



# BEA WebLogic Portal

コマース テンプレート  
コマース および  
キャンペーン ツアー

リリース 7.0  
マニュアルの日付 : 2002 年 7 月 17 日

## 著作権

Copyright © 2002 BEA Systems, Inc. All Rights Reserved.

## 限定的権利条項

本ソフトウェアおよびマニュアルは、BEA Systems, Inc. 又は日本ビー・イー・イー・システムズ株式会社（以下、「BEA」といいます）の使用許諾契約に基づいて提供され、その内容に同意する場合にのみ使用することができ、同契約の条項通りにのみ使用またはコピーすることができます。同契約で明示的に許可されている以外の方法で同ソフトウェアをコピーすることは法律に違反します。このマニュアルの一部または全部を、BEA からの書面による事前の同意なしに、複製、複製、翻訳、あるいはいかなる電子媒体または機械可読形式への変換も行うことはできません。

米国政府による使用、複製もしくは開示は、BEA の使用許諾契約、および FAR 52.227-19 の「Commercial Computer Software-Restricted Rights」条項のサブパラグラフ (c)(1)、DFARS 252.227-7013 の「Rights in Technical Data and Computer Software」条項のサブパラグラフ (c)(1)(ii)、NASA FAR 補遺 16-52.227-86 の「Commercial Computer Software--Licensing」条項のサブパラグラフ (d)、もしくはそれらと同等の条項で定める制限の対象となります。

このマニュアルに記載されている内容は予告なく変更されることがあり、また BEA による責務を意味するものではありません。本ソフトウェアおよびマニュアルは「現状のまま」提供され、商品性や特定用途への適合性を始めとする（ただし、これらには限定されない）いかなる種類の保証も与えません。さらに、BEA は、正当性、正確さ、信頼性などについて、本ソフトウェアまたはマニュアルの使用もしくは使用結果に関していかなる確約、保証、あるいは表明も行いません。

## 商標または登録商標

BEA、Jolt、Tuxedo および WebLogic は BEA Systems, Inc. の登録商標です。BEA Builder、BEA Campaign Manager for WebLogic、BEA eLink、BEA Manager、BEA WebLogic Commerce Server、BEA WebLogic Enterprise、BEA WebLogic Enterprise Platform、BEA WebLogic Express、BEA WebLogic Integration、BEA WebLogic Portal、BEA WebLogic Server および How Business Becomes E-Business は、BEA Systems, Inc. の商標です。

その他の商標はすべて、関係各社がその権利を有します。

## コマース テンプレート—コマースおよびキャンペーン ツアー

マニュアルの版数	日付	ソフトウェアのバージョン
なし	2002 年 7 月 17 日	WebLogic Portal 7.0

---

# 目次

まえがき vii

## 第1章 コマース テンプレート ツアーとは

JSP について .....	1-1
データベース スキーマについて .....	1-2
サンプル データベースについて .....	1-2
税金の計算について .....	1-2
支払処理について .....	1-3

## 第2章 ツアーの開始

Netscape 4.7x ユーザの前提条件 .....	2-1
Commerce Server とテンプレートの起動 .....	2-3
ホームページの概要 .....	2-4
匿名ユーザ用 main.jsp の特徴 (前半) .....	2-5
テクニカル ノート タグ ライブラリと Java パッケージのインポート .....	2-7
匿名ユーザ用 main.jsp のその他の特徴 (後半) .....	2-8
テクニカル ノート プロダクト カタログ .....	2-9

## 第3章 新規ユーザによる商品の購入

新規ユーザ ロードマップ .....	3-2
ステップ 1: 商品の検索 .....	3-6
ステップ 2: 商品の選択 .....	3-6
searchresults.jsp の特徴 .....	3-7
ステップ 3: ショッピング カート内容の表示 .....	3-8
shoppingcart.jsp の特徴 .....	3-8
テクニカル ノート Pipeline コンポーネント、Pipeline セッション	

ン、および Webflow .....	3-9
ステップ 4: 新規ユーザ プロファイルの作成 .....	3-10
login.jsp の特徴 .....	3-11
newuser.jsp の特徴 ( 1/3 ) .....	3-12
テクニカル ノート 入力プロセッサ .....	3-13
newuser.jsp によるデモグラフィック ( 2/3 ) .....	3-14
newuser.jsp のその他の特徴 ( 3/3 ) .....	3-15
ステップ 5: ログイン成功の表示 .....	3-16
newusercreation.jsp の特徴 .....	3-16
ステップ 6: 出荷オプションの選択 .....	3-18
shipping.jsp の特徴 .....	3-19
selectaddress.jsp の特徴 .....	3-20
ステップ 7: 支払いに関する情報の入力 .....	3-22
payment.jsp の特徴 .....	3-23
paymentnewcc.jsp の特徴 .....	3-23
テクニカル ノート テンプレート内にインクルードされるテンプレ レート .....	3-25
ステップ 8: クレジットカードの選択 .....	3-27
payment.jsp のその他の特徴 .....	3-27
ステップ 9: 商品の購入 .....	3-28
checkout.jsp の特徴 .....	3-28
ステップ 10: 注文確認のレビュー .....	3-29
confirmorder.jsp の特徴 .....	3-30

## 第 4 章 登録ユーザによる商品の購入

登録ユーザ ロードマップ .....	4-2
ステップ 1: カタログのブラウズと商品の選択 .....	4-5
browse.jsp の特徴 ( 前半 ) .....	4-5
browse.jsp のその他の特徴 ( 後半 ) .....	4-7
details.jsp の特徴 .....	4-8
ステップ 2: ショッピング カートのレビューとチェックアウト .....	4-9
テクニカル ノート セキュリティ モデル .....	4-11
寄り道 : 認証済ユーザ用ホーム ページの復習 .....	4-13
認証済ユーザ用の main.jsp の特徴 .....	4-13
ステップ 3: 出荷オプションの選択 .....	4-14

---

ステップ 4: 新しい届け先住所の入力と選択 .....	4-14
ステップ 5: クレジットカードの選択 .....	4-16
ステップ 6: 商品の購入 .....	4-17
ステップ 7: 注文確認のレビュー .....	4-19

## 第 5 章 登録ユーザによるクレジットカードの追加

ユーザ プロファイル ロードマップ .....	5-2
ステップ 1: ユーザ プロファイルへのアクセス .....	5-3
ステップ 2: ユーザ プロファイルのレビュー .....	5-3
viewprofile.jsp の特徴 .....	5-4
ステップ 3: クレジットカードの追加 .....	5-5
profilenewcc.jsp の特徴 .....	5-6
ステップ 4: 注文の履歴とステータスの表示 .....	5-7
orderhistory.jsp の特徴 .....	5-9
orderstatus.jsp の特徴 .....	5-10
ステップ 5: ログアウト .....	5-11

## 第 6 章 テンプレートのカスタマイズ例

ステップ 1: [Add to cart] リンクを理解する .....	6-2
リンク アンカが Webflow を呼び出す仕組み .....	6-2
Webflow がリンクを解決する仕組み .....	6-3
ステップ 2: [Add to cart] リンクの変更 .....	6-5
getTopCategoriesIP 入力プロセッサの出力先を設定する .....	6-6
ステップ 3: 変更の検証 .....	6-6
ここから先の話 .....	6-8

