IBM WebSphere Commerce

NewFashion サンプル・ストア: オンライン・ヘルプ・ファイル

Version 5.4

IBM WebSphere Commerce

NewFashion サンプル・ストア: オンライン・ヘルプ・ファイル

Version 5.4

ご注意

本書の情報およびそれによってサポートされる製品をご使用になる前に、101 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
NewFashion Sample Store: Online Help Files
Version 5.4

発　行： 日本アイ・ビー・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

目次

第 1 章 Creating a store using a sample 1
サンプルを使用したストアの作成 1
サンプル・ストア・アーカイブの作成 1
ストア・データベース資産の変更 2
Web 資産の変更 10
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 10
コマンド行を使用したストア・アーカイブの発行 (Windows). 12
ストアのための Payment Manager のセットアップ 14
サンプル・ストア用のスケジュールされたジョブの作成 15
サンプル・ストアの E メール通知の構成. 17

第 2 章 Sample store database assets 21
ストア・データベース資産 21
NewFashion カタログ・データベース資産. 22
NewFashion 契約およびビジネス・ポリシー・データベース資産 24
NewFashion 配送データベース資産 24
サンプル・ストアのアクセス制御データベース資産 26

第 3 章 NewFashion sample store 29
NewFashion サンプル・ストア 29

第 4 章 NewFashion store pages 31
NewFashion サンプル・ストア検索ページ. 31
サンプル・ストア住所録ページ 33
NewFashion サンプル・ストア住所録ページ 34
WebFashion と NewFashion のサンプル・ストア・バンドル表示ページ 36
サンプル・ストア・カタログ・グループ・ページ 37
サンプル・ストア・ページ: 共通インプリメンテーション手法 39
サンプル・ストア問い合わせ先ページ 42
サンプル・ストア・フッター 42
サンプル・ストア・ヘッダー 44
サンプル・ストア・ヘルプ・ページ 45
サンプル・ストア・ホーム・ページ 45
サンプル・ストア・ログイン・ページ 46
サンプル・ストア・プライバシー・ページ 49
サンプル・ストア商品ページ 50
サンプル・ストア登録ページ 51
サンプル・ストア新着商品ページ. 54
NewFashion サンプル・ストア・オーダーの確認ページ. 55
WebFashion および NewFashion の E メール通知ページ 55
サンプル・ストア・オーダー要約ページ 56
NewFashion サンプル・ストア製品納期情報のチェック・ページ 57
WebFashion と NewFashion のサンプル・ストア・パッケージ表示ページ 58
サンプル・ストア - 左側のナビゲーション・フレーム 60
NewFashion サンプル・ストアの左側のナビゲーション・フレーム 61
サンプル・ストア請求先住所の選択ページ 63
NewFashion サンプル・ストア請求先住所の選択ページ. 64
サンプル・ストア・ショッピング・カート 65
NewFashion サンプル・ストア・ショッピング・カート. 67

サンプル・ストア配送方法ページ. 68
NewFashion サンプル・ストア「配送方法の選択」ページ 70
サンプル・ストア「配送先住所の選択」ページ. 71
NewFashion サンプル・ストア「配送先住所の選択」ページ 72
サンプル・ストア・クイック・チェックアウト要約ページ 73
WebFashion および NewFashion サンプル・ストア「オーダーの表示」ページ 74
WebFashion および NewFashion サンプル・ストア購入希望商品リスト・ページ 77
サンプル・ストアの使用事例 79

第 5 章 Sample store use cases. 81
新規住所追加の使用事例 81
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 82
商品カテゴリー・ビューの使用事例. 82
ショッピング・カートのチェックアウトの使用事例 83
WebFashion および NewFashion バンドル表示ページの使用事例 86
WebFashion および NewFashion パッケージ・ページ表示の使用事例. 87
住所編集の使用事例 (Business Edition) 88
ホーム・ページの使用事例 89
ログオンの使用事例. 90
個人アカウント管理の使用事例 90
WebFashion および NewFashion オーダー表示の使用事例 92
商品ページ表示の使用事例 92
個人情報変更の使用事例 93
登録の使用事例 94
ショッピング・カート表示の使用事例 95
WebFashion と NewFashion の購入希望商品リスト表示の使用事例 96

第 6 章 NewFashion shopping flows 99
NewFashion サンプル・ストアのショッピング・カート・フロー 99
NewFashion サンプル・ストアのショッピング・フロー 100

特記事項 101

第 1 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、ストア間で共用されます。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、共用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:

1. 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。
2. 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。
3. 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。
4. 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。
5. 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になります。

