ul style="list-style-type: none"> ストア・アーカイブ XML ファイルを編集する。 	<ul style="list-style-type: none"> WebSphere Commerce アクセラレーターのキャンペーン管理ツール

ストア・データベース資産	データベース・テーブル	ストア・アーカイブ編集オプション	データベース編集オプション
store-catalog-shipping	CATENCALCD	• ストア・アーカイブ XML ファイルを編集する。	• WebSphere Commerce アクセラレーターの商品管理ツール
	CATGPCALCD	• ストア・アーカイブ XML ファイルを編集する。	
store-defaults	STOREDEF	• ストア・アーカイブ XML ファイルを編集する。	
consistency check		• ストア・アーカイブ XML ファイルを編集する。	
payment (支払い)	CMDREG	• 支払い設定を変更する。 • ストア・アーカイブ XML ファイルを編集する。	
	VIEWREG	• 支払い設定を変更する。 • ストア・アーカイブ XML ファイルを編集する。	
ビジネス・ポリシー	POLICY	• ストア・アーカイブ XML ファイルを編集する。	
	POLICYCMD	• ストア・アーカイブ XML ファイルを編集する。	
 組織	ORGENTITY	• ストア・アーカイブ XML ファイルを編集する。	
	MBRREL	• ストア・アーカイブ XML ファイルを編集する。	
	ADDRBOOK	• ストア・アーカイブ XML ファイルを編集する。	
	ADDRESS	• ストア・アーカイブ XML ファイルを編集する。	
 ビジネス・アカウント	TERMCOND	ストア・アーカイブを使用してアカウント資産を更新したり再発行したりすることはできません。アカウント資産に変更を加える必要がある場合は、WebSphere Commerce アクセラレーターのビジネス関係管理ツールを使用して、データベース内の資産を編集する必要があります。	WebSphere Commerce アクセラレーターのビジネス関係管理ツール 重要: ローダー・パッケージを使用してアカウントをロードすることはできません。
	ACCOUNT		
	TRADING		
	TCDESC		
	PATTRVALUE		
	CREDITLINE		
	TRDDDESC		
	POLICYTC		
	BUYERPO		
	PARTICIPNT		
	ATTACHMENT		
	TRDATTACH		

ストア・データベース資産	データベース・テーブル	ストア・アーカイブ編集オプション	データベース編集オプション
Business 契約	CONTRACT TERMCOND PRODUCTSET TRADING TCDESC PATTRVALUE TRDDESC POLICYTC PARTICIPNT TRADEPOSCN ATTACHMENT OFFER TRDATTACH OFFERPRICE STORECNTR PURCHASELT STOREDEF (ストア・サービスによって使用される) APRVSTATUS FLINSTANCE	<ul style="list-style-type: none"> ストア・アーカイブ中の契約 XML ファイルを編集する。 	WebSphere Commerce アクセラレーターのビジネス関係管理ツール 重要: ローダー・パッケージを使用して契約をロードすることはできません。

Web 資産の変更

ストア・アーカイブで Web 資産を変更するには、以下のオプションがあります。

- ストア・アーカイブを WebSphere Studio にインポートし、ページ・デザイナーまたは好みのツールを使用して、Web 資産に必要な変更を行う。必要な場合、新規ストア・ページを作成する。それから、Web 資産をストア・アーカイブあるいは実行中のストアに戻してエクスポートします。
- ストア・サービスで「Web 資産」ダイアログを使用して、ストア・アーカイブから任意のロケーションに Web 資産をダウンロードし、その後、任意のツールでこれらを変更する。あるいは、ストア・サービスで「Web 資産」ダイアログを使用して、ストア・アーカイブにある Web 資産を既存の Web 資産に置換する。必要であれば、新しいストア・ページを作成してください。
- ストア・アーカイブにある Web 資産の圧縮アーカイブ・ファイルを手動でオープンし、既存のファイルを変更するか、あるいは新しいファイルを追加する。

ストア・サービスからのサンプル・ストア・アーカイブの発行

サンプル・ストア・アーカイブを WebSphere Commerce サーバーに発行することにより、ストアを実働環境にすることができます。ストア・アーカイブを発行するには、以下のステップを実行してください。

1. サイト管理者またはストア管理者のアクセス権を持っていることを確認する。ストア管理者アクセス権がある場合には、そのアクセス権がすべてのストアに対するものであることを確認してください。
2. 「該当するサービスとサーバーが実行していることを確認」の項目にある作業が完了していることを確認する。
3. ストア・サービスをオープンする。
4. 「ストア・アーカイブ」リストから、発行しようとするストア・アーカイブを選択する。

5. 「発行」をクリックする。
「ストア・アーカイブの発行」ページが表示されます。
6. 希望する発行オプションを選択する。発行オプションの詳細については、「ヘルプ」をクリックしてください。
ヒント: 完全な機能を備えたストアを作成するには、最初にストア・アーカイブを発行するときに、商品データ・オプションを含め、すべての発行オプションを選択してください。
7. **OK** をクリックする。
ストアが発行されている間、「ストア・アーカイブ」をリストしたページに戻ります。発行状態は、「発行」状況列に反映されます。「最新表示」をクリックして、状況を更新します。
8. リストからストア・アーカイブを選択し、「発行の要約」をクリックして、発行結果を表示する。
9. 発行が完了したら「ストアの立ち上げ」をクリックし、ストアを表示してテストする。終了したら、サイトにブックマークを当て、ブラウザをクローズする。

重要:

1. Web アプリケーションの Web パスや Web アプリケーションの文書ルートを変更する場合は、それらが、WebSphere Commerce サーバー内に定義されているパスと一致することを確認する必要があります。
2. 一度には1 つのストア・アーカイブしか発行できません。同時発行はサポートされていません。それは、両方のストアの発行が失敗する原因となります。
3. 発行中に整合性チェッカーは、ストア・アーカイブで参照されるファイルが存在することを確認します。整合性検査でエラーがあると、エラーはログに書き込まれます。発行は正常として継続されません。
4. ストアを再発行する前に次のディレクトリーからファイルを削除してください。

▶ NT

`drive:¥WebSphere¥CommerceServer¥instances¥instancename¥cache`

▶ 2000

`drive:¥Program Files¥WebSphere¥CommerceServer¥instancename¥cache`

▶ AIX

`/usr/WebSphere/CommerceServer/instances/instancename/cache`

▶ SOLARIS ▶ LINUX

`/opt/WebSphere/CommerceServer/instances/instancename/cache`

▶ 400

`/QIBM/UserData/WebCommerce/instances/instancename/cache`

ストア開発フェーズでは、キャッシングを使用不可にしたい場合があります。詳しくは、キャッシングの構成を参照してください。

5. ストア・サービスからストアを立ち上げると、ストア・サービスにログインする時に使ったのと同じユーザー名およびパスワードをもつストアにログインされます。そのストアでパスワードを変更すると、ストア・サービスに対してもパスワードを変更することになります。あるいは、パスワードの変更を含め、ストアのフィーチャーをテストするには、サイト・アドレスを保管し、すべてのブラウザ・ウィンドウをクローズしてから、ストアに再びログオンします。

6. デフォルトの管理者としてログインしている場合は、企業間取り引きサンプル・ストアを基にしたストアをブラウズできません。この場合は、デフォルト組織に属する新規ユーザーを作成してから、ストアをブラウズしてください。
7. カタログや配送センターのような特定のストア・データ資産は、ストア間で共有されます。その結果、同じサンプル・ストアに基づいて複数のストアを発行する場合、カタログおよび配送センターは各ストアとも同じになります。同じサンプルで他のストアを発行した場合には、あるカタログで変更を行うとそれらの変更が上書きされます。詳細情報および、変更の上書きを避ける方法については、共有データ資産を参照してください。

コマンド行を使用したストア・アーカイブの発行 (Windows)

ストア・アーカイブを発行する基本的な方法は、ストア・サービスによる方法ですが、コマンド行を使用してストア・アーカイブを発行することもできます。コマンド行を使用して発行するには、以下のようにします。

1. サイト管理者またはストア管理者のアクセス権を持っていることを確認する。ストア管理者アクセス権がある場合には、そのアクセス権がすべてのストアに対するものであることを確認してください。
2. ストア・アーカイブの有効なパラメーターを使用して以下のコマンドを入力する。

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

ここでは以下のようにになっています。

- `sarName` はストア・アーカイブ名です。 `sarName` は大文字小文字の区別があります。必ず大文字小文字を正しく区別しなければなりません。
- `hostname`。 `hostname`は WebSphere Commerce サーバーの完全修飾 TCP/IP 名です。インスタンスのツール・ポート番号。 ツール・ポート番号は、構成マネージャー (インスタンス・プロパティ > **WebSphere** にある) で見つかります。デフォルトでは、これは `hostname:8000` です。
- `logonId` は WebSphere Commerce ユーザー ID です。
- `logonPwd` は WebSphere Commerce のユーザー・ログオン・パスワードです。
- `insert|update` はストアを作成 (`insert`) か更新 (`update`) するかを決定します。
- `{ALL|NOCATLG}` は XML ファイル内のどのXML ファイルを発行するかを決定します。すべてを発行するには、`ALL` を使用します。カタログ以外をすべて発行するには、`NOCATLG` を使用します。
- `destination1=webapp.zip,destination2=properties.zip`は、SAR 内のファイル資産ファイル (たとえば `webapp.zip`) と、発行先のパスのリストです。発行先のパスは、たとえば次のようなものです。 `destination1` は、次のとおりです。

▶ NT

```
drive:%WebSphere%AppServer%installedApps%
WC_Enterprise_App_instancename.ear%wcstores.war
```

▶ 2000

```
drive:%Program Files%WebSphere%AppServer%installedApps%
WC_Enterprise_App_instancename.ear%wcstores.war
destination2 is
```

▶ NT

```
drive:%WebSphere%AppServer%installedApps%WC_Enterprise_App_demo.ear%
wcstores.war%WEB-INF%classes
```

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

以下はこのコマンドの例です:

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

3. Internet Explorer をオープンし、以下の Web アドレスにアクセスする。

http://hostname/webapp/wcs/stores/store_directory/index.jsp (ここで *store_directory* は、直前に発行したストアのディレクトリーです。)

作成したストアが表示される。

注:バージョン 5.1 で作成されたストア・アーカイブを発行する場合は、URL からストアを立ち上げる前に、以下のステップを実行する必要があります。

1. DB2 を使用している場合の手順は、以下のとおり。Oracle を使用している場合はステップ 2 に進みます。
 - a. コマンドの実行が終わったら、**スタート・メニューからプログラム、DB2 for Windows NT、コマンド・ウィンドウ**の順に選択する。
 - b. **DB2 CLP** ウィンドウで `db2 connect to dbname` と入力する。ここで `dbname` はストアを発行しようとする宛先のデータベースです。 **Enter** を押す。
 - c. コマンド行で、`db2 select * from store`と入力し、 **Enter** を押す。ストアのリストが表示される。作成したストアの番号を書き取る。
 - d. コマンド行で、`db2 select * from catalog`と入力し、 **Enter** を押す。カタログのリストが表示されます。 サンプル・ストアのカタログの番号を書き取る。
 - e. ステップ 3 に進む。
2. Oracle を使用している場合に手順は、以下のとおり。
 - a. コマンドの実行が終わったら、**スタート・メニューからプログラム、Oracle - HomeOra81、アプリケーション開発、SQL Plus** の順に選択する。
 - b. ウィンドウで、`user name`、`password`、および `host string` を入力します。
 - c. 「SQL Plus」ウィンドウで、`select * from store;`と入力し、 **Enter** を押す。ストアのリストが表示される。 作成したストアの番号を書き取る。
 - d. SQL Plus ウィンドウで、`select * from catalog;`と入力し、 **Enter** を押す。 カatalogのリストが表示される。 サンプル・ストアのカタログの番号を書き取る。
 - e. ステップ 3 に進む。
3. Internet Explorer をオープンし、以下の URL にアクセスする。

<http://hostname/webapp/wcs/stores/servlet/StoreCatalogDisplay?storeId=storeId from step1c or 2c&langId=-1&catalogId=catalogId from step1d or 2d>

作成したストアが表示される。

ストアの表示で問題がある場合は、発行に関するトラブルシューティングを参照してください。

重要:

1. 発行中に整合性チェッカーは、ストア・アーカイブで参照されるファイルが存在することを確認します。 整合性検査でエラーがあると、エラーはログに書き込まれます。 発行は正常として継続されません。

2. ストアを再発行する前に次のディレクトリーからファイルを削除してください。

▶ NT

drive:¥WebSphere¥CommerceServer¥instances¥instancename¥cache

▶ 2000

drive:¥Program Files¥WebSphere¥CommerceServer¥instances¥instance name¥cache

3. ストア開発フェーズでは、キャッシング・トリガーを使用不可にします。キャッシュをオンのままにしておくと、以下のような結果になる場合があります。
 - JSP ファイルに加えた変更がブラウザで表示されない。
 - データベースが更新されると、発行中にキャッシング・トリガーが呼び出される。キャッシング・トリガーは、データベース・トランザクション・ログ・オーバーフローを引き起こす可能性のある不必要なデータベース・アクティビティーを生成する場合があります。詳しくは、キャッシングの構成を参照してください。
4. デフォルトの管理者としてログインしている場合は、企業間取り引きサンプル・ストアを基にしたストアをブラウズできません。この場合は、デフォルト組織に属する新規ユーザーを作成してから、ストアをブラウズしてください。

ストアのための Payment Manager のセットアップ

ストアのための Payment Manager のセットアップは、管理コンソールまたは Payment Manager ユーザー・インターフェースを使用して完成することができます。管理コンソールを使用する場合、メニュー項目は **Payment Manager** メニュー上に現れます。Payment Manager ユーザー・インターフェースを使用する場合、メニュー項目は、ナビゲーション・フレームの**管理**の下に現れます。

サンプル・ストア・アーカイブ (これが推奨です) を使用してストアを作成すると、Payment Manager が部分的に構成されます。

ストアのための Payment Manager のセットアップを完了するには、以下のようになります。

1. 管理コンソールまたは Payment Manager ユーザー・インターフェースをオープンする。
2. 必要であれば、Payment Manager ユーザー役割を WebSphere Commerce ユーザーに割り当てる。Payment Manager ユーザー役割を割り当てるには、**ユーザー**を選択します。

デフォルトでは、WebSphere Commerce サイト管理者のデフォルトの wcsadminが、Payment Manager 管理者役割に割り当てられます。別の WebSphere Commerce ユーザーをさまざまな Payment Manager ユーザーに割り当てたくなることもあります。

3. 以下を実行して、ストアのカセットを許可する。
 - a. **マーチャント設定**を選択する。
 - b. ストア名を**マーチャント名**列でクリックする。
 - c. ストアに許可したいカセットを選択する。
 - d. **更新**をクリックする。

重要: 自分のストアを手動で作成した場合、そのストア用のカセットを許可するには、新規マーチャント (自分のストア) を追加する必要があります。新規マーチャントを作成するとき、指定するマーチャント番号は、WebSphere Commerce ストア ID と一致している必要があります。新規マーチャントは、**マーチャント設定**を選択し、**マーチャントの追加**をクリックして作成できます。

4. 以下を実行して、ストアのカセットを構成する。

- a. マーチャント設定を選択する。
- b. 自分のストアの行の、構成したいカセットの列に現れたアイコンをクリックして、構成するカセットを選択する。
- c. ストアのカセットのページで、**アカウント**をクリックして、以下のいずれかを実行する。
 - 既存のアカウントを変更するには、「アカウント名」をクリックする。
 - 新規アカウントを作成するには、**アカウントの追加**をクリックする。


BankServACH のカセットの構成については、*IBM WebSphere Payment Manager for Multiplatforms, Cassette for BankServACH Supplement, Version 3.1* を参照してください。

OfflineCard および CustomOffline カセットの構成の詳細は、*IBM WebSphere Payment Manager for Multiplatforms 管理者ガイド バージョン 3.1* を参照してください。

Cassette for SET の構成の詳細は、*IBM WebSphere Payment Manager for Multiplatforms Cassette for SET 補足 バージョン 3.1* を参照してください。

Cassette for CyberCash の構成の詳細は、*IBM WebSphere Payment Manager for Multiplatforms Cassette for CyberCash 補足 バージョン 3.1* を参照してください。

Cassette for VisaNet の構成の詳細は、*IBM WebSphere Payment Manager for Multiplatforms, Cassette for VisaNet Supplement, Version 3.1* を参照してください。

管理コンソールまたは Payment Manager ユーザー・インターフェースから、Payment Manager を使用しているときにヘルプが必要であれば、使用している Payment Manager のページの右上隅にある  をクリックします。

上記の Payment Manager 管理タスクまたはそれ以外の Payment Manager 管理タスクの詳細は、*IBM WebSphere Payment Manager for Multiplatforms 管理者ガイド バージョン 3.1* を参照してください。


WebSphere Payment Manager のインストールの詳細は、次の資料を参照してください。

- *IBM WebSphere Commerce Business Edition, Installation Guide, Version 5.4*
- *IBM WebSphere Payment Manager for Multiplatforms インストール・ガイド バージョン 3.1.*

サンプル・ストア用のスケジュールされたジョブの作成

サンプル・ストアを発行した後で、スケジュールされたジョブをストア用に作成する必要があります。以下の表では、各サンプル・ストア用に作成しなければならない、スケジュールされたジョブが示されています。

サンプル・ストア	必要なスケジュールされたジョブ
InFashion	<ul style="list-style-type: none"> • BalancePayment • PayCleanup • ReturnCreditAndCloseScan

サンプル・ストア	必要なスケジュールされたジョブ
NewFashion	<ul style="list-style-type: none"> • BalancePayment • PayCleanup • ProcessBackorders • RAReallocate • ReleaseExpiredAllocations • ReleaseToFulfillment • ReturnCreditAndCloseScan
WebFashion	<ul style="list-style-type: none"> • BalancePayment • PayCleanup • ReturnCreditAndCloseScan
 ToolTech	<ul style="list-style-type: none"> • BalancePayment • PayCleanup • ProcessBackorders • RAReallocate • ReleaseExpiredAllocations • ReleaseToFulfillment • ReturnCreditAndCloseScan
WebAuction	<ul style="list-style-type: none"> • BalancePayment • PayCleanup • ReturnCreditAndCloseScan

これらのジョブの簡単な説明を、以下に示します。

BalancePayment

このジョブは、DoDepositCmd タスク・コマンドを呼び出し、オーダーが配送されるたびに支払いを資金化します。このコマンドは、WebSphere Commerce の自動的支払いの資金化機能を実装しています。

PayCleanup

このジョブは、ストア構成期間よりも長い期間それぞれの金融機関によって拒否されている支払い許可要求について、WebSphere Commerce オーダーをキャンセルします。

ProcessBackorders

このジョブは、在庫が使用できないときに作成されたバック・オーダーに在庫を割り振ります。

RAReallocate

(予定在庫に対する再配布の割り振り) このジョブは、既存のバック・オーダーに対して、オープンな予定在庫レコード (EIR) を再配布します。これは、EIR 情報が追加または変更され、すでにバック・オーダーされているアイテムが削除または割り振られたときに、バック・オーダーされたオーダー・アイテムがいつ入手可能になるかを、より正確に予測するために必要です。

ReleaseExpiredAllocations

このジョブは、有効期限の時間制限を超過している、以前に割り振られたオーダー・アイテムから、受け取りテーブルに割り振り済みの在庫を戻します。

ReleaseToFulfillment

このジョブは、オーダー上の割り振られたアイテムを配送にリリースします。

ReturnCreditAndCloseScan

このジョブは、クレジットに適格で、クローズ済みとしてマーク付けされる、返品商品取引権限を走査します。

スケジュールされたジョブを作成する場合、ジョブごとにストア・レベル・ジョブのスケジューリングにある指示に従ってください。以下の表は、ジョブごとの推奨パラメーターをリストしています。

スケジュールされたジョブ	推奨されている開始時刻	推奨されている インターバル (秒単位)	推奨されている優先順位
BalancePayment	00:00	86400	1
PayCleanup	00:00	86400	1
ProcessBackorders	00:00	43200	8
RAREallocate	00:00	86400	1
ReleaseExpiredAllocations	00:00	3600	8
ReleaseToFulfillment	00:00	600	10
ReturnCreditAndCloseScan	00:00	86400	1

開始時刻を 00:00 に設定すると、スケジュールされたジョブがすぐに開始されます。

注: これらのジョブに関して、「スケジュール・ジョブ」ウィンドウのジョブ・パラメーター・フィールドに入力する必要はありません。

サンプル・ストアの E メール通知の構成

以下の手順を使用すると、支払いが許可されたとき、オーダーが許可されたとき、そしてオーダーがキャンセルされたときに、顧客へ E メールで通知できます。顧客へ E メールを送信するには、メール・サーバーをセットアップする必要があることに注意してください。

注: メール・サーバーをセットアップしていない場合、自分のストアから E メール通知を送信できませんが、サンプル・ストアの残りの機能は使用できます。

それぞれのサンプル・ストアは、別々の E メール通知をサポートします。以下の表は、サポートされている E メール通知をストア別に示します。

サンプル・ストア	サポートされている E メール通知	メッセージ・タイプ
InFashion	パスワードのリセット	パスワード・リセットの通知メッセージ
NewFashion	与信済みオーダー	与信済みオーダーの通知メッセージ
	パスワードのリセット	パスワード・リセットの通知メッセージ
	Submission order (送信オーダー)	受け取り済みオーダーの通知メッセージ
	Canceled order (キャンセルされたオーダー)	キャンセルされたオーダーの通知メッセージ
	Shipping notification (配送通知)	顧客にオーダー・リリースの公開を通知するメッセージ
	購入希望リスト	ブロードキャスト・メッセージ

サンプル・ストア	サポートされている E メール通知	メッセージ・タイプ
WebFashion	与信済みオーダー	与信済みオーダーの通知メッセージ
	パスワードのリセット	パスワード・リセットの通知メッセージ
	購入希望リスト	ブロードキャスト・メッセージ

注: WebAuction サンプル・ストアは WebFashion を基にしています。E メール通知をセットアップするには、WebFashion でのすべてのステップと、さらにオークション関連のステップを実行する必要があります。WebAuction での E メール通知のステップの詳細は、下記の『関連タスク』を参照してください。

E メール通知を使用可能にするには、以下のようにします。

1. IBM WebSphere Application Server の管理サーバーが始動していることを確認する。
2. サイト管理者 ID を使用して、管理コンソールをオープンする。
3. 「管理コンソールのサイト/ストアの選択」ページで、「ストア」を選択する。「ストアおよび言語の選択」セクションが表示されます。
4. 「名前」ドロップダウンから、ストアを選択する。
5. 「言語」ドロップダウン・リストから、言語を選択する。「OK」をクリックします。「ストア管理コンソール」ホーム・ページが表示されます。
6. 「構成」メニューから、「トランスポート」をクリックする。「トランスポートの構成」ページが表示されます。
 - a. E メール・トランスポートの状況が「アクティブ」であることを確認する。
E メールが非アクティブの場合、その E メールを選択して、「状況の変更」をクリックします。
 - b. E メールを選択して、「構成」をクリックする。「トランスポート構成パラメーター」ページが表示されます。
 - c. 「ホスト」フィールドに、使用する完全修飾メール・サーバー名を入力する。たとえば、myserver.ibm.com など。
 - d. 「プロトコル」フィールドに、小文字の smtpか、選択したプロトコルを入力する。OK をクリックします。
7. 「構成」メニューから、「メッセージ・タイプ」をクリックする。「メッセージ・タイプ構成」ページが表示されます。
8. 以下のようにして、支払いが許可されたときに送信する通知を作成する。
 - a. 「新規」をクリックする。「メッセージ・トランスポート割り当て」ページが表示されます。
 - b. 「メッセージ・タイプ」ドロップダウン・リストからメッセージ・タイプを選択する。使用するストアで使うメッセージ・タイプについては、上記の表を参照してください。
 - c. 「メッセージ重大度」フィールドに、0 to 0 と入力する。
 - d. 「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。
 - e. 「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。
 - f. 「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。
 - g. 以下のようにしてフィールドを完了させます。

ホスト

使用するメール・サーバーの完全修飾名。たとえば、example.ibm.com。

プロトコル宛先	smtp (小文字を使用のこと) か、使用するプロトコルを入力します。有効な E メール・アドレスを入力します。このアドレスは、実行時には顧客 E メール・アドレスに置き換えられます。
差出人	メッセージの差出人として使用する E メール・アドレスを入力します。たとえば、orders@example.ibm.comなど。このアドレスは、メール・サーバーで有効なユーザーの E メール・アドレスでなければなりません。
件名	メッセージの件名行として表示するテキストを入力します。たとえば、「オーダーをお受けしました」など。

h. 「終了」をクリックする。「メッセージ・タイプ構成」ページが表示されます。

9. サンプル・ストアでのメッセージ・タイプごとに、ステップ 8 を繰り返す。

注:

- キャンセル済みオーダー通知の E メールを送信するまでには、大抵かなりの時間がかかります。以下のようにして、この時間を短縮できます。
 1. STORE データベース・テーブルの REJECTEDORDEREXPIRY カラムの値を、より小さい値に設定する。
 2. PayCleanup スケジュール済みジョブを、スケジュール間隔が短くなるよう変更する。
- 配送通知の詳細は、ReleaseShipNotify メッセージを参照してください。