## 第 7 章 サンプルのキャンペーン

概論と前提条件 .....	7-1
キャンペーンの考察 : TourCampaign1 .....	7-2

## 索引



---

# まえがき

『BEA コマース テンプレート — コマースおよびキャンペーン ツアー』へようこそ。このマニュアルの他に、以下のリソースも活用されることをお勧めします。

**オンライン マニュアルの参照** BEA 製品マニュアルは、BEA 社の Web サイトで公開しています。BEA Home ページで [製品のドキュメント] リンクをクリックするか、「e-docs」製品ドキュメント ページ (<http://edocs.beasys.co.jp/e-docs/>) に直接アクセスしてください。

**マニュアルについてのフィードバック** BEA WebLogic Portal マニュアルについてのフィードバックをお寄せください。ご質問やコメントがあれば、電子メールで [docsupport-jp@bea.com](mailto:docsupport-jp@bea.com) までお送りください。なお、お送りいただく電子メールには、WebLogic Portal リリース 7.0 のマニュアルをお使いであることを明記してください。

**BEA WebSUPPORT への連絡** このバージョンの WebLogic Portal について質問がある場合、または WebLogic Portal のインストールや実行に問題がある場合には、BEA WebSUPPORT (<http://support.bea.com/welcome.jsp>) を通じて BEA カスタマ サポートにご連絡ください。製品パッケージに同梱のカスタマ サポート カードに記載されている連絡先にお問い合わせいただいても結構です。





---

# 第1章 コマース テンプレート ツアー とは

BEA WebLogic Portal には、インストール中にサンプルを選択していれば、e コマース Web サイトをすばやくセットアップするのに使えるコマース テンプレートのセットが用意されています。このツアーでは、これらの多数のテンプレートを順に紹介して、それらテンプレートの適用範囲と設計について説明します。ツアーは、e コマース Web サイトをシミュレートするサンプルデータを使用します。手始めとして、このツアー文書にもとづいて、テンプレートに習熟するための推奨する手順を完了します。

これらのテンプレートは、JavaServer Pages (JSP) と画像ファイルから成ります。Java プログラミング スキルがなくても、テンプレートのレイアウトとこの製品でのページの表示順序を変更することができます。デフォルトのビジネス ロジックの変更といった、それ以外の低レベルの変更には、Java プログラミング スキルが必要になります。

JSP やこの製品の基本コンセプトに詳しくなければ、この章でこのあと取り上げる以下のトピックスを一読されることをお勧めします。

- [JSP について](#)
- [データベーススキーマについて](#)

## JSP について

JavaServer Pages (JSP) は、適格な (well-formed) HTML と、Java 対応の Web ブラウザで表示される際に HTML の機能を拡張する特殊なタグ (JSP タグ) が記述されているテキスト ファイルです。Campaign サービスおよび Commerce サービスでは、JSP タグを使って、Web ページの特定領域で動的コンテンツを提供したり、顧客とデータを交換するためのインタフェースを作成します。

JSP の詳細については、<http://java.sun.com/products/jsp/index.html> を参照してください。

# データベース スキーマについて

Commerce サービスには、PointBase データベースと Oracle データベース用のスキーマが用意されています。これらのスキーマは、Commerce サービス JSP テンプレートから利用できるトランザクションをすべてサポートしています。

**注意：** WebLogic Server をインストールすると、評価版の PointBase データベースがセットアップされます。運用 Web サイトで PointBase データベースを使いたい場合には、PointBase データベースを購入しなければなりません。また、Oracle データベースを使いたい場合には、WebLogic Portal でサポートされている Oracle プラットフォームを購入してインストールしなければなりません。

# サンプル データベースについて

WebLogic Portal はデフォルトでは、PointBase データベースを使うようにコンフィグレーションされています。WebLogic Portal をインストールすると、サンプルの PointBase データベースが作成され、そこにデータが格納されます。このサンプル データは、工具やさまざまな金物類を販売する e コマース Web サイトをシミュレートするものです。

# 税金の計算について

支払うべき売上税の金額を計算するために、WebLogic Portal では、税金サービス ベンダとの統合が可能な API を提供しています。運用 Web サイトでこの税金 API を利用したい場合は、税金サービス ベンダに対する専用のアカウントをセットアップする必要もあります。このアカウントを使えば、最新のデータ

ファイルをダウンロードすることができます。これらのファイルには、米国および選択されたその他の国におけるあらゆる税務関係の要件についての情報が記載されています。

## 支払処理について

クレジットカード取引の検証と発行のために、WebLogic Portal では、支払ベンダとの統合が可能な支払 API を提供しています。運用 Web サイトでこの支払 API を利用したい場合は、支払ベンダに対する専用のアカウントをセットアップする必要もあります。デフォルトでは、支払ベンダとの統合は利用できなくなっています。統合が利用できなくても購入を完了することはできますが、WebLogic Portal はデータを支払ベンダには送信しません。



---

## 第2章 ツアーの開始

自分で、あるいはシステム管理者に依頼して WebLogic Portal をインストールしてからでなければ、ツアーを開始することはできません。

Web ブラウザでテンプレートやサンプルデータを参照するには、使用しているオペレーティングシステムのタイプに応じて、以下のいずれかの作業を行います。

- [Netscape 4.7x ユーザの前提条件](#)
- [Commerce Server とテンプレートの起動](#)

そのあと、この章の最後のトピックである以下の節に進みます。

- [ホームページの概要](#)

## Netscape 4.7x ユーザの前提条件

最低必要な Netscape ブラウザのバージョンは 4.77 です。ツアーを Netscape 4.7x ブラウザで実行しようとする場合、この節をお読みください。前もって必要なステップがいくつかあります。これらのステップは、Microsoft Internet Explorer ブラウザを使用する場合は適用されません。この節は、Netscape 6.0 以降のバージョンを使用する場合も適用されません。

サンプルの JSP テンプレートは、この製品のデフォルトのコンフィグレーションで有効なクッキーでは動作しません。ツアーを開始する前に、Web アプリケーションの `weblogic.xml` ファイルで、`CookieDomain` の値を設定し、その値を含む行のコメントを解除します。

このアプリケーションの `weblogic.xml` ファイルには次の行が含まれています。

```
<!--  
<session-param>  
  <param-name>CookieDomain</param-name>  
  <param-value>.mycompany.com</param-value>
```

```
</session-param>
-->
```

Netscape で Web アプリケーションを使用する前に、これらのコメント開始文字とコメント終了文字を削除し、weblogic.xml ファイル内の CookieDomain パラメータ値を .mycompany.com から、アプリケーションが実行されるドメインの名前に変更します。

weblogic.xml ファイルは次のディレクトリに入っています。

<BEA\_HOME>/weblogic700/samples/portal/wlcsDomain/beaApps/wlcsApp/wlcs/WEB-INF/  
(UNIX の場合)

<BEA\_HOME>\weblogic700\samples\portal\wlcsDomain\beaApps\wlcsApp\wlcs\WEB-INF\  
(Windows の場合)