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)	• 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「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	• ストア・アーカイブ XML ファイルを編集する。 • 「ストア・プロファイル」ノートブックを使用してストア情報を変更する。「ストア・プロファイル」ノートブックは以下のデータベース列を編集します。 – IDENTIFIER – DIRECTORY – SETCURR	
	STADDRESS	• 「ストア・プロファイル」ノートブックを使用してストア情報を変更する。	• 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	• 「ストア・プロファイル」ノートブックを使用してストア情報を変更する。「ストア・プロファイル」ノートブックは以下のデータベース列を編集します。 – DESCRIPTION – DISPLAYNAME	
	STORELANG	• 「ストア・プロファイル」ノートブックを使用してストア情報を変更する。「ストア・プロファイル」ノートブックは以下のデータベース列を編集します。 – LANGUAGE_ID	
	STORELANGDS	• ストア・アーカイブ XML ファイルを編集する。	
	STORE	• 「ストア・プロファイル」ノートブックを使用してストア情報を変更する。「ストア・プロファイル」ノートブックは以下のデータベース列を編集します。 – 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)	• 「ストア・プロファイル」ノートブックを使用してストア情報を変更する。「ストア・プロファイル」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)	• 「ストア・プロファイル」ノートブックを使用してストア情報を変更する。「ストア・プロファイル」ノートブックは PAGENAME データベース列を編集します。	
	VENDOR	• 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	• 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。	• WebSphere Commerce アクセラレーターの商品管理ツール
	CATGRPATTR	• 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	• 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。	• WebSphere Commerce アクセラレーターの商品管理ツール
	CATALOG	• 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	• 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	• 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	• 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。	• WebSphere Commerce アクセラレーターの商品管理ツール
	CATENTRY	• 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。	• WebSphere Commerce アクセラレーターの商品管理ツール
	CATENTDESC	• 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。	• WebSphere Commerce アクセラレーターの商品管理ツール
	ATTRIBUTE	• 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。	• WebSphere Commerce アクセラレーターの商品管理ツール
	ATTRVALUE	• 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。	• WebSphere Commerce アクセラレーターの商品管理ツール
	CATGPENREL	• 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。	• WebSphere Commerce アクセラレーターの商品管理ツール
	CATENTREL	• 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	• 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。	• WebSphere Commerce アクセラレーターの商品管理ツール
	ITEMSPC	• 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。	• WebSphere Commerce アクセラレーターの商品管理ツール
	VERSIONSPC	• 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	• 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	• 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	• 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	• 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。	• 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)	• 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。 – JURSTGROUP_ID – DESCRIPTION – SUBCLASS – STOREENT_ID – CODE	
	JURST (税と配送で共通)	• 「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。 – JURST_ID – COUNTRY – STOREENT_ID – CODE – SUBCLASS – STATE	
	JURSTGPREL (税と配送で共通)	• 「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。 – JURST_ID – JURSTGROUP_ID – SUBCLASS	
	CALMETHOD (税と配送で共通)	• 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ー化している CALCODE の場合のみ。)	• ストア・アーカイブ XML ファイルを編集する。	
	TAXCGRY	• 「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。 – TAXCGRY_ID – STOREENT_ID – NAME	
	CALCODE (税と配送で共通)	• 「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。 – CALCODE_ID – CODE – CALUSAGE_ID – STOREENT_ID – GROUPBY – CALMETHOD_id – CALMETHOD_id_app – CALMETHOD_id_qfy	
	CALCODEDSC (税と配送で共通。オプション)	• 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)	• 「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。 – CALRULE_ID – CALCODE_ID – TAXCGRY_ID – CALMETHOD_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
tax	CALSCALE (税と配送で共通)	• 「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。 – CALSCALE_ID – CODE – STOREENT_ID – CALUSAGE_ID – SETCURR – CALMETHOD_ID	
	CALSCALEDS (税と配送で共通)	• 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)	• 「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。 – CALRANGE_ID – CALSCALE_ID – CALMETHOD_ID – RANGESTART	
	CALRLOOKUP (税と配送で共通)	• 「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。 – CALSCALE_ID – CALRULE_ID	
	CRULESCALE (税と配送で共通)	• 「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。 – CALSCALE_ID – CALRULE_ID	
	CALRULEMGP (オプション)	• ストア・アーカイブ XML ファイルを編集する。	
	CALCODTXEX (税免除、割引により使用)	• 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 エントリー: 販売税 1、配送税 1)	• 「税」ノートブックを使用して、税設定を変更する。	
taxfulfillment	TAXJCRULE	• 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。 – CALRULE_ID – FFMCENTER_ID – JURSTGROUP_ID	
store-catalog-tax	CATENCALCD (オプション)	• 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 (オプション)	• 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
配送	JURSTGROUP (税と配送で共通)	• 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。 – JURSTGROUP_ID – DESCRIPTION – SUBCLASS – STOREENT_ID – CODE	
	JURST (税と配送で共通)	• 「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。 – JURST_ID – COUNTRY – STOREENT_ID – CODE – SUBCLASS – STATE	
	JURSTGPREL (税と配送で共通)	• 「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。 – JURST_ID – JURSTGROUP_ID – SUBCLASS	
	SHIPMODE	• 「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。 – CODE – CARRIER – SHIPMODE_ID	
	SHPMODEDSC	• 「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。 – SHIPMODE_ID – LANGUAGE_ID	
	CALMETHOD (税と配送で共通)	• 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)	• 「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。 – CALCODE_ID – CODE – CALUSAGE_ID – STOREENT_ID – GROUPBY – CALMETHOD_ID	
	CALCODEDSC (オプション)	• 「配送」ノートブックを使用して配送設定を変更する。	
	CALRULE (税と配送で共通)	• 「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。 – CALRULE_ID – CALCODE_ID – TAXCGRY_ID – CALMETHOD_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
配送	CALSCALE (税と配送で共通)	• 「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。 – CALSCALE_ID – CODE – STOREENT_ID – CALUSAGE_ID – SETCURR – CALMETHOD_ID	
	CALRULEMGP (オプション)	• 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)	• 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)	• 「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。 – CALRANGE_ID – CALSCALE_ID – CALMETHOD_ID	
	CALRLOOKUP (税と配送で共通)	• 「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。 – CALRLOOKUP_ID – SETCURR – CALRANGE_ID – VALUE	
	CRULESCALE (税と配送で共通)	• 「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。 – 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。)	• 「配送」ノートブックを使用して配送設定を変更する。	
ship fulfillment	SHPJCRULE (ストアに少なくとも 1 つのデフォルト・ルール)	• 「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。 – CALRULE_ID – SHPARRANGE_ID – JURSTGROUP_ID	
	SHPARRANGE	• 「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。 – SHARRAND_ID – STORE_ID – FFMCENTER_ID – SHIPMODE_ID	
store-catalog	STORECAT	• ストア・アーカイブ XML ファイルを編集する。	
	STORECENT	• ストア・アーカイブ XML ファイルを編集する。	
	STORECGRP	• ストア・アーカイブ XML ファイルを編集する。	
	DISPENTREL	• ストア・アーカイブ XML ファイルを編集する。	
	DISPCGPREL	• ストア・アーカイブ XML ファイルを編集する。	
store fulfillment	INVENTORY	• 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
offering	TRADEPOSCN (1)	• 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)	• ストア・アーカイブ XML ファイルを編集する。	
	OFFER	• 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
	OFFERDESC	• ストア・アーカイブ XML ファイルを編集する。	
	OFFERPRICE	• 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
	LISTPRICE	• ストア・アーカイブ XML ファイルを編集する。	
command	URLREG	• ストア・アーカイブ XML ファイルを編集する。	
	CMDREG	• ストア・アーカイブ XML ファイルを編集する。	
	VIEWREG	• ストア・アーカイブ XML ファイルを編集する。	
currency	CURCONVERT	• ストア・アーカイブ XML ファイルを編集する。	
	CURFORMAT	• ストア・アーカイブ XML ファイルを編集する。	
	CURFMTDESC	• ストア・アーカイブ XML ファイルを編集する。	
	CURCVLIST	• ストア・アーカイブ XML ファイルを編集する。	
	CURLIST	• 「ストア・プロファイル」ノートブックを使用してストア情報を変更する。「ストア・プロファイル」ノートブックは CURRSTR を編集します。	
campaign	EMSPOT	• 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 e-マーケティング・スポット管理ツール
	CAMPAIGN	• 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キャンペーン管理ツール
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。	
Business 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。	
Business 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		
	TRDDESC		
	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
 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	• 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、ブラウザーをクローズする。

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す。
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、すべてのブラウザー・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、ストア間で共用されます。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、共用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・プロパティー > 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

     destination2is

     NT

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

     2000

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

   以下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。 カタログ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す。
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¥*instancename*¥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ーで表示されない。

- 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、以下のようにします。

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	• BalancePayment • PayCleanup • ReturnCreditAndCloseScan
NewFashion	• BalancePayment • PayCleanup • ProcessBackorders • RAReallocate • ReleaseExpiredAllocations • ReleaseToFulfillment • ReturnCreditAndCloseScan
WebFashion	• 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
 ToolTech	• BalancePayment • PayCleanup • ProcessBackorders • RAReallocate • ReleaseExpiredAllocations • ReleaseToFulfillment • ReturnCreditAndCloseScan
WebAuction	• 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

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
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
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	`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。
宛先	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 2 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ーをサポートす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式 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 ではエンティティーと呼ぶ) 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*

SOLARIS /opt/WebSphere/CommerceServer/xml/sar

400 /QIBM/ProdData/WebCommerce/xml/sar

NewFashion カタログ・データベース資産

WebSphere Commerce で、catalog.xmlファイルは、サンプル・ストアのカタログ情報を保管しています。各サンプル・ストアには独自の catalog.xml ファイルがあります。 詳細は、catalog.xml ファイルを参照してください。

NewFashion カタログ・データベース資産は以下のセクションに分かれています。

- カタログ・グループ
- カタログ・エンティティー
- ATP フィーチャー

カタログ・グループ

カタログ・グループは、カテゴリーや商品のグループのことです。 カテゴリーは、それ自体がカタログ・グループです。 たとえば、メンズ・ファッションは、パンツやシャツといったメンズ・ファッションを構成するカテゴリーの集合ですが、一方パンツというカテゴリーは商品の集合です。

NewFashion サンプル・ストアには、いくつかのカタログ・グループがあります。

- メンズ・ファッション
- レディース・ファッション
- 新着
- Homepromo

カタログ・グループは、CATGROUP テーブルに作成され、 CATTOGRP テーブル内の特定のカタログに関連付けられます。カタログ・グループは先頭カテゴリーとサブカテゴリーを持つことができます。 たとえば、先頭カテゴリーをメンズ・ファッションにして、サブカテゴリーをパンツにできます。サブカテゴリーがある場合、CATGRPREL テーブルでは、先頭カテゴリーの下に置く必要があります。

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。

カタログ・エンティティー

NewFashion サンプル・ストアのカタログは、カタログ・エンティティーで構成されています。 これらのエンティティーには、以下のものがあります。

- products (商品)
- アイテム
- パッケージ
- バンドル

カタログ・エンティティーについての情報は、CATENTRY テーブルにあります。 CATENTREL テーブルには、商品アイテム、バンドル、およびパッケージの関連など、カタログ・エンティティー間の関連が示されています。 エンティティーの詳細は、 パッケージおよびバンドルを参照してください。

商品

BASEITEM テーブルには、NewFashion ストアの各商品ごとに 1 つのエントリーがあります。基本アイテムは商品です。 各商品には、有効期限によって判別される 1 つのバージョンだけが入ります。 したがっ

て、サイズが 12 で有効期限が 2010 年 1 月 1 日であるレディース用の赤いシャツだけを入れられます。
商品ごとのバージョン情報は、ITEMVERSN テーブルに保管されます。

NewFashion ストアでは、在庫は他のマーチャントと共有されていません。 その結果、DISTARRANG テーブルでは、カタログ内の各商品につき 1 つのエントリーしかありません。 DISTARRANG テーブルの各行は配布調整を表し、これによりストアは独自の在庫を販売できるようになります。

アイテム

アイテムは、属性によって定義される商品の特定インスタンスです。 特定アイテムについて記される情報は、ITEMSPC テーブルに保管されます。各アイテムの在庫情報は、RECEIPT テーブルに保管されます。バック・オーダー・アイテムを受け取る予定の日付は、RADETAIL テーブルに示されます。

注: NewFashion ストアでは、カタログは、OFFERPRICE テーブルのアイテムごとに価格を設定します。詳細は、`offering.xml` ファイルを参照してください。