第 3 章 Sample store database assets

ストア・データベース資産

ストア・データは WebSphere Commerce Server データベースにロードされる情報であり、ストアの機能を可能にするものです。正常に稼働できるようにするには、ストアでデータが適所に置かれて、すべての顧客アクティビティをサポートできるようにしなければなりません。たとえば、顧客が購入できるようにするには、ストアに販売する商品のカタログ (カタログ・データ)、処理するオーダーに関連するデータ (税および配送データ)、そして、要求を実行する在庫 (在庫およびフルフィルメント・データ) を組み込まなければなりません。

データは 1 つのストアに占有させることも、ストア間で共用することもできます。詳細については、共用データ資産を参照してください。

WebSphere Commerce で提供されるサンプル・ストア・アーカイブにあるストア・データベース資産は整形 XML ファイルでありローダー・パッケージで有効です。ただし例外として、ストア・アーカイブ XML ファイルは移植可能であることを意図しているため、データベースの特定インスタンスに固有の生成された 1 次キーはこれに含めません。代わりに、発行時に IDResolver によって解決される内部別名 (ストア・アーカイブ・ローダーの規則を参照) が使用されます。ストア・アーカイブでは一連の DTD マクロ (XML ではエンティティと呼ぶ) も使用します。これらのマクロは、ストア作成時にストア・サービスで選択する値のプレースホルダーとして作用します。次の 2 つの規則に従うと、サンプル・ストア・アーカイブを複数回コピーして発行することができます。

サンプル・ストア・アーカイブには、機能的なストアの作成に必要なデータベース資産がすべて組み込まれています。これらのファイルは、独自のストア・アーカイブで使用するために変更でき、また独自の XML ファイルを作成するガイドとして使用できます。WebSphere Commerce は、特定のデータを WebSphere Commerce データベースにロードして機能的なストアを作成すること、そしてこのデータをスキーマで決定した順序でロードすることを要求します。たとえば、FFMCENTER テーブルは、STOREENT テーブルの前に取り込みを行う必要があります。サンプル・ストアにはすべての必須データが WebSphere Commerce が必要とする順番と構造で組み込まれているため、独自のストアの基礎あるいはガイドとしてデータベース資産を使用することによって、最初の作成期間の相当の時間を節約することができます。

サンプル・ストア・アーカイブで使用されるデータベース資産ファイルのリストについては、サンプル・ストア・アーカイブ・データベース資産を参照してください。ストア・データの詳細は、*IBM WebSphere Commerce ストア開発者ガイド*を参照してください。

注: サンプル・ストア・データベース資産 XML ファイルの DTD は、ストア・アーカイブ・ファイルにありません。これらのファイルは、以下のディレクトリーにあります。

NT

drive:\%WebSphere%\CommerceServer\xml\sar

2000

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

AIX

/usr/WebSphere/CommerceServer/xml/sar

サンプル・ストアのアクセス制御データベース資産

WebSphere Commerce では、すべてのストアにアクセス制御ポリシーがあります。アクセス制御ポリシーは、ユーザーまたはユーザー・グループが特定のアクションを実行する許可を与えます。各ストアごとに、2つのアクセス制御ポリシー・ファイルがあります。

- *samplestorenameAccessPolicies.xml*
- *samplestorenameAccessPolicies_locale.xml*

AccessPolicies.xml および *AccessPolicies_locale.xml* のどちらも、ネイティブの高水準アクセス制御ファイルです。*AccessPolicies.xml* は各国語に依存しないのに対して、*AccessPolicies_locale.xml* は各国語に依存します。どちらのファイルもサンプル・ストアで使用される可能なアクション、アクション・グループ、リソース、およびポリシー定義で構成されています。これらのファイルはそれぞれ、*AccessPoliciesOut.xml* ファイルと *AccessPoliciesOut_locale.xml* ファイルに変換されます。変換されたファイルはそれぞれデータベースに移植されます。アクセス・ポリシー・ファイルの変換に関する詳細は、以下の関連するリンクを参照してください。

注: 変換済みファイルのみが、大量ロードしたり SAR ファイル内で直接使用することができます。変換前のファイルはそうにできません。

サンプル・ストアのためのデータベース資産情報は以下のセクションに分けることができます。

- アクション
- リソース・カテゴリ
- リソース・グループ
- アクション・グループ
- ポリシー定義

アクション

アクセス制御ポリシーのもとで実行が可能なアクションは、各サンプル・ストアの *AccessPolicies.xml* に定義されます。

リソース・カテゴリ

リソース・カテゴリはプロテクト可能なリソースを定義します。

リソース・グループ

リソース・グループは、アクセス制御ポリシーによって統制されるリソースを含みます。リソース・グループには、「契約」または「取引ポジション」などのビジネス・オブジェクトや、それに関連した一連のコマンドを含めることができます。各サンプル・ストアの *AccessPolicies.xml* ファイルは、ポリシー内のリソース・グループを定義します。

アクション・グループ

アクション・グループは、アクセス制御ポリシー内のリソース・グループ上で実行が可能なアクションを定義します。これらのグループは、各ストアの `AccessPolicies.xml` ファイルに定義されます。

ポリシー定義

各サンプル・ストアのポリシーは、個々のストアの `AccessPolicies.xml` ファイルで定義されます。ToolTech サンプル・ストアには 2 つのポリシーがあります。

第 4 章 Sample store pages

サンプル・ストア住所録ページ

サンプル・ストア住所録ページによって、登録済み顧客は配送先および請求先住所を含む住所を住所録に追加できます。

登録済み顧客はログインして、アカウント (`account.jsp`) ページの**住所録の編集**をクリックします。住所録 (`addressbookform.jsp`) ページが表示され、そこから顧客は新しい住所を追加し、あるいは既存の住所を編集することができます。詳細については、住所の新規追加の使用事例および住所の編集の使用事例を参照してください。

サンプル・ストア住所録手順は以下の JSP ファイルを使用します。

- `account.jsp` (アカウント・ページ)
- `addressbookform.jsp` (住所録ページ)
- `addressform.jsp` (`AddressForm` コマンドのパラメーターを含む。顧客には表示されません。)
- `address.jsp`(住所の追加ページおよび住所の更新ページ)

注: `address.jsp` は、住所の追加ページと住所の更新ページの両方で使用されます。 `addressId`が指定されると、`address.jsp` が住所の更新ページとしてロードされます。そうでない場合は、「住所の追加」ページとしてロードされます。 `AddressAdd` コマンドのパラメーターとして `addressId` を指定すると、このコマンドは、指定された `addressId` のアドレスを更新します。そうでない場合には、新しい住所が作成されます。

コマンド

`account.jsp` は、以下のコマンドを使用します。

- `UserRegistrationForm`
- `AddressBookForm`

`addressbook.jsp` は、以下のコマンドを使用します。

- `AddressForm`
- `AddressDelete`

`address.jsp` は、以下のコマンドを使用します。

- `AddressAdd`
- `PrivacyView`

bean

`addressbook.jsp` は、以下の bean を使用します。

- `AddressAccessBean`

`address.jsp` は、以下の bean を使用します。

- `ErrorDataBean`
- `AddressDataBean`

インプリメンテーション詳細情報

注: 多文化情報を含むすべてのサンプル・ストア・ページに共通なインプリメンテーション手法については、サンプル・ストア・ページ: 共通のインプリメンテーション手法を参照してください。

顧客がアカウント・ページの住所録の編集をクリックすると、AddressBookForm コマンドが呼び出されます。次に、AddressBookForm は住所録ページ (addressbook.jsp) をロードします。顧客が新規住所の追加をクリックした場合には、AddressForm コマンドが呼び出されます。AddressForm はデータベース中に AddressForm.jsp と一緒に登録され、page パラメーターがチェックされます。page に newshipaddress を設定した場合には、チェックアウト 1: 請求先住所の追加ページ (billingaddress.jsp) がロードされますが、そうでない場合には、住所の追加ページ (address.jsp) がロードされます。

注: AddressForm は、請求先住所追加時のエラー処理のせいで newshipaddress にセットされる場合、billingaddress.jsp ページをロードします。page が newshipaddress であれば、顧客は、請求先住所ページから新規住所を作成し、配送先住所ページへ移動しようとしたのですが、エラーが生じたことを示しています。そのため、顧客は「請求先住所」ページへ戻されます。

顧客が住所の追加ページ (address.jsp) のフィールドを完了した後で、addressId があるかどうかをチェックします。addressId がある場合には、住所録が更新されますが、ない場合には新しい住所が作成されません。

addressId が指定されると、address.jsp が住所の更新ページとしてロードされます。そうでない場合は、「住所の追加」ページとしてロードされます。「住所の更新」ページでは、入力フィールドには、以下のように、以前に入力した値が入っています。

新しい住所が作成される前に、以下の JavaScript を使用して、住所のニックネーム（時刻と日付を含む住所の固有 ID）が作成されます。

顧客が住所を完了した時には、住所の追加ページ (address.jsp) と住所の更新ページ (address.jsp) の両方で送信をクリックして、AddressAdd コマンドを呼び出します。住所録ページ (addressbook.jsp) は既存の住所を表示します。

エラー処理

顧客が、住所の追加ページ (address.jsp) または住所の更新ページ (address.jsp) のどちらかで、必須フィールドを完成させていない場合には、システムがそのフィールドに再入力するよう要求します。

サンプル・ストア・フッター

サンプル・ストアには各ページの下にフッター (footer.jsp) があり、以下のリンクが含まれています。

- ホーム
- ショッピング・カート
- アカウント
- 問い合わせ先
- プライバシー・ポリシー
- ヘルプ

コマンド

footer.jsp は以下のコマンドを使用します。

- StoreCatalogDisplay

- OrderItemDisplay
- LogonForm
- ContactView
- PrivacyView
- HelpView

bean

footer.jspは以下の bean を使用します。

- UserRegistrationDataBean
- ErrorDataBean

インプリメンテーションの詳細情報

注: 多文化情報を含むすべてのサンプル・ストア・ページに共通なインプリメンテーション手法については、サンプル・ストア・ページ: 共通のインプリメンテーション手法を参照してください。

ショッピング・カート

フッターの **SHOPPING CART** リンクは、 OrderItemDisplay コントローラー・コマンドにリンクし、これは、 OrderItemDisplayViewShiptoAssoc 表示コマンドを戻します。 OrderItemDisplayViewShiptoAssoc 表示コマンドは、データベース中に OrderItemDisplay.jsp と一緒に登録されています。 OrderItemDisplay.jsp は、ショッピング・カート・ページを表示するために shoppingcart.jsp をロードします。

コマンドの詳細については、コマンド、および “WebSphere Commerce プログラマーズ・ガイド” を参照してください。

アカウント

ユーザーが登録済み顧客である場合には、**アカウント**をクリックすると、アカウント・ページ (account.jsp) が表示されます。ユーザーが未登録の顧客である場合には、**アカウント**をクリックすると、登録またはログオン・ページ (myaccount.jsp) が表示されます。これは、以下のコードを使用して実行されます。

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

問い合わせ先

問い合わせ先をクリックすると、ContactView コマンドが呼び出され、これは問い合わせ先ページ (contact.jsp) をロードします。

ヘルプ ヘルプをクリックすると、HelpView コマンドが呼び出され、これは Helpページ (help.jsp) をロードします。

プライバシー・ポリシー

「プライバシー・ポリシー」をクリックすると、PrivacyView コマンドが呼び出され、これはプライバシー・ポリシー・ページ (privacy.jsp) をロードします。

サンプル・ストア・ヘッダー

サンプル・ストアには各ページの上部にヘッダー (header.jsp) があり、以下のリンクが含まれています。

- ショッピング・カート
- アカウント
- 問い合わせ先
- ヘルプ
- メンズ
- レディース
- 新着

コマンド

header.jspは、以下のコマンドを使用します。

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

bean

header.jspは、以下の bean を使用します。

- UserRegistrationDataBean
- CatalogDataBean
- CategoryDataBean

インプリメンテーションの詳細情報

注: 多文化情報を含むすべてのサンプル・ストア・ページに共通なインプリメンテーション手法については、サンプル・ストア・ページ: 共通のインプリメンテーション手法を参照してください。

ショッピング・カート

ヘッダーのショッピング・カート・リンクは、OrderItemDisplay コマンドにリンクし、これは、OrderItemDisplayViewShiptoAssoc 表示コマンドを戻します。 OrderItemDisplayViewShiptoAssoc 表示コマンドは、データベース中に OrderItemDisplay.jsp と一緒に登録されています。 OrderItemDisplay.jsp は、ショッピング・カート・ページを表示するために shoppingcart.jsp をロードします。

アカウント

ユーザーが登録済み顧客である場合には、**アカウント**をクリックすると、アカウント・ページ (account.jsp) が表示されます。ユーザーが未登録の顧客である場合には、**アカウント**をクリックすると、登録またはログオン・ページ (myaccount.jsp) が表示されます。これは、以下のコードを使用して実行されます。

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

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

問い合わせ先

問い合わせ先をクリックすると、ContactView コマンドが呼び出され、これは問い合わせ先ページ (contact.jsp) をロードします。

ヘルプ ヘルプをクリックすると、HelpView コマンドが呼び出され、これは Helpページ (help.jsp) をロードします。

トップレベル・カテゴリー (メンズ、レディース、新着商品)

顧客がヘッダーのトップレベル・カテゴリーの 1 つをクリックした時には、CategoryDisplayコマンドが呼び出されます。CategoryDisplay は、データベース中に CategoryDisplay.jsp と一緒に登録済みです。top パラメーターを Y に設定した時には、以下の例のように topcategory.jsp がロードされ、該当するカテゴリー・ページが以下のとおり表示されます。

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

サンプル・ストア・ヘルプ・ページ

ヘルプ・ページ (help.jsp) は、顧客がヘルプをクリックすると表示されます。

インプリメンテーション詳細情報

注: 多文化情報を含むすべてのサンプル・ストア・ページに共通なインプリメンテーション手法については、サンプル・ストア・ページ: 共通のインプリメンテーション手法を参照してください。

HelpView という新しい表示は、VIEWREG テーブルに作成され、help.jsp に関連付けられます。プライバシー・ページの URL は、http://machine_name/webapp/wcs/v5/stores/HelpView?parameter_listです。

サンプル・ストア・ホーム・ページ

ホーム・ページ (storecatalogdisplay.jsp) は顧客をストアに引き付けるストアフロントとして機能します。サンプル・ストア・ホーム・ページは、メンズおよびレディース用など、ストアのトップレベル・カテゴリーのすべてを表示し、フィーチャー特別ご提供品をいくつかプロモートします。サンプル・ストア・ホーム・ページの詳細については、ホーム・ページの使用事例を参照してください。

コマンド

storecatalogdisplay.jspは次のコマンドを使用します。

- CategoryDisplay
- ProductDisplay

bean

storecatalogdisplay.jspは、次の bean を使用します。

- CatalogDataBean
- CategoryDataBean
- ProductDataBean

- EMarketingSpotBean (WebFashion のみ)

インプリメンテーションの詳細情報

注: 多文化情報を含むすべてのサンプル・ストア・ページに共通なインプリメンテーション手法については、サンプル・ストア・ページ: 共通のインプリメンテーション手法を参照してください。

storecatalogdisplay.jspは、サンプル・ストアのホーム・ページを呼び出す URL を指定している、index.jspページによって立ち上げられます。index.jsp は、ストアを立ち上げるために必要なパラメーターを含む parameters.jsp ファイルを呼び出します。storecatalogdisplay.jsp は、以下を表示します。

トップ・カテゴリー

トップ・カテゴリーは、CATTOGRP テーブルに登録済みです。storecatalogdisplay.jsp は、CatalogDataBean 中の getTopCategories() メソッドでトップ・カテゴリーを検索します。

フィーチャー特別ご提供品

InFashion および NewFashion に含まれるフィーチャー特別ご提供品は、WebSphere Commerce アクセラレーターではなく、カタログ中に作成されています。セールを作成するために、スペシャル・トップレベル・カテゴリーが、ID HOMEPAGE_PROMO としてカタログに追加されています。次に、このカテゴリーに属している商品が、フィーチャー特別ご提供品として表示されます。

WebFashion では、ホーム・ページ上のターゲット商品は、キャンペーンの一部です。キャンペーンは、e-マーケティング・スポットと WebSphere Commerce アクセラレーターを使用して作成されます。ホーム・ページ上の e-マーケティング・スポットを StoreHomePage といい、これは性別の販売促進を作成するのに使用されます。e-マーケティング・スポットをアクティブにするには、WebSphere Commerce アクセラレーターを使用してキャンペーンを作成する必要があります。性別情報を入力していないゲスト顧客と登録済み顧客には、デフォルトの商品 (HOMEPAGE_PROMO) が表示されます。

注:HomePromo カテゴリーまたは新着カテゴリーの商品は、特別なカテゴリーにのみ現れます。販売促進が終了したら、これらの商品を再び正規のカテゴリーへ戻す必要があります。

サンプル・ストア - 左側のナビゲーション・フレーム

サンプル・ストアの、左側のナビゲーション・フレーム (sidebar.jsp) により、顧客はストアを表示する言語を選択することができます。ナビゲーション・フレームには、登録およびヘルプ・ページへのリンクも含まれます。

詳細については、ホーム・ページの使用事例を参照してください。

コマンド

sidebar.jspは、以下のコマンドを使用します。

- StoreCatalogDisplay
- LogonForm
- HelpView

bean

sidebar.jspは、以下の bean を使用します。

- StoreLanguageAccessBean

- LanguageDescriptionAccessBean

インプリメンテーションの詳細情報

注: サンプル・ストアに関する一般的なインプリメンテーション技術の詳細については、多文化的な情報も含めて サンプル・ストアのページ: 共通するインプリメンテーション技術をご覧ください。

顧客が、国 / 地域の選択のドロップダウン・リストから国 / 地域と言語を選択して **GO!** をクリックすると、以下のコードを使用して、languageId に応じてストア・カタログ・ページが再ロードされます。

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

JavaScript コードは、StoreCatalogDisplay コマンドと SetCurrencyPreference コマンドを相互にチェーニングするために、動的に作成されます。このことは、特定の言語をデフォルト通貨にリンクするために行われます。顧客が特定の言語を選択する場合、その言語の通貨を間接的に選択することになるため、結果として、shoppingcart.jsp ページで OrderPrepare コマンドを呼び出す必要はなくなります。これは、以下のコードで実現できます。

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

以下に示すのは、選択した通貨に基づいて特定の言語へリンクする、動的に生成した javascript の一例です。

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

サンプル・ストア・ログイン・ページ

サンプル・ストア・ログイン・ページによって、登録済み顧客はログインすることができます。

登録済み顧客がサイド・バーのリンク「**Register now and receive advance notice of promotions!** (今すぐ登録して販売上の詳しい注意を入手しよう!)」をクリックすると、「登録」または「ログイン」ページ (account.jsp) が表示されます。次に顧客は自分の E メール・アドレスとパスワードを入力して、**ログイン**をクリックします。「アカウント」ページ (myaccount.jsp) が表示されます。詳細については、ログオンの使用事例を参照してください。

サンプル・ストア・ログイン手順は、以下の JSP ファイルを使用します。

- account.jsp (「登録またはログイン」ページ)
- myaccount.jsp (「アカウント」ページ)
- LoginForm.jsp (Logon コマンドのパラメーターを含む。顧客には表示されません。)
- Logoff.jsp (Logoff コマンドのパラメーターを含む。顧客には表示されません。)
- forgetpassword.jsp (「パスワードを忘れました」ページ)
- ResetPasswordForm.jsp(パスワードのリセット・コマンドのパラメーターを含む。顧客には表示されません。)
- password.jsp (顧客に、そのパスワードの設定を示すメッセージを表示します。)

- ChangePasswordForm.jsp (「パスワード変更」ページ)
- ResetPasswordError.jsp(パスワードのリセットの際に問題が生じると呼び出されます。顧客には表示されません。)

コマンド

account.jspは、以下のコマンドを使用します。

- ログオン
- ログオフ

myaccount.jspは、以下のコマンドを使用します。

- UserRegistrationForm (InFashion、WebFashion、NewFashion、および WebAuction)
- AddressBookForm (InFashion、WebFashion、NewFashion、および WebAuction)
- InterestItemDisplay (WebFashion、NewFashion、および WebAuction)
- ProfileFormView (WebFashion および WebAuction)
- TrackOrderStatus (WebFashion、NewFashion、および WebAuction)

forgetpassword.jspは、以下のコマンドを使用します。

- ログオフ
- ResetPassword

ChangePasswordForm.jspは、以下のコマンドを使用します。

- ResetPassword

forgetpassword_err.jspは、以下のコマンドを使用します。

- ResetPassword
- ログオフ

password.jsp は、以下のコマンドを使用します。

- LogonForm

bean

forgetpassword.jspは、以下の bean を使用します。

- ErrorDataBean

forgetpassword_err.jspは、以下の bean を使用します。

- ErrorDataBean

インプリメンテーションの詳細情報

注: 多文化情報を含むすべてのサンプル・ストア・ページに共通なインプリメンテーション手法については、サンプル・ストア・ページ: 共通のインプリメンテーション手法を参照してください。

顧客が登録ページで登録を行なったり、ログイン・ページ (account.jsp) に E メール・アドレスとパスワードを入力したりした後は、以下のコードによって値が小文字に変換されます。

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

また、account.jsp は、Logon コマンドに必要なフィールドを設定します。たとえば、

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

顧客が登録またはログオン・ページで「ログイン」をクリックした時には、Logon コマンドが呼び出されます。Logon は、データベース中に LoginForm.jsp と一緒に登録済みです。LoginForm.jsp は、page パラメーターを使用して、どのページ (myaccount.jsp または account.jsp) をロードするかを判別します。

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

正しい E メール・アドレスおよびパスワードの組み合わせが入力された場合には、LoginForm.jsp がアカウント・ページ (account.jsp) をロードします。正しくない E メール・アドレスおよびパスワードの組み合わせが入力した場合には、LoginForm.jsp が登録またはログオン・ページ (account.jsp) を再ロードします。

顧客が自分のパスワードを忘れた場合は、パスワードをお忘れですか? をクリックします。Logoff コマンドが state=forgetpassword パラメーターとともに呼び出されます。Logoff コマンドは、データベース中に Logoff.jsp と一緒に登録済みです。Logoff.jsp は、下の説明のように、パラメーターの状態を検査します。