**重要:** この変更後は、サーバがすでに実行中の場合は、サーバを再起動しなければなりません。サーバの起動とサンプル アプリケーションの先頭ページを開く方法の詳細については、次の節を参照してください。

**注意:** ツアーを実行するためのもう 1 つの選択として、Netscape 4.77 以降でクッキーを無効にする方法があります。しかし、実際の Web 運用環境では、Web サイト ユーザがどのブラウザを使用するか、またはクッキーを有効にしたか無効にしたかが予測できないので、この方法は現実的ではありません。したがって、実際の運用では、Web アプリケーションの weblogic.xml ファイル内で CookieDomain を変更する必要があります。

この章とすべての製品マニュアルにおいて、BEA 製品がインストールされる親ディレクトリを表すために、変数 <BEA\_HOME> が使用されます。変数 <BEA\_HOME> は、製品のインストール ディレクトリのトップレベルを表します。

これ以降、ツアーでは UNIX のパス名構文を使います。Windows マシンから WebLogic Portal を使う場合には、Windows で必要とされるパス名構文に置き換えてください (パス名要素をバックスラッシュで区切り、変数名を「%」記号で囲む)。

# Commerce Server とテンプレートの起動

ツアーを開始する前に、必ず、以下がインストールされていることを確認します。

- WebLogic Server 7.0
- WebLogic Portal 7.0 ( Portal Examples を含む )

ツアーを開始するには、まず、サーバを起動してから Commerce Server を起動することが必要です。

1. 次のとおりにして WebLogic Portal を起動します。

**Windows の場合:** [ スタートメニュー | プログラム | BEA WebLogic Platform 7.0 | WebLogic Portal 7.0 | Portal Examples | Commerce Template | Launch Commerce Server ]

**UNIX の場合:** < インストール ディレクトリ

>/weblogic700/samples/portal/wlcsDomain/startWLCS.sh を実行します。

2. WebLogic Portal 稼働後、次のとおりにしてコマース テンプレートを開始します。

**Windows の場合:** [ スタートメニュー | プログラム | BEA WebLogic Platform 7.0 | WebLogic Portal 7.0 | Portal Examples | Commerce Templates | Start Commerce Templates ]

**UNIX の場合:** コマース テンプレート クライアントにログ オンします。WebLogic Portal をデフォルト設定でインストールした場合には、サーバと同じマシン上で起動したブラウザに下記 URL を指定することで開始できます。

`http://localhost:7501/wlcs`

localhost とは、Commerce Server が稼働しているサーバの名前です。

これでツアーの準備が整いました。

## ホームページの概要

`http://<WLCS-host:7501>/wlcs` にアクセスすると、ブラウザに `main.jsp` が表示されます。これが Web サイトのホームページです。[図 2-1](#) は、匿名ユーザから利用できる情報とナビゲーション オプションを示しています。[第 4 章「登録ユーザによる商品の購入」](#)では、認証済ユーザの場合に `main.jsp` がどのように変わるかを示しています。



## 匿名ユーザ用 main.jsp の特徴（前半）

図 2-1 匿名ユーザ用の Home ページ



main.jsp の以下の特徴に注目してください。

- このファイルの絶対パス名は、以下のとおり。  
`<BEA_HOME>\weblogic700\samples\portal\wlcsDomain\beaApps\wlcsApp\wlcs\commerce\main.jsp`
- トップバナー、ナビゲーションヘッダー、およびフッターは、別個の JSP インクルードファイルに定義されている。これらの JSP タグには、ブラウザから main.jsp が要求された時点で WebLogic Portal がこれらのファイルをインクルードする（組み込む）ように指定しています。

- `main.jsp` の先頭に表示される `admin.inc` インクルード ファイルでは、
  - 上部バナーに JSP ファイル名が表示される。この表示はインクルード ファイル `admin.inc` に基づきます。どのテンプレートでも、JSP ファイル名はこの位置に表示されます。
  - オンライン ヘルプが用意されており、その中には、このテンプレートの目的、さまざまなユーザ コンテキストで利用可能なナビゲーション オプション、およびこのテンプレートのカスタマイズに関する注意事項が記載されています。
  - WebLogic Portal の機能のセットアップと保守を行うためのツールへ移動するための [Administration] リンク。このツアーでは、管理ツールについては説明しません。管理ツールにアクセスするには、WebLogic Portal 管理者用のユーザ名とパスワードが必要になります。詳細については、<http://edocs.beasys.co.jp/e-docs/wlp/docs70/admin/commerce.htm#1167188> にある『管理者ガイド』の「カタログの作成と管理」、および <http://edocs.beasys.co.jp/e-docs/wlp/docs70/dev/commerce.htm> にある『開発者ガイド』の「Commerce サービスのセットアップ」を参照してください。
  - `templateindex.html` へ移動するための [テンプレート インデックス] リンク。このページでは、WebLogic Portal に用意されているすべての JSP テンプレートについて簡潔に説明しています。
- インクルードされている `header.inc` ファイルには広告ブレースホルダがあります。`main.jsp` ファイルでは、ブラウザがバナー広告を表示する前に、JSP タグを使ってバナー広告の場所が確保されます。次はその例です。

```
<td colspan="7" height="75" align="center" valign="middle">  
<ph:placeholder name="<%= banner %>" />  
</td>
```

`header.inc` には、以下のように、サイトのさまざまな領域に移動するためのボタンも用意されています。

- `search.jsp` へ移動するための [サーチ] リンク。この JSP では、ブール式を使った検索をサポートしています。
- `shoppingcart.jsp` へ移動するための [カートの中を見る] リンク。この JSP には、現在のセッション中に購入された商品の一覧が記録されています。

- `login.jsp` へ移動するための [ ログイン ] リンク。この JSP では、ユーザのログイン、または新規ユーザのサイトへの登録ができるようになっています。

## テクニカル ノート タグ ライブラリと Java パッケージのインポート

JSP タグを使う各 JSP テンプレートの冒頭には、タグ ライブラリをインポートする一連のコード行が記述されます。[コード リスト 2-1](#) では、Pipeline、Catalog (cat)、User Management (um) の各タグ ライブラリを `main.jsp` でインポートしている部分を示しています。

### コード リスト 2-1 タグ ライブラリのインポート

---

```
<%@ taglib uri="webflow.tld" prefix="webflow" %>
<%@ taglib uri="cat.tld" prefix="catalog" %>
<%@ taglib uri="um.tld" prefix="um" %>
<%@ taglib uri="pz.tld" prefix="pz" %>
<%@ taglib uri="es.tld" prefix="es" %>
<%@ taglib uri="cm.tld" prefix="cm" %>
```

---

同様に、Java クラスを使うテンプレートでは、Java パッケージをインポートしなければなりません。[コード リスト 2-2](#) では、`main.jsp` において Java パッケージをインポートしている部分を示しています。