パッケージ

パッケージは、ゴルフ・パッケージのように、一式で販売されている商品です。 パッケージは、ショッピング・カートの中では 1 行で示されます。 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。 商品は、`product.xml`ファイルでセットアップされていて、 CATENTRY テーブルでパッケージとしてグループ化されています。 パッケージにある商品のエントリーはそれぞれ異なります。

バンドル

バンドルは、一緒にグループ化されている商品ですが、個別に販売することも可能です。バンドルの各商品は、ショッピング・カートの中では 1 行で示されます。 バンドルは CATENTRY テーブルに保管され、バンドルと関連付けられた商品は CATGPENREL テーブルに保管されます。 商品とバンドルの関係は、CATENTREL テーブルで定義されています。

カタログ・エンティティー、パッケージ、およびバンドルの属性は、 PKGATTR および PKGATTRVAL テーブルで設定されます。 PKGATTR テーブルは、パッケージ内の商品に属する既存の属性テーブルを再び参照します。 PKGATTRVAL テーブルは、パッケージの商品の属性値をもう一度参照します。

ATP フィーチャー

NewFashion ストアには、ATP フィーチャーを例示する特別なアイテムがいくつかあります。以下はその例です。

- メンズ、アクティブ・ウェア、TeamShirt は手元の在庫がゼロです。
- レディース、アクセサリー、イブニング・バッグは強制バック・オーダーを抱えています。
- メンズ、アクティブ・ウェア、TeamShirt には RADetail があります。

手元のアイテムがゼロの場合、またはアイテムが強制バック・オーダーの場合、顧客がいくらこのアイテムをオーダーしても、自動的にバック・オーダーになります。 RADetail がないということは、バック・オーダーの予想日が、現在日付にストアのデフォルトのバック・オーダー・オフセット (ストアのテーブルにある DEFAULTBOOFFSET 列) を加えた日付になるということです。つまり、バック・オーダーの予想日は、30 日に設定されます。

NewFashion 契約およびビジネス・ポリシー・データベース資産

WebSphere Commerce では、すべてのストアにデフォルトの契約が必要です。 デフォルトの契約は、contract.xmlファイルに保管されている使用条件で構成されています。 使用条件がビジネス・ポリシーに言及している場合、そのビジネス・ポリシー情報は policy.xml ファイルに保管されています。

NewFashion 契約およびビジネス・ポリシー・データベース資産情報は、以下のセクションに分かれています。

- Contract
- ビジネス・ポリシー

契約

使用条件

サンプル・ストアのすべての契約には、使用条件が必要です。 使用条件は、契約の中で同意の得られたルール群であり、バイヤーとセラーとの間の購入プロセスを制御します。 それぞれの契約ごとに、価格についての一群の使用条件が必要になります。

さらに、配送料など、他のタイプの課金についても使用条件を設定できます。 各契約には、配送料の使用条件を 1 つだけ含める必要があります。

contract.xmlの中でバイヤーとセラーについて言及されていなければなりません。

ビジネス・ポリシー

使用条件

ポリシーは、特定のビジネス・プロセスで、ビジネス上従う必要のあるルールを略述したものです。 契約の使用条件で、特定のビジネス・ポリシーに言及している場合、 contract.xmlファイルをインポートする前に、 businesspolicy.xml を取り込む必要があります。

注: contract.xmlで言及されているポリシーの中には、 businesspolicy.xml の中に含まれないものがあります。 このようなポリシーは、ブートストラップ・データの一部です。 ブートストラップ・データの詳細は、ブートストラップ・データを参照してください。

NewFashion 配送データベース資産

NewFashion 配送データベース資産は、以下の XML ファイルに保管されます。

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

NewFashion 配送データベース資産は以下のセクションに分かれています。

- 取扱範囲
- 配送モード
- 計算コード
- 計算ルール

- 計算スケール
- 計算範囲
- 計算ルックアップ
- 計算組み合わせ
- 配送実行

取扱範囲

shipping.xmlファイルは、配送の取扱範囲を示します。 取扱範囲は JURST テーブルで定義されており、JURSTGROUP はその取扱範囲をグループおよびサブクラスへ割り当てます。また、JURSTPREL はその取扱範囲と取扱範囲グループを同じサブクラスへ割り当てます。

配送モード

配送モードは、運送会社とその配送サービスの組み合わせです。 たとえば、「XYZ Carrier、翌日配送」で 1 つの配送モードということになります。 配送モードについての情報は、SHIPMODE テーブルに保管されます。

計算コード

計算コードは、割引、配送料、消費税、および配送税を計算するために使用されます。 shipping.xml ファイルには、配送のためのすべての計算コードが示されています。 CALCODE テーブルは、配送の計算コードをセットアップします。 displaylevel フィールドには、計算対象を示す数が示されます。

0 =オーダー・アイテム
1 = オーダー
2 = 商品
3 = アイテム
4 = 契約

計算ルール

各計算コードには、計算がどのように実行されるかを定義する計算ルールのセットがあります。 たとえば、商品を 1 つの地域に配送しようとする場合には、計算に一定のルールを適用できます。また商品を別の地域に配送しようとする場合には、計算に別のルールを適用できます。 The CALRULE テーブルは、配送のための計算ルールを保管します。flag フィールドでは、特定 CalculationCode の CalculationCodeQualifyMethod を呼び出すかどうかを指定します。

0 = このメソッドは呼び出されません。

1 = このメソッドは呼び出されます。

計算スケール

計算スケールは、計算に適用される範囲のセットです。 たとえば、配送料金の場合、それぞれが料金に対応する重量範囲のセットがある場合があります。 (配送料金として、重量が 0 kg から 5 kg の商品は 1,000 円、5 kg から 10 kg のものは 1,500 円など。) CALSCALE テーブルは、配送のためのスケール・コードを、オーダーごとに 1 つ、そしてアイテムごとに 1 つ保管します。

注:CALSCALE テーブルは、通貨を対応する範囲に適用する場合に、その通貨のスケール・コードを保管します。

計算範囲

スケール・コードの範囲は、CALRANGE テーブルに保管されます。 calmethod_id_10はオーダー配送ごとに使用され、 calmethod_id_11 はアイテム配送ごとに使用されます。

計算ルックアップ

計算ルックアップ値は、計算スケールに関連した値です。計算ルックアップ値は、配送料金として、重量が 0 kg から 5 kg の商品で 1,000 円、 5 kg から 10 kg のもので 1,500 円の場合に、それぞれ、1,000 円と 1,500 円になります。 指定した CALRANGE ID の通貨ごとに、1 つのルックアップ値があります。CARLOOKUP テーブルは、ルックアップ ID と値を定義します。

計算組み合わせ

計算ルールとスケール範囲は、以下のコードに示されているように、組み合わされて CRULESCALE テーブルに保管されます。 計算メソッドおよびルールは組み合わされて STENCALUSG テーブルに保管されます。 計算についてのストアのデフォルトも、このテーブルに保管されます。 usageflag フィールドは OrderPrepare コマンドがどのように計算を使用するかを制御します。

1 = 使用 - この CalculationUsage を使用する。

2 = チェック - この計算でオーダー・アイテムの値を生成しない場合、ECApplicationException を送出します。

配送実行

配送実行資産は、配送取扱範囲グループを計算ルールに、 または配送センターをストアの shipmodeに関連付けます。 配送実行情報は、SHPJCRULE および SHPARRANGE テーブルに保管され、shipfulfill.xml の中で以下のように記されます。

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。

- *samplestorename*AccessPolicies.xml
- *samplestorename*AccessPolicies_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のようにできません。

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・カテゴリー
- 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・カテゴリー

リソース・カテゴリー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 3 章 NewFashion sample store

NewFashion サンプル・ストア

NewFashion は、WebSphere Commerce に備えられた企業顧客間のオンライン・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を実装しています。 NewFashion サンプル・ストアに実装されているフィーチャーのいくつかを以下に示してあります。

- 多文化サポート
- オーダー・アイテムの販売開始日
- 現在在庫のないアイテムのバック・オーダー
- 顧客の好みに基づいたオーダーの分割
- オーダー状況のトラッキング
- オーダー状況の E メール通知
- 検索機能
- リアルタイム顧客サポートによるコラボレーション

NewFashion ストアは、機能的なストアに必要なページおよびフィーチャーをすべて装備しています。 NewFashion は、ストア・アーカイブとして WebSphere Commerce 内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・ストアを表示するのに必要なのは、ストア・サービス・ツールを使用して、 NewFashion ベースの新規ストア・アーカイブを作成してから、それを 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、 NewFashion 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。 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。

NewFashion 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・チャートおよびその使用事例を参照してください。 使用事例は、商品の登録や表示など、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。 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 4 章 NewFashion store pages