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

if (state == null || state.length() == 0)
{
incfile = "/" + storeDir + "/UserRegistrationForm.jsp";
}
else if (state.equals("forgetpassword"))
{
incfile = "/" + storeDir + "/forgetpassword.jsp";
}
}
```

状態が forgetpassword と等しい場合には、「パスワードを忘れました」ページ (forgetpassword.jsp) がロードされます。顧客がページのフィールドを完了し、「パスワードの送信」をクリックした時には、ResetPassword コマンドが呼び出されます。データベースで旧パスワードが「有効期限切れ」に設定され

ると、顧客に新規パスワードが E メールで送信されます。新規パスワードを使用して顧客がログインすると、パスワードの変更が要求され、パスワード変更ページ (ChangePasswordForm.jsp) が表示されます。

注: 顧客のパスワードが有効納期切れに設定されると、次のログイン時に自動的に「パスワード変更」ページが表示されます。

エラー処理

顧客が正しくない E メール・アドレスまたはパスワードを指定した場合、あるいはフィールドが未完了の場合には、エラー・メッセージが表示され、Logon コマンドが page パラメーターの設定なしで登録またはログインを再ロードします。誤ったパスワードを入力した場合、顧客はログインするまで 2、3 秒待機する必要があるか、以下のエラーが表示されます。

Please wait a few seconds before attempting to login again. (再ログインするまで 2、3 秒お待ちください。)

サンプル・ストア新着商品ページ

顧客が**新着商品**をクリックした時には、新着商品ページ (newarrivals.jsp) が表示されます。

コマンド

newarrivals.jspは、以下のコマンドを使用します。

- ProductDisplay

bean

newarrivals.jspは、以下の bean を使用します。

- CategoryDataBean
- ProductDataBean
- CatalogDataBean
- EMarketingSpotBean

インプリメンテーションの詳細情報

注: 多文化情報を含むすべてのサンプル・ストア・ページに共通なインプリメンテーション手法については、サンプル・ストア・ページ: 共通のインプリメンテーション手法を参照してください。

新着商品ページ (newarrivals.jsp)は、プロモートしたアイテムのリスト表示を意味します。 InFashion に含まれる新着商品は、カタログ内で作成されたものです。結果として、新着商品と呼ばれるトップレベル・カテゴリが作成され、プロモートされた商品のすべてがこのカテゴリに追加されます。 InFashion が、他のトップレベル・カテゴリとは別の方法でこのカテゴリのコンテンツを表示するので、newarrivals.jsp ファイルは、このカテゴリを表示するためにデータベース (DISPCGPREL テーブル) 中に登録されています。このカテゴリ ID によって CategoryDisplay コマンドが呼び出された時には、newarrivals.jsp がロードされ、新着商品として指定された商品を表示します。

newarrivals.jspページは、CategoryDataBean を使用してカテゴリ情報を表示します。 NewFashion および InFashion では、このページの商品は「ホット情報」という ID の付いた特別なカテゴリからのものです。販売促進商品のリストが CategoryDataBean の getProducts() メソッドを使用して検索されます。

WebFashion では、新着商品ページにホット・セール・キャンペーンが表示されます。このキャンペーンの目的は、シーズンのトップ・トレンドを販売することです。ホット・セール・キャンペーンでは、お勧め商品提示商法のイニシアチブを使って、登録済みの顧客をターゲットにして顧客の年齢に基づいてアイテムを提案します。登録済み顧客が 29 歳を超えていると、1 セット 3 商品が表示され、29 歳以下の顧客には別のセットが表示されます。年齢情報を入力していないゲスト顧客と登録済み顧客には、デフォルトの商品 (ホット情報) が表示されます。

ホット・セール・キャンペーンは、WebSphere Commerce アクセラレーターおよび e-マーケティング・スポットを使用して作成されます。e-マーケティング・スポットは、NewArrivalsPage と呼ばれ、新着商品ページに位置しています。e-マーケティング・スポットをアクティブにするには、WebSphere Commerce アクセラレーター内にキャンペーンをセットアップする必要があります。

サンプル・ストア・オーダー要約ページ

サンプル・ストアのチェックアウト処理の 4 番目のステップ「チェックアウト 4 オーダー要約ページ (orderdisplaypending.jsp)」では、顧客が、数量、単価と合計価格、配送先住所、および配送料金の他に、購買アイテムの説明を含む詳細オーダー情報を確認することができます。次に、顧客は支払い情報を指定し、**オーダー**をクリックすることによって、オーダーを完了する必要があります。NewFashion ストアでは、予定配送日が表示されます。

詳細については、ショッピング・カートのチェックアウトの使用事例を参照してください。

コマンド

orderdisplaypending.jsp は、以下のコマンドを使用します。

- OrderProcess
- PrivacyView
- ContactView
- MultiOrderProcess (NewFashion のみ) (MultiOrderProcess は MultiOrderProcess.jsp に関連付けられている)

bean

orderdisplaypending.jsp は、以下の bean を使用します。

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

インプリメンテーションの詳細情報

NewFashion のみ

顧客が「**オーダー**」をクリックすると、MultiOrderProcess コマンドが呼び出されます。

MultiOrderProcess は、VIEWREG テーブルに登録されている表示コマンドで、MultiOrderProcess.jsp と関連付けられています。

MultiOrderProcess.jspは、オーダー要約ページのオーダー数に応じて、OrderProcess を複数回実行します。NewFashion では、チェックアウト・フローは、オーダー要約ページ内のオーダーを 2 つまでしか許可していません。

注: 多文化情報を含むすべてのサンプル・ストア・ページに共通なインプリメンテーション手法については、サンプル・ストア・ページ: 共通のインプリメンテーション手法を参照してください。

すべてのサンプル・ストア

顧客は、ショッピング・カート・ページの**チェックアウト**をクリックすることによって、チェックアウト処理を開始すると、一連のチェックアウト・ページを移動しますが、その 4 番目のページがチェックアウト 4 オーダー要約ページ (orderdisplaypending.jsp) です。このページには、顧客がクレジット・カード情報を送信できるフォームが入っています。UsablePaymentTCListDataBean を使用して、Payment Manager から利用可能なクレジット・カード名が入手され、フォームのアクションは、OrderProcess に設定されます。フォームが送信された後で、オーダー処理が正常に実行されると、OrderOKView が呼び出されます。OrderOKView コマンドは、データベース中の VIEWREG テーブルに登録されていますが、これは、オーダーを確認するための情報を表示する confirmation.jsp と関連付けられています。NewFashion ストアでは、2 つのオーダーが処理されると、両方のオーダーの情報が確認ページに表示されます。

エラーがある場合には、DoPaymentErrorView が呼び出されます。DoPaymentErrorView は、データベース中に OrderDisplayPending.jsp と関連付けられています。結果として、エラーがある時には、チェックアウト 4 オーダー要約ページ (orderdisplaypending.jsp) がエラー・メッセージと一緒に再表示されます。

サンプル・ストア・プライバシー・ページ

顧客が**プライバシー・ポリシー**をクリックすると、プライバシー・ページ (privacy.jsp) が表示されます。

インプリメンテーションの詳細情報

注: 多文化情報を含むすべてのサンプル・ストア・ページに共通なインプリメンテーション手法については、サンプル・ストア・ページ: 共通のインプリメンテーション手法を参照してください。

PrivacyView と呼ばれる新しい表示は、privacy.jsp と関連付けられた VIEWREG テーブルに作成されます。プライバシー・ページを表示する URL は、
`http://machine_name/webapp/wcs/v5/stores/PrivacyView?parameter_list` です。

注: 独自のプライバシー・ポリシーを作成して、それをオンライン・ストアに組み込む必要があります。詳しくは、IBM のプライバシー・サイト <http://www.ibm.com/privacy/> を参照してください。

サンプル・ストア商品ページ

商品ページは、オンライン・ストア内の 1 つの特定商品の特徴を示します。ここには通常は、説明、価格、およびイメージが含まれており、その商品に属性 (たとえば、異なるサイズやカラー) がある場合、顧客は属性を選択することができます。

商品ページとその機能の詳細については、商品表示ページの使用事例を参照してください。

コマンド

productdisplay.jspは、以下のコマンドを使用します。

- OrderItemAdd
- InterestItemAdd (WebFashion および NewFashion のみ)

bean

productdisplay.jspは、以下の bean を使用します。

- CategoryDataBean
- ProductDataBean

インプリメンテーションの詳細情報

注: 多文化情報を含むすべてのサンプル・ストア・ページに共通なインプリメンテーション手法については、サンプル・ストア・ページ: 共通のインプリメンテーション手法を参照してください。

ProductDisplay.jsp は、データベース (DISPENTREL テーブルに) 中に登録され、ストアにすべての商品を表示します。 ProductDisplay.jsp以下のとおり表示されます。

- 説明、イメージ、属性、および商品の属性値
- 親カテゴリの簡略説明
- 「ショッピング・カートに追加」リンク
- 「数量」テキスト・ボックス (NewFashion のみ)
- 「購入希望商品リストに追加」リンク (WebFashion および NewFashion のみ)

説明、イメージ、属性、および商品の属性値

商品説明およびイメージは、ProductDataBean プロパティを使用して表示されます。

商品の属性は、ProductDataBean の getAttributes() メソッドを使用して検索されます。各属性の値は、AttributeAccessBean 中の getDistinctAttributeValues() メソッドを使用して検索されます。ProductDataBean がこの列から情報を検索して見つけた場合、「商品表示」ページには完全イメージの代わりに Hotmedia イメージが表示されます。

親カテゴリの簡略説明

親カテゴリの ID は、parent_category_rn パラメーターによって商品ページに提供されます。親カテゴリの簡略説明は、CategoryDataBean によって検索されます。デフォルトでは、CategoryDataBean は categoryIdパラメーターからカテゴリ ID を入手します。以下の例では、パラメーター名は parent_category_rn であり、カテゴリ ID は明示的に設定されます。

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

ショッピング・カートに追加

ショッピング・カートに追加リンクは、OrderItemAdd コマンドを呼び出すフォームを作成することによってインプリメントされます。InFashion と WebFashionでは、オーダーされた商品の数量は、以下のように隠しフィールドを使用して、デフォルトで 1 に設定されます。

```
<input type="hidden" name="quantity" value="1">
```

顧客が別の数量を入力できるように、テキスト・ボックスで潜在フィールドを置き換えることができます。

NewFashion ストアでは、オーダーされた商品の数量は、以下のようにテキスト・フィールドを使用し、デフォルトで 1 に設定されます。

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

このテキスト・フィールドによって、顧客は異なる数量を入力することができます。

ショッピング・カートに追加 および購入希望商品リストに追加

顧客がショッピング・カートに追加または購入希望商品リストに追加を選択すると、以下の javascript が呼び出されます。

```
<SCRIPT language="javascript">
  function Add2ShopCart(form){
    form.action='OrderItemAdd'
    form.URL.value='OrderItemDisplay'
    form.submit()
  }
  function Add2WishList(form){
    form.action='InterestItemAdd'
    form.URL.value='InterestItemDisplay'
    form.submit()
  }
</SCRIPT>
```

顧客がショッピング・カートに追加した場合は OrderItemAdd コマンドが呼び出されます。顧客が購入希望商品リストに追加した場合は InterestItemAdd コマンドが呼び出されます。

注:WebSphere Commerce アクセラレーターを使用して商品を作成することができます。商品を作成するとき、サンプル・ストア商品ページで価格を表示させるために、商品価格を作成する必要があります。価格を作成しない場合、ProductDataBean の getCalculatedContractPrice メソッドを削除しなければなりません。

サンプル・ストア登録ページ

サンプル・ストア登録ページでは、顧客がサンプル・ストアに登録することができます。顧客が登録するには、氏名および E メール・アドレスを入力してから、パスワードを作成する必要があります。

顧客が**登録して購入する!**をクリックすると、登録またはログオン・ページが表示されます。次に、顧客が**登録**をクリックすると、登録ページ (register.jsp) が表示されます。

顧客が自分の登録情報を更新しようとする場合は、「アカウント」ページにある「**個人情報の変更**」をクリックします。個人情報の変更ページ (edit_registration.jsp) が表示されます。

コマンド

register.jspは、以下のコマンドを使用します。

- UserRegistrationAdd
- PrivacyView

bean

register.jspは、以下の bean を使用します。

- ErrorDataBean

edit_registration.jsp は、以下の bean を使用します。

- DemographicsAccessBean

- UserRegistrationDataBean
- ErrorDataBean

インプリメンテーションの詳細情報

注: 多文化情報を含むすべてのサンプル・ストア・ページに共通なインプリメンテーション手法については、サンプル・ストア・ページ: 共通のインプリメンテーション手法を参照してください。

登録 「登録」または「ログイン」ページ (account.jsp) からの「登録」リンクは、登録フォームを表示します。 WebSphere Commerce は UserRegistrationAdd コマンドを使用して、新規登録を作成します。 顧客がログインする場合、UserRegistrationAdd は、 UserRegistrationUpdate と同様の動作をします。 すなわち、ログインした顧客は再度登録し新規アカウントを作成することができません。 サンプル・ストアではユーザーの複数回の登録と複数のアカウントの作成が可能のため、ログインした顧客は、再度登録をする前に、ログオフする必要があります。 このために、ログインした顧客にはログオフ・コマンド、ゲスト顧客には UserRegistrationForm ビューへの登録リンクが自動的に設定されます。 ログオフ・コマンドは、自動的に LogoffView タスクを呼び出します。 LogoffView タスクは、データベース中に Logoff.jsp と一緒に登録されています。 Logoff.jsp は、state URL パラメーターを検査します。 それが forgetpassword に設定されていれば、forgetpassword.jsp がロードされ、そうでない場合は、UserRegistrationForm.jsp がロードされます。 UserRegistrationForm ビューは、VIEWREG テーブル内の UserRegistrationForm.jsp ファイルに関連付けられています。 UserRegistrationForm.jsp は “new” URL パラメーターを検査します。 それが Y に設定されていれば、register.jsp を組み込んで、新規登録フォームが表示されます。 そうでない場合は、edit_registration.jsp を組み込んで、更新登録フォームが表示されます。

register.jspには、新規登録フォームが含まれます。 新規登録フォームは UserRegistrationAdd コマンドにより送信されます。 デフォルトでは UserRegistrationAdd コマンドは、サンプル・ストアでは不要な、いくつかの必須フィールドが必要です。 そのため、これらのフィールドは隠し HTML フィールドとしてセットアップされ、以下の例のように、その値は "-"に設定されます。

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

注:NewFashion ストアでは、名前値のペアは送信されません。 名前値のペアが送信されないと UserRegistrationAdd コマンドは URL パラメーターをチェックしません。 顧客が自分のプロフィールを登録して記入したら、 WebSphere Commerce アクセラレーター-のいずれかのプロフィールと一致させられます。 WebSphere Commerce アクセラレーター・プロフィールに、関連付けられた割引がある場合、顧客はその割引を受け取ります。

E メール・アドレスでログイン

サンプル・ストアでは顧客はログイン ID として E メール・アドレスを入力する必要がありますが、 WebSphere Commerce では、ログインとしてユーザー ID が必要です。 ソリューションとして、サンプル・ストアは登録フォームで顧客に E メール・アドレスの入力を要求します。 そのためフォームを送信する前に、以下の JavaScript を使用して E メール・アドレスの値に E メールとユーザー ID フィールドの両方をセットします。

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

```

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

```

個人情報の変更

個人情報の変更をクリックすると UserRegistrationForm コマンドが呼び出されますが、これは、データベース中に UserRegistrationForm.jspと関連付けられます。新規 パラメーターが非ヌルの場合は、register.jspがロードされます。そうでない場合は、edit_registration.jsp がロードされます。

UserRegistrationUpdate コマンドは、ユーザーの登録情報を更新するために edit_registration.jsp で使用されます。顧客がパスワードまたは確認パスワード・フィールドを完了していない場合には、以下のコードを使用して、システムが登録パスワードを提供します。