### コード リスト 2-2 Java パッケージのインポート

---

```
<%@ page import="com.beasys.commerce.webflow.HttpRequestConstants" %>
<%@ page import="com.beasys.commerce.webflow.PipelineSessionConstants" %>
<%@ page import="com.beasys.commerce.ebusiness.catalog.Category" %>
<%@ page import="com.beasys.commerce.ebusiness.catalog.ViewIterator" %>
<%--@ page import="com.beasys.commerce.axiom.contact.*" --%>
<%--@ page import="com.beasys.commerce.ebusiness.customer.*" --%>

<%@ page import="com.bea.pl3n.content.ContentHelper"%>
<%--@ page import="java.util.Collection" --%>
<%--@ page import="java.util.Iterator" --%>
```

## 匿名ユーザ用 main.jsp のその他の特徴（後半）

図 2-2 プロダクト カタログのトップレベル



main.jsp は、プロダクト カタログのトップレベルでもあります。カタログのトップレベルにおける以下の特徴に注目してください。

- 左の欄の検索機能は JSP タグによって提供される。
- [ショッピング カタログ] セクションに表示される情報は、プロダクト カタログ内の情報と、カタログ階層におけるユーザの現在位置に基づいて動的に生成される。

## テクニカル ノート    プロダクト カタログ

WebLogic Portal には、データベーススキーマと、プロダクトカタログを実装するための一連のビルドスクリプトが用意されています。これらのスキーマとスクリプトは、Web ベースのカタログサイトでよく使われる機能と属性を定義しています。たとえば、スキーマでは、商品カテゴリと商品から成る階層を確立するのに使うメタデータが指定されています。この階層を使えば、カタログにおけるユーザの現在位置に基づいて商品データを表示することができます。図 2-2 は、WebLogic Portal によって生成された、この階層のトップレベルのコンテンツを示しています。

コードリスト 2-3 は、main.jsp に記述されている JSP タグのうち、プロダクトカタログにクエリを発行してカテゴリを検索し、そのクエリ結果を表示するものを示しています。このコードリストには、それ以外に、コンテンツの書式付けと表示のための HTML 要素も示されています。(JSP の構文に詳しくなければ、`<!--` と `-->` で囲まれるコメントタグ部分だけを見てください)

### コードリスト 2-3    プロダクトカタログにクエリを発行する JSP タグ

```
<webflow:getProperty id="topCategory"
property="<%=PipelineSessionConstants.CATALOG_CATEGORY%>"
type="com.beasys.commerce.ebusiness.catalog.Category" scope="request"
namespace="sampleapp_main" />

<webflow:getProperty id="subcategories"
property="<%=PipelineSessionConstants.CATALOG_CATEGORIES%>"
type="com.beasys.commerce.ebusiness.catalog.ViewIterator" scope="request"
namespace="sampleapp_main" />

<p class="head1">ショッピング カタログ</p>

<ul type="square">

<!-- 共通のブラウザ パラメータを含んだ String を宣言する -->

<%! static final String commonParameters =
HttpRequestConstants.CATALOG_SOURCE_KEY + "=" +
PipelineSessionConstants.CATALOG_CATEGORIES + "&" +
HttpRequestConstants.CATALOG_DESTINATION_KEY + "=wlcs_siblings&"; %>

<catalog:iterateViewIterator iterator="<%= subcategories %>"
id="currentCategory"
returnType="com.beasys.commerce.ebusiness.catalog.Category">
```

## 第 2 章 ツアーの開始

---

```
<% String browseParameters = commonParameters +
HttpRequestConstants.CATALOG_CATEGORY_ID + "=" +
java.net.URLEncoder.encode(currentCategory.getKey().getIdentifier()); %>

<li>

    <a href="<webflow:createWebflowURL event="link.browse"
namespace="sampleapp_main" extraParams="<%=browseParameters%"/>" />">

<b><catalog:getProperty object="<%= currentCategory %>"
propertyName="Name" /></b>

</a>

<br>

</catalog:iterateViewIterator>
```

---

WebLogic Portal には、既存のプロダクトカタログデータベースからデータをインポートするためのユーティリティも用意されています。あるいは、それ以外のユーティリティを使ってデータをインポートすることもできます。WebLogic Portal データベースより適したカタログシステムをすでに使っている場合には、既存のカタログシステムを使うように JSP テンプレートを変更することができます（ただし、こうした低レベルの変更には、Java プログラミングのスキルが必要になります）。

### ツアーを続行するには

ブラウザで `main.jsp` を開いたまま、[第 3 章「新規ユーザによる商品の購入」](#)に進みます。このセクションでは、新規ユーザが遭遇しそうな JSP テンプレートを示します。

ツアーの他のセクション、[第 4 章「登録ユーザによる商品の購入」](#)と [第 5 章「登録ユーザによるクレジットカードの追加」](#)では、コマース サービス JSP テンプレートの使い方の付加的なシナリオを示します。

---

## 第3章 新規ユーザによる商品の購入

ツアーのこのセクションでは、家の修繕計画に必要なハンマーを購入したいと仮定しましょう。品揃えがよく値段の安い新しいe コマース Web サイトのことを聞きつけ、そのサイトにアクセスすることに決めます。この章では、この Web サイト内を以下の順に進んでいきます。

**注意：** 新規ユーザ ロードマップを別のブラウザ ウィンドウに開いておくとういでしょう。ツアー中にマップの内容を参照したほうがよい場合が多いからです。

- [新規ユーザ ロードマップ](#)
- [ステップ 1: 商品の検索](#)
- [ステップ 2: 商品の選択](#)
- [ステップ 3: ショッピング カート内容の表示](#)
- [ステップ 4: 新規ユーザ プロファイルの作成](#)
- [ステップ 5: ログイン成功の表示](#)
- [ステップ 6: 出荷オプションの選択](#)
- [ステップ 7: 支払いに関する情報の入力](#)
- [ステップ 8: クレジット カードの選択](#)
- [ステップ 9: 商品の購入](#)
- [ステップ 10: 注文確認のレビュー](#)

## 新規ユーザ ロードマップ

ツアーのこのセクションを開始する前に、JSP テンプレートの処理フローを以下のロードマップ ( [図 3-1](#)、[図 3-2](#)、および [図 3-3](#) ) で確認しておいてください。これらのロードマップに示されている矢印は、Web サイトでのナビゲーション上の選択肢を表しています。そのうち、濃い矢印は、ツアーのこのセクションで説明する経路を示しています。