NewFashion サンプル・ストア検索ページ

検索結果ページ resultlist.jsp は、顧客がホーム・ページ sidebar.jsp、または拡張検索ページ advancedsearch.jsp から検索要求を入力した後に表示されます。

注:subcategory.jspページでは、ページのサイドバーに sidebar.jspが組み込まれていません。 そのため、検索機能がこのページに直接に組み込まれています。この検索機能は sidebar.jsp の検索機能と同じです。

bean

resultlist.jsp は、以下の bean を使用します。

- CatalogDataBean
- CategoryDataBean
- CatEntrySearchListDataBean
- CatalogEntryDataBean

advancedsearch.jsp は、以下の bean を使用します。

- CatalogDataBean
- CategoryDataBean

コマンド

resultlist.jsp は、以下のコマンドを使用します。

- ProductDisplay
- AdvancedSearchView
- CatalogSearchResultView

advancedsearch.jsp は、以下のコマンドを使用します。

- CatalogSearchResultView

sidebar.jsp は、以下のコマンドを使用します。

- CatalogSearchResultView
- AdvancedSearchView
- LogonForm (検索のみ)
- HelpView (検索のみ)
- StoreCatalogDisplay (検索のみ)
- SetCurrencyPreference (検索のみ)

subcategory.jsp は、以下のコマンドを使用します。

- CatalogSearchResultView
- AdvancedSearchView

注:subcategory.jspページでは、ページのサイドバーに sidebar.jspが組み込まれていません。 そのため、検索機能がこのページに直接に組み込まれています。この検索機能は sidebar.jsp の検索機能と同じです。

インプリメンテーション

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。

顧客が左側のナビゲーション画面 sidebar.jsp か subcategory.jspで **GO** をクリックするか、拡張検索ページ advancedsearch.jspで**送信**をクリックすると、 resultlist.jsp が表示されます。

CatEntrySearchListDataBean がアクティブになると、データベースにアクセスして次のコードで検索のすべての結果を検索します。

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

request パラメーターは URL ストリングで、resultlist.jspページに渡されます。 CatEntrySearchListDataBean では、メソッド getResultList() が CatalogEntryDataBean の配列を戻します。 各 CatalogEntryDataBean は、検索結果ページにリストされる 1 つの結果です。 getResultList()メソッドは、ページあたりの結果の特定の数を戻します。 この数は、CatEntrySearchListBean がアクティブになるときに、pageSize 変数を使用して URL 要求アドレスに指定されます。 result.jsp が表示されて顧客が**「前へ」**か**「次へ」**ボタンをクリックすると、 CatalogSearchResultView は結果を表示し、 beginindexURL パラメーターが CatalogSearchResultView に渡されます。 CatalogSearchResultView は、表示を開始する検索結果リストの中にある対象となる数の結果を CatEntrySearchListDataBean へ通知します。

NewFashion サンプル・ストアでは、検索結果には商品だけが表示されます。 CatalogSearchResultView コマンドが呼び出されると、resulttype 変数が URL 引き数としてそのコマンドに渡されます。 resulttype は、商品、アイテム、またはその両方をリストするよう CatEntrySearchListBean に命令します。 resulttype の値は、以下のとおりです。

- 1 = アイテムのみをリスト。
- 2 = 商品のみをリスト。
- 3 = アイテムと商品の両方をリスト。

以下のコードは商品のみをリストします。

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

制限

ストア・カタログは、階層構造になっており、メンズ、レディス、新着商品などの先頭カテゴリーが最上部にあり、シャツ、パンツなどのサブカテゴリーがその下に位置します。

顧客が**拡張検索**を選択すると、選択された特定の先頭カテゴリーまたはサブカテゴリー内しか検索できません。 たとえば、顧客がメンズ・カテゴリーの下を検索する場合、直接そのカテゴリーの下にある商品が表示されることになります。 メンズ・シャツまたはメンズ・パンツのカテゴリーの下にある商品は表示されません。 シャツまたはパンツのサブカテゴリーを検索するには、その特定のカテゴリーを選択し、検索結果が表示されるようにする必要があります。

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ましたが、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。 「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。

NewFashion 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、登録済み顧客は住所録に配送先および請求先住所を追加できるようになります。

登録済み顧客はログインして、アカウント (myaccount.jsp) ページの**住所録の編集**をクリックします。 住所録 (addressbookform.jsp) ページが表示されます。 そこで顧客は、新規住所を追加したり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。そうでない場合には、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。

- 住所録フォーム
- InterestItemDisplay
- オーダー状況追跡

addressbookform.jspは以下のコマンドを使用します。

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

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
- AddressBookDataBean

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 (addressbookform.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ましたが、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) のフィールドに情報を入力した後、システムは addressIdがあるかどうかをチェックします。 addressId がある場合は住所録が更新されます。 ない場合には新しい住所が

作成されます。 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。「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。

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

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 (addressbookform.jsp) は既存の住所を表示します。

顧客は、「配送先住所の選択」ページから**「住所録の編集」**をクリックして、住所録へ移動できます。 顧客が「配送先住所の選択」ページから「住所録」ページへ移動する場合、「住所録」ページには**「チェックアウトに戻る」**リンクが表示されます。 これらの顧客は、「住所録」ページから住所を除去することを許可されていません。 そのため、顧客が「配送先住所の選択」ページから来たかどうかを確認するため、以下のような新しいコードが追加されます。

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

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了していない場合には、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。 以下のコードはエラーを処理します。

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

if (hshErrorProperties != null)
{
//送信済み住所にエラーがあります。
strErrorCode = hshErrorProperties.getString(ECConstants.EC_ERROR_CODE, "");
if (strErrorCode.equals(ECUserConstants.EC_ADDR_ERR_BAD_NICKNAME))
strErrorMessage = infashiontext.getString("ERROR_MESSAGE1");

...