```

function prepareSubmit (フォーム)
{
form.<%= ECUserConstants.EC_ADDR_EMAIL1
%>.value=form.<%=ECUserConstants.EC_UREG_LOGONID%>.value
form.<%=ECUserConstants.EC_UREG_LOGONID%>.value =
form.<%= ECUserConstants.EC_UREG_LOGONID%>.value.toLowerCase()
if (form.<%=ECUserConstants.EC_UREG_LOGONPASSWORD%>.value.length == 0)
{
form.<%=ECUserConstants.EC_UREG_LOGONPASSWORD%>.value = '<%=strPassword%>'
}
if (form.<%=ECUserConstants.EC_UREG_LOGONPASSWORDVERIFY%>.value.length == 0)
{
form.<%=ECUserConstants.EC_UREG_LOGONPASSWORDVERIFY%>.value =
'<%=strPassword%>'
}
form.submit()
}

```

顧客が以前に性別と年齢の情報を入力している場合は、性別および年齢フィールドは事前に埋められます。DemographicsAccessBean がデータベースから性別および年齢の情報を抽出します。

エラー処理

どのようなエラーでも UserRegistrationAdd コマンドは、UserRegistrationErrorView を呼び出しますが、これはデータベースの UserRegistrationForm.jsp に登録されています。しかし、同じエラーが UserRegistrationUpdate によっても呼び出されます。新規登録フォームと編集フォームを区別するため、新規登録フォームには、newという隠しパラメーターが含まれています。そのパラメーターがあると UserRegistrationForm.jsp は、register.jspを組み込みます。そうでない場合は、edit_registration.jsp がロードされます。

register.jsp は、通常時、エラー発生時の両方に使用されます。ErrorDataBean およびエラー検査は、どのような状況下で register.jsp が実行されているかを判別するために使用されます。エラーが発生した場合、register.jsp は、エラー・メッセージを表示します。

注: パスワード・エラーがあると、AuthenticationPolicyErrorView が呼び出されますが、実際には UserRegistrationAdd および UserRegistrationUpdate コマンドで捕そくされます。

サンプル・ストア・ショッピング・カート

顧客は、ショッピング・カートの表示の使用事例で説明されているように、ショッピング・カート (shoppingcart.jsp) で選択済みのアイテムを表示、編集することができます。

コマンド

shoppingcart.jspは、以下のコマンドを使用します。

- OrderItemUpdate
- OrderItemDelete
- StoreCatalogDisplay
- QuickCheckoutView

bean

shoppingcart.jspは、以下の bean を使用します。

- OrderDataBean

インプリメンテーションの詳細情報

注:サンプル・ストアに関する一般的なインプリメンテーション技術の詳細については、多文化的な情報も含めて サンプル・ストアのページ: 共通するインプリメンテーション技術をご覧ください。

顧客がヘッダーまたはフッターのショッピング・カートをクリックすると、OrderItemDisplay コマンドが呼び出されますが、これは、OrderItemDisplayViewShiptoAssoc 表示コマンドを戻します。

OrderItemDisplayViewShiptoAssoc 表示コマンドは、データベース中に OrderItemDisplay.jspと一緒に登録されています。OrderItemDisplay.jsp は、pageパラメーターに基づいて、別の JavaServer Page ファイルを組み込みます。page の値が入っていない場合は、ショッピング・カート・ページ (shoppingcart.jsp) がロードされます。

注: OrderItemDisplay.jsp は OrderPrepare を実行します。これはオーダー合計価格を再計算し、デフォルトの通貨を顧客が使用している通貨に変換します。単一通貨のストアでは OrderPrepare を実行する必要はありません。

ショッピング・カート・ページには、**合計の更新**ボタン、および **チェックアウト**・ボタンがあります。**合計の更新**をクリックし、オーダー済みアイテムの数量更新してから、ショッピング・カート・ページを表示します。**チェックアウト**をクリックし、オーダー済みアイテムの数量更新してから、チェックアウト 1 請求先住所の選択ページを表示します。

チェックアウトおよび**合計の更新**ボタンはどちらも、同じ HTML フォームを使用します。ただし、顧客がフォームを送信する前に**チェックアウト**をクリックすると JavaScript が使用されて、値が billingaddress に設定された page パラメーターがさらに追加されます。

クイック・チェックアウト・オプションを完了する際に使用されるコマンドは QuickCheckoutView で、VIEWREG に登録されている表示コマンドで、QuickCheckout.jspに関連付けられています。

QuickCheckout.jsp は、以下のサーバー・コマンドを実行します。

- OrderItemUpdate (配送先住所の更新)
- OrderItemUpdate (配送モードの更新)
- OrderCopy (請求先住所および支払い情報の更新)

- OrderPrepare

QuickCheckout.jspは、OrderAccessBean を使用して、請求先および配送先住所、配送方法、および支払い情報を顧客のクイック・チェックアウト・プロフィールから検索します。次に、この情報を orderId で指定されたオーダーに割り当て、OrderPrepare コマンドを実行します。

クイック・チェックアウト処理が完了した後、このコマンドは URL で指定されたビューに転送されません。WebFashion では、QuickCheckoutSummaryView が QuickCheckout コマンドの URL として指定されています。そのため、クイック・チェックアウトが完了すると、「クイック・チェックアウトの要約」ページが表示されます。

エラー処理

ストアに配送センターが接続されていない場合、または商品が在庫切れの場合、OrderItemAdd/OrderItemUpdate コマンドが ResolveFulfillmentCenterErrorView を呼び出します。これはデータベース中に shoppingcart.jsp で登録されています。顧客が数量フィールドに無効な文字を入力する場合、InvalidInputErrorView が呼び出されます。InvalidInputErrorView も、データベースの shoppingcart.jsp に登録されています。

OrderItemUpdate コマンドと OrderItemAdd コマンドの両方によってエラーが起こされる可能性があるため、shoppingcart.jspは、どちらのコマンドがエラーを起こしたかを確認し、それによってエラーを表示します。最後のコマンドが OrderItemUpdate の場合、「ショッピング・カート」ページが再表示されて、エラー・メッセージが示されます。そうではない場合、個別のエラー・ページが表示されて、エラー・メッセージが示されます。これは以下のコードで実現させることができます。

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

shoppingcart.jspは、通常時、エラー発生時の両方に使用されます。ErrorDataBean およびエラー検査は、どの状態で shoppingcart.jspが表示されるかを決定します。エラーがある場合、shoppingcart.jsp は該当するエラー・メッセージを表示します。

QuickCheckout.jsp(QuickCheckoutView) がクイック・チェックアウト・プロフィールを検索できない場合、QuickCheckoutError.jspを呼び出してエラー・メッセージを表示します。その結果として、quickcheckouterrorview.jsp が表示されます。

quickcheckouterrorview.jspは、顧客が登録済みであるかどうかを検査します。顧客が登録済みである場合は、システムにより顧客にクイック・チェックアウト・プロフィールの作成を促すプロンプトが出されません。そうでない場合は、システムにより顧客にまず登録し、それからクイック・チェックアウト・プロフィールを作成するよう促すプロンプトが出されます。

サンプル・ストア請求先住所の選択ページ

サンプル・ストア・チェックアウト処理の最初のステップ「チェックアウト 1 請求先住所の選択ページ (billingaddress.jsp)」では、顧客は請求先住所として既存の住所を選択するか、あるいは新しい住所を作成して請求先住所として使用することができます。

詳細については、ショッピング・カートのチェックアウトの使用事例を参照してください。

コマンド

billingaddress.jspは、以下のコマンドを使用します。

- アドレス・フォーム
- OrderItemDisplay
- OrderCopy
- AddBillAddressView

bean

billingaddress.jspは、以下の bean を使用します。

- OrderDataBean
- AddressAccessBean
- OrderItemAccessBean
- ErrorDataBean

インプリメンテーションの詳細情報

注: 多文化情報を含むすべてのサンプル・ストア・ページに共通なインプリメンテーション手法については、サンプル・ストア・ページ: 共通のインプリメンテーション手法を参照してください。

顧客は、ショッピング・カート・ページの**チェックアウト**をクリックすることによって、チェックアウト処理を開始すると、一連のチェックアウト・ページを移動しますが、その最初のページがチェックアウト 1 請求先住所の追加ページ (billingaddress.jsp) です。 billingaddress.jsp は、顧客が住所録に既存の住所を持っているかどうかを検査します。 現在、住所が住所録にある場合には、住所が表示され、顧客は請求先住所としてこれを選択することができます。

また、顧客は、**新規住所の作成**をクリックすることによって、新しい住所を作成することができます。**新規住所の作成**をクリックすると、AddressForm コマンドが呼び出されますが、これは、データベース中に AddressForm.jsp と関連付けられます。 AddressForm.jspは、「住所の追加」ページをロードする address.jsp を呼び出します。 address.jsp は、ロードする次のページを判別するためにページパラメータを検査します。 ページの値が billingaddressに設定されると、 AddressAdd フォームの URL 値は、OrderItemDisplay に設定されます。 OrderItemDisplay は、billingaddress.jsp を呼び出しますが、これによって、顧客が「送信」をクリックすると、チェックアウト 1 請求先住所の選択ページに戻され、ロードする住所フォームを判別するために、 ページパラメーターがチェックされます。

どの住所も住所録にない場合には、請求先住所の追加フォームが表示され、顧客に新しい住所を入力するようプロンプトが出されます。 また、この場合の請求先住所の追加フォームは、billingaddress.jspにより生成されます。この HTML フォームのアクションは、AddBillAddressView に設定されますが、これは、VIEWREG テーブルに登録されています。 AddBillAddressView は、AddBillAddress.jspと関連付けられています。 AddBillAddressView のフォームをサブミットすると、AddBillAddress.jsp が呼び出されます。

AddBillAddress.jspは、以下のコマンドを実行します。

- AddressAdd
- OrderCopy

AddressAdd を実行した後で、AddBillAddress.jsp は、OrderCopy コマンドへの入力の一部として AddressAdd により、戻されたアドレス ID を使用します。 次に、OrderCopy は、現行オーダーの請求先住所にアドレス ID を割り当て、OrderItemDisplay.jsp を呼び出します。 ページ パラメーターの値が、newshipaddressに設定されているので、 OrderItemDisplay.jsp が shipaddress.jsp を呼び出します。

注:WebSphere Commerce は、登録時に住所の作成を必要とします。サンプル・ストアは、顧客登録時に住所を必要としないので、住所 1 などの必須フィールドのいくつかは unused に設定されます。住所を検査するとき、**billingaddress.jsp** は、住所 1 の値が unused であるかどうか検査します。その場合には、住所が表示されません。

サンプル・ストア「配送先住所の選択」ページ

サンプル・ストアのチェックアウト処理の 2 番目のステップ、「チェックアウト 2 配送先住所の選択ページ (shipaddress.jsp)」では、顧客は配送先住所として既存の住所を選択する、その住所を編集する、または配送先住所として使用する新しい住所を作成することができます。

詳細については、ショッピング・カートのチェックアウトの使用事例を参照してください。

コマンド

shipaddress.jsp は、以下のコマンドを使用します。

- OrderItemUpdate
- アドレス・フォーム
- OrderItemDisplay

bean

shipaddress.jsp は、以下のコマンドを使用します。

- OrderBean
- AddressAccessBean
- OrderItemDataBean

インプリメンテーションの詳細情報

注:サンプル・ストアに関する一般的なインプリメンテーション技術の詳細については、多文化的な情報も含めて サンプル・ストアのページ: 共通するインプリメンテーション技術をご覧ください。

顧客は、ショッピング・カート・ページの**チェックアウト**をクリックすることによって、チェックアウト処理を開始すると、一連のチェックアウト・ページを移動します。その 2 番目のページがチェックアウト 2 配送先住所の選択ページ (shipaddress.jsp) です。shipaddress.jsp は、すべての既存の住所を表示し、これによって顧客が配送先住所として選択することができます。

また、顧客は、**新規住所の作成**をクリックすることによって、新しい住所を作成することができます。**新規住所の作成**をクリックすると、AddressForm コマンドが呼び出されますが、これは、データベース中に AddressForm.jsp と関連付けられます。AddressForm.jsp は、住所の追加ページをロードする address.jsp を呼び出します。address.jsp は、ロードする次のページを判別するためにページパラメーターを検査します。ページの値が shipaddress に設定されると、AddressAdd フォームの URL 値は、OrderItemDisplay に設定されます。OrderItemDisplay は、shipaddress.jsp を呼び出しますが、これによって、顧客が**送信**をクリックすると、チェックアウト 2 配送先住所の選択ページに戻されます。

注:WebSphere Commerce は、登録時に住所の作成を必要とします。サンプル・ストアは、顧客登録時に住所を必要としないので、住所 1 などの必須フィールドのいくつかは "-" に設定されます。住所を検査するとき、**billingaddress.jsp** は、住所 1 の値が "-" であるかどうか検査します。その場合には、住所が表示されません。

サンプル・ストア配送方法ページ

サンプル・ストア・チェックアウト処理の 3 番目のステップでは、ショッピング・カートのチェックアウトの使用事例で説明されているように、顧客は配送方法 (shipping.jsp) を選択する必要があります。

コマンド

shipping.jspは、以下のコマンドを使用します。

- AddShipModeView
- OrderItemDisplay

bean

shipping.jspは、以下の bean を使用します。

- OrderBean
- OrderItemAccessBean
- ShipModeAccessBean

インプリメンテーションの詳細情報

注:サンプル・ストアに関する一般的なインプリメンテーション技術の詳細については、多文化的な情報も含めて サンプル・ストアのページ: 共通するインプリメンテーション技術をご覧ください。

顧客は、ショッピング・カート・ページの**チェックアウト**をクリックすることによって、チェックアウト処理を開始すると、一連のチェックアウト・ページを移動します。 OrderItemDisplay コマンドは、次にロードするページを判別するために使用されます。 OrderItemDisplay は、データベース中の OrderItemDisplay.jspに戻されます。 OrderItemDisplay.jsp は、ページパラメーターに基づいた、別の JSP ファイルに入っています。 ページの値が shipmethod である場合には、3 番目の ページ 「チェックアウト 3 配送方法の選択 (shipping.jsp) 」がロードされます。

チェックアウト 3 「配送方法の選択」 ページ (shipping.jsp) には、顧客が配送方法を選択できるフォームがあります。 フォームのアクションは、AddShipModeView に設定されますが、AddShipMode.jsp と関連付けるために VIEWREG データベース・テーブルに登録されます。 AddShipModeView でこのフォームが送信されると、AddShipMode.jspが呼び出されます。

AddShipMode.jsp は、以下のコマンドを実行します。

- OrderItemUpdate
- OrderPrepare.

OrderItemUpdate コマンドは、配送方法の選択でオーダー・アイテムを更新します。次に、OrderPrepare コマンドは、オーダーをプリプロセスするために呼び出されます。 その後、OrderDisplay は、チェックアウト処理に次ページを表示するために呼び出されます。 statusパラメーターに P を設定した場合には、チェックアウト 4 オーダー要約 (OrderDisplayPending.jsp) が、次に表示されます。

チェックアウト 3 「配送方法の選択」 ページ (shipping.jsp) には、それぞれの配送方法の料金構成および配達のおよその時間が表示されます。 この情報は、次のフィールドの SHPMODEDSC テーブルに保管されます。

- DESCRIPTION は配送方法の説明を保管する。
- SHPMODEDSC.FIELD1 は構成の説明を保管する。

- SHPMODEDSC.FIELD2 は配達のおよその時間を保管する。

データベースの配送料を変更した場合には、更新された値がこのページにも表示されるように、SHPMODEDSC テーブルの説明も変更するようにしてください。

注:SHPMODEDSC.FIELD1 および SHPMODEDSC.FIELD2 は SHPMODEDESC テーブルのカスタム・フィールドを使用して作成されました。

WebFashion と NewFashion のサンプル・ストア・バンドル表示ページ

バンドル表示ページは、オンライン・ストア内の商品の販売促進グループをフィーチャーしています。それには通常、説明、バンドルを構成するコンポーネントのリスト、各コンポーネントの価格、イメージ、およびバンドルを構成する商品にバリエーションがある場合は属性 (サイズやカラー) のリスト、さらに各属性ごとの値 (カラーでは赤、青、サイズでは L、LL など) のリストが含まれます。

顧客は、シングルクリックでバンドルの商品をすべてショッピング・カート、または購入希望商品リストに追加することができます。バンドルの各商品は、ショッピング・カート内では別々の明細アイテムとして表示されます。これは、パッケージ全体が単一の明細アイテムとして表示される、パッケージの場合とは異なります。

サンプル・ストア・バンドル表示ページとその機能の詳細については、バンドル・ページ表示の使用事例を参照してください。

コマンド

bundledisplay.jspは、以下のコマンドを使用します。

- OrderItemAdd
- InterestItemAdd

bean

bundledisplay.jspは、以下の bean を使用します。

- BundleDataBean
- CompositeProductDataBean

インプリメンテーションの詳細情報

注: 多文化情報を含むすべてのサンプル・ストア・ページに共通なインプリメンテーション手法については、サンプル・ストア・ページ: 共通のインプリメンテーション手法を参照してください。

Bundledisplay.jsp は、データベース (DISPENTREL テーブルに) 中に登録され、ストアにすべての商品を表示します。 Bundledisplay.jspは、以下を表示します。

- 説明 (バンドルを構成するコンポーネントのリストを含む) およびバンドルのイメージ
- バンドルの各商品ごとのイメージ、価格、および属性のリスト (サイズとカラー)。 さらに、各属性ごとの値 (カラーでは赤、青、サイズでは L、LL) のリスト。
- 「ショッピング・カートに追加」リンクおよび「購入希望リストに追加」リンク
- 「数量」フィールド (NewFashion のみ)

説明 (バンドルを構成するコンポーネントのリストを含む) およびバンドルのイメージ

BundleDataBean はバンドルの説明とイメージを検索します。

商品イメージ、価格、属性および属性値

BundleDataBean は ProductDataBean を呼び出して、バンドルを構成する各商品についての情報を検索します。ProductDataBean の機能についての詳細は、サンプル・ストア商品ページ を参照してください。

「ショッピング・カートに追加」および「購入希望商品リストに追加」

詳細はサンプル・ストア商品ページを参照してください。

「数量」フィールド (NewFashion のみ)

バンドルされている商品ごとに数量フィールドがあるため、顧客は各商品の数を指定してショッピング・カートに追加することができます。

WebFashion と NewFashion のサンプル・ストア・パッケージ表示ページ

パッケージ表示ページは、オンライン・ストア内の商品の販売促進グループをフィーチャーしています。それには通常、説明、パッケージのイメージ、パッケージを構成するコンポーネントのリストとそれぞれのイメージ、パッケージの価格、およびパッケージを構成する商品にバリエーションがある場合は属性 (サイズやカラー) のリスト、さらに各属性値 (カラーでは赤、青、サイズでは L、LL など) のリストが含まれます。

顧客は、シングルクリックでパッケージをショッピング・カート、または購入希望商品リストに追加することができます。パッケージは、ショッピング・カート内では単一の明細アイテムとして表示されます。これは、バンドル内の各アイテムが別々の明細アイテムとして表示される、バンドルの場合とは異なります。パッケージ内の商品を個別に購入することはできません。

サンプル・ストア・パッケージ表示ページとその機能の詳細については、パッケージ・ページ表示の使用事例を参照してください。

コマンド

PackageDisplay.jspは、以下のコマンドを使用します。

- OrderItemAdd
- InterestItemAdd

bean

PackageDisplay.jspは、以下の bean を使用します。

- PackageDataBean
- CompositeProductDataBean

インプリメンテーションの詳細情報

注: 多文化情報を含むすべてのサンプル・ストア・ページに共通なインプリメンテーション手法については、サンプル・ストア・ページ: 共通のインプリメンテーション手法を参照してください。

packagedisplay.jspは、データベース (DISPENTREL テーブルに) 中に登録され、ストアのすべての商品を表示します。packagedisplay.jspは、以下を表示します。

- パッケージの説明 (パッケージを構成するコンポーネントのリストを含む)、イメージ、および価格
- パッケージを構成する商品の属性および属性値

- 「ショッピング・カートに追加」および「購入希望商品リストに追加」のリンク
- 「数量」テキスト・ボックス (NewFashion のみ)

パッケージの説明 (パッケージを構成するコンポーネントのリストを含む)、イメージ、および価格
 パッケージの説明、イメージ、および価格は PackageDataBean を使用して検索されます。

パッケージを構成する商品の属性および属性値

PackageDataBean は ProductDataBeans を検索してパッケージ内の各商品についての情報を表示します。詳細はサンプル・ストア商品ページを参照してください。

パッケージを正確に表示するには、パッケージ内の各商品の属性値に同じパラメーター名を使用する必要があります。例:

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

「ショッピング・カートに追加」および「購入希望商品リストに追加」
 詳細はサンプル・ストア商品ページを参照してください。

「数量」テキスト・ボックス

顧客は、ショッピング・カートまたは購入希望商品リストに追加するパッケージの数を指定することができます。

WebFashion サンプル・ストア - クイック・チェックアウト・プロフィールの作成ページ

登録済みの顧客はクイック・チェックアウト・フィーチャーを使用することができます。これを使用すると、顧客は配送および請求の情報を入力せずにオーダーを完了することができます。クイック・チェックアウトを完了するには、顧客はクイック・チェックアウト・プロフィール (ProfileForm.jsp) を作成する必要があります。これは、請求先および配送先住所、配送方法、および支払い情報を入力して行います。

詳細については、クイック・チェックアウト・プロフィールの作成の使用事例を参照してください。

コマンド

ProfileForm.jspは以下のコマンドを使用します:

- UpdateQuickCheckoutProfileView (VIEWREG テーブルに登録されていて、UpdateQuickCheckoutProfile.jsp と関連付けられているコマンド表示)
- PrivacyView

bean

ProfileForm.jspは、以下の bean を使用します。

- ErrorDataBean
- AddressAccessBean
- OrderAccessBean
- OrderPaymentInfoAccessBean
- ShippingDataBean
- ShippingModeDescriptionAccessBean
- ProfileCassetteAccountDataBean
- StoreDataBean
- OrderItemAccessBean

インプリメンテーションの詳細情報

注: 多文化情報を含むすべてのサンプル・ストア・ページに共通なインプリメンテーション手法については、サンプル・ストア・ページ: 共通のインプリメンテーション手法を参照してください。

UpdateQuickCheckoutProfile は、以下のサーバー・コマンドを実行します。

- AddressAdd (請求先住所の追加または更新)
- AddressAdd (配送先住所の追加または更新)
- OrderProfileUpdate

UpdateQuickCheckoutProfile.jspは、配送先住所および請求先住所を更新、または作成します。次に、新規に生成された配送先住所 ID、新規に生成された請求先住所 ID、配送モード ID、および支払い情報 (カード・ブランド、カード番号、有効期限月および年) をクイック・チェックアウト・プロファイルに対して更新または作成します。デフォルトで作成される配送先住所には default_shipping というニックネームが付けられ、デフォルトの請求先住所には default_billing というニックネームが付けられます。

クイック・チェックアウト・プロファイルは、状況が 'Q' のダミー・オーダーとして機能します。その後プロファイル情報はこのダミー・オーダーのオーダー ID に保管されます。

UpdateQuickCheckoutProfile.jsp には以下のパラメーターが必須です。

- shipping_nickName=s &
- shipping_firstName=s &
- shipping_lastName=s &
- shipping_address1=s &
- shipping_address2=s &
- shipping_city=s &
- shipping_state=s &
- shipping_zipCode=s &
- shipping_country=s &
- shipping_phone1=s &
- billing_nickName=s &
- billing_firstName=s &
- billing_lastName=s &

- billing_address1=s &
- billing_address2=s &
- billing_city=s &
- billing_state=s &
- billing_zipCode=s &
- billing_country=s &
- billing_phone1=s &
- shipModeId=s &
- URL=s &

以下のパラメーターはオプションです。

- shipping_addressId=s &
- billing_addressId=s &
- pay_payMethodId=s &
- pay_cardBrand=s &
- pay_cardNumber=s &
- pay_cardExpiryMonth=s &
- pay_cardExpiryYear=s

注:

1. “shipping_” で始まるパラメーターは配送先住所用です。
2. “billing_” で始まるパラメーターは請求先住所用です。
3. shipping_addressId および billing_addressId は、デフォルトの配送先住所およびデフォルトの請求先住所を更新する場合にのみ必須です。

顧客が**請求先住所と同じ**のチェック・ボックスを選択した場合、請求先住所が配送先住所になります。以下の javascript で、このアクションが実行されます。

```
<SCRIPT language="javascript">
function UpdateProfile(form)
{
if ( form.sameaddress.checked
&& form.shipping_firstName.value == ''
&& form.shipping_lastName.value == ''
&& form.shipping_address1.value == ''
&& form.shipping_address2.value == ''
&& form.shipping_city.value == ''
&& form.shipping_state.value == ''
&& form.shipping_zipCode.value == ''
&& form.shipping_country.value == ''
&& form.shipping_phone1.value == '' )
{
form.shipping_firstName.value = form.billing_firstName.value
form.shipping_lastName.value = form.billing_lastName.value
form.shipping_address1.value = form.billing_address1.value
form.shipping_address2.value = form.billing_address2.value
form.shipping_city.value = form.billing_city.value
form.shipping_state.value = form.billing_state.value
form.shipping_zipCode.value = form.billing_zipCode.value
form.shipping_country.value = form.billing_country.value
form.shipping_phone1.value = form.billing_phone1.value
}
```

```
}  
form.submit()  
}  
</SCRIPT>
```

エラー処理

UpdateQuickCheckoutProfile.jspの AddressAdd コマンドが失敗すると、 AddressErrorView が呼び出されます。 AddressErrorView は、AddressForm.jspと関連付けられています。 AddressForm.jsp は、quickcheckout が page パラメーターの値として渡されることを確認します。 quickcheckout が渡される場合、ProfileForm.jspが組み込まれ、エラー・メッセージの示された「クイック・チェックアウト・プロフィール」ページが再表示されます。 たとえば、AddressForm.jsp (WebFashion) では、以下のようになります。

```
if (state.equals("quickcheckout")) {  
incfile = storeDir + "ProfileForm.jsp";
```

サンプル・ストア・クイック・チェックアウト要約ページ

顧客がショッピング・カーとの「クイック・チェックアウト」をクリックすると、オーダー詳細と、事前に記入された配送、請求、および支払い情報が含まれた「Quick Checkout: Order summary (クイック・チェックアウト: オーダー要約)」ページ (quickcheckoutsummary.jsp) が表示されます。 顧客は次に「オーダー」をクリックしてオーダーを完了する必要があります。

詳細については、クイック・チェックアウトの使用事例を参照してください。

コマンド

quickcheckoutsummary.jspは、以下のコマンドを使用します。

- OrderProcess
- OrderItemDisplay
- PrivacyView
- ContactView

bean

quickcheckoutsummary.jspは以下の bean を使用します。

- OrderBean
- AddressDataBean
- ErrorDataBean
- ProfileCassetteAccountDataBean
- OrderPaymentInfoAccessBean

インプリメンテーションの詳細情報

注:サンプル・ストアに関する一般的なインプリメンテーション技術の詳細については、多文化的な情報も含めて サンプル・ストアのページ: 共通するインプリメンテーション技術をご覧ください。

クイック・チェックアウト要約ページは、以下の相違点を除いて、サンプル・ストア・オーダー要約ページと同様に動作します。

- 「支払い情報」フィールドには、クイック・チェックアウト・プロフィールの情報が事前に記入されています。
- 支払い情報は OrderPaymentInfoAccessBean を使用して検索されます。

WebFashion および NewFashion サンプル・ストア購入希望商品リスト・ページ

サンプル・ストア購入希望商品リスト・ページにより、登録された顧客はアイテムを購入希望商品リスト(買い物候補リスト)に追加した後、リスト(interestitemdisplay.jsp)を表示して編集し、希望の商品をショッピング・カートに追加することができます。また、顧客はEメール(sendwishlistmsg.jsp)で家族または友達に購入希望商品リストを送ることもできます。NewFashion ストアでは、購入希望商品リストとともに個人メッセージを送ることができます。

注:NewFashion では、ゲスト・ショッパーと登録済み顧客の両方が購入希望商品リストにアイテムを追加することができます。WebFashion で購入希望商品リストにアイテムを追加できるのは、登録済み顧客だけです。

顧客がEメールで購入希望商品リストを受け取ると、購入希望商品リスト・ページ(sharedwishlist.jsp)を参照できます。このページは、「購入希望商品リストの送信」と「アイテムの除去」ボタンがないことを除いて、interestitemdisplay.jsp と同一です。顧客は購入するアイテムを選択し、それをショッピング・カートに追加することができます。

詳細については、購入希望商品リスト表示の使用事例および購入希望商品リストへのアイテム追加の使用事例を参照してください。

サンプル・ストア購入希望商品リスト手順は、以下のJSPファイルを使用します。

- interestitemdisplay.jsp(購入希望商品リストページ)
- sharedwishlist.jsp
- sendwishlistmsg.jsp

コマンド

interestitemdisplay.jspは、以下のコマンドを使用します。

- OrderItemAdd
- SendWishListMsg

sharedwishlist.jspは、以下のコマンドを使用します。

- OrderItemAdd

sendwishlistmsg.jspは、以下のコマンドを使用します。

- SendMsgCmd

bean

interestitemdisplay.jspは、以下のbeanを使用します。

- InterestItemListDataBean
- InterestItemDataBean
- CatalogEntryAccessBean

- UserRegistrationDataBean

sharedwishlist.jspは、以下の bean を使用します。

- InterestItemListDataBean
- InterestItemDataBean
- CatalogEntryAccessBean

sendwishlistmsg.jspは、以下の bean を使用します。

- UserRegistrationDataBean (WebFashion のみ)

インプリメンテーションの詳細情報

注: 多文化情報を含む、すべてのサンプル・ストア・ページに共通なインプリメンテーション手法については、サンプル・ストア・ページ: 共通のインプリメンテーション手法を参照してください。

追加希望リスト

顧客が「追加希望リスト」をクリックすると、InterestItemAdd コマンドが呼び出されます。詳しくは、サンプル・ストア商品ページを参照してください。

アイテムを購入希望商品リストに追加する前に、システムは UserRegistrationDataBean を使用して、顧客がストアに登録されているかどうかを検査します。登録されていない場合、JSP は registerfirst.jspを組み込みます。これは、顧客に登録と再試行を促すメッセージを表示します。

購入希望商品リストの表示

顧客が「購入希望商品リストの表示」をクリックすると、購入希望商品リスト・ページ (interestitemdisplay.jsp)が、購入希望商品リストの内容とともに表示されます。

InterestItemListDataBean および InterestItemDataBean は、購入希望商品リストの内容に関する情報を検索します。

ショッピング・カートに追加およびアイテムの除去

顧客は購入希望商品リスト・ページ (interestitemdisplay.jsp)から、ショッピング・カートへのアイテムの追加か、または購入希望商品リストからのアイテムの削除を選択することができます。顧客が「選択したアイテムをショッピング・カートへ追加」をクリックすると、OrderItemAdd コマンドが呼び出されます。「アイテムの除去」をクリックすると、InterestItemDelete コマンドが呼び出されます。

購入希望商品リストの送信

顧客が「名前」フィールドと「E メール・アドレス」フィールドを完成させた後「購入希望商品リストの送信」をクリックすると、SendWishListMsg コマンドが呼び出されます。

SendWishListMsg は VIEWREG テーブルに登録され、sendwishlistmsg.jspを表示します。

インスタンスを作成してからでなければ、SendWishListMsg コマンドは実行できません。

WebFashion および NewFashion は、以下を使用してインスタンスを作成します。 SendMsgCmd sendMsgCmd = (SendMsgCmd) CommandFactory.createCommand(cmdEntry);

以下に例を示します。

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


“sendMsgCmd” タスク・コマンドのパラメーターを、即時にメッセージを送信するように設定することができます。