図 3-1 商品の検索と選択のための JSP

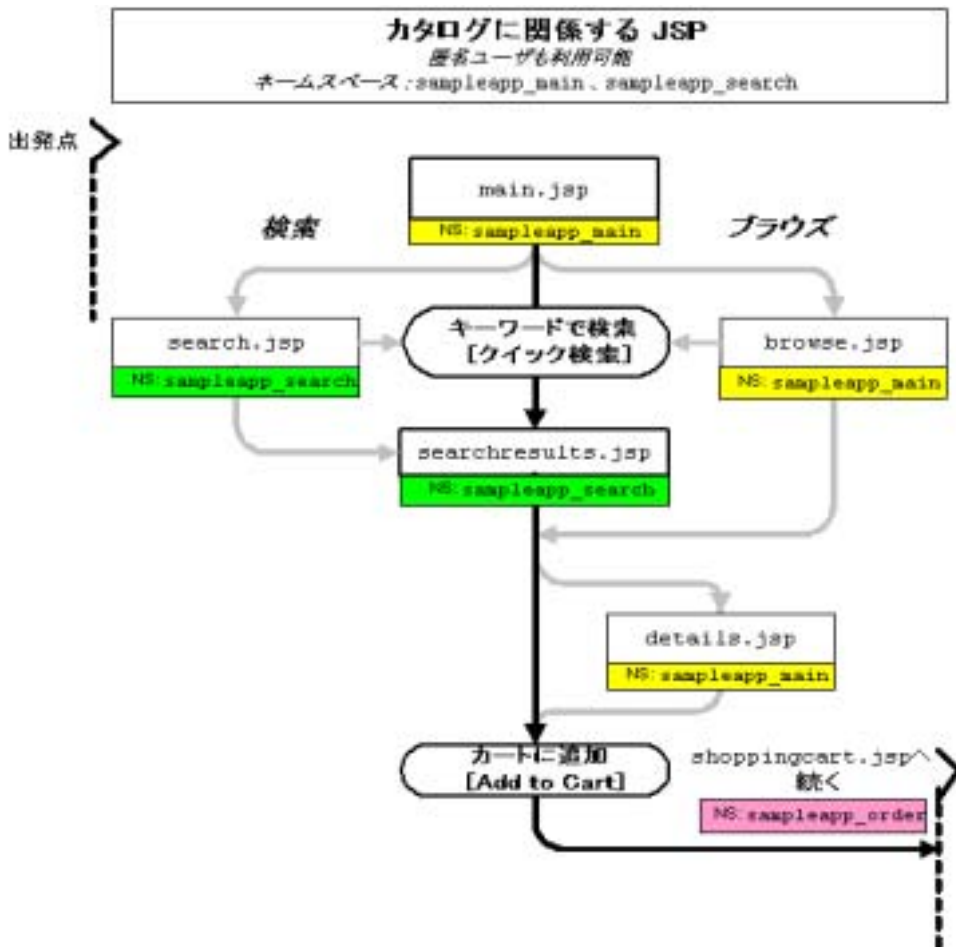
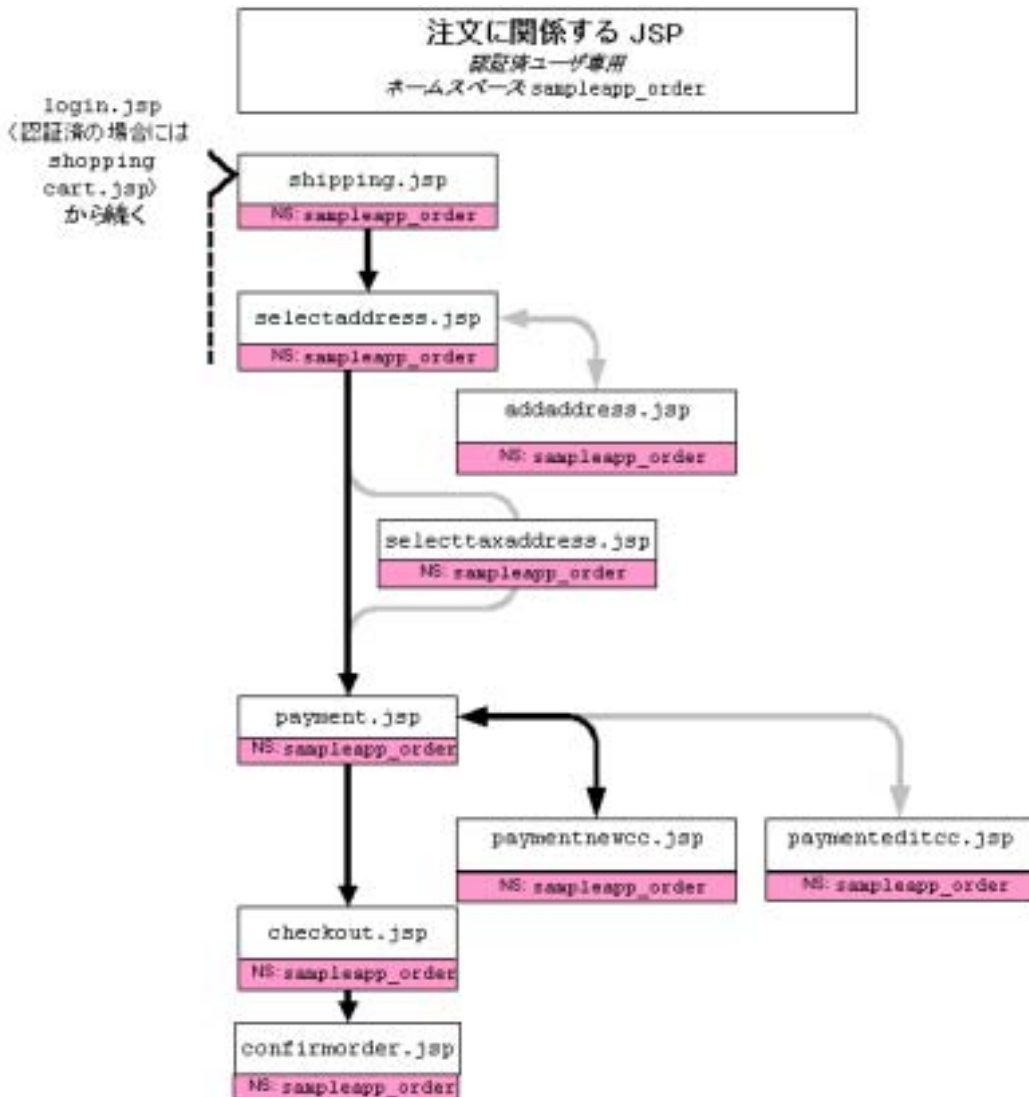




図 3-3 注文完了のための JSP



## ステップ 1: 商品の検索

この Web サイトのホーム ページに初めてアクセスすると、プロダクト カタログのブラウズか検索のどちらかを行えることがわかります。今回探しているのがハンマーであることがはっきりしており、また、この新しい Web サイトの全体像がよくわかっていないので、[クイック検索] ボックスを使うことにします (図 3-4 を参照)。

図 3-4 main.jsp での検索



ツアーを続行するには

ホームページで、[クイック検索] ボックスに「hammer」と入力します。そのあと、[Find] をクリックします。

## ステップ 2: 商品の選択

[Find] をクリックすると、WebLogic Portal の検索エンジンが、プロダクト カタログにクエリを発行して、キーワードが「hammer」に一致する商品を検索し、その結果を searchresults.jsp に表示します (図 3-5 を参照)。

## searchresults.jsp の特徴

図 3-5 searchresults.jsp での検索結果



searchresults.jsp の以下の特徴に注目してください。

- このファイルの絶対パス名は、以下のとおり。  
<BEA\_HOME>/weblogic700/samples/portal/wlcsDomain/beaApps/wlcsApp/wlcs/commerce/catalog/searchresults.jsp (UNIX)
- コンテンツはすべて動的に生成される。商品の説明、価格、入手可能性、および関連画像への URL は、プロダクト カタログ内の商品レコードから得ることができます。