}
```

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。

サンプル・ストア・カタログ・グループ・ページ

カタログ・グループ・ページは、顧客がそのストア内にある各種の売り場、つまり、商品やサービスのグループをナビゲートするのに役立ちます。 最初のカタログ・グループ・ページでは、ショッピングしたいエリアへ顧客を案内し、各カタログ・グループの名前、簡単な説明、およびイメージを表示します。以降のカタログ・グループ・ページでは、選択した商品タイプをさらに顧客がブラウズしたいタイプに限定していきます。 ショッピング・パスの最後のカタログ・グループ・ページには商品ページへのリンクが含まれます。

通常、カタログ・グループ・ページには 3 つのタイプがあります。

- サブカタログ・グループ・ページにリンクするページ (カタログ・グループ・ページ)
- 商品ページにリンクするページ (商品リスト・ページ)
- サブカタログ・グループ・ページと商品ページの両方にリンクするページ

サンプル・ストア・カタログ・グループ・ページとその操作方法の詳細については、商品カテゴリーの表示の使用事例を参照してください。

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。

カタログ・グループ・ページの表示と商品リスト・ページの表示

サンプル・ストアでは、カタログ・グループ・ページをフィーチャーしており、フィーチャー特別ご提供商品の他に各トップレベル・カテゴリー（メンズとレディース）ごとにサブカテゴリーを表示します。また、商品リスト・ページもフィーチャーしており、各サブカテゴリー（パンツやシャツ）ごとに商品のリストを表示します。

トップレベル・カテゴリーへのすべてのハイパーリンクには、topと呼ばれる追加のパラメーターが含まれ、これは Yに設定されています。 JavaServer Pages (JSP) ファイル categorydisplay.jsp は、ストアのすべてのカテゴリーを表示するページとして、データベース（DISPCGREL テーブル）に登録されます。 このページで、top パラメーターの存在をチェックします。 存在する場合には、このページに topcategory.jspが入っています。そうでない場合には、subcategory.jsp は入っていません。 topcategory.jsp は、カタログ・グループ・ページであり、subcategory.jsp は、商品リスト・ページです。

カタログ・グループ・ページ

topcategory.jspは、CategoryDataBean の getSubCategories() メソッドを使用して、サブカテゴリーのリストを検索します。 InFashion に含まれるフィーチャー特別ご提供品は、WebSphere Commerce アクセラレーターのキャンペーンの一部ではなく、カタログ中に作成されています。 フィーチャー特別ご提供商品はカタログ・グループに追加された後、topcategory.jspによって、CategoryDataBean の getProducts() メソッドを使用して検索されます。
WebFashion および NewFashion では、メンズの topcategory.jspがバンドルを表示し、レディースの topcategory.jspがパッケージを表示します。 バンドルは CategoryDataBean の getBundles() メソッドを使用して検索され、パッケージは CategoryDataBean の getPackages() メソッドを使用して検索されます。

商品リスト・ページ

subcategory.jsp は、カタログ・グループの全商品のリスト、および左のナビゲーション・バーのすべての兄弟カテゴリー（同じトップレベル・カテゴリーの下にある他のカテゴリーのすべて）のリストを表示します。 各商品ごとに、subcategory.jspが商品の簡略説明、完全なイメージ、および価格を、次のメソッドを使用して表示します。 getDescription().getShortDescription()、getCalculatedContractPrice()、および getDescription().getFullImage()。 親カテゴリー ID は、兄弟カテゴリーのリストを表示するために必要です。 親カテゴリー ID を検索するには、パラメーター parent_category_rnをハイパーリンクに指定しますが、これにより親カテゴリーの CategoryDataBean が構成されます。 デフォルトでは、CategoryDataBean は CategoryId パラメーターからカテゴリー ID を検索します。 しかし、このケースでは、ID が parent_category_rn に保管されます。 結果として、カテゴリー ID は、次のように明示的に設定しなければなりません。

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

このカタログ・グループに属する商品のリストは、CategoryDataBean の getProducts() メソッドを使用して検索されます。

注:NewFashion ストアでは、ストア・サービスで使用可能になっていれば、 subcategory.jspページにもコラボレーション・カスタマー・ケア機能へのリンクが含まれます。 subcategory.jspページでは、独自のサイド・バーが備えられていて、 他のページのように sidebar.jspは含まれていないため、リンクが追加されています。

顧客が**「Live Chat with Customer Care (カスタマー・ケアとのライブ・チャット)」**をクリックすると、画面にポップアップ・ウィンドウが表示され、顧客はオンライン上で顧客サービス担当者とリアルタイムにチャットできます。このリンクは、この機能がストア・サービスで使用可能になっている場合にのみ表示されます。 **「Live Chat with Customer Assistance (顧客アシスタンスとのライブ・チャット)」**リンクは、カスタム・タグのペア (<flow:ifEnabled feature="customerCare"> タグと </flow:ifEnabled>タグ) の本体内に含まれ、ストア・サービスで選択するオプションに基づいて使用可能または使用不可にすることができます。 カスタム・タグが残されている限り、JavaServer Pages (JSP) を変更せずに、ストア・サービスを使用して、コラボレーション・サポートを持つサイトと持たないサイトを自動的に切り替えることができます。ページ内のコラボレーション・サポートを永続的に使用可能または使用不可にするために、ストア・サービスの GUI で**「Apply Permanently (永続的に適用)」**をクリックすることにより、JavaServer Pages (JSP) からカスタム・タグと**「Live Chat with Customer Care (カスタマー・ケアとのライブ・チャット)」**リンクを除去できます。 カスタム・タグやタグの間のコードを手動で除去したり変更することはお勧めできません。その代わり、ストア・サービスの「**Apply Permanently (永続的に適用)**」を使用してください。

注:「Live Chat with Customer Care (カスタマー・ケアとのライブ・チャット)」リンクの前後にあるカスタム・タグは、他のストアの JavaServer Pages (JSP) へコピーすることはできません。これらのタグは、元々それらのタグが含まれていたストアで機能することを意図しています。コラボレーションの詳細は、以下の関連リンクを参照してください。

サンプル・ストア・ページ: 共通インプリメンテーション手法

サンプル・ストア・ページの大部分は、以下のインプリメンテーション手法を使用します。 このページでは例として InFashion ストアを使用しますが、すべてのサンプル・ストアに適用することができます。 個々のページに対する特定の手法の詳細については、そのページの参照ファイルを参照してください。

多文化コンテンツ

サンプル・ストア・ページは、多文化コンテンツを表示するためにも使用されます。 すなわち、同じページのセットは、幾つかの異なったロケールで使用することができます。 多文化表示を使用可能にするコードの大部分は、getResource.jspに入っています。 getResource.jspは以下のことを行います。

- コマンド・コンテキストから現在のロケールを検索し、locale 変数として、それを保管します。
- ストア・ディレクトリーを検索し、storeDir 変数として、それを保管します。
- ストア名を検索し、storeName 変数として、それを保管します。

- ResourceBundle API を使用して言語特定のプロパティー・ファイルをロードし、 infashiontext 変数として、それを保管します。

JavaServer Pages (JSP) ファイルが上の変数をアクセスするには、 getResource.jspは JavaServer Pages (JSP) ファイルに入っていなければなりません。これは、コンパイル時インクルード・アクションを使用します。

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

getResource.jspは、サンプル・ストア JavaServer Pages (JSP) ファイルのほとんどすべてに入っているため、単一要求で数回実行されることがあります。 繰り返す必要がないように、このページに検索された情報の多くは、要求コンテキストに保管されています。例:

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

String storeName = "";

if (storeDir == null) {

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

言語特定メッセージ

「ご注文有難うございました」などの言語特定メッセージは、リソース・バンドル・プロパティー・ファイルに保管されています。 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¥AppServer¥installedApps¥WC_Enterprise_App_*instance_name*.ear¥wcstores.war¥WEB-INF¥classes¥*storedir*

2000

drive:¥Program Files¥WebSphere¥AppServer¥installedApps¥WC_Enterprise_App_*instance_name*.ear¥wcstores.war¥WEB-INF¥classes¥*storedir*

AIX

/usr/WebSphere/AppServer/installedApps/WC_Enterprise_App_*instance_name*.ear/wcstores.war/WEB-INF/classes/storedir

SOLARIS

/opt/WebSphere/Appserver/installedApps/WC_Enterprise_App_instance_name.ear/wcstores.war/WEB-INF/classes/*storedir*
/opt/WebSphere/AppServer/installedApps/WC_Enterprise_App_*instance_name*.ear/wcstores.war/WEB-

INF/classes/*storedir*

400

/QIBM/UserData/WebASAdv4/WAS_instance_name/installedApps/WC_Enterprise_App_*instance_name*.ear/wcstores.war/WEB-INF/classes/*storedir*

たとえば、ストア・ディレクトリーが “storedir” の場合、英語のプロパティー・ファイルは以下のようになります。

NT

drive:¥*drive*:¥WebSphere¥AppServer¥installedApps¥WC_Enterprise_App_*instance_name*.ear¥wcstores.war¥WEB-INF¥classes¥*storedir*
¥infashiontext_en_US.properties.

2000

drive:¥Program Files¥WebSphere¥AppServer¥installedApps¥WC_Enterprise_App_*instance_name*.ear¥wcstores.war¥WEB-INF¥classes¥*storedir*

AIX

/usr/WebSphere/AppServer/installedApps/WC_Enterprise_App_*instance_name*.ear/wcstores.war/WEB-INF/classes/*storedir*/infashiontext_en_US.properties.

SOLARIS

/opt/WebSphere/AppServer/installedApps/WC_Enterprise_App_*instance_name*.ear/wcstores.war/WEB-INF/classes/*storedir*/infashiontext_en_US.properties.

このファイルのコンテンツは、JSP ファイル getResource.jsp から、Java java.util.ResourceBundle API を使用して、ロードされます。 このバンドルは、変数 infashiontextとして保管されます。 言語特定メッセージは以下のように表示されます。

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

コンテンツ・エンコードの設定

最近のブラウザーの多くは、UTF-8 でエンコードされた HTML データを理解します。 ただし、旧式のブラウザーの一部は、ネイティブ・エンコードしか理解できません。 たとえば、旧式の日本語ブラウザーは、“Shift_JIS” でエンコードされた HTML データしか理解できません。 この問題を解決するために、サンプル・ストアは、プロパティー名 ENCODESTATEMENTで、リソース・バンドル・プロパティー・ファイルに言語特定のエンコード方式を指定します。 たとえば、英語プロパティー・ファイル infashiontext_en_US.properties には、以下のエントリーが入っています。

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

個々の JSP ファイルに応じて、エンコードは以下のとおり JSP 要求オブジェクトを使用して設定されます。

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

HTTP ヘッダーはどの HTML コンテンツよりも前に送り出されるので、JSP ファイル中でのエンコード・タイプの設定は、できるだけ早く実行する必要があります。 結果として、どの HTML コンテンツよりも後に、このコンテンツ・タイプ（HTTP ヘッダーとして送信される）を設定した場合には、所要の効果はありません。このブラウザーは、適切なデータを表示できない可能性があります。

ヘッダー、フッター、および左のナビゲーション・フレームの組み込み

サンプル・ストア・ページのほとんどすべては、ヘッダー (header.jsp)、フッター (footer.jsp)、および左のナビゲーション・フレーム (sidebar.jsp) ページを表示します。これらは、以下の実行時インクルード・ディレクティブを使用して、他の JSP ファイルに入っています。

```
<% String incfile;

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

JSP ファイルの正確な位置が分かっている場合には、以下を使用して、インクルード処理が簡単にできます。
<jsp:include page="/*storedir*/include/header.jsp"/>