```
sendMsgCmd.sendImmediate();
```

注: WebFashion には、宛先の E メール・アドレスのフィールドだけがあります。NewFashion には、宛先の E メール・フィールド、送信側の名前のフィールド、および個人用メッセージのフィールドがあります。宛先の E メール・フィールドと、送信側の名前のフィールドは必須です。

エラー処理

顧客が購入希望商品リストにアイテムを選択していないのに「**選択したアイテムをショッピング・カートへ追加**」をクリックすると、エラー・メッセージが表示されます。以下のコードはこのアクションを実行します。

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

WebFashion および NewFashion の E メール通知ページ

登録済み顧客が「**オーダー**」をクリックしてオーダーを完了し（詳細はショッピング・カートのチェックアウトの使用事例を参照してください）、オーダーが Payment Manager によって与信済みになると、システムは顧客に E メール OrderAuthorized.jsp を送信し、オーダーが受諾済みになり、支払いが与信済みになったことを知らせます。

NewFashion ストアでは、以下の E メール通知が送信されます。

- オーダー・キャンセル E メール (OrderCanceledNotification.jsp)。クレジット・カードが拒否されたため、オーダーを完了できませんでした。
- 配送通知 E メール (ReleaseShipNotify.jsp)。オーダーがショッパーに配送されました。
- オーダー送信 E メール (OrderReceived.jsp)。登録済みの顧客が、「**オーダー**」をクリックすることにより、オーダーを完了したら、オーダー送信 E メールが送信されます。

bean

OrderAuthorized.jsp は、以下の bean を使用します。

- StoreDataBean

- OrderDataBean
- OrderItemDataBean
- OrderBean
- StoreEntityDescriptionAccessBean
- AddressDataBean

注: 上記の bean は NewFashion ストアだけで使用されます。

インプリメンテーションの詳細情報

注: 多文化情報を含むすべてのサンプル・ストア・ページに共通なインプリメンテーション手法については、サンプル・ストア・ページ: 共通のインプリメンテーション手法を参照してください。

システムは、OrderDataBean を使用して顧客のオーダーについての情報を検索します。ストア情報は StoreDataBean を使用して検索されます。StoreEntityDescriptionAccessBean は、ストアの名前をシステムに渡します。AddressDataBean は配送先住所をシステムに渡します。

WebFashion および NewFashion サンプル・ストア「オーダーの表示」ページ

登録済みの顧客がオーダーを発行した後は、いつでもそのオーダーの状況を表示することができます。オーダーを表示するには、顧客が「アカウント」ページでアカウントをクリックし、**オーダーの表示**をクリックします。顧客の発行したオーダーのリストが載せられた「オーダーの状況」ページ (trackorderstatus.jsp) が表示されます。顧客が特定のオーダーをクリックすれば、そのオーダーの詳細を表示できます。「オーダー詳細情報」ページ (orderdetail.jsp) が表示されます。

注: 一時、保留、CSR 編集、クイック・オーダー・プロフィール、個人用要求リスト、共有可能要求リスト、在庫なしオーダー、およびキャンセル済みオーダーは表示できません。

詳細は、オーダー表示の使用事例を参照してください。

Bean

trackorderstatus.jspは以下の bean を使用します。

- OrderDataBean
- PayStatusListPMDDataBean
- OrderAccessBean
- PriceDataBean

orderdetail.jsp は以下の bean を使用します。

- OrderDataBean
- OrderItemDataBean

インプリメンテーションの詳細情報

注: サンプル・ストアに関する一般的なインプリメンテーション技術の詳細については、多文化的な情報も含めて サンプル・ストアのページ: 共通するインプリメンテーション技術をご覧ください。

顧客が**オーダーの表示**をクリックすると、顧客の発行したオーダーのリストが載せられた「オーダーの状況」ページ (trackorderstatus.jsp) が表示されます。 OrderDataBean は、以下のコードを使用してリスト内のすべてのオーダーをループすることによって、顧客が発行したすべてのオーダーを検索します。

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

OrderDataBean は以下のオーダー情報を表示します。

- オーダー番号
- オーダー日付
- 合計
- 支払い状況

OrderDataBean は、.getStatus() メソッドを呼び出すことによって支払い状況を検索します。
.getStatus() メソッドは、以下の表に示す支払い状況を表す状況ストリングを返します。

状況	要旨	意味
P	保留	顧客はオーダーを変更できます。
I	送信済み	顧客はオーダーを変更できません。
W	承認保留中	CheckOrderApproval タスク・コマンドが、いくつかのオーダー・アイテムが承認されていないことを示しました。
N	承認拒否	CheckOrderApproval タスク・コマンドが、いくつかのオーダー・アイテムの承認が拒否されたことを示しました。
M	支払い 与信が保留中	支払い与信を待っています。
A	支払い与信の確認が必要	支払い与信において、住所確認警告などの異常事態が発生しました。「オーダー管理」ユーザー・インターフェースを使用して、支払い与信を確認して受諾するか、オーダーをキャンセルする必要があります。与信が受諾されると、ユーザー・インターフェースがオーダー状況を B または C に適宜変更します。
B	バック・オーダー済み	バック・オーダー在庫割り振りの結果オーダー金額が変更された可能性があるため、支払い与信を再実行する必要がある可能性があります。
C	支払い 与信は完了	支払い与信が完了しました。すべてのオーダー・アイテムが既存在庫から割り振られました。オーダー金額は変わりません。
E	CSR 編集	顧客サービス担当者がオーダーを処理しています。

R	リリース済み	すべてのオーダー・アイテムがオーダー処理のためにリリースされました。
S	配送済み	すべてのオーダー・アイテムが配送されました。
D	入金済み	支払いが資金化されました。
L	在庫なし	1 つ以上のオーダー・アイテムの在庫がありません。
T	一時	「オーダー管理」インターフェースが、オーダーを一時的にバックアップするために使用します。
Q	クイック・オーダー・プロファイル	オーダーはデフォルトのオーダー情報を保持しており、新規オーダーはこれをコピーすることで手早く作成できます。
F	リモート・オーダー処理の実行が可能	オーダーをリモート・システムに送信してオーダー処理することが可能です。この状況は、MQAdapter フィーチャーによって使用されます。
G	リモート・オーダー処理が保留中	オーダーがオーダー処理のためにリモート・システムに送信されました。この状況は、MQAdapter フィーチャーによって使用されます。
Y	個人用要求リスト	オーダーは個人用の要求リストです。
Z	共有可能要求リスト	オーダーは共有可能な要求リストです。
X	キャンセル済み	オーダーはキャンセル済みです。

注: Payment Manager は、支払いが拒否されているかどうか確認します。支払い状況が「拒否されていない」であれば、.getStatus() メソッドの戻りに応じて、メッセージが表示されます。

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

顧客は、オーダーのリストを検索した後、特定のオーダーをクリックして詳細 (orderdetail.jsp) を表示できます。OrderDataBean は顧客のすべてのオーダー・アイテムを検索し、OrderItemDataBean は各アイテムに関する詳細情報を検索します。OrderItemDataBean は以下の詳細情報を検索します。

- オーダーの価格
- オーダーの内容
- オーダーの予定または実際の配送日付
- 各アイテムの追跡番号

各アイテムは、複数の部分で構成できますが、それぞれを個別のボックスに入れて配送し、複数の追跡番号を付けることができます。複数のアイテムで、同じ追跡番号を共用することも可能です。

OrderDataBean は、以下のコードで示されているようにオーダーの予定または実際の配送日付を検索します。

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

第 5 章 Auction pages

「オークション - すべての入札」ページ

「オークション - すべての入札」ページ (auc_all_bids.jsp) は、オークションのすべての参加者によって送信された入札のリストを表示します。このリストには、以下のものが含まれます。

- アクティブな入札
- 落札
- オーダーが完了しなかった落札
- オーダーが完了した入札

顧客はオークションについて以下の詳細を見ることができます。

- 入札を送信した顧客の名前
- アイテムごとの入札値
- 入札数量
- 入札番号

顧客は「[入札の変更](#)」または「[自動入札の変更](#)」をクリックして、入札または自動入札を変更することもできます。

「オークション - すべての入札」ページは、各ページに最大 5 件の入札を表示します。「[次ページ](#)」および「[前ページ](#)」をクリックして、ページ間をナビゲートできます。

コマンド

auc_all_bids.jsp は、以下のコマンドを使用します。

- AutoBidUpdateForm
- BidUpdateForm

bean

auc_all_bids.jsp は、次の bean を使用します。

- BidDataBean
- AuctionDataBean
- ItemDataBean
- UserRegistrationDataBean
- BidListBean

インプリメンテーション

入札の変更

「[入札の変更](#)」リンクをクリックすると、BidUpdateForm コマンドが呼び出され、「オークション - [入札の変更](#)」ページ (auc_bid_modify.jsp) が表示されます。

自動入札の変更

「[自動入札の変更](#)」リンクをクリックすると AutoBidUpdateForm コマンドが呼び出され、「オー

クション - 自動入札の変更」ページ (auc_autobid_modify.jsp) が表示されます。選択されたオークションに対して送信された入札がない場合、auc_all_bids.jsp は適切なメッセージを表示します。

auc_all_bids.jspは BidDataBean および BidListBean を使用し、オークションに参加したすべてのユーザーの入札関連情報を検索します。 AuctionDataBean および ItemDataBean は、オークションの名前とタイプ、購入可能な数量、およびオークション中のアイテムなど、オークションについての情報を検索するときに使用されます。 UserRegistrationDataBean は、顧客の登録情報とユーザー ID に関連した情報を検索するときに使用されます。

「オークション - 自動入札の作成」ページ

「オークション - 自動入札の作成」ページ (auc_autobid_create.jsp) は、顧客が新規の自動入札を作成するために使用できる、最高入札の情報 (auc_show_best_bid.jsp) および自動入札送信フォーム (auc_autobidform.jsp) を表示するフレーム・セットです。自動入札の送信がエラーになった場合、エラー・メッセージが先頭フレームに表示されます (auc_err_bid_submit_frame.jsp)。自動入札送信フォームが最終フレームに表示されるので、顧客は情報を訂正して自動入札を再送信することができます。

コマンド

auc_autobid_create.jsp は、コマンドを使用しません。

bean

auc_autobid_create.jsp は、bean を使用しません。

インプリメンテーション

auc_autobid_create.jsp は、以下のページで「新規の自動作成」をクリックすることによって AutoBidCreateForm コマンドが呼び出されるときに表示されます。

- 「すべてのオークション・リスト」ページ (auc_all_auction_list.jsp)
- 「オークション - メッセージ・フォーラムの追加」ページ (auc_msg_add.jsp)
- 「オークション - メッセージ・フォーラムの付加」ページ (auc_msg_append.jsp)
- 「オークション・ルール」ページ (auc_rule.jsp)
- 「オークション・ギャラリー」ページ (auc_shopper_auction_list.jsp)
- 「オークション - メッセージ・フォーラム・リスト」ページ (auc_shopper_msg_list.jsp)

「オークション - 自動入札フォーム」ページ

「オークション - 自動入札フォーム」ページ (auc_autobidform.jsp) は、顧客が新規の自動入札を作成したり既存の自動入札を更新するために使用できる、最高入札の情報および自動入札送信フォームを表示するフレーム・セットです。

このフォームでは、自動入札の参照番号が表示されていて、顧客は以下の情報を入力することができます。

- アイテムごとの入札値
- 自動入札の上限
- 要求数量
- クレジット・カードのブランド

- クレジット・カード番号
- クレジット・カードの有効期限の (月および年)
- 請求情報
- 配送情報
- 運送会社および配送モード

顧客は「リセット」をクリックしてフォームに入力した情報を消去するか、または「送信」をクリックして入札をオークションに送信することができます。自動入札の送信がエラーになった場合、エラー・メッセージが先頭フレームに表示されます (auc_err_bid_submit_frame.jsp)。自動入札送信フォームが最終フレームに表示されるので、顧客は情報を訂正して自動入札を再送信することができます。

コマンド

auc_autobidform.jsp は以下のコマンドを使用します。

- AutoBidSubmit

bean

auc_autobidform.jsp は以下の bean を使用します。

- AddressBookDataBean
- AuctionDataBean
- AutoBidDataBean
- BidDataBean
- ItemDataBean
- PriceDataBean
- ShippingDataBean
- ShippingModeAccessBean
- ShippingModeDescriptionAccessBean

インプリメンテーション

auc_autobidform.jspは、AuctionDataBean、ItemDataBean および PriceDataBean を使用し、商品詳細情報、購入可能な数量、およびデポジットが必要かどうかといったオークション関連情報を検索します。顧客が自動入札を更新する場合、auc_autobidform.jspは AutoBidDataBean と BidDataBean を使用し、既存の入札情報を検索します。

auc_autobidform.jspは、ShippingDataBean、ShippingModeAccessBean、ShippingModeDescriptionAccessBean、および AddressbookDataBean を使用して、配送、請求、配送モードについての情報、そして配送先と請求先として利用可能な住所を検索します。

「送信」をクリックすると AutoBidSubmit コマンドが呼び出され、「入札の確認」ページ (auc_bid_ack.jsp) が表示されます。

auc_autobidform.jsp も auc_bid_validate.jsファイルを使用して、入力された情報の妥当性を検査し、関係するエラー・メッセージが示されたメッセージ・ボックスを表示します。

「オークション - 自動入札の変更」ページ

「オークション - 自動入札の変更」ページ (auc_autobid_modify.jsp) は、最高入札の情報 (auc_show_best_bid.jsp) および自動入札の送信を変更するための自動入札フォーム (auc_autobidform.jsp) を表示するフレーム・セットです。自動入札の送信がエラーになった場合、エラー・メッセージ (auc_err_bid_submit_frame.jsp) が先頭フレームに表示されます。顧客は自動入札の送信フォームを使用して、情報を訂正したり入札を再送信することができます。

コマンド

auc_autobid_modify.jsp は、コマンドを使用しません。

bean

auc_autobid_modify.jsp は、bean を使用しません。

インプリメンテーション

auc_autobid_modify.jsp は、顧客が「オークション - すべての入札」ページ (auc_all_bids.jsp) および「ユーザーのオークションの入札」ページ (auc_shopper_bid_list.jsp) から、「自動入札の変更」をクリックすると表示されます。

「オークション - 入札の変更」ページ

「オークション - 入札の変更」ページ (auc_bid_modify.jsp) は、顧客が既存の入札を変更するために使用する、最高入札情報および入札送信フォームを表示するフレームセットです。

入札の送信がエラーになった場合、エラー・メッセージが先頭フレームに表示されます (auc_err_bid_submit_frame.jsp)。入札送信フォームが最終フレームに表示されるので、顧客は情報を訂正して入札を再送信することができます。

コマンド

auc_bid_modify.jsp は、コマンドを使用しません。

bean

auc_bid_modify.jsp は、以下の bean を使用します。

- AuctionDataBean

インプリメンテーション

auc_bid_modify.jspは AuctionDataBean を使用し、オークションの名前とタイプ、購入可能な数量などの、オークション関連情報を検索します。

「オークション」ページ

「オークション」ページ (Auction.jsp) は、顧客をオークション専用ページへ案内し、「オークション・ログオン・フォーム」ページ (auctionLogonForm.jsp) へのリンクが張られた開始ページです。この「オークション」ページは、会社およびブランドの情報やその他の説明を表示したり、スプラッシュ画面として使用することができます。

コマンド

Auction.jsp は、コマンドを使用しません。

bean

Auction.jsp は、bean を使用しません。

インプリメンテーション

「ログオン」リンクをクリックすると、「オークション・ログオン・フォーム」ページ (auctionLogonForm.jsp) が表示されます。

「オークション - 入札の確認」ページ

「オークション - 入札の確認」ページ (auc_bid_ack.jsp) は、顧客が正常に入札または自動入札を作成または更新するときに、確認通知メッセージを表示します。確認メッセージには、アイテムごとの入札または自動入札参照番号、数量、および入札値が含まれています。

コマンド

auc_bid_ack.jsp は、コマンドを使用しません。

bean

auc_bid_ack.jsp は、以下の bean を使用します。

- AuctionDataBean

インプリメンテーション

顧客が「オークション - 入札のフォーム」ページ (auc_bidform.jsp) または「オークション - 自動入札のフォーム」ページ (auc_autobidform.jsp) から、「送信」をクリックすると、auc_bid_ack.jsp が表示されます。

auc_bid_ack.jspは AuctionDataBean を使用し、入札または自動入札の確認メッセージの一部として表示されるオークション関連情報をデータベースから検索します。

「オークションのイメージ・マップ」ページ

「オークションのイメージ・マップ」ページ (auctionbuttonbar.jsp) は、すべてのオークション・ストア・ページの先頭フレームにオープンし、以下へのリンクが示されます。

- ショッピング・カート
- アカウント
- 問い合わせ先
- ヘルプ

コマンド

auctionbuttonbar.jspは、以下のコマンドを使用します。

- StoreCatalogDisplay
- OrderItemDisplay

- LogonForm

bean

auctionbuttonbar.jspは、bean を使用しません。

インプリメンテーション

ホーム 「ホーム」をクリックすると StoreCatalogDisplay コマンドが呼び出され、ストアのホーム・ページがオープンします。

ショッピング・カート

「ショッピング・カート」をクリックすると OrderItemDisplay コマンドが呼び出され、OrderItemDisplay.jsp がロードされます。 OrderItemDisplay.jsp は shoppingcart.jspをロードし、「ショッピング・カート」ページを表示します。

アカウント

「アカウント」をクリックすると LogonForm コマンドが呼び出されます。登録済みの顧客の場合、LogonForm コマンドで「アカウント」ページ (account.jsp) がオープンします。登録されていない顧客の場合、LogonForm コマンドで「登録」または「ログオン」ページ (myaccount.jsp) がオープンします。

問い合わせ先

「問い合わせ先」をクリックすると、「問い合わせ先」ページ (contact.jsp) が表示されます。

ヘルプ 「ヘルプ」をクリックすると、「ヘルプ」ページ (help.jsp) がロードされます。

「オークション - 入札の作成」ページ

「オークション - 入札の作成」ページ (auc_bid_create.jsp) は、顧客が新しい入札を作成するために使用する、最高入札情報 (auc_show_best_bid.jsp) および入札送信フォーム (auc_bidform.jsp) を先頭フレームに表示するフレームセットです。

入札の送信がエラーになった場合、エラー・メッセージが先頭フレームに表示されます (auc_err_bid_submit_frame.jsp)。入札送信フォームが最終フレームに再表示されるので、顧客は情報を訂正して入札を再送信することができます。

コマンド

auc_bid_create.jsp は、コマンドを使用しません。

bean

auc_bid_create.jsp は、以下の bean を使用します。

- AuctionDataBean

インプリメンテーション

auc_bid_create.jspは AuctionDataBean を使用して、オークション関連情報を検索します。

顧客は、他のオークション・ページから「新規入札」をクリックして、 auc_bid_create.jsp へアクセスできます。

「オークション - 入札のフォーム」 ページ

「オークション - 入札のフォーム」 ページ (auc_bidform.jsp) は、顧客が新しい入札の作成または既存の入札の更新を実行するための入力フォームを表示します。

このフォームは、入札参照番号を表示し、顧客が以下を入力できるようにします。

- アイテムごとの入札値
- 要求数量
- クレジット・カードのブランド
- クレジット・カード番号
- クレジット・カードの有効期限の (月および年)
- 請求情報
- 配送情報
- 運送会社および配送モード

顧客は、フォーム上に入力した情報をクリアするためには「リセット」を、オークションに入札するためには「送信」を使用することができます。入札の送信がエラーになった場合、エラー・メッセージが先頭フレームに表示されます (auc_err_bid_submit_frame.jsp)。入札送信フォームが最終フレームに表示されるので、顧客は情報を訂正して入札を再送信することができます。

コマンド

auc_bidform.jsp は以下のコマンドを使用します。

- BidSubmit

bean

auc_bidform.jsp は、以下の bean を使用します。

- AddressBookDataBean
- AuctionDataBean
- BidDataBean
- ItemDataBean
- PriceDataBean
- ShippingDataBean
- ShippingModeAccessBean
- ShippingModeDescriptionAccessBean

インプリメンテーション

auc_bidform.jspは、AuctionDataBean、ItemDataBean および PriceDataBean を使用し、商品詳細情報、購入可能な数量、およびデポジットが必要かどうかといったオークション関連情報を検索します。顧客が入札を更新する場合、auc_bidform.jspは BidDataBean を使用し、既存の入札情報を検索します。

auc_bidform.jspは、ShippingDataBean、ShippingModeAccessBean、ShippingModeDescriptionAccessBean、および AddressbookDataBean を使用して、配送、請求、配送モードについての情報、そして配送先と請求先として利用可能な住所を検索します。

「送信」をクリックすると BidSubmit コマンドが呼び出され、「入札の確認」ページ (auc_bid_ack.jsp) が表示されます。

auc_bidform.jsp も auc_bid_validate.js ファイルを使用して、入力された情報の妥当性を検査し、関係するエラー・メッセージが示されたメッセージ・ボックスを表示します。

「入札のオーバーライド件名」ページ

オークション中に顧客の入札がオーバーライドされる場合、メッセージがオーバーライドを知らせます。このメッセージは、件名および本体の 2 つの部分からなります。

「入札のオーバーライド件名」ページ (BidOverrideSubject.jsp) は、メッセージの件名を形式化するために使用されます。

コマンド

BidOverrideSubject.jsp は、コマンドを使用しません。

bean

BidOverrideSubject.jsp は、以下の bean を使用します。

- StoreDataBean
- LanguageAccessBean

インプリメンテーション

BidOverrideSubject.jsp は、顧客により直接呼び出されることはありません。このページは、メッセージ・システムによりスケジューラー経由で呼び出されます。「入札のオーバーライド」メッセージの件名情報は、StoreDataBean と LanguageAccessBean を使用して、データベースから検索され、ストリングとして保管されます。「オークション - メールを表示」ページ (auc_mail_display.jsp) は、このデータベースからストリング情報を検索し、「入札のオーバーライド」オークション通知メッセージの件名として表示します。

「オークション - 共通」ページ

「オークション - 共通」ページ (auc_common.jsp) は、ロケール、言語、ストア ID、および顧客 ID など、オークション・ストア・ページに関する共通情報を提供します。「オークション - 共通」ページでは、顧客が選択するストアおよび言語に基づいて、オークション・ページが使用するプロパティ・ファイルも判別します。

コマンド

auc_common.jsp は、コマンドを使用しません。

bean

auc_common.jsp は、以下の bean を使用します。

- StoreDataBean

インプリメンテーション

auc_common.jspは、すべてのオークション・ページに組み込まれていて、顧客およびストアについての情報を獲得するときに使用されます。 auc_common.jspは StoreDataBean を使用して、データベースから顧客およびストアの情報を検索します。

「オーダーの完了」ページ

顧客がオークションを落札した場合、オークションのコンポーネントは落札した商品に対するオーダーを作成し、オーダーを作成したことを落札した顧客に通知するメッセージを送信します。 このメッセージは、件名および本体の 2 つの部分からなります。

オークションのコンポーネントは、「オーダーの完了」ページ (CompleteOrder.jsp) を使用して、メッセージの本体を形式しますが、以下の情報を提供します。

- 数量、開始日および終了日、説明、および更新時刻を含む、オークション情報。
- ストア名、説明、E メール・アドレス、FAX および電話番号、番地、市区町村、都道府県、および郵便番号を含む、マーチャントまたはセラー情報。
- 名、ミドル・ネーム、および姓を含む、顧客情報。
- SKU、説明、重量、重量単位、長さ、幅、配送コード、および URL を含む、商品情報。
- 数量、値、日時、状況、支払いメソッド、および落札値を含む。入札情報。
- 商品合計、商品税額、配送料金、配送税、日時、を含むオーダー情報。

コマンド

CompleteOrder.jsp は、コマンドを使用しません。

bean

CompleteOrder.jsp は、以下の bean を使用します。

- StoreDataBean
- BidDataBean
- AuctionDataBean
- UserRegistrationDataBean
- ItemDataBean
- OrderDataBean
- LanguageAccessBean
- StoreEntityDescriptionAccessBean
- StoreAddressAccessBean
- CatalogEntryShippingAccessBean

インプリメンテーション

CompleteOrder.jspは、顧客により直接呼び出されることはありません。 このページは、スケジューラーを通してメッセージング・システムにより呼び出されます。「オーダーの完了」ページの情報の内容は、上記でリストされている databean を使用して、データベースから検索され、ストリングとして保管されます。

「オークション - メールが表示」ページ (auc_mail_display.jsp) は、このデータベースからストリング情報を検索し、「オーダーの完了」オークション通知メッセージの内容として表示します。

「完了オーダーの件名」ページ

顧客がオークションを落札した場合、オークションのコンポーネントは落札した商品に対するオーダーを作成し、オーダーを作成したことを落札した顧客に通知するオークション通知メッセージを送信します。このメッセージは、件名および本体の 2 つの部分からなります。

「完了オーダーの件名」ページ (CompleteOrderSubject.jsp) は、メッセージの件名を形式化するために使用されます。

コマンド

CompleteOrderSubject.jsp は、コマンドを使用しません。

bean

CompleteOrderSubject.jsp は、以下の bean を使用します。

- StoreDataBean
- LanguageAccessBean

インプリメンテーション