ツアーを続行するには

ハンマーを選んで [Add to cart] をクリックします。

## ステップ 3: ショッピング カート内容の表示

[Add to cart] をクリックすると、WebLogic Portal は、サイトに現在アクセスしている間に選択した商品をすべて `shoppingcart.jsp` に列挙します ( 図 3-6 参照 )。

### shoppingcart.jsp の特徴

図 3-6 ショッピング カート

The screenshot shows a web page for a shopping cart. At the top, there is a banner for 'Storage Boxes!' and a navigation menu with links for 'Home', 'Search', 'View Cart', and 'Login'. Below the navigation is a search box labeled 'クイック検索: キーワードを入力' with a 'Find' button. The main content area is titled 'ショッピング カート' and contains text instructions for checking out, deleting items, and updating quantities. Below this text are two buttons: 'Empty cart' and 'Check out >'. A table below the buttons displays the cart items with columns for quantity, item name, price, shipping fee, discount, and subtotal. A 'Remove' button is next to each item. At the bottom of the table, the total price is shown as '総額 ¥ 13,200 (送料、消費税別)'. Annotations on the right side of the image indicate that the top part of the page is '静的 HTML' (static HTML) and the table and buttons are 'JSP タグとスクリプトレットで生成される' (generated by JSP tags and scriptlets).

`shoppingcart.jsp` の以下の特徴に注目してください。

- このファイルの絶対パス名は、以下のとおり。  
`<BEA_HOME>/weblogic700/samples/portal/wlcsDomain/beaApps/wlcsApp/wlcs/commerce/shoppingcart.jsp` (UNIX)
- HTML として書式付けされる静的テキストのセクションが含まれている。
- 商品と価格のリスト、小計、およびカート関連のすべてのボタンは、現在の Pipeline セッション内の情報に基づいて動的に生成される。

## テクニカル ノート Pipeline コンポーネント、Pipeline セッション、および Webflow

このツアーではこれまで、HTML 要素と JSP タグに焦点を合わせてきました。WebLogic Portal はこれらを使って、Web ブラウザに表示可能なフォーマットで情報を提示するのです。

こうしたプレゼンテーション層の裏では、Pipeline コンポーネント、Pipeline セッション、および Webflow がユーザとの間でデータを交換し、ビジネス ロジックを適用してデータを処理し、個々のショッピングセッションにおける状態を維持管理し、そして、コンテキスト固有のルール セットに基づいてユーザを誘導し、Web サイト内を進ませるのです。これらの各メカニズムを以下に説明します。

- **Pipeline コンポーネント**: 税金の計算や注文の委託といったサーバサイド ビジネス ロジックを表す別個の構成単位。複数の Pipeline コンポーネントを順に並べて、一方のコンポーネントから他方のコンポーネントにデータが入力されるようにしたものを Pipeline と呼びます。Pipeline 内の Pipeline コンポーネントの順序は、`namespace.pln` ファイル内の設定により決定します。
- **Pipeline セッション**: 現在のセッションに関する情報（たとえば、現在のショッピングカートなど）や、もっと一時的なデータ（たとえば、顧客の最新の入力に関するエラー メッセージなど）の格納場所。
- **Webflow**: WebLogic Portal によってユーザに表示される JSP の順序を Pipeline セッションの現在の状態に基づいて決定するメカニズムのこと。WebLogic Portal では、`namespace.wf` ファイルを使って、Web サイトでのページの表示順序を決定します。Webflow によって Web サイトでのナビゲーションが集中管理されることで、Web サイトの保守と更新が容易になります。

たとえば、[コード リスト 3-1](#) に示す JSP タグは `shoppingcart.jsp` から抜粋したもので、現在の Pipeline セッションからショッピングカートの中身を取得しています。

### コード リスト 3-1 ショッピングカートの中身を取得するための JSP タグ

```
<webflow:getProperty id="shoppingCart"  
property="<%=PipelineSessionConstants.SHOPPING_CART%>"
```

```
type="com.beasys.commerce.ebusiness.shoppingcart.ShoppingCart" scope="session"
namespace="sampleapp_main" />

<webflow:getProperty id="savedShoppingCart"
property="<%=PipelineSessionConstants.SAVED_SHOPPING_CART%>"
type="com.beasys.commerce.ebusiness.shoppingcart.ShoppingCart" scope="session"
namespace="sampleapp_main" />
```

---

ツアーを続行するには

[Check out] をクリックします。

## ステップ 4: 新規ユーザ プロファイルの作成

ショッピング カートのページで [Check out] をクリックすると、Weblogic Portal は `login.jsp` を開いて、ログインするようユーザに促します。新規ユーザであるので、まずユーザ プロファイルを作成しなければなりません ( [図 3-7](#) を参照 )。



## login.jsp の特徴

図 3-7 新規ユーザのログイン



- このファイルの絶対パス名は、以下のとおり。  
`<BEA_HOME>/weblogic700/samples/portal/wlcsDomain/beaApps/wlcsApp/wlcs/commerce/login.jsp (UNIX)`
- 広告プレースホルダを使用してヘッダーに広告を表示する。パワー ユーザは、特定の顧客のタイプに合わせた広告を、与えられた広告プレースホルダへ指定できます。
- サンプルテンプレートをブラウザしやすくするために、[ユーザ名]と[パスワード]の各ボックスにはデフォルト値が指定されている。サンプルデータベースには、ユーザ名 democustomer のユーザ プロファイルが用意されています。



- ユーザがすべての必須フィールドに情報を入力したかどうかは、入力プロセッサを使って決定される（入力プロセッサについては、この節の次のトピック「[テクニカル ノート 入力プロセッサ](#)」で説明します）。
- 問い合わせ先住所と届け先住所を入力するためのフィールドが含まれている。ユーザは、`selectaddress.jsp` または `viewprofile.jsp` で、これ以外の届け先住所を入力することができます（[図 5-2](#) を参照）。
- [ 都道府県 ] リストと [ 国 ] リストに表示される値を提供するために、以下のファイルがインクルードされる。
 

```
<BEA_HOME>/weblogic700/samples/portal/wlcsDomain/beaApps/wlcsApp/wlcs/commerce/includes/states.inc および
<BEA_HOME>/weblogic700/samples/portal/wlcsDomain/beaApps/wlcsApp/wlcs/commerce/includes/countries.inc
```

[ 都道府県 ] リストと [ 国 ] リストは複数の JSP テンプレートで使われるので、別個のインクルード ファイルに定義されています。
- チェックボックスによって、電子メールを受け取るための設定を顧客が指定できる。Campaign サービスを使用すれば、JSP を使用するシナリオ アクションを作成することで、パーソナライズされた電子メールを作成できます。

## テクニカル ノート 入力プロセッサ

入力プロセッサは、フォームの発行を処理する Java クラスです。入力プロセッサの中には、顧客データの妥当性確認を行うものがありますが、入力プロセッサの主な役割は、顧客データを Pipeline セッションに格納したあと、Pipeline コンポーネントで利用できるようにすることです。ユーザ入力を入力プロセッサに送るかどうかと、入力プロセッサが出力をどこに送るかは、Webflow によって決定されます。