ここで header.jsp は Web アプリケーション文書ルートの *storedir* ディレクトリーにあります。

サンプル・ストア問い合わせ先ページ

顧客が**問い合わせ先**リンクをクリックすると、問い合わせ先ページ (contact.jsp) が表示されます。"brick and mortar" ストアの電話番号、場所など、そのストアのさまざまな種類の連絡先情報を表示します。

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。

ContactView という新しい表示は、VIEWREG テーブルに作成され、contact.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/stores/servlet/ContactView?parameter_list です。

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 topcatagory.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・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 が登録またはログイン・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、次回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・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
- 親カテゴリーの簡略説明
- 「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 プロパティー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。

親カテゴリーの簡略説明

親カテゴリーの 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。親カテゴリーの簡略説明は、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ー 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、カテゴリー 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て、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なため、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ァ

イ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ァイルと一致させられます。 WebSphere Commerce アクセラレーター・プロファイルに、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・カテゴリーが作成され、プロモートされた商品のすべてがこのカテゴリーに追加されます。 InFashion が、他のトップレベル・カテゴリーとは別の方法でこのカテゴリーのコンテンツを表示するので、newarrivals.jsp ファイルは、このカテゴリー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。 このカテゴリー 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ー情報を表示します。 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ーからのものです。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。

NewFashion サンプル・ストア・オーダーの確認ページ

顧客は、サンプル・ストアのチェックアウト処理の 4 番目のステップでオーダーを完了した後、**オーダー**をクリックします。 オーダーの確認ページ (confirmation.jsp)が表示され、顧客は、オーダー全部についてのオーダー番号、小計、税合計、配送料、総計、および価格を見ることができます。 2 つのオーダーを処理する場合、両方のオーダーの情報が確認ページに表示されます。

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。

bean

confirmation.jsp は、以下の bean を使用します。

- PayStatusPMDataBean
- OrderDataBean

インプリメンテーション

PayStatusPMDataBean は、システムからオーダー情報を受け取ります。 OrderDataBean は情報を受け取り、形式を設定し、システムに表示します。

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。

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。

NewFashion サンプル・ストア製品納期情報のチェック・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、パート 3a. で (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ます)、顧客は製品納期情報をチェックして、オーダーの各アイテムごとの予定配送日を見ることができます (ProductAvailability.jsp)。

注: このページは、オーダー・アイテムの一部が購入不可の場合のみ表示されます。

現在、アイテムの一部の在庫がない場合、顧客は配送設定を選択して、オーダーを分割するか、またはアイテムをバック・オーダーにすることができます。 また、顧客にはオーダーからそのアイテムを除去するオプションもあります。

コマンド

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

- ProductDisplay
- OrderItemDelete
- OrderPrepare
- OrderItemMove
- OrderItemDisplay

bean

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

- OrderDataBean
- OrderItemDataBean
- CatalogEntryDataBean

インプリメンテーション

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。

チェックアウト 3a の商品納期情報のチェックのページ (ProductAvailability.jsp) は、オーダー・アイテムの一部が購入不可の場合のみ表示されます。 AllocationCheck.jspでは、購入不可のアイテムがないかチェックします。 すべてのアイテムが購入不可の場合、またはすべてのアイテムが購入可能の場合、OrderDisplayPending.jspへ転送されます。それ以外の場合は ProductAvailability.jspが組み込まれます。 ProductAvailability.jspによって、顧客はオーダーの中のアイテムごとの販売開始日を見ることができます。 オーダーのアイテムのいずれかが在庫にない場合、顧客はオーダーの**配送の設定の選択**フィールドで配送オプションを選択します。

顧客は以下の 3 つのオプションを選択することができます。

1. オーダー全部が配送準備完了になるまで待つ。
2. 在庫にあるアイテムをすぐ配送し、残りは入手可能になったときに配送する。
3. 在庫にあるアイテムをすぐ配送し、残りはショッピング・カートに残したままにする。

顧客がオプション 2 または 3 を選択した場合、 OrderItemMove コマンドがそのオーダーを 2 つの別のオーダーに分割します。

- NewFashion サンプル・ストア

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。

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>
```

NewFashion サンプル・ストアの左側のナビゲーション・フレーム

顧客は、NewFashion ストアの左側のナビゲーション・フレーム (sidebar.jsp) で、ストアが表示する言語と通貨を選択して、アイテムのカタログを検索することができます。 ナビゲーション・フレームには、登録、ヘルプ、拡張検索、およびカスタマー・ケアとのライブ・チャットの各ページへのリンクも含まれています。

注: 「**Live Chat with CustomerCare (カスタマー・ケアとのライブ・チャット)**」へのリンクは、ストア・サービスを介してストアでこのリンクが使用可能な場合のみ表示されます。

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
- SetCurrencyPreference
- LogonForm
- HelpView
- CatalogSearchResultView
- AdvancedSearchView

bean

sidebar.jspは、以下の bean を使用します。

- SupportedLanguageAccessBean
- LanguageDescriptionAccessBean
- CurrencyDescriptionAccessBean

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、言語の選択ドロップダウン・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、languageId に応じてストア・カタログ・ページが再ロードされます。 次のコードは、ストアがサポートする言語を取得して、それらをドロップダウン・ボックスに表示します。

```
<jsp:useBean id="supportedLanguageDataBean"
class="com.ibm.commerce.common.objects.SupportedLanguageAccessBean"
scope="page" />
<%
Enumeration enStoreLangList =
supportedLanguageDataBean.findByStore(new Integer(storeId));
```

```
while (enStoreLangList.hasMoreElements()) {
SupportedLanguageDataBean storeLang =
(SupportedLanguageDataBean) enStoreLangList.nextElement();
String storelangId = storeLang.getLanguageId();
//Get the display name of the language in the language
//currently selected by the shopper.
LanguageDescriptionDataBean langDesc =
new LanguageDescriptionDataBean();
langDesc.setInitKey_languageId(languageId);
langDesc.setInitKey_descriptionLanguageId(storelangId);

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

顧客が、通貨の選択ドロップダウン・リストから通貨を選択して **GO!**をクリックすると、supportedCurrencies に応じてストア・カタログ・ページが再ロードされます。 次のコードは、サポートされている通貨を取得して、それらをドロップダウン・ボックスに表示します。

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

顧客が検索フィールドにキーワードを入力して **GO** をクリックすると、 CatalogSearchResultView コマンドによって検索基準が送信され、 ResultList.jsp ページがその検索結果とともに表示されます。

顧客が**「Live Chat with Customer Care (カスタマー・ケアとのライブ・チャット)」**をクリックすると、画面にポップアップ・ウィンドウが表示され、顧客はオンライン上で顧客サービス担当者とリアルタイムにチャットできます。 このリンクは、この機能がストア・サービスで使用可能になっている場合にのみ表示されます。 **「Live Chat with Customer Assistance (顧客アシスタンスとのライブ・チャット)」**リンクは、カスタム・タグのペア (<flow:ifEnabled feature="customerCare"> タグと </flow:ifEnabled>タグ) の本体内に含まれ、ストア・サービスで選択するオプションに基づいて使用可能または使用不可にすることができます。 カスタム・タグが残されている限り、JavaServer Pages (JSP) を変更せずに、ストア・サービスを使用して、コラボレーション・サポートを持つサイトと持たないサイトを自動的に切り替えるこ

とができます。 ページ中のコラボレーション・サポートを永続的に使用可能または使用不可にするには、ストア・サービス GUI で「**Apply Permanently (永続的に適用)**」をクリックして、カスタム・タグおよび「**Live Chat with Customer Care (カスタマー・ケアとのライブ・チャット)**」リンクを JavaServer ページから除去することができます。 カスタム・タグやタグの間のコードを手動で除去したり変更することはお勧めできません。 その代わり、ストア・サービスの「**Apply Permanently (永続的に適用)**」を使用してください。

注:「Live Chat with Customer Care (カスタマー・ケアとのライブ・チャット)」リンクの前後にあるカスタム・タグは、他のストアの JavaServer Pages (JSP) へコピーすることはできません。 これらのタグは、元々それらのタグが含まれていたストアで機能することを意図しています。 コラボレーションの詳細は、以下の関連リンクを参照してください。

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も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ーを検査します。 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。

NewFashion 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
- AddressAdd

bean

billingaddress.jspは、以下の bean を使用します。

- OrderDataBean
- AddressDataBean
- OrderItemDataBean

- 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も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。

OrderItemDisplay コマンドは、次にロードするページを判別する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。

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。顧客が**送信**をクリックすると、住所の追加ページから、チェックアウト 1 請求先住所の追加ページに戻ります。