CompleteOrderSubject.jspは、顧客により直接呼び出されることはありません。このページは、スケジューラーを通してメッセージング・システムにより呼び出されます。「オーダーの完了」メッセージの件名情報は、StoreDataBean と LanguageAccessBean を使用して、データベースから検索され、ストリングとして保管されます。「オークション - メールを表示」ページ (auc_mail_display.jsp) は、このデータベースからストリング情報を検索し、「オーダーの完了」メッセージの件名として表示します。

「オークション - アイテムの表示」ページ

「オークション - アイテムの表示」ページ (auc_ItemDisplay.jsp) は、オークション中の商品の名前、説明、およびイメージを表示します。この「オークション - アイテムの表示」ページでは、顧客は商品に対する入札のためのオークション・ルールにアクセスすることもできます。オークション・ルールのページからは、顧客は他のリンクを使用して、オークション中の商品の入札を作成することができます。

コマンド

auc_ItemDisplay.jsp は以下のコマンドを使用します。

- DisplayAuctionRules

bean

auc_ItemDisplay.jsp は、以下の bean を使用します。

- AuctionInfoDataBean
- AuctionInfoListBean
- ItemDataBean

インプリメンテーション

auc_ItemDisplay.jspは、AuctionInfoDataBean、AuctionInfoListBean、および ItemDataBean を使用して、データベースから、オークションの名前、オークション中のアイテムの簡略および詳細説明、そして商品のイメージの情報を検索します。

顧客が「オークション・ルール」をクリックすると、DisplayAuctionRules コマンドが呼び出され、「オークション・ルール」ページ (auc_rule.jsp) が表示されます。

「オークション - エラー入札送信のフレーム」ページ

「オークション - エラー入札送信のフレーム」ページ (auc_err_bid_submit_frame.jsp) は、最高入札情報、エラー・メッセージ、および入札または自動入札送信フォームを表示するフレームセットです。オークション・タイプがオープン・クライ・オークションである場合、トップ・フレームには、「オークション - 最高入札の表示」ページ (auc_show_best_bid.jsp) を使用して、最高入札が表示されます。シールド・ビッド・オークションおよびダッチ・オークションの場合、トップ・フレームは隠されます。

入札の送信がエラーになった場合、エラー・メッセージが先頭フレームに、入札送信フォームが最終フォームに表示されます。顧客は、入札送信フォームを使用して、情報を訂正し、入札を再送信することができます。

顧客が自動入札を送信する場合、入札送信フォームの代わりに最終フレーム内に自動入札フォーム・ページ (auc_authobidform.jsp) が表示されます。

コマンド

auc_err_bid_submit_frame.jsp はこのコマンドを使用しません。

bean

auc_err_bid_submit_frame.jsp は、以下の bean を使用します。

- ErrorDataBean

インプリメンテーション

入札フォーム・ページ (auc_bidform.jsp) または自動入札フォーム・ページ (auc_autobidform.jsp) で BidSubmit または AutoBidSubmit コマンドが失敗すると、auc_err_bid_submit_frame.jsp は最高入札 (auc_show_best_bid.jsp) ページの上部のフレームに、該当するエラー・メッセージを表示します。auc_err_bid_submit_frame.jspは ErrorDataBean を使用して、オークション - 最高入札の表示ページ (auc_show_best_bid.jsp) で表示される、該当するエラー・メッセージを検索します。

「オークション - ErrorDataBean 共通」ページ

「オークション - ErrorDataBean 共通」ページ (auc_edb_common.jsp) は、異なる各国語およびストアをサポートするために、「オークション - 入札の確認」ページ (auc_bid_ack.jsp) および「オークション - 最高入札の表示」ページ (auc_show_best_bid.jsp) によって使用されます。

「オークション - 共通」ページは、顧客が選択するストアと言語に基づき、メッセージを表示するときに使用するプロパティ・ファイルを決定します。このページには、ロケール、言語、ストア名、ストア ID、および顧客 ID といった共通情報も示されます。

コマンド

auc_edb_common.jsp は、コマンドを使用しません。

bean

auc_edb_common.jsp は、以下の bean を使用します。

- StoreDataBean
- ErrorDataBean

インプリメンテーション

auc_edb_common.jsp は、顧客に直接情報を表示しません。その代わりに、auc_edb_common.jspは ErrorDataBean を使用して、顧客およびストアについての情報を入手します。

「オークション・ヘッダー」ページ

「オークション・ヘッダー」ページ (auctionheader.jsp) には、ストア ID によって判別されたストアの名前とロゴが示されます。「オークション・ヘッダー」ページは、すべてのオークション・ページの左上のフレームに示されます。

コマンド

auctionheader.jsp はコマンドを使用しません。

bean

auctionheader.jsp は bean を使用しません。

「オークション・ホーム」ページ

「オークション・ホーム」ページ (auc_home.jsp) は、オークション・ストアのホーム・ページで、オークション・ギャラリー、将来のオークション、現在のオークション、最近クローズされたオークション、すべての参加可能なオークションの入札へのリンクを提供します。この「オークション・ホーム」ページには、顧客が初めてオークションを表示するときの初期ナビゲーションがあります。「オークション・ナビゲーション」ページ (auc_navigation.jsp) にも同じページがありますが、これはすべてのオークション・ページの左側のフレームに示されます。

コマンド

auc_home.jsp は、コマンドを使用しません。

bean

auc_home.jsp は、以下の bean を使用します。

- UserRegistrationDataBean

インプリメンテーション

顧客は、「オークション・ログオン・フォーム」ページ (auctionLogonForm.jsp) を使用してログオンするか、ストアフロントからログオンすることにより、auc_home.jsp を表示します。

オークション・ギャラリー

「オークション・ギャラリー」をクリックすると、「オークション・ギャラリー」ページ (auc_shopper_auction_list.jsp) が表示されます。

将来のオークション

「将来のオークション」をクリックすると、「すべてのオークション・リスト」ページ (auc_all_auction_list.jsp) が表示され、将来のオークションのリストが示されます。

現在のオークション

「現在のオークション」をクリックすると、「すべてのオークション・リスト」ページ (auc_all_auction_list.jsp) が表示され、現在のオークションのリストが表示されます。

最近クローズされたオークション

「最近クローズされたオークション」をクリックすると、「すべてのオークション・リスト」ページ (auc_all_auction_list.jsp) が表示され、最近クローズされたオークションのリストが表示されます。

お客様の入札

「お客様の入札」をクリックすると、「ユーザーのオークションの入札」ページ (auc_shopper_bid_list.jsp) が表示されます。

auc_home.jspは UserRegistrationDataBean を使用して、ユーザー ID などの顧客の登録情報を検索します。

「オークション - ログイン・フォーム」ページ

「オークション - ログイン・フォーム」ページ (AuctionLoginForm.jsp) は、ストアのオークションを表示するためにログインするために顧客によって使用されます。顧客が有効なユーザー名およびパスワードを入力し、「ログイン」をクリックすると、「オークション - ストア選択」ページ (auc_storeselection.jsp) が表示されます。

コマンド

AuctionLoginForm.jsp は以下のコマンドを使用します。

- ログオン

インプリメンテーション

顧客が有効なユーザー名およびパスワードを提供し、「ログイン」をクリックすると、Logon コマンドが呼び出され、「オークション - ストア選択」ページ (AuctionStoreSelection.jsp) が表示されます。顧客が無効なユーザー名またはパスワードを入力する場合、AuctionLoginForm.jsp はエラー・メッセージを表示し、顧客は再びログインすることができます。

「すべてのオークション・リスト」ページ

「すべてのオークション・リスト」ページ (auc_all_auction_list.jsp) は、ストアで使用可能な将来のオークション、現在のオークション、そして最近クローズされたオークションのリストを表示します。

顧客は、以下の情報を含むこれらのオークションの詳細を見ることができます。

- 商品名
- 購入可能な数量
- オークション・タイプ
- 開始の日付と時刻
- 終了の日付と時刻
- 最高入札
- 最低落札価格

さらに、顧客はリンクをクリックすることにより、オークションの商品詳細情報やオークション・ルールを表示し、新規入札または自動入札を作成し、すでに送信した入札のリストや、同じオークションで他の顧客によって送信された入札のリストを表示できます。

コマンド

`auc_all_auction_list.jsp` は以下のコマンドを使用します。

- BidCreateForm
- AutoBidCreateForm
- ProductDisplay
- DisplayAuctionItem
- DisplayAuctionRules

bean

`auc_all_auction_list.jsp` は、次の bean を使用します。

- AuctionInfoListBean
- AuctionInfoDataBean
- ItemDataBean
- BidDataBean

インプリメンテーション

簡略説明

「簡略説明」をクリックすると `DisplayAuctionItem` コマンドが呼び出され、「オークション・アイテム表示」ページ (`auc_ItemDisplay.jsp`) が表示されます。

オークション・ルール

「オークション・ルール」をクリックすると `DisplayAuctionRules` コマンドが呼び出され、「オークション・ルール」ページ (`auc_rule.jsp`) が表示されます。

新規入札

「新規入札」をクリックすると `BidCreateForm` コマンドが呼び出され、「オークション - 入札のフォーム」ページ (`auc_bidform.jsp`) が表示されます。

新規自動入札

「新規自動入札」をクリックすると `AutoBidCreateForm` コマンドが呼び出され、「オークション - 自動入札のフォーム」ページ (`auc_autobidform.jsp`) が表示されます。

すべての入札を参照

「すべての入札を参照」をクリックすると、「オークション - すべての入札」ページ (`auc_all_bids.jsp`) が表示されます。

お客様の入札

「お客様の入札」をクリックすると、「ユーザーのオークションの入札」ページ (`auc_shopper_bid_list.jsp`) が表示されます。

`auc_shopper_auction_list.jsp` は **AuctionInfoListBean**、**AuctionInfoDataBean**、および **ItemDataBean** を使用して、オークション中のアイテムの名前、オークションのタイプ、購入可能な数量を含め、ギャラリー内にリストされている各オークションについての情報を検索します。**BidDataBean** は、入札送信期限だけでなく、最高入札情報を検索するときにも使用されます。

「Auction Message Forum Append (オークション - メッセージ・フォーラムの付加)」 ページ

「Auction Message Forum Append (オークション - メッセージ・フォーラムの付加)」 ページ (auc_msg_append.jsp) を使用して、顧客は、オークションに関連した既存のフォーラム・メッセージに回答することができます。このページは、顧客がメッセージを入力することができるフォームを表示します。

「Auction Message Forum Append (オークション - メッセージ・フォーラムの付加)」 ページから、ディスカッション・フォーラム・メッセージに回答するには「送信」を、入力した情報を消去するには「リセット」を、メッセージに回答しないで「Auction Message Forum List (オークション - メッセージ・フォーラム・リスト)」 ページ (auc_shopper_msg_list.jsp) を表示するには「メッセージ・リスト」を、それぞれクリックすることができます。

コマンド

auc_msg_append.jsp は以下のコマンドを使用します。

- ProductDisplay
- DisplayAuctionItem
- UpdateGallery
- 入札作成フォーム
- AutoBidCreateForm

bean

auc_msg_append.jsp は、以下の bean を使用します。

- AuctionDataBean
- ItemDataBean
- ForumMessageDataBean
- UserRegistrationDataBean

インプリメンテーション

「新規メッセージの追加」をクリックすると、「オークション - メッセージの追加」 ページ (auc_shopper_msg_add.jsp) が表示されます。

「新規入札」をクリックすると BidCreateForm コマンドが呼び出され、「オークション - 入札のフォーム」 ページ (auc_bidform.jsp) が表示されます。

「新規自動入札」をクリックすると AutoBidCreateForm コマンドが呼び出され、「オークション - 自動入札のフォーム」 ページ (auc_autobidform.jsp) が表示されます。

「商品詳細情報に移動」をクリックすると DisplayAuctionItem コマンドが呼び出され、「オークション・アイテム表示」 ページ (auc_ItemDisplay.jsp) が表示されます。

「オークション・ギャラリーに追加」リンクは、オークションが顧客のギャラリー内にない場合にだけ表示されます。「Add to Gallery (ギャラリーに追加)」をクリックすると UpdateGallery コマンドが呼び出され、オークション・ギャラリーに追加されたオークションが載せられた「オークション・ギャラリー」 ページ (auc_shopper_list.jsp) が表示されます。

「送信」をクリックすると、メッセージが追加された「Auction Message Forum List (オークション - メッセージ・フォーラム・リスト)」ページ (auc_shopper_msg_list.jsp) が表示されます。

「リセット」をクリックすると、メッセージ・ボックス内が消去されます。

「メッセージ・リスト」をクリックすると、「Auction Message Forum List (オークション - メッセージ・フォーラム・リスト)」ページ (auc_shopper_msg_list.jsp) が表示されます。

auc_msg_append.jspは AuctionDataBean および ItemDataBean を使用して、オークションの名前とタイプ、購入可能な数量、およびオークション中のアイテムなど、オークションに関連した情報をデータベースから検索します。ForumMessageDataBean は、顧客が応答する既存のフォーラム・メッセージの内容を検索するときに使用します。 UserRegistrationDataBean は、ユーザー ID など、顧客の登録情報を検索するときに使用します。

「オークション - メッセージ共通」ページ

「オークション - メッセージ共通」ページ (auc_msg_common.jsp) は、ロケール、各国語、ストア ID、および顧客 ID などのオークション通知メッセージの共通情報を提供します。「オークション - 共通」ページでは、顧客が選択するストアおよび言語に基づいて、オークション・ページが使用するプロパティ・ファイルも判別します。

コマンド

auc_msg_common.jsp は、コマンドを使用しません。

bean

auc_msg_common.jsp は、bean を使用しません。

インプリメンテーション

auc_msg_common.jspは、顧客により直接呼び出されることはありません。このページは、以下のオークション・ページがスケジューラーを介して呼び出されるときに、呼び出されます。

- 「オークションの開始」ページ (StartAuction.jsp)
- 「開始オークションの件名」ページ (StartAuctionSubject.jsp)
- 「落札者」ページ (Winner.jsp)
- 「落札者件名」ページ (WinnerSubject.jsp)
- 「入札のオーバーライド」ページ (BidOverride.jsp)
- 「入札のオーバーライド件名」ページ (BidOverrideSubject.jsp)
- 「オーダーの完了」ページ (CompleteOrder.jsp)
- 「完了オーダーの件名」ページ (CompleteOrderSubject.jsp)
- 「オークション - メールを表示」ページ (auc_mail_display.jsp)

「オークション - メールを表示」ページ

「オークション - メールを表示」ページ (auc_mail_display.jsp) では、オークション通知メッセージを顧客に表示します。メッセージには以下の 4 つのタイプがあります。

メッセージ・タイプ 1

オークションが開始されました。 以下を含むオークション開始メッセージが表示されます。

- オークションの参照番号
- 数量
- 開始および終了日付、時刻
- オークションの説明
- マーチャントまたはセラー情報 (名前および住所など)
- 商品情報 (説明、重量、長さ、幅、および配送コード)

さらに、このメッセージによって、顧客は「オークション・ルール」ページ (auc_rule.jsp) および「オークション - アイテムの表示」ページ (auc_ItemDisplay.jsp) にアクセスできます。

メッセージ・タイプ 2

オークションを落札しました。 以下を含むオークション落札者メッセージが表示されます。

- オークション情報
- マーチャントまたはセラー情報
- 顧客情報 (名、姓、およびミドル・ネームなど)
- 顧客入札情報 (数量、アイテムごとの値、日時、状況、および支払方法)

さらに、このメッセージによって、顧客は「オークション - アイテムの表示」ページ (auc_ItemDisplay) にアクセスできます。

メッセージ・タイプ 3

お客様の入札は変更されました。 以下を含む入札オーバーライド・メッセージが表示されます。

- 数量および入札情報を含む、オークション情報
- 商品詳細情報
- マーチャントまたはセラー情報
- 顧客情報
- 現在の最高入札

さらに、このメッセージによって、「オークション - すべての入札」ページ (auc_all_bids.jsp) および「Auction Item Display」ページにアクセスできるようになります。

メッセージ・タイプ 4

オーダーが作成されました。 以下を含むオーダー作成メッセージが表示されます。

- オークション情報
- マーチャントまたはセラー情報
- 顧客情報
- 商品詳細情報
- 入札情報
- Order

さらに、このメッセージによって、顧客は「オークション - アイテムの表示」ページ (auc_ItemDisplay.jsp) にアクセスできます。

「オークション - メール表示」ページでは、顧客が自分のメールボックスからメッセージを削除したり、「オークション - メール・リスト」ページ (auc_maillist.jsp) にアクセスすることができます。

コマンド

auc_mail_display.jsp は以下のコマンドを使用します。

- DeleteMail
- ProductDisplay
- DisplayAuctionItem
- DisplayAuctionRules
- DisplayAuctionBids

bean

auc_mail_display.jsp は、以下の bean を使用します。

- MessageInfoDataBean
- BidDataBean
- AuctionDataBean
- ItemDataBean

インプリメンテーション

auc_mail_display.jspは、オークションの状況に基づき、特定のオークションについての適切なオークション情報、入札情報、および商品情報を顧客に表示します。

このメッセージの削除

「このメッセージの削除」をクリックすると DeleteMail コマンドが呼び出され、選択したメッセージを削除し、auc_mail_display.jsp を最新表示します。

メールボックスに戻る

「メールボックスに戻る」をクリックすると、「オークション - メール・リスト」ページ (auc_maillist.jsp) が表示されます。

Short description of item (アイテムの簡略説明)

「Short description of item (アイテムの簡略説明)」をクリックすると DisplayAuctionItem コマンドが呼び出され、「オークション - アイテムの表示」ページ (auc_ItemDisplay.jsp) が表示されます。

オークション・ルール

「オークション・ルール」をクリックすると DisplayAuctionRules コマンドが呼び出され、「オークション・ルール」ページ (auc_rule.jsp) が表示されます。

入札情報

「入札情報」をクリックすると DisplayAuctionBids コマンドが呼び出され、「オークション - すべての入札」ページ (auc_all_bids.jsp) が表示されます。

auc_mail_display.jspは MessageInfoDataBean を使用して、表示するメッセージのタイプに固有のデータベースから、送信側と宛先の情報、メッセージの件名、およびその内容などの情報を検索します。AuctionDataBean および ItemDataBean は、オークションの名前とタイプ、購入可能な数量、およびオークション中のアイテムなど、オークションに関連した情報を検索するときに使用されます。 BidDataBean は、入札情報を検索するときに使用されます。

「Auction Message Forum Add (オークション - メッセージ・フォーラムの追加)」 ページ

「Auction Message Forum Add (オークション - メッセージ・フォーラムの追加)」 ページ (auc_msg_add.jsp) により、顧客は、新規ディスカッション・メッセージをオークションのフォーラムに追加することができます。このページは、顧客が件名およびメッセージを入力できるフォームを表示します。

「Auction Message Forum Add (オークション - メッセージ・フォーラムの追加)」 ページでは、顧客は、メッセージをフォーラムへ送信するときには「送信」を、入力した情報を消去するときには「リセット」を、そして新規メッセージを追加せずに「Auction Message Forum List (オークション - メッセージ・フォーラム・リスト)」 ページ (auc_shopper_msg_list.jsp) を表示するときには「キャンセル」をクリックできます。

コマンド

auc_msg_add.jsp は、以下のコマンドを使用します。

- ProductDisplay
- DisplayAuctionItem
- GalleryAdd
- BidCreateForm
- AutoBidCreateForm

bean

auc_msg_add.jsp は、以下の bean を使用します。

- AuctionDataBean
- ItemDataBean

インプリメンテーション

「新規入札」をクリックすると BidCreateForm コマンドが呼び出され、「オークション - 入札のフォーラム」 ページ (auc_bidform.jsp) が表示されます。

「新規自動入札」をクリックすると AutoBidCreateForm コマンドが呼び出され、「オークション - 自動入札のフォーラム」 ページ (auc_autobidform.jsp) が表示されます。

「商品詳細情報に移動」をクリックすると DisplayAuctionItem コマンドが呼び出され、「オークション・アイテム表示」 ページ (auc_ItemDisplay.jsp) が表示されます。

「オークション・ギャラリーに追加」リンクは、オークションが顧客のギャラリー内にはない場合にだけ表示されます。「Add to Gallery (ギャラリーに追加)」をクリックすると UpdateGallery コマンドが呼び出され、オークション・ギャラリーに追加されたオークションが載せられた「オークション・ギャラリー」 ページ (auc_shopper_list.jsp) が表示されます。

「メッセージ・リスト」をクリックすると、「Auction Message Forum List (オークション - メッセージ・フォーラム・リスト)」 ページ (auc_shopper_msg_list.jsp) が表示されます。

auc_msg_append.jspは AuctionDataBean および ItemDataBean を使用して、オークションの名前とタイプ、購入可能な数量、およびオークション中のアイテムなど、オークションに関連した情報をデータベースから検索します。

「オークション・メイン・ホーム」ページ

「オークション・メイン・ホーム」ページ (auc_main_home.jsp) は、すべてのオークション・サンプル・ストア・ページの基本フレームセットです。左上のフレームには、ストアの名前と会社のロゴなどのヘッダー情報が表示されます (auctionheader.jsp)。右上のフレームには、ショッピング・カートなど、ストアの残りの 1 次機能へリンクするツールバーが表示されます (auctionbuttonbar.jsp)。左方フレームには、オークション機能、メッセージ、およびメイン・ストアへのリンクや、現在の日付と時刻が示されたナビゲーション・ページが表示されます (navigation.jsp)。

右方フレームは、顧客との対話に応じて、リスト、送信入札フォーム、およびその他のオークション・サンプル・ページを最新表示します。

コマンド

auc_main_home.jsp は以下のコマンドを使用します。

- DisplayAuctionRules

bean

auc_main_home.jsp は、bean を使用しません。

インプリメンテーション

顧客は、ストアフロントのツールバーからメインの「オークション」リンクをクリックするか、オークション中の商品の商品ページから「オークション・ルール」リンクをクリックすることにより、オークションをオープンできます。

顧客がメインの「オークション」リンクをクリックすると、auc_main_home.jsp がオープンし、右方フレームに現在のオークションのリストが表示されます (all_auctions_list.jsp)。

顧客が「オークション・ルール」をクリックすると、DisplayAuctionRules コマンドが呼び出され、auc_main_home.jspがオープンし、右方フレームにオークション中の商品のオークション・ルールが表示されます (auc_rule.jsp)。

「オークション - メール・リスト」ページ

「オークション - メール・リスト」ページ (auc_maillist.jsp) では、顧客が、オークションに関連したメッセージを表示および削除することができます。「オークション - メール・リスト」ページは、ナビゲーション・ページ (auc_navigation.jsp) から、「メッセージ」リンクをクリックしてアクセスします。

「オークション・メール・リスト」ページでは、以下を含む、顧客入札に関連した通知メッセージのリストを表示します。

- オークションの自動入札の開始
- 入札のオーバーライド
- オークションの入札落札
- 完了したオーダー詳細情報

このページには、日付、メールボックス内にあるメッセージ顧客の合計数、および未読メッセージ数も表示されます。未読メッセージには、アスタリスク (*) が付いています。さらに、このページでは、顧客が自分のメールボックスからメッセージを削除することができます。

「オークション - メール・リスト」ページでは、各ページに最大 5 つのメッセージを表示します。「次ページ」および「前ページ」をクリックして、ページ間をナビゲートできます。

コマンド

auc_maillist.jsp は、以下のコマンドを使用します。

- DisplayMessage
- DeleteMail

bean

auc_maillist.jsp は、以下の bean を使用します。

- MessageInfoListBean
- MessageCountListBean
- MessageCountDataBean
- MessageInfoDataBean
- UserRegistrationDataBean

インプリメンテーション

auc_maillist.jspは、MessageInfoListBean、MessageInfoDataBean および UserRegistrationDataBean を使用し、WebSphere Commerce 構成マネージャーで指定した SMTP サーバーから、登録ユーザーのメッセージ・リストとメッセージ情報を検索します。

メッセージの内容を表示する場合、顧客は件名ヘッダーをクリックします。顧客が件名ヘッダーをクリックすると、Display Message コマンドが呼び出されます。これは、新しいメッセージから古いメッセージまでメッセージの状況を変更し、メッセージの内容の表示 (auc_mail_display.jsp) を行います。

顧客がメッセージを表示したら、そのメッセージは既読とマークされ、削除が可能になります。該当するチェック・ボックスをチェックし、「アイテムの除去」をクリックして、メールボックスから 1 つ以上のメッセージを除去することができます。「アイテムの除去」をクリックすると、DeleteMail コマンドが呼び出され、チェックしたメッセージを除去してページを最新表示します。

auc_maillist.jspは、MessageCountListBean および MessageCountDataBean を使用し、顧客が使用可能なメッセージの数についての情報を検索します。表示するメッセージがなければ、auc_maillist.jsp は適切なメッセージを表示します。

「オークション・ナビゲーション」ページ

「オークション・ナビゲーション」ページ (auc_navigation.jsp) では、オークション・ギャラリー、将来についての詳細情報、現在のオークションと最近クローズされたオークション、そして顧客がすべてのオークションで送信した入札のリストを含め、オークション・ストア・ページへアクセスできます。

コマンド

- LogonForm

- StoreCatalogDisplay

bean

auc_navigation.jsp は、bean を使用しません。

インプリメンテーション

登録 「登録」をクリックすると、LoginForm コマンドが呼び出され、「登録」ページまたはログイン・ページがオープンします。