たとえば、[コード リスト 3-2](#) に示すのは `newuser.jsp` から抜粋したもので、ユーザが [Save] ボタンをクリックするとブラウザがデータを Webflow に送ると指定しています。

### コード リスト 3-2 データを入力プロセッサに送るフォーム

```
<!-- 顧客データを入力するためのフォームを開始する。
```

このフォームは、起点 "newuser.jsp" で発生する "button(save)" イベントを Webflow Advisor に送る。-->

```
<form method="post" action="<webflow:createWebflowURL
event="button.save" httpsInd="calculate"
namespace="sampleapp_user" />">

<input type="hidden" name="origin" value="newuser.jsp">
<input type="hidden" name="event" value="button.save">
<input type="hidden" name="httpsInd" value="calculate">
<input type="hidden" name="namespace" value="sampleapp_user">
```

---

## newuser.jsp によるデモグラフィック (2/3)

newuser.jsp は顧客に関するデモグラフィック情報を収集します (図 3-9 参照)。このデモグラフィック情報を使用して顧客を階層に分類でき、これをシナリオとコンテンツ セレクタ が使用します。

シナリオはイベント、システム状態、および顧客階層を識別し、特定のアクションを実行します。コンテンツ セレクタ はパーソナライズされた情報を任意の顧客または顧客の階層に表示します。

図 3-9 デモグラフィック情報の入力

The screenshot shows a web form titled "個人情報" (Personal Information). It contains the following fields and options:

- 性別 \*** (Gender): Radio buttons for  女性 (Female) and  男性 (Male).
- 生年月日 \*** (Date of Birth): A text input field containing "1948/10/0" with a placeholder "(yyyy/mm/dd)".
- 職業 \*** (Occupation): Radio buttons for  公務員 (Public Servant),  会社役員 (Company Director),  専門家 (Specialist),  エンジニア (Engineer),  管理職 (Manager), and  営業 (Sales).

## newuser.jsp のその他の特徴 (3/3)

図 3-10 ユーザ名とパスワードの入力

**お支払い情報(オプション)**

クレジットカードの種類  \* リストの値は newcctemplate.inc に定義されている。

クレジットカード名義人  \*

カード番号  \*

有効期限 (mm/yyyy)   \* リストの値はインクルードファイルから得られる。

請求先住所

郵便番号  \*

都道府県  \*

市区町村  \*

番地  \*

マンション/アパート  \*

国  \*

**ユーザ名とパスワード**

ユーザ名  \*

パスワード  \*

パスワードの確認  \*

\* のマークが付いたフィールドは必須です。

Save

Cancel

サンプル JSP テンプレートでは、ユーザ名とパスワードに使える文字タイプに制限はない。

newuser.jsp の後半部分の以下の特徴に注目してください。

- newcctemplate.inc がインクルードされる。この中には、新規のクレジットカード情報を入力するためのフィールドが記述されています。

[クレジットカードの種類] リストと [有効期限] リストに表示される値は、newcctemplate.inc テンプレートでしか使われないので、そのテンプレートに定義されています。また、[都道府県] リストと [国] リストに表示される値は、複数の JSP テンプレートで使われるので、別個の JSP ファイルに定義されています。

- ユーザ プロファイルをセットアップするのに、支払に関する情報を入力する必要はない。

- ユーザ名およびパスワードに使える最大文字数は、JSP ファイルに 50 と設定されている。サンプル JSP テンプレートでは、これ以外の制限事項（たとえば、必須の文字タイプや禁止されている文字タイプなど）はありません。こうした制限をさらに課したい場合には、独自のフィールド検証機能 (validator) をセットアップしなければなりません。その場合には、おそらく入力プロセッサの形でそれを実現することになります。

ツアーを続行するには

必須フィールドに情報を入力し、[Save] をクリックします。ツアーに自動的にログインするはずですが。

# ステップ 5: ログイン成功の表示

## newusercreation.jsp の特徴

図 3-11 newusercreation.jsp



newusercreation.jsp の以下の特徴に注目してください。

- このファイルの絶対パス名は以下のとおり。  
<BEA\_HOME>/weblogic700/samples/portal/wlcsDomain/beaApps/wlcsApp/wlcs/commerce/user/newusercreation.jsp (UNIX)

- ユーザが新規ユーザ プロファイルを作成するときのみ表示し、ユーザが現在登録され認証されていることを確認する。
- 現在の Pipeline セッションのショッピング カート内に商品がある場合にのみ、[Check out] ボタンを表示する。[コード リスト 3-3](#) は、Pipeline セッションをクエリする JSP タグおよびクエリの結果を評価して [Check out] ボタンをトグルする Java スクリプトレットを示しています。

### コード リスト 3-3 [Check out] ボタンの表示

```

<!-- ショッピング カートを取得すれば、チェックアウトが選択可能かどうか分かる -->

<webflow:getProperty id="shoppingCart"
property="<%=PipelineSessionConstants.SHOPPING_CART%>"
type="com.beasys.commerce.ebusiness.shoppingcart.ShoppingCart" scope="session"
namespace="sampleapp_main" />

<p class="head1">
おめでとうございます。

</p>
<p>
ようこそ <%= request.getRemoteUser() %>, ご登録が完了し、ログインしました。

</p>
<p>
以下のアクションから一つお選びください：

</p>
<a href="<webflow:createWebflowURL event="link.shoppingcart"
namespace="sampleapp_order" />">" border="0" vspace="2"
hspace="3"></a>

<% if (shoppingCart != null && shoppingCart.isEmpty() == false) { %>
<a href="<webflow:createWebflowURL event="button.checkout" httpsInd="calculate"
namespace="sampleapp_order" />">" border="0" vspace="2"
hspace="3"></a>

```

```
<% } %>
```

```
<a href="<webflow:createWebflowURL event="link.home" namespace="sampleapp_main"
/>">" border="0" vspace="2"
hspace="3"></a>
```

---

ツアーを続行するには

[Check out] をクリックします。

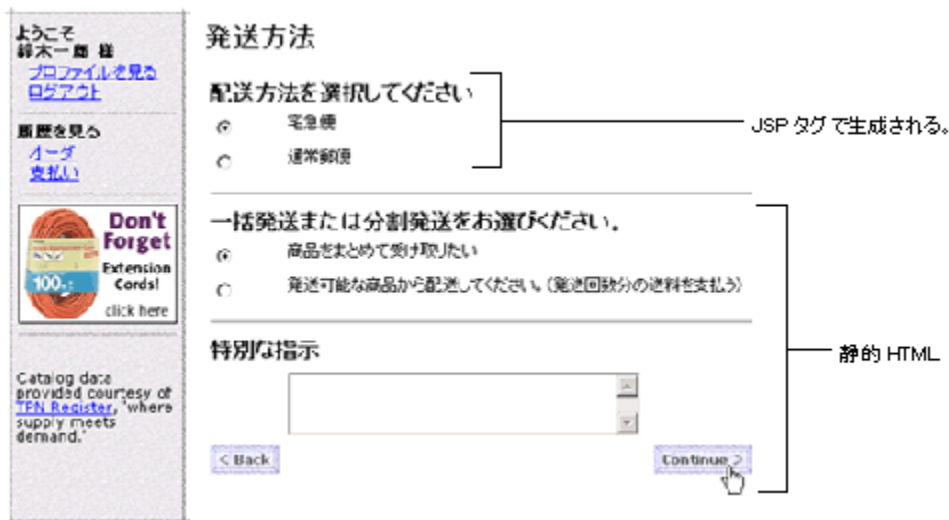
## ステップ 6: 出荷オプションの選択

[Check out] をクリックすると、WebLogic Portal は、`shipping.jsp` に出荷オプションを表示することで、チェックアウト プロセスを開始します ( [図 3-12](#) を参照 )。