住所録に既存の住所がない場合は、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、この場合請求先住所の追加フォームも billingaddress.jsp によって生成されます。

AddressAdd が実行されて OrderCopy コマンドが呼び出されると、 OrderCopy は現在のオーダーの請求先住所にアドレス ID を割り当て、 OrderItemDisplay コマンドを呼び出します。このコマンドは 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:InFashion および WebFashion ストアでは、 AddAddress および OrderCopy コマンドを呼び出すのに AddBillAddress.jsp が使用されます。 NewFashion ストアでは、AddBillAddress.jsp は、AddAddress コマンドと OrderCopy コマンドを *URL* パラメーターを使ってチェーニングすることにより省かれています。この方法は余分な JSP が必要ないためにインプリメントが容易ですが、余分なリダイレクトのためにシステム実行パスが長くなります。

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・プロファイルから検索します。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す。 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検査し、それに従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・プロファイ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・プロファイルの作成を促すプロンプトが出されます。 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・プロファイルを作成するよう促すプロンプトが出されます。

NewFashion 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。ショッピング・カートには、オーダーを 2 つまで入れておくことができます。

コマンド

shoppingcart.jsp は、以下のコマンドを使用します。

- OrderItemUpdate
- ProductDisplay
- StoreCatalogDisplay
- OrderItemMove

bean

shoppingcart.jspは、以下の bean を使用します。

- OrderItemDataBean
- ErrorDataBean

- CatalogEntryDataBean
- OrderDataBean
- FormattedMonetaryAmountDataBean
- OrderAccessBean

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。

ショッピング・カートは、必ず 1 つのオーダーとして表示されますが、そのショッピング・カートには、分割された複数のオーダーを入れることができます。 このことは、顧客が以前にチェックアウト・フローへ移動していて、オーダーを分割している場合に行なわれます。 顧客が**「チェックアウト」**をクリックすると、OrderItemMove が呼び出され、すべてのオーダーが 1 つのオーダーに移動します。 OrderItemMove は、deleteIfEmpty パラメーターを使用して、移動後に残された空のオーダーをすべて削除します。

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。これはデータベース内の `OrderItemDisplay.jsp` で登録されています。 `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。 ページの値が入っていない場合には、ショッピング・カート・ページ (`shoppingcart.jsp`) がロードされます。

OrderItemUpdate コマンドは、ショッピング・カートの中のアイテムごとの数量を更新します。 FormattedMonetaryAmountDataBean は合計価格および金額の形式を設定します。 OrderDataBean は、オーダーごとにショッピング・カートのアイテムを取得します。 各ショッピング・カート・アイテムは OrderItemBean です。

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**をクリックすると、オーダー済みアイテムの数量が更新され、チェックアウト 1 請求先住所の選択ページが表示されます。

注: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。

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、ErrorDataBean がエラー情報を提供します。 ErrorDataBean は、顧客が非数値文字などの無効な数量を入力した場合にも情報を提供します。

`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。 エラー処理の詳細は、以下の関連概念を参照してください。

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
- ShipModeAccesBean

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、AddShipModeVew 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。 AddShipModeVew 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。

NewFashion サンプル・ストア「配送方法の選択」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 (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ます)、顧客は配送方法 (shipping.jsp) を選択する必要があります。 顧客は、オーダーの各アイテムごとに別々の配送方法を選択することができます。

コマンド

shipping.jspは、以下のコマンドを使用します。

- OrderItemUpdate
- Product Display
- OrderItemDisplay
- AllocationCheck

bean

shipping.jspは、以下の bean を使用します。

- OrderDataBean
- ShippingDataBean
- OrderItemDataBean
- ShippingModeDescriptionDataBean
- CatalogEntryDataBean

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)」 で、顧客はオーダーの各アイテムについて配送方法を選択することができます。 顧客は、各アイテムの**配送方法**フィールドで配送方法を選択します。

OrderItemUpdate コマンドは、選択された配送方法の選択でオーダー・アイテムを更新します。 OrderItemDisplay コマンド が呼び出され、顧客が**次へ**をクリックするとチェックアウト処理の次のページが表示されます。 いくつかのアイテムが購入不可である場合は、チェックアウト 3a「オーダー可能商品一覧 (ProductAvailability.jsp)」ページが次に表示されます。この「オーダー可能商品一覧」ページは AllocationCheck.jsp に転送され、購入可能な在庫があるかどうかに応じて、移動先のページが決まります。

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。

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 は SHPMODEDESC テーブルのカスタム・フィールドです。

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。

NewFashion 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、顧客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。 **配送先**フィールドの下の、ニックネームを選択して既存の住所を選択します。 オーダーの各アイテムに別々の配送先住所を指定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
- AddressBookForm
- Product Display
- OrderItemDisplay

bean

shipaddress.jspは、以下のコマンドを使用します。

- OrderBean
- AddressDataBean
- 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はオーダーのアイテム、および既存の住所にリンクしているニックネームをすべて表示します。これにより、顧客は配送先住所として各アイテムごとに別々のニックネームを選択することができます。

OrderItemDisplay コマンドは、次にロードするページを判別する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。

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。「住所の追加」で「**送信**」をクリックすると、顧客は「配送先住所の選択」ページに戻されます。

顧客は、**住所録の編集**をクリックすることによって住所を編集することもできます。**住所録の変更**をクリックすると、AddressBookForm コマンドが呼び出されます。これはデータベース中で AddressBookForm.jsp と関連付けられています。 AddressBookForm.jsp は AddressBook.jspを呼び出し、これは Address Book ページをロードします。 その顧客に関連した住所がすべて表示されます。 顧客は選択した住所の下の**編集**をクリックすることができます。 **編集**をクリックすると、「住所の更新」ページが表示され、このページで変更を加えることができます。 **送信**をクリックすると、顧客は住所録ページに戻されます。 住所録ページで、顧客が**チェックアウトに戻る**をクリックすると、顧客は「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、登録時に作成された住所のフィールドは空になります。 住所を調べるときに、shipaddress.jspは、 LogonID が NickName と同じかどうかを検査します。 同じであれば、その住所は自分の住所、すなわち登録時に作成された住所ということになり表示されません。 その場合には、住所が表示されません。

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・プロファイ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「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・プロファイル、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
- PayStatusListPMDataBean
- 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();
```

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 メールのフィールド、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 メールのフィールドと、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>
```

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
- パッケージ・ページ表示の使用事例
- 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
- 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
- オーダー表示の使用事例
- 購入希望商品リストへのアイテム追加の使用事例
- 購入希望商品リスト表示の使用事例
- 高速チェックアウト・プロファイル作成の使用事例

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。

第 5 章 Sample store use cases

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、パンドル・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。

WebFashion および NewFashion バンドル表示ページの使用事例

バンドル・ページは、バンドルを構成する商品またはアイテムに関する詳細を表示します。

実行者

顧客

メイン・フロー

顧客はカテゴリー・ページでバンドルを選択します。 そこでシステムはデータベースからバンドル情報を検索し、選択されたバンドルのより詳しい情報のページを表示します。 バンドル・ページは以下の情報を表示します。

- 簡略説明
- 詳細説明
- バンドルのフル・サイズ・イメージ
- バンドルを構成する商品のイメージ
- バンドル内の各商品の価格
- バンドルを構成する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。 NewFashion ストアでは、顧客はショッピング・カートへ追加する数量を指定できます。 システムはバンドルをショッピング・カード (E-1) 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・ページを表示します。

注:デフォルトの数量は 1 です。顧客は、バンドルがショッピング・カートへ追加されてから数量を変更できます。

顧客は**「追加希望リスト」**を選択することもできます。 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バンドルを追加して、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

WebFashion および NewFashion パッケージ・ページ表示の使用事例

パッケージ・ページは、パッケージを構成する商品またはアイテムに関する詳細を表示します。

実行者

顧客

メイン・フロー

顧客はカテゴリー・ページでパッケージを選択します。 そこでシステムはデータベースからパッケージ情報を検索し、選択されたパッケージのより詳しい情報のページを表示します。 パッケージ・ページは以下の情報を表示します。

- 簡略説明
- 詳細説明
- パッケージを構成する商品のイメージ
- パッケージの価格
- パッケージを構成する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。システムはパッケージをショッピング・カード (E-1) 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・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。

注: パッケージ内のアイテムはすべて個別に販売できませんので、ショッピング・カートでは 1 つのパッケージは 1 つのアイテムとカウントされます。

顧客は**「追加希望リスト」**を選択することもできます。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

E1: 属性が選択されていない

属性値が複数ある商品の場合、顧客は各属性値を選択する必要があります。 顧客が属性値を選択しないと、システムはデフォルト値を選択し、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します。