ヘルプ 「ヘルプ」をクリックすると、WebSphere Commerce のヘルプ・ページがオープンします。

将来のオークション

「将来のオークション」をクリックすると、「すべてのオークション・リスト」ページ (auc_all_auction_list.jsp) が表示され、将来のオークションのリストが示されます。

現在のオークション

「現在のオークション」をクリックすると、「すべてのオークション・リスト」ページ (auc_all_auction_list.jsp) が表示され、現在のオークションのリストが示されます。

クローズされたオークション

「クローズされたオークション」をクリックすると、「すべてのオークション・リスト」ページ (auc_all_auction_list.jsp) が表示され、最近クローズされたオークションのリストが示されます。

Your Auction Gallery (ユーザーのオークション・ギャラリー)

「オークション・ギャラリー」をクリックすると、「オークション・ギャラリー」ページ (auc_shopper_auction_list.jsp) が表示されます。

ユーザーのオークションの入札

「お客様の入札」をクリックすると、「ユーザーのオークションの入札」ページ (auc_shopper_bid_list.jsp) が表示されます。

メッセージ

「メッセージ」をクリックすると、「オークション - メールの表示」ページ (auc_mail_display.jsp) が表示されます。

Return to Store (ストアに戻る)

「Return to Store (ストアに戻る)」をクリックすると StoreDisplay コマンドが呼び出され、ストアのホーム・ページがオープンします。

「入札のオーバーライド」ページ

オークション中に顧客の入札がオーバーライドされる場合、オークション・コンポーネントは顧客オーバーライドを知らせるメッセージを送信します。このメッセージは、件名および本体の 2 つの部分からなります。

「入札のオーバーライド」ページ (BidOverride.jsp) は、メッセージの本体を形式化するために使用され、以下の情報を含んでいます。

- 数量、開始日および終了日、説明、および更新時刻を含む、オークション情報。
- ストア名、説明、E メール・アドレス、FAX および電話番号、番地、市区町村、都道府県、および郵便番号を含む、マーチャントまたはセラー情報。
- 名、ミドル・ネーム、および姓を含む、顧客情報。

- SKU、説明、重量、重量単位、長さ、幅、配送コード、および URL を含む、商品情報。
- 数量、値、日時、状況、支払いメソッド、および落札値を含む。入札情報。

コマンド

BidOverride.jsp は、コマンドを使用しません。

bean

BidOverride.jspは、以下の bean を使用します。

- StoreDataBean BidDataBean
- AuctionDataBean
- UserRegistrationDataBean
- ItemDataBean
- LanguageAccessBean
- StoreEntityDescriptionAccessBean
- StoreAddressAccessBean
- CatalogEntryShippingAccessBean

インプリメンテーション

BidOverride.jspは、顧客により直接呼び出されることはありません。このページは、スケジューラーを通してメッセージング・システムにより呼び出されます。「入札のオーバーライド」ページの情報の内容は、上記でリストされている databean を使用して、データベースから検索され、ストリングとして保管されません。「オークション - メールを表示」ページ (auc_mail_display.jsp) は、このデータベースからストリング情報を検索し、「入札のオーバーライド」オークション通知メッセージの内容として表示します。

「開始オークションの件名」ページ

オークションを開始する際、オークションのコンポーネントはギャラリーにそのオークションを書き込んでいる顧客にメッセージを送信し、オークションを開始したことを通知します。このメッセージは、件名および本体の 2 つの部分からなります。

「開始オークションの件名」ページ (StartAuctionSubject.jsp) は、メッセージの件名を形式化するために使用されます。

コマンド

StartAuctionSubject.jsp は、コマンドを使用しません。

bean

StartAuctionSubject.jsp は、以下の bean を使用します。

- AuctionDataBean
- StoreDataBean
- LanguageAccessBean

インプリメンテーション

StartAuctionSubject.jspは、顧客により直接呼び出されることはありません。このページは、スケジューラーを通してメッセージング・システムにより呼び出されます。「Start Auction (オークションの開始)」メッセージの件名情報は、StoreDataBean と LanguageAccessBean を使用して、データベースから検索され、ストリングとして保管されます。「オークション - メールを表示」ページ (auc_mail_display.jsp) は、このデータベースからストリング情報を検索し、「Start Auction (オークションの開始)」通知メッセージの件名として表示します。

「オークションの開始」ページ

オークションを開始する際、オークションのコンポーネントはギャラリーにそのオークションを書き込んでいる顧客にメッセージを送信し、オークションを開始したことを通知します。このメッセージは、件名および本体の 2 つの部分からなります。

「オークションの開始」ページ (StartAuction.jsp) は、メッセージの本体を形式化するために使用され、以下の情報を提供します。

- 数量、開始日および終了日、説明、および更新時刻を含む、オークション情報。
- ストア名、説明、E メール・アドレス、FAX および電話番号、番地、市区町村、都道府県、および郵便番号を含む、マーチャントまたはセラー情報。
- SKU、説明、重量、重量単位、長さ、幅、配送コード、および URL を含む、商品情報。

コマンド

StartAuction.jsp は、コマンドを使用しません。

bean

StartAuction.jsp は、以下の bean を使用します。

- AuctionDataBean
- StoreDataBean
- ItemDataBean
- StoreEntityDescriptionAccessBean
- StoreAddressAccessBean
- CatalogEntryShippingAccessBean
- LanguageAccessBean

インプリメンテーション

StartAuction.jspは、顧客により直接呼び出されることはありません。このページは、スケジューラーを通してメッセージング・システムにより呼び出されます。「Start Auction (オークションの開始)」ページの内容は、上記でリストされている databean を使用して、データベースから検索され、ストリングとして保管されます。「オークション - メールを表示」ページ (auc_mail_display.jsp) は、このデータベースからストリング情報を検索し、「Start Auction (オークションの開始)」通知メッセージの内容として表示します。

「オークション - 最高入札の表示」ページ

「オークション - 最高入札の表示」ページ (auc_show_best_bid.jsp) では、オークションの最高および最低の入札値を表示することができます。さらにこのページでは、顧客が送信済みの入札または自動入札が、無効データのために正常に作成または変更されない場合に、該当するメッセージが表示されます。

コマンド

auc_show_best_bid.jsp は、コマンドを使用しません。

bean

auc_show_best_bid.jsp は以下の bean を使用します。

- BidDataBean
- AutoBidDataBean
- AuctionDataBean
- ItemDataBean

インプリメンテーション

auc_show_best_bid.jspは、BidDataBean、AutoBidDataBean、AuctionDataBean、および ItemDataBean を使用し、最高および最低落札額、オークションの名前、およびオークション中の商品についての情報を検索します。顧客が、「オークション - 入札フォーム」ページ (auc_bidform.jsp) か、「オークション - 自動入札フォーム」ページ (auc_autobidform.jsp) を使用して、無効な情報を送信する場合、auc_show_best_bid.jsp は関連するエラー・メッセージも表示します。その他の場合には、auc_show_best_bid.jsp には高額および最高の入札値のみが表示されます。

ユーザーのオークションの入札

「ユーザーのオークションの入札」ページ (auc_shopper_bid_list.jsp) には、現在のオークション、将来のオークション、または最近クローズされたオークションで顧客が送信した、すべての自動入札および入札のリストが表示されます。顧客は自動入札および入札を確認し、まだアクティブな入札を変更または撤回することができます。

「オークション・ナビゲーション」ページ (auc_navigation.jsp)で「お客様の入札」をクリックすると、すべてのオークションの自動入札および入札の詳細を含むリストが、「ユーザーのオークションの入札」ページに表示されます。そうでない場合、「すべてのオークション・リスト」ページ (auc_all_auction_list.jsp) か、「オークション・ギャラリー」ページ (auc_shopper_auction_list.jsp)で「お客様の入札」をクリックすると、「ユーザーのオークションの入札」ページには、選択したオークションだけに関連する詳細が示されます。

「ユーザーのオークションの入札」ページから、顧客は以下のような自動入札および入札の詳細情報を表示することができます。

- 商品説明
- 自動入札の上限
- 自動入札の初期金額
- 入札の商品数量
- 自動入札または入札が送信された日付と時刻

- 自動入札番号または入札番号

さらにこのページでは、顧客が「**アイテムの説明**」、「**すべての入札を参照**」、「**自動入札の変更**」、および「**入札の変更**」をクリックして、商品詳細情報を表示したり、オークションに送信済みの入札を表示し、自動入札または入札を変更することができます。加えてこのページでは、顧客を現在、将来、および最近クローズされたオークションのリスト間をナビゲートします。

「ユーザーのオークションの入札」ページには、1 ページにつき最大 5 件の自動入札および入札が表示されます。「**次ページ**」および「**前ページ**」をクリックして、ページ間をナビゲートできます。

コマンド

auc_shopper_bid_list.jsp は、以下のコマンドを使用します。

- AutoBidDelete
- BidDelete
- ProductDisplay
- DisplayAuctionItem
- AutoBidUpdateForm
- BidUpdateForm

bean

auc_shopper_bid_list.jsp は、次に bean を使用します。

- UserRegistrationDataBean
- AutoBidDataBean
- AutoBidListBean
- AuctionDataBean
- BidDataBean
- BidListBean
- ItemDataBean

インプリメンテーション

自動入札の撤回

「**自動入札の撤回**」をクリックすると AutoBidDelete コマンドが呼び出され、選択した自動入札がリストから除去され、ページが最新表示されます。「**入札の撤回**」をクリックすると BidDelete コマンドが呼び出され、選択した入札がリストから除去され、ページが最新表示されます。auc_shopper_bid_list.jsp では、ダッチ・オークションまたは最近クローズされたオークションへ送信された入札を撤回することはできません。

自動入札の変更

「**自動入札の変更**」をクリックすると AutoBidUpdateForm コマンドが呼び出され、変更可能な既存の自動入札情報が示された「オークション - 自動入札のフォーム」ページ (auc_autobidform.jsp) が表示されます。「**入札の変更**」をクリックすると BidUpdateForm コマンドが呼び出され、「オークション - 入札のフォーム」ページ (auc_bidform.jsp) が表示されます。

すべての入札を参照

「すべての入札を参照」をクリックすると、「オークション - すべての入札」ページ (auc_all_bids.jsp) が表示されます。

アイテムの説明

「アイテムの説明」をクリックすると DisplayAuctionItem コマンドが呼び出され、「オークション・アイテム表示」ページ (auc_ItemDisplay.jsp) が表示されます。

auc_shopper_bid_list.jspは、AuctionDataBean および ItemDataBean を使用し、オークションの名前とタイプ、購入可能な数量、オークション中のアイテムの名前と説明など、オークション関連情報を検索します。AutoBidDataBean、AutoBidListBean、BidDataBean、および BidListBean は、入札および自動入札値、その入札または自動入札を設定した日付を含め、顧客が送信した入札および自動入札に関連した情報を検索するときに使用されます。UserRegistrationDataBean は、ユーザー ID など、ユーザーの登録情報を検索するときに使用されます。

「オークション・ギャラリー」ページ

「オークション・ギャラリー」ページ (auc_shopper_auction_list.jsp) では、顧客は関心のあるオークションを見て回ることができます。このページでは、顧客は特定のストアに関連した将来のオークション、現在のオークション、および最近クローズされたオークションのリストを確認できます。

「オークション・ギャラリー」ページから、以下のようなオークションの詳細を表示できます。

- 商品説明
- 数量
- オークション・タイプ
- 開始日付と時刻
- 終了日付と時刻
- 最高入札
- 最低の落札

また「アイテムの説明」、「オークション・ルール」、「すべての入札を参照」、「新規入札」、「新規自動入札」、および「お客様の入札」をクリックすると該当するページにアクセスできます。さらに、「除去」をクリックすると、ギャラリーから 1 つ以上のオークションを除去することができます。

コマンド

auc_shopper_auction_list.jsp は以下のコマンドを使用します。

- ProductDisplay
- DisplayAuctionItem
- DisplayAuctionRules
- 入札作成フォーム
- 自動入札作成フォーム
- GalleryDelete

bean

auc_shopper_auction_list.jsp は以下の bean を使用します。

- UserRegistrationDataBean

- AuctionInfoListBean
- AuctionInfoDataBean
- BidDataBean
- ItemDataBean

インプリメンテーション

簡略説明

オークションが現在行なわれている場合、「簡略説明」をクリックすると ProductDisplay コマンドが呼び出され、商品の簡略説明が表示されます (ProductDescription.jsp)。オークションが将来であるかクローズされている場合、「簡略説明」をクリックすると DisplayAuctionItem コマンドが呼び出され、「オークション・アイテム表示」ページ (auc_ItemDisplay.jsp) が表示されます。

オークション・ルール

「オークション・ルール」をクリックすると DisplayAuctionRules コマンドが呼び出され、「オークション・ルール」ページ (auc_rule.jsp) が表示されます。

新規入札

「新規入札」をクリックすると BidCreateForm コマンドが呼び出され、「オークション - 入札のフォーム」ページ (auc_bidform.jsp) が表示されます。

新規自動入札

「新規自動入札」をクリックすると AutoBidCreateForm コマンドが呼び出され、「オークション - 自動入札のフォーム」ページ (auc_autobidform.jsp) が表示されます。

すべての入札を参照

「すべての入札を参照」をクリックすると、「オークション - すべての入札」ページ (auc_all_bids.jsp) が表示されます。

お客様の入札

「お客様の入札」をクリックすると、「ユーザーのオークションの入札」ページ (auc_shopper_bid_list.jsp) が表示されます。

アイテムの除去

「アイテムの除去」をクリックすると、GalleryDelete コマンドが呼び出され、選択したオークションがギャラリーから除去され、auc_shopper_auction_list.jsp が最新表示されます。

- UserRegistrationDataBean
- AuctionInfoListBean
- AuctionInfoDataBean
- BidDataBean
- ItemDataBean

auc_shopper_auction_list.jspは UserRegistrationDataBean を使用して、登録したユーザー ID についての情報を検索します。 AuctionInfoListBean、AuctionInfoDataBean、および ItemDataBean は、オークション中のアイテムの名前、オークションのタイプ、購入可能な数量を含め、ギャラリー内にリストされている各オークションについての情報を検索するときに使用されます。 BidDataBean は、入札送信期限だけでなく、最高入札情報を検索するときにも使用されます。

「Auction Message Forum List (オークション - メッセージ・フォーラム・リスト)」ページ

「オークション - ショッパーのメッセージ・リスト」ページ (auc_shopper_msg_list.jsp) では、オークションに関連したフォーラム・メッセージのリストを表示できます。

「オークション - ショッパーのメッセージ・リスト」には、各メッセージまたは応答についての以下の事柄が表示されます。

- メッセージ ID
- メッセージを送信したまたはそれに応答した人の姓
- メッセージを送信したまたはそれに応答した日付と時刻
- メッセージの件名

加えて、「オークション - ショッパーのメッセージ・リスト」ページでは、以下のタスクを実行できます。

- メッセージへの応答
- 新規メッセージの作成
- 新規入札または自動入札の作成
- ギャラリーへのオークションの追加
- 商品詳細情報の表示

「オークション - ショッパーのメッセージ・リスト」ページでは、1 ページに最大 15 のメッセージを表示します。「次ページ」および「前ページ」をクリックして、ページ間をナビゲートできます。

コマンド

auc_shopper_msg_list.jsp は以下のコマンドを使用します。

- ProductDisplay
- DisplayAuctionItem
- UpdateGallery
- 入札作成フォーム
- AutoBidCreateForm

bean

auc_shopper_msg_list.jsp は以下の bean を使用します。

- AuctionDataBean
- ItemDataBean
- UserRegistrationDataBean
- ForumMessageLightListBean

インプリメンテーション

「**Subject of message (メッセージの件名)**」をクリックすると、「Auction Message Forum Append (オークション - メッセージ・フォーラムの付加)」ページ (auc_msg_append.jsp) が表示されます。

「新規メッセージの追加」をクリックすると、「Auction Message Forum Add (オークション - メッセージ・フォーラムの追加)」ページ (auc_shopper_msg_add.jsp) が表示されます。

「新規入札」をクリックすると BidCreateForm コマンドが呼び出され、「オークション - 入札のフォーム」ページ (auc_bidform.jsp) が表示されます。

「新規自動入札」をクリックすると AutoBidCreateForm コマンドが呼び出され、「オークション - 自動入札のフォーム」ページ (auc_autobidform.jsp) が表示されます。

オークションが現在行なわれている場合、「商品詳細情報に移動」をクリックすると ProductDisplay コマンドが呼び出され、「商品表示」ページが表示されます。オークションが将来であるかクローズされている場合、「商品詳細情報に移動」をクリックすると DisplayAuctionItem コマンドが呼び出され、「オークション・アイテム表示」ページ (auc_itemdisplay.jsp) が表示されます。

「オークション・ギャラリーに追加」リンクは、オークションが顧客のギャラリー内にない場合にだけ表示されます。「Add to Gallery (ギャラリーに追加)」をクリックすると UpdateGallery コマンドが呼び出され、オークション・ギャラリーに追加されたオークションが載せられた「オークション・ギャラリー」ページ (auc_shopper_list.jsp) が表示されます。

auc_msg_append.jspは AuctionDataBean および ItemDataBean を使用して、オークションの名前とタイプ、購入可能な数量、およびオークション中のアイテムなど、オークションに関連した情報をデータベースから検索します。 ForumMessageDataBean は、顧客が応答する既存のフォーラム・メッセージの内容を検索するときに使用します。 UserRegistrationDataBean は、ユーザー ID など、顧客の登録情報を検索するときに使用します。

「落札者」ページ

顧客がオークションを落札すると、オークションのコンポーネントは顧客が落札したことを通知する、オークション通知メッセージを送信します。このメッセージは、件名および本体の 2 つの部分からなります。

「落札者」ページ (winner.jsp) は、落札者オークション通知メッセージの本体を形式化するために使用され、以下の情報を提供します。

- 数量、開始日、終了日、時刻、説明、および更新時刻を含む、オークション情報。
- ストア、説明、E メール・アドレス、FAX 番号、電話番号、番地、市区町村、都道府県、および郵便番号を含む、マーチャントまたはセラー情報。
- 名、ミドル・ネーム、および姓を含む、顧客情報。
- SKU、説明、重量、重量単位、長さ、幅、配送コード、および URL を含む、商品情報。
- 数量、値、日時、状況、支払いメソッド、および落札値を含む。入札情報。

コマンド

winner.jsp は、コマンドを使用しません。

bean

winner.jsp は、以下の bean を使用します。

- BidDataBean
- StoreDataBean
- ItemDataBean

- AuctionDataBean
- LanguageAccessBean
- StoreAddressAccessBean
- StoreEntityDescriptionAccessBean
- CatalogEntryShippingAccessBean
- StoreAddressAccessBean
- UserRegistrationDataBean

インプリメンテーション

winner.jspは、顧客により直接呼び出されることはありません。このページは、スケジューラーを通してメッセージング・システムにより呼び出されます。「落札者」ページの情報の内容は、上記でリストされている databean を使用して、データベースから検索され、ストリングとして保管されます。「オークション - メールが表示」ページ (auc_mail_display.jsp) は、このデータベースからストリング情報を検索し、「落札者」オークション通知メッセージの内容として表示します。

「落札者件名」ページ

顧客がオークションを落札すると、オークションのシステムは顧客が落札したことを通知するメッセージを送信します。このメッセージは、件名および本体の 2 つの部分からなります。

「落札者件名」ページ (WinnerSubject.jsp) は、メッセージの件名を形式化するために使用されます。

コマンド

WinnerSubject.jsp は、コマンドを使用しません。

bean

WinnerSubject.jsp は、以下の bean を使用します。

- StoreDataBean
- LanguageAccessBean

インプリメンテーション

Winner.jspは、顧客により直接呼び出されることはありません。このページは、スケジューラーを通してメッセージング・システムにより呼び出されます。落札者メッセージの件名情報は、StoreDataBean と LanguageAccessBean を使用して、データベースから検索され、ストリングとして保管されます。「オークション - メールが表示」ページ (auc_mail_display.jsp) は、このデータベースからストリング情報を検索し、落札者オークション通知メッセージの件名として表示します。

第 6 章 Sample store use cases

サンプル・ストアの使用事例

使用事例は、登録やチェックアウトなど、ストアにおける各ユーザーの対話のフローを詳述しています。サンプル・ストア InFashion、WebFashion、および NewFashion での対話を詳述している一連の使用事例は、オンライン・ヘルプに提供されています。これらの使用事例は、サンプル・ストアのフローをより完全に理解するのに役立てることができ、独自のストアの使用事例を作成するためのガイドとしても使用できます。

以下の使用事例が提供されています。

- ホーム・ページの使用事例
- 登録の使用事例
- ログオンの使用事例
- 個人アカウント管理の使用事例
- 個人情報変更の使用事例
- 商品カテゴリー・ビューの使用事例
- 商品ページ表示の使用事例
- **Professional** パッケージ・ページ表示の使用事例
- **Professional** バンドル・ページ表示の使用事例
- ショッピング・カート表示の使用事例
- ショッピング・カートのチェックアウトの使用事例
- **Professional** 高速チェックアウトの使用事例
- 住所編集の使用事例
- 新規住所追加の使用事例
- **Professional** オーダー表示の使用事例
- **Professional** 購入希望商品リストへのアイテム追加の使用事例
- **Professional** 購入希望商品リスト表示の使用事例
- **Professional** 高速チェックアウト・プロフィール作成の使用事例

ページがどのように機能するかについて、詳しくは、サンプル・ストアのショッピング・フロー・チャートを参照してください。

WebFashion クイック・チェックアウト・プロフィール作成の使用事例

登録済み顧客がクイック・チェックアウト機能を使用するには、まずクイック・チェックアウト・プロフィールを作成しなければなりません。

実行者

登録済み顧客

メイン・フロー

顧客は「アカウント」をクリックした後、「プロフィールの作成または更新」をクリックして、使用事例を開始します。

システムは「クイック・チェックアウト・プロフィール」ページを表示します。

1. 請求先住所。これには以下のフィールドが含まれます。

- 名
- 姓
- 番地
- 市区町村
- 都道府県
- 郵便番号
- 国/地域
- 電話番号

注: デフォルトの請求先住所がすでに存在する場合、ページはシステムによって事前に埋められています。

2. 配送先住所。これには以下のフィールドが含まれています。

- 名
- 姓
- 番地
- 市区町村
- 都道府県
- 郵便番号
- 国/地域
- 電話番号

注: デフォルトの配送先住所がすでに存在する場合、ページはシステムによって事前に埋められています。

配送先住所には、「請求先住所と同じ」チェック・ボックスも含まれています。チェック・ボックスが選択される場合、配送先住所は請求先住所と同じになります。

3. 利用可能な配送方法。各配送方法には、以下が含まれます。

- 配送方法の名前
- 配送料金の詳細
- 配達時間

注: 顧客が選択した最後の配送方法がデフォルトで選択されます。

4. 支払い情報。これには、以下が含まれます。

- クレジット・カード・タイプ
- カード番号
- 有効期限月
- 有効期限年

顧客は情報を入力または編集した後、「送信」をクリックします。システムは情報を保存し、住所の新規追加の使用事例で説明されているようにアドレス情報を追加します。システムは「アカウント」ページを表示します。

WebFashion クイック・チェックアウトの使用事例

顧客は以前にクイック・チェックアウト・プロフィールを作成している場合、配送情報および請求情報を入力せずにオーダーを完了することができます。

実行者

顧客

メイン・フロー

一度顧客が商品をショッピング・カートに追加したら、「クイック・チェックアウト」をクリックして、使用事例 (E1、E2) を開始します。システムは、クイック・チェックアウト: オーダー要約ページを表示します。ショッピング・カート内の各アイテムごとに、以下の情報が表示されます。

- 数量
- 商品の簡略説明
- 単価
- 合計価格

また、クイック・チェックアウト: オーダー要約ページには、オーダーに関する以下の情報も含まれます。

- 小計
- 割引 (ある場合)
- 合計税。税が複数ある場合は、個別に表示されます。
- 関税 (ある場合)
- 配送方法を含めた配送料金
- 総合計
- 請求先住所
- 配送先住所

システムは、顧客に以下の支払い情報の入力を促すプロンプトを出します。

- クレジット・カード・タイプ
- カード番号
- 有効期限月:
- 有効期限年

注:クイック・チェックアウト・プロフィール作成の使用事例で説明されているように、顧客がクイック・チェックアウト・プロフィールにクレジットカード情報を作成すると、[支払い情報] フィールドがあらかじめ入力されています。

顧客はオーダー要約を確認し、「オーダー」をクリックします。システムは、オーダー (E3) を完了します。

そして、オーダーの確認ページを表示します。

代替フロー

なし

例外フロー

E1: 顧客にクイック・チェックアウト・プロフィールがない場合

顧客はクイック・チェックアウトを完成させる前に、クイック・チェックアウト・プロフィール作成の使用事例で説明されているように、クイック・チェックアウト・プロフィールを作成しなければなりません。システムが請求先住所、配送先住所、または配送方法を見つけることができない場合、顧客にクイック・チェックアウト・プロフィールの作成を命じるエラー・メッセージを表示します。

E2: 顧客が登録されていない場合

ゲスト・ショッパーがクイック・チェックアウトの使用を試みると、システムは顧客に、クイック・チェックアウト・オプションを使用する前に登録とクイック・チェックアウト・プロフィールの作成するように命じるエラー・メッセージを表示します。

E3: クレジット・カード情報が無効な場合

クイック・チェックアウト・プロフィールに入力したクレジットカード番号が無効である場合、またはその有効期限が切れている場合、システムはエラー・メッセージを表示します。

新規住所追加の使用事例

 顧客は自分の住所録に新しい住所を追加することができます。

実行者

顧客

メイン・フロー

顧客は**新規住所の追加**をクリックします。システムは以下のフィールドのあるページを表示します。

- ニックネーム (NewFashion のみ)
- 名
- 姓
- 番地 (2 つのテキスト・ボックスで構成)
- 市区町村
- 都道府県
- 郵便番号
- 国/地域
- 電話番号

顧客はフィールドに情報を入力し、**送信**をクリックします。システムはアドレス・ブックに新規のアドレスを追加します (E1)。

代替フロー

なし

例外フロー

E1: 必須フィールドが欠落している場合

以下のフィールドが 1 つでも欠落しているとシステムはエラー・メッセージを出します。

- 名
- 姓
- 番地
- 市区町村
- 都道府県
- 郵便番号
- 国/地域

入力したニックネームが顧客の住所録にすでに存在する場合、システムはエラー・メッセージを報告します。

そしてその使用事例を最初から再開します。

WebFashion と NewFashion の購入希望商品リストへのアイテム追加の使用事例

顧客はショッピング・カートで、購入しようとして選択した商品を表示したり編集したりすることができます。

WebFashion ストアでは、購入希望商品リスト (または買い物候補リスト) により、登録済みの顧客は将来注文しようとしている商品をリストに追加することができます。NewFashion ストアでは、アイテムを購入希望商品リストへ追加する際に、顧客が登録されている必要はありません。また、購入希望商品リストは家族または友達に E メールで送ることもできます。家族または友達は、アイテムを顧客への贈答品として購入することができます。購入希望商品リストは、ショッピング・カートとは違います。ショッピング・カート内にある商品は、そのショッピング・セッション中に顧客が購入しようと考えている商品です。

実行者

登録済み顧客

メイン・フロー

顧客は商品、パッケージ、またはバンドルを表示して (詳しくは、商品ページの表示の使用事例、バンドル・ページの表示の使用事例、および パッケージ・ページの表示の使用事例を参照してください)、その後「追加希望リスト」をクリックします。購入希望商品リスト表示の使用事例で説明されているように、システムは購入希望商品リスト (E1) に商品、バンドル、またはパッケージを追加して、購入希望商品リスト・ページを表示します。

例外フロー

E1: ゲスト・ショッパーが購入希望商品リストへのアイテムの追加を試みます。

ゲスト・ショッパーが購入希望商品リストにアイテムを追加しようとする場合、システムは以下のメッセージを表示します。購入希望商品リストにアイテムを保管する場合、WebFashion に登録してからもう一度やり直してください。

NewFashion ストアでは、ゲスト・ショッパーは登録されていなくても購入希望商品リストに追加することができます。

商品カテゴリー・ビューの使用事例

カタログ・グループ・ページはサブカテゴリーおよび商品のリストを表示します。カタログ・グループ・ページは顧客が商品を比較しやすいようナビゲートします。カタログ・グループ・ページは、広い分野から始まり、続くカタログ・グループ・ページで、狭い分野を検索していきます。

カタログ・グループ・ページには通常 3 つのタイプがあります。

- 上位カテゴリーでサブカテゴリーを表示するページ
- サブカテゴリーで商品を表示するページ
- サブカテゴリーと商品の両方を表示するページ

実行者

顧客

メイン・フロー

トップレベルの商品カテゴリーはホーム・ページに表示されます。たとえばメンズ・ファッション、レディース・ファッション、新着情報などです。顧客がメンズ・ファッションかレディース・ファッションをクリックすると、システムは、選択したカテゴリー内のサブカテゴリーをデータベースから検索し、対応するカテゴリー情報を表示します。顧客が「新着」リンクをクリックすると、新しい商品が表示されます。以下の情報はそれぞれのカテゴリーおよび製品ごとに表示されます。

- サムネール・イメージ (商品のみ)
- 各カテゴリー名、または各商品名
- 商品を識別する簡単な説明
- 商品の価格

NewFashion および WebFashion のカテゴリー・ページには、パッケージまたはバンドルの形式で、フィーチャー特別ご提供商品が含まれます。フィーチャー特別ご提供商品には、パッケージまたはバンドルのイメージおよび簡単な説明が含まれます。InFashion のフィーチャー特別ご提供商品は、1 つの商品だけです。

顧客はサムネール・イメージや名前をクリックします。それによりシステムは、一致したページ (A1、A2、A3、A4) を表示します。

代替フロー

A1: 顧客がカテゴリーを選択

顧客がカテゴリーを選択すると、システムははじめからこの使用事例を要約し、別のカテゴリー・ページを表示します。

A2: 顧客が商品を選択

顧客が商品を選択すると、商品ページは商品ページ表示の使用事例で説明する商品ページを表示します。

A3: 顧客がバンドルを選択 (WebFashion および NewFashion のみ)

顧客がバンドルを選択すると、バンドル・ページ表示の使用事例で説明するバンドル・ページを表示します。

A4 顧客がパッケージを選択 (WebFashion および NewFashion のみ)

顧客がパッケージを選択すると、パッケージ・ページ表示の使用事例で説明するパッケージ・ページを表示します。

例外フロー

なし

ショッピング・カートのチェックアウトの使用事例

顧客がショッピング・カートにリストされている商品の支払いを完了するとチェックアウトし、オーダーが送信されます。

実行者

顧客

メイン・フロー

顧客が**チェックアウト**をクリックして、使用事例を開始します。

システムは請求先住所ページを表示します。顧客が住所録に少なくとも 1 つの住所を持っている場合、システムは、顧客に請求先住所として、住所録から住所を選択するようプロンプトを出します。顧客はアドレスを選択するか、「**新規住所の作成**」をクリックします。顧客が「**新規住所の作成**」をクリックすると、住所の新規追加の使用事例を使用して新しい住所が追加されます。そうすると、顧客は住所を 1 つ選択します。システムは選択された住所をオーダーの請求先住所にセットアップします。顧客が住所録に住所を持っていない場合、A2 Enter Billing Address (A2 請求先住所の入力) が実行されます。

システムは配送先住所ページを表示します。住所録の住所のリストが表示されます。顧客がリストで該当する配送先住所を見つけた場合には、その住所を選択します。そうでない場合には、顧客が「**新規住所の作成**」をクリックして、新規住所追加の使用事例を使用して新しい住所を追加します。次に、システムは選択された住所をオーダーの配送先住所にセットアップします。

システムは配送先住所に応じた配送方法のリストを表示します。以下の情報はそれぞれの配送方法ごとに表示されます。

- 簡略説明
- 配送料金
 - オーダーごとの固定配送料金
 - オーダーされた項目ごとの配送料金
- 配達のおよその時間 (InFashion および WebFashion のみ)

顧客は配送方法を選択します。システムは、オーダーの配送方法として選択された方法を設定します。

注: NewFashion ストアでは、配送先住所と配送方法は各オーダー・アイテムごとに指定されます。

For NewFashion のみ

NewFashion サンプル・ストアでは、システムは、オーダーに含まれる各アイテムの販売開始日だけでなく、その販売開始日に基づくアイテム配送の選択項目をいくつか表示します。オーダーのそれぞれの商品ごとに、システムは以下を表示します。

- 数量
- 簡略説明
- 属性値 (たとえば、サイズが x = ラージ、カラーが青)
- 将来の在庫に基づく予想販売開始日

- (オーダーからアイテムを除去するための) 除去ボタン

特定のアイテムに関して要求された数量の在庫がない場合、システムはオーダーを、使用可能なオーダー・アイテムを含むオーダーと使用不可能な在庫部分を含むオーダーの、2つのオーダー・アイテムに分割します。顧客は3つの配送設定のいずれかを選択します。

- オプション 1、オーダー全部が配送準備完了になるまで待つ。システムは全オーダーの予想販売開始日を示します。
- オプション 2、配送可能なアイテムだけすぐに配送し、残りは後で配送する。
- オプション 3、購入可能なアイテムを配送し、残りは後日の購入のためにショッピング・カートに残しておく。

その後、顧客はオーダーの配送設定を選択します。

注: すべてのアイテムが購入可能であれば、各アイテムの販売開始日は表示されません。

NewFashion 在庫サブシステムの詳細は、在庫サブシステムを参照してください。

InFashion、WebFashion、および NewFashion の場合

システムは、商品情報を表示するだけでなく、オーダー情報の詳細も表示します。オーダーのそれぞれのアイテムごとに、システムは以下を表示します。

- 簡略説明
- 属性値 (たとえば、サイズが x = ラージ、カラーが青)
- 数量
- 単価
- 合計価格
- システムは以下も表示します。
 - 請求先住所
 - 配送先住所
 - 小計 (オーダーされた全商品の合計額)
 - 割引 (ある場合)
 - 税額合計 (ある場合)(複合税は別に表示されます)
 - 割引 (ある場合)
 - 配送方法を含めた配送料金
 - 総合計 (顧客に課金される)
- 配送日付の見積もり (先頭に表示される) (NewFashion のみ)

注: NewFashion ストアでは、オーダーの各アイテムごとに配送先住所と配送方法が表示されます。

システムは、ユーザーにクレジット・カード情報を入力するようプロンプトを出します。以下の情報が必要になります。

- クレジット・カード・タイプ (Visa(R) や MasterCard(R) など)
- カード番号
- 有効期限月:
- 有効期限年

顧客はクレジットカード情報を入力し、「**オーダー**」をクリックします。システムは支払い情報 (E1) を保存し、オーダーに関する以下の情報を含めた確認ページを表示します。

- オーダー番号
- 小計 (オーダーされた全商品の合計額)
- 税額合計
- 配送
- 割引 (ある場合)
- 総合計

顧客は後で参照できるようにこのページを印刷します。

代替フロー

A1: 請求先住所の追加

顧客に新しい住所を追加するようプロンプトが出されます。顧客は **新規住所追加の使用事例** を使用してアドレスを入力します。システムは新しい住所をオーダーの請求先住所にセットアップします。使用事例は続きます。

例外フロー

E1: 無効なクレジットカード番号

システムはクレジットカード番号の妥当性を検査します。検査に失敗すると、システムは以下の情報を示したエラー・メッセージを表示します。

無効なクレジットカード番号

さらに、顧客に情報を再入力するようプロンプトを出します。使用事例は続きます。

E2: クレジット・カードの有効納期切れ

クレジットカードの有効期限日付が現在日付より古い場合、システムはエラー・メッセージを表示します。そしてその使用事例を最初から再開します。

注: アイテムが購入できない場合、「**オーダー可能商品一覧**」ページが表示されます。

住所編集の使用事例 (Business Edition)

 顧客はその住所録の住所を編集することができます。

実行者

顧客

メイン・フロー

顧客は編集する住所を住所録から選択します。システムは、選択された住所の詳細を検索して表示します。

- ニックネーム (NewFashion のみ)
- 名
- 姓
- 番地
- 市区町村

- 都道府県
- 郵便番号
- 国または地域
- 電話番号

NewFashion では、顧客が入力したニックネームがページの上部に表示されます。

顧客は必要に応じて変更を加え、送信をクリックします。システムはアドレス (E1) を更新します。

代替フロー

なし

例外フロー

E1: システムは、必須の情報が欠落していることを顧客へ通知し、以下を含む欠落情報を要求します。

- 名
- 姓
- 番地
- 市区町村
- 都道府県
- 郵便番号
- 国または地域

ニックネームがすでに住所録に存在する場合、エラー・メッセージが表示されます。

顧客は、欠落した必須情報を入力します。

ホーム・ページの使用事例

ホーム・ページはストアフロントとして機能し、顧客をストアに引き付けます。

実行者

顧客

メイン・フロー

顧客は Web ブラウザーにストアの URL を入力します。次に、システムはホーム・ページを表示します。

ホーム・ページは以下の情報があります。

- 以下のページへのリンクをはったナビゲーション・バー
 - ホーム・ページ
 - 連絡先情報ページ
 - セキュリティおよびプライバシー・ページ
 - ショッピング・カート・ページ
 - 登録ページ

- アカウント・ページ
- ヘルプ・ページ
- 上位レベル、または最上位レベルのカテゴリーへのリンク。それぞれのカテゴリーごとに以下の情報が表示されます。
 - イメージ
 - ターゲットの商品へのリンク

顧客が登録の際に性別を指定した場合、顧客の性別をターゲットとした商品が表示されます。指定しない場合は、商品の汎用セットが表示されます。

それぞれの製品ごとに以下の情報が表示されます。

 - イメージ
 - 簡略説明

顧客はイメージをクリックします。それによりシステムは、一致したページ (A1、A2) を表示します。

代替フロー

A1: 顧客がカテゴリーを選択

顧客がカテゴリーを選択すると、システムは商品カテゴリー・ビューの使用事例で説明するカテゴリー・ページを表示します。

A2: 顧客が商品を選択

顧客が商品を選択すると、商品ページは商品ページ表示の使用事例で説明する商品ページを表示します。

ログオンの使用事例

ログオン処理により、登録済みショッパーは自分の名前およびパスワードを入力することで、自分のアカウントにアクセスすることができます。

実行者

顧客

メイン・フロー

顧客は「アカウント」を選択します。そこでシステムは、以下のフィールドでページを表示します。

- E メール・アドレス
- パスワード

顧客が上記フィールドに該当する情報を入力し、「ログイン」を選択します。システムは、顧客の E メール・アドレスとパスワードが正しいことを確認してから、顧客がそのアカウントに入ることを許可します。顧客がパスワードを忘れた場合、「パスワードをお忘れですか」をクリックすると代替フロー A1 が実行されます。

代替フロー

A1: パスワードを忘れた場合

顧客がパスワードを忘れた場合は、「パスワードをお忘れですか」をクリックします。システムはここで顧客に E メール・アドレスの入力を促すページを表示します。顧客は自分の E メール・アドレスを入力し、「パスワードの送信」をクリックします。システムはその E メール・アドレス (E1) あてにパスワードを送信します。

例外フロー

E1: システムに一致する E メール・アドレスが存在しない場合

システムに一致する E メール・アドレスを持つ顧客が存在しない場合は、以下のメッセージが表示されます。「該当する E メール のお客様が見つかりません。 使用事例は異常終了します。」

顧客が同じユーザー名でログイン使用として 6 回失敗すると、その顧客はロックアウトされます。 アカウントがロックされたこと、顧客はアカウントを再びアクティブにするために、ストアの担当者に連絡しなければならないことを示すメッセージが表示されます。

個人アカウント管理の使用事例

顧客は自分のアカウントをアカウント・ページで管理します。

実行者

顧客

メイン・フロー

顧客は**アカウント**を選択します。次に、システムは以下のオプションで、アカウント・ページを表示します。

- 個人情報の変更
- 住所録の編集
- クイック・チェックアウトの作成または更新 (WebFashion のみ)
- 購入希望商品リストの表示
- オーダーの表示

顧客が**個人情報の変更**をクリックすると、代替フロー A1: E メールおよびパスワードの変更が実行されます。

顧客が「**住所録の編集**」を選択すると代替フロー A2: 住所録の編集が実行されます。

クイック・チェックアウト・プロフィールの作成の使用事例で説明されているように、顧客が「**プロフィールの作成または更新**」(クイック・チェックアウト・プロフィール)を選択すると、クイック・チェックアウト・プロフィール・ページが表示されます。

購入希望商品リスト表示の使用事例で説明されているように、顧客が「**購入希望商品リストの表示**」を選択すると、購入希望商品リスト・ページが表示されます。

オーダー表示の使用事例で説明されているように、顧客が「**オーダーの表示**」を選択すると、オーダーの状況ページが表示されます。

代替フロー

A1: E メールおよびパスワードの編集

システムは個人情報の変更の使用事例に説明されている手順を使用して、顧客に E メール・アドレスおよびパスワードの変更を促すプロンプトを出します。そしてその使用事例を最初から再開します。

A2: 住所録の編集

システムはすでに住所録に追加されている全住所をリストしたページを表示します。各アドレスのとなりには編集および削除の 2 つのボタンがあります。住所リストの下部には「新規住所の追加」ボタンがあります。

顧客が削除をクリックすると、システムはデータベースからそれに一致する住所を削除し、その住所が正しく削除されたことを表示するメッセージを表示します。そして代替フローが最初から再開します。

顧客が編集をクリックすると、システムは住所の編集の使用事例に説明されている手順を使用して顧客に住所の編集を促すプロンプトを出します。そこでシステムは住所が正しく更新されたことを確認するメッセージを表示し、代替フローが最初から再開します。

顧客が「新規住所の追加」をクリックすると、システムは新規の住所追加の使用事例で説明している手順を使用して顧客に新規住所の入力を促すプロンプトを出します。ここで代替フローが最初から再開します。

例外フロー

なし

WebFashion および NewFashion オーダー表示の使用事例

顧客は、オーダーの状況をトラックし、表示することができます。

実行者

顧客

メイン・フロー

顧客は、「アカウント」、「オーダーの表示」をクリックします。システムは、オーダーの状況ページを表示します。そのページでは、顧客が発行したオーダーすべてがリストされています。顧客がオーダーを発行しなかった場合、E1: オーダーがなかった場合が実行されます。

以下の情報は、それぞれのオーダーごとに表示されます。

- オーダー番号 (オーダー詳細情報ページにリンク)
- オーダー日付
- オーダー状況
- 合計

オーダーについてさらに情報を調べるには、オーダー番号をクリックします。システムはオーダー詳細情報ページを表示します。オーダー詳細情報ページで、オーダーの各商品に関する以下の情報が表示されます。

- 数量
- 簡略説明
- サイズや色などの商品属性や、カラーでは赤、青、サイズでは L、LL などの、属性値
- 単価
- 合計価格
- オーダー全体の配送日 (NewFashion のみ)
- 追跡 ID 番号 (NewFashion のみ)

例外フロー

E1: オーダーがなかった場合

システムがその顧客に関連したオーダーを検出できない場合、システムは以下のエラー・メッセージを表示します。

オーダーが見つかりません

ここで、顧客は「ホーム・ページに戻る」または「アカウント」をクリックできます。

商品ページ表示の使用事例

商品ページは商品をより詳しく表示します。

実行者

顧客

メイン・フロー

顧客はカテゴリ・ページで商品を選択します。そこでシステムはデータベースから商品情報を検索し、選択された商品のより詳しい情報のページを表示します。商品ページは以下の情報を表示します。

- 商品名
- 詳細記述
- 価格
- 商品のフル・サイズ・イメージ
- 商品のバリエーションに応じて、サイズや色などの商品属性の一覧や カラーでは赤、青、サイズでは L、LL などの、属性値の一覧

ここで顧客は属性値からふさわしいものを選択し、「ショッピング・カートに追加」をクリックします。システムは選択されたアイテムをショッピング・カート (E1) に追加し、ショッピング・カートの表示の使用事例に説明されているショッピング・カート・ページを表示します。NewFashion ストアでは、顧客はショッピング・カートに追加する数量を指定できます。デフォルトの数量は 1 です。

WebFashion および NewFashion では、顧客は「追加希望リスト」を選択することができます。これを選択する場合、購入希望商品リストへのアイテム追加の使用事例で説明されているように、システムは購入希望商品リストにパッケージを追加して、購入希望商品リストを表示します。

代替フロー

なし

例外フロー

なし

個人情報変更の使用事例

顧客は E メール・アドレスやパスワードのような個人情報を変更することが可能です。

実行者

顧客

メイン・フロー

顧客は「**個人情報の変更**」(E1)をクリックします。システムは以下のフィールドを含む、個人情報の変更ページを表示します。

- 名、現在の名があらかじめ入力されています
- 姓、現在の姓があらかじめ入力されています
- E メール・アドレス、現在の E メール・アドレスがあらかじめ入力されています
- パスワード
- 確認パスワード
- 年齢、年齢があらかじめ入力されています (WebFashion のみ)
- 性別、性別があらかじめ入力されています (WebFashion のみ)
- 優先言語 (NewFashion のみ)
- 希望する通貨 (NewFashion のみ)

顧客は必要に応じてフィールドを変更します。パスワード・フィールドは空欄にしておくことも可能です。顧客がパスワード・フィールドを空欄のままにしておくと、システムは現在のパスワードを変更しません。次に顧客は「**送信**」を選択し、システムは E メール・アドレスやパスワード (E2) を変更します。パスワード・フィールドが空欄だと、システムは直前のパスワードを変更しません。

代替フロー

なし

例外フロー

入力したパスワードは、少なくとも 6 文字でなければならず、少なくとも 1 つの数字を含み、最低 1 つの文字を含むものでなければなりません。ただし、同じ文字を合計で 5 回以上含めたり、連続して 4 回続けたりはならず、そのようにするとエラー・メッセージが表示されます。エラー・メッセージは、パスワードが E メール・アドレスと同じ場合や、顧客がパスワードを変更する場合で、そのパスワードが前のパスワードと同じである場合にも表示されます。詳細は、以下の関連タスクを参照してください。

E1: 顧客がログインしていない

顧客は必ずログインして E メール・アドレスおよびパスワードを変更する必要があります。顧客がログインしていない場合、システムはログオンの使用事例に説明されている手順を使用して、顧客にログインを促すプロンプトを出します。この使用事例を最初から再開します。

E2: E メールがすでに別の顧客として存在する。

入力された E メール・アドレスは既存の他の顧客と一致することはありません。システムが既存アドレスとの一致を検出すると、顧客に別の E メール・アドレスの選択を促すプロンプトを出します。そしてその使用事例を最初から再開します。

E3: パスワードが確認できない

入力されたパスワードはパスワードの確認に指定されたパスワードに一致する必要があります。

一致しない場合、システムは以下のエラー・メッセージを表示します。「パスワードが、「パスワードの確認」に入力されたものに一致しません。もう一度入力してください。」
そしてその使用事例を最初から再開します。

登録の使用事例

登録処理により顧客はデータベースに個人情報を入力することができます。

実行者

顧客

メイン・フロー

顧客が登録を選択します。次に、システムは以下のフィールドのあるページを表示します。

- E メール
- パスワード
- 確認パスワード
- 名
- 姓
- 年齢 (オプション)(WebFashion のみ)
- 性別 (オプション)(WebFashion のみ)
- 優先言語 (NewFashion のみ)
- 希望する通貨 (NewFashion のみ)

顧客が上記フィールドに該当する情報を入力し、送信を選択します。システムはシステムに新規顧客を作成し、顧客の情報 (E1, E2, E3) を保存します。

システムは顧客に、個人アカウント管理の使用事例の手順に従い、アカウントを管理するよう促すプロンプトを出します。

代替フロー

なし

例外フロー

E1: E メール・アドレスがすでに存在する場合

E メール・アドレスがすでにシステムに存在する場合、システムは、ユーザーに別の E メール・アドレスの入力を求めるエラー・メッセージを表示します。そしてその使用事例を最初から再開します。

E2: 必須フィールドが抜けている場合

以下のフィールド (E メール、パスワード、確認パスワード、名、姓) の 1 つでも指定されていない場合、システムはエラー・メッセージを表示します。そしてその使用事例を最初から再開します。

E3: パスワードが無効な場合

パスワードが確認パスワードと一致しない場合、システムは警告を表示します。

注:入力したパスワードは、少なくとも 6 文字でなければならず、少なくとも 1 つの数字と文字を含むものでなければなりません。ただし、同じ文字を合計で 5 回以上含めたり、連続して 4 回続けたりしてはならず、そのようにするとエラー・メッセージが表示されます。エラー・メッセージは、パスワードが E メール・アドレスと同じ場合にも表示されます。

ショッピング・カート表示の使用事例

顧客はショッピング・カートで、購入しようとして選択した商品を表示したり編集したりすることができます。

実行者

顧客

メイン・フロー

顧客は「ショッピング・カート」をクリックします。これにより、システムはショッピング・カートの内容を示したページを表示します。ショッピング・カートにある商品のリスト (E1) が表示されます。それぞれの製品ごとに以下の情報が表示されます。

- 簡略説明
- アイテムの属性名と属性値のペア (サイズや大きさなど)
- 数量
- 単価
- 合計価格

システムは、税額、配送料を含めない、オーダーの合計額も表示します。

顧客が「合計の更新」をクリックすると、合計の更新が実行されます。

顧客が「アイテムの除去」をクリックすると、A2: カートからアイテムを除去が実行されます。

顧客がチェックアウトをクリックすると、A3: カートを更新しチェックアウトが実行されます。

顧客が「ショッピングに戻る」をクリックすると、システムは、顧客がショッピング・カートに入ってきたときと同じページを表示します。そして使用事例は終了します。

クイック・チェックアウトの使用事例で説明されているように、顧客が「クイック・チェックアウト」をクリックすると、システムはクイック・チェックアウト: オーダー要約ページを表示します。

サブフロー

A1: 合計の更新

任意のアイテムの数量を更新するには、顧客はテキスト・ボックスに新規数量を入力し、「合計の更新」をクリックします。システムはショッピング・カート内のアイテムの数量を更新します。アイテムの新規数量に 0 が入力されると、このアイテムはカートから除去されます。

A2: カートからアイテムを除去

カートからアイテムを削除するには、顧客はこのアイテムの「アイテムの除去」をクリックします。システムはカートからアイテムを除去します。

A3: カートを更新しチェックアウト

システムはショッピング・カート内のアイテムの数量を更新し、チェックアウト処理を開始します。ショッピング・カートのチェックアウトの使用事例である、ショッピング・カートのチェックアウトの使用事例を参照してください。

例外フロー

E1: ショッピング・カートが空

ショッピング・カートにアイテムが無い場合、システムは以下のメッセージを表示します。「カートは空です。」

数量フィールドに、文字などの無効な数値が入力される場合、ショッピング・カートのページが再表示されて、エラー・メッセージが示されます。

WebFashion と NewFashion の購入希望商品リスト表示の使用事例

顧客は購入希望商品リスト（または買い物候補リスト）に追加した商品を表示したり編集したりすることができます。

実行者

顧客

メイン・フロー

顧客は、「アカウント」、「購入希望商品リストの表示」をクリックします。これにより、システムは購入希望商品リストの内容を示したページを表示します。購入希望商品リストにある商品のリスト (E1) が表示されます。購入希望商品リストのそれぞれの商品、バンドル、またはパッケージごとに、以下の情報が表示されます。

- 簡略説明
- アイテムの属性と値、たとえば、サイズ、カラーなど
- 単価
- 「除去」リンク

システムは同じページで以下の項目も表示します。

- 「ショッピングに戻る」ボタン
- 「選択したアイテムをショッピング・カートへ追加」ボタン
- 「購入希望商品リストの送信」ボタン
- 「E メール・アドレス」 フィールド (宛先)
- 「E メール・メッセージ」 フィールド (NewFashion のみ)
- 「From:/sender Name (送信者名)」 (NewFashion のみ)
- 「E メール・アドレス」 (NewFashion のみ)

購入希望商品リストからアイテムを削除するには、顧客は「除去」をクリックします。システムは A1 アイテムの除去を完了します。

E メールで友達や家族に購入希望商品リストを送信するには、顧客は「E メール・アドレス」フィールドに宛先の E メール・アドレスを入力し、「購入希望商品リストの送信」をクリックします。システムは A2 購入希望商品リストの送信を完了します。NewFashion ストアでは、顧客は「E メール・メッセージ」

フィールドを使用して E メール・メッセージを送信することができます。顧客は、それぞれの名前を指定する必要がありますが、E メール・アドレスは任意で指定できます。登録済みの顧客の場合、「名前」フィールドはあらかじめ記入されています。

購入希望商品リストのアイテムをショッピング・カートに追加するには、顧客はショッピング・カートに追加したい各アイテムを選択し、「**選択したアイテムをショッピング・カートへ追加**」をクリックします。システムは A3 カートへの追加を完了します。

代替フロー

A1: アイテムの除去

カートからアイテムを削除するには、顧客はアイテムの「**除去**」をクリックします。システムは購入希望商品リストからアイテムを除去します。そしてその使用事例を最初から再開します。

A2: 購入希望商品リストの送信

E メールで友達や家族に購入希望商品リストを送信するには、顧客は「**E メール・アドレス**」フィールドに宛先の E メール・アドレスを入力します。複数の友達に購入希望商品リストを送信する場合、各 E メール・アドレスをコンマで分ける必要があります。たとえば、info@infashion.com,wcs@infashion.com のようにします。それから顧客は「**購入希望商品リストの送信**」をクリックします。

システムは、以下の情報を含む E メール・メッセージを構成します。

- 購入希望商品リストのアイテムの購入方法に関する指示
- 購入希望商品リストへのリンク。これには以下の情報が含まれます。
 - 購入希望商品リスト内の各商品、パッケージ、またはバンドルの簡略説明。
 - 商品、バンドル、またはパッケージ表示ページへのリンク
 - price
 - 「**選択したアイテムをショッピング・カートへ追加**」ボタン

NewFashion ストアでは、顧客は「**E メール・メッセージ**」フィールドにメッセージを入力することによって、システム・メッセージとともに個別設定された E メール・メッセージを送信することができます。

A3: カードへの追加

購入希望商品リストのアイテムをショッピング・カートに追加するには、顧客は追加する各アイテムを選択し、「**選択したアイテムをショッピング・カートへ追加**」をクリックします。システムは選択したすべてのアイテムをショッピング・カートに追加し、ショッピング・カート表示の使用事例に説明されているショッピング・カートを表示します。

例外フロー

E1: 購入希望商品リストが空

アイテムにアイテムが無い場合、システムは以下のメッセージを表示します。「購入希望商品リストは空です」。

そして使用事例は終了します。

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。



Printed in Japan