## shipping.jsp の特徴

図 3-12 出荷オプションの選択



shipping.jsp の以下の特徴に注目してください。

- このファイルの絶対パス名は、以下のとおり。  
`<BEA_HOME>/weblogic700/samples/portal/wlcsDomain/beaApps/wlcsApp/wlcs/commerce/order/shipping.jsp` (UNIX)
- Pipelines JSP タグを使い、データベースに定義されている出荷方法を表示する。パフォーマンスを向上させるため、WebLogic Portal は、これらの JSP タグでデータが要求されるたびにデータベースにアクセスするのではなく、出荷方法のリストをキャッシュします。
- HTML を使って、[一括発送または分割発送をお選びください。] ラジオ ボタンと [特別な指示] 入力ボックスを実装している。

### ツアーを続行するには

1. 出荷オプションを選択します。[特別な指示] ボックスにコメントを入力してもかまいません。

2. [Continue] をクリックします。

[Continue] をクリックすると、WebLogic Portal は、selectaddress.jsp に届け先住所を表示します ( 図 3-13 を参照 )。

## selectaddress.jsp の特徴

図 3-13 届け先住所の選択



selectaddress.jsp の以下の特徴に注目してください。

- このファイルの絶対パス名は、以下のとおり。  
<BEA\_HOME>/weblogic700/samples/portal/wlcsDomain/beaApps/wlcsApp/wlcs/commerce/order/selectaddress.jsp (UNIX)
- Java スクリプトレットを使い、ユーザ プロファイルに記載されている届け先住所のリストに対して反復処理を行います ( コード リスト 3-4 を参照 )。
- **Taxation サービス** を呼び出す。このサービスは、WebLogic Portal におけるビジネス ロジックとプレゼンテーション ロジックの実装の 1 つです。

---

**コードリスト 3-4 selectaddress.jsp における Java スクリプトレット**

---

```
<um:getProperty propertySet="CustomerProperties"
propertyName="shippingAddressMap" id="shippingAddressMap" />

<%
if(shippingAddressMap == null) shippingAddressMap = new HashMap();
%>

<p><font color="red"><webflow:getException/></font></p>
  <!-- Iterate through all addresses -->

  <%
    Iterator iterator = ((Map)shippingAddressMap).keySet().iterator();
    while(iterator.hasNext())
    {
      String addressKey = (String)iterator.next();
      Address shippingAddress = (Address)((Map)shippingAddressMap).get(addressKey);
    }
  %>

  <table width="90%" border="0" cellpadding="6" cellspacing="0">

    <tr>
      <td align="left" valign="top" width="40%" nowrap>

        <p><%= shippingAddress.getStreet1() %><br>

        <% if( shippingAddress.getStreet2().length() != 0) {%>
```

```
<%= shippingAddress.getStreet2() %><br>
<% } %>

<%= shippingAddress.getCity() %><br>
<%= shippingAddress.getState() %> <%= shippingAddress.getPostalCode()
%><br>

<%= shippingAddress.getCountry() %>

</td>

<td align="left" valign="top" width="5%" >
<%
String extraParams = HttpRequestConstants.ADDRESS_KEY + "=" + addressKey;
%>
```

---

ツアーを続行するには

[Use] をクリックします。

## ステップ 7: 支払いに関する情報の入力

[Use] をクリックすると、WebLogic Portal は `payment.jsp` を表示し、この中に、ユーザ プロファイルに記載されているクレジットカードがすべて列挙されます (図 3-14 を参照)。

## payment.jsp の特徴

図 3-14 クレジットカードの追加を選択する場合



payment.jsp の以下の特徴に注目してください。

- このファイルの絶対パス名は、以下のとおり。  
`<BEA_HOME>/weblogic700/samples/portal/wlcsDomain/beaApps/wlcsApp/wlcs/commerce/order/payment.jsp (UNIX)`
- ユーザ プロファイルをセットアップする際に支払に関する情報を付け加えなかったため、このページで選択できるのは、クレジットカード情報を付け加えることだけである。

ツアーを続行するには

[Add card] をクリックします。

## paymentnewcc.jsp の特徴

[Add card] をクリックすると、WebLogic Portal は paymentnewcc.jsp を開いて、支払に関する情報を入力するようユーザに促します (図 3-15 を参照)。

図 3-15 クレジットカード情報の入力

YOUR Logo Here

ホーム

ようこそ  
鈴木一郎様  
プロフィールを見る  
ログアウト

履歴を見る  
モード  
変更

### 新規クレジットカード

クレジットカードの種類	VEA *
クレジットカード名義人	JOHRO SUZUKI *
カード番号	4111111111111111 *
有効期限 (mm/yyyy)	6 2001 *
請求先住所	
郵便番号	105-0001 *
都道府県	東京都 *
市区町村	港区虎ノ門 *
番地	2-10-1 *
マンション・アパート	新日ビル東棟14F
国	日本 *

\* のマークが付いたフィールドは必須です。

Save

ユーザープロフィールに追加する。

このセクションは、以下のインクルードファイルから得られる。  
/commerce/includes/newcctemplate.inc

paymentnewcc.jsp の以下の特徴に注目してください。

- このファイルの絶対パス名は、以下のとおり。  
<BEA\_HOME>/weblogic700/samples/portal/wlcsDomain/beaApps/wlcsApp/wlcs/commerce/order/paymentnewcc.jsp (UNIX)
- このページは、  
<BEA\_HOME>/weblogic700/samples/portal/wlcsDomain/beaApps/wlcsApp/wlcs/commerce/user/profilenewcc.jsp および  
<BEA\_HOME>/weblogic700/samples/portal/wlcsDomain/beaApps/wlcsAp

p/wlcs/commerce/register/  
newuser.jsp と似ている。

これら 3 つの JSP ファイルのすべてが、インクルード ファイル

<BEA\_HOME>/weblogic700/samples/portal/wlcsDomain/beaApps/wlcsAp  
p/wlcs/commerce/includes/

newcctemplate.inc を使用して、クレジットカード入力フィールドを定義  
します。テンプレート インクルード ファイルの詳細については、この節の次  
のトピック「[テクニカル ノート テンプレート内にインクルードされるテ  
ンプレート](#)」を参照してください。

- [Save] をクリックすると、ユーザ プロファイルが更新される。

## テクニカル ノート    テンプレート内にインクルードされる テンプレート