住所編集の使用事例 (Business Edition)

Business 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
  - セキュリティー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・プロファイルの作成の使用事例で説明されているように、顧客が**「プロファイルの作成または更新」**(クイック・チェックアウト・プロファイル) を選択すると、クイック・チェックアウト・プロファイル・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

顧客はカテゴリー・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さす。

- 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

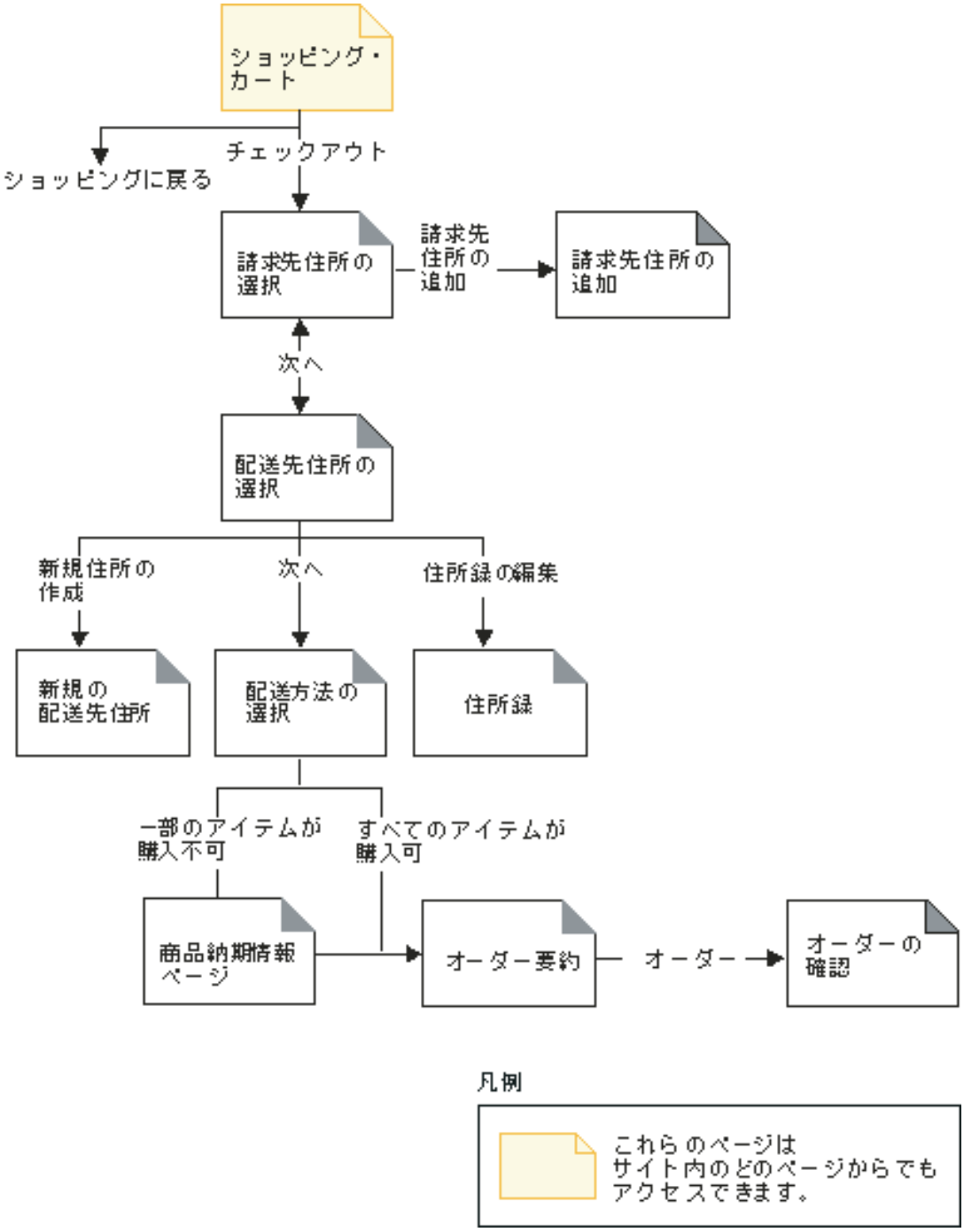
E1: 購入希望商品リストが空

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。

第 6 章 NewFashion shopping flows

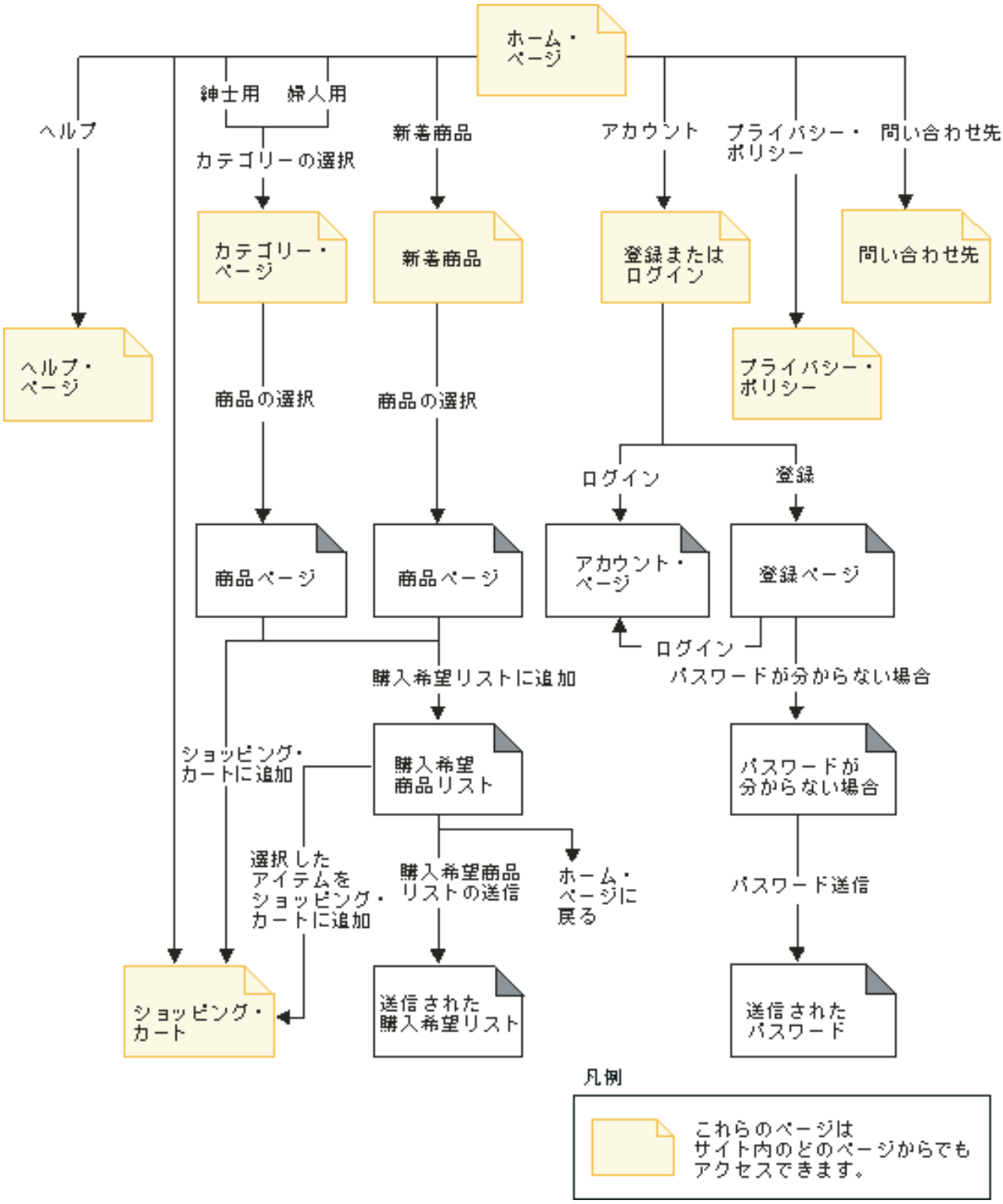
NewFashion サンプル・ストアのショッピング・カート・フロー

以下の図は、NewFashion ショッピング・カート・フローの詳細情報です。 完全なショッピング・フローの詳細は、以下の関連参照を参照してください。



NewFashion サンプル・ストアのショッピング・フロー

以下の図は、NewFashion ショッピング・フローの先頭ページの詳細情報です。ショッピング・カートとアカウントのフローについての詳細は、以下の関連参照セクションを参照してください。



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 VisualAge WebSphere
400

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

日本アイ・ビー・エム株式会社
〒106-8711　東京都港区六本木3-2-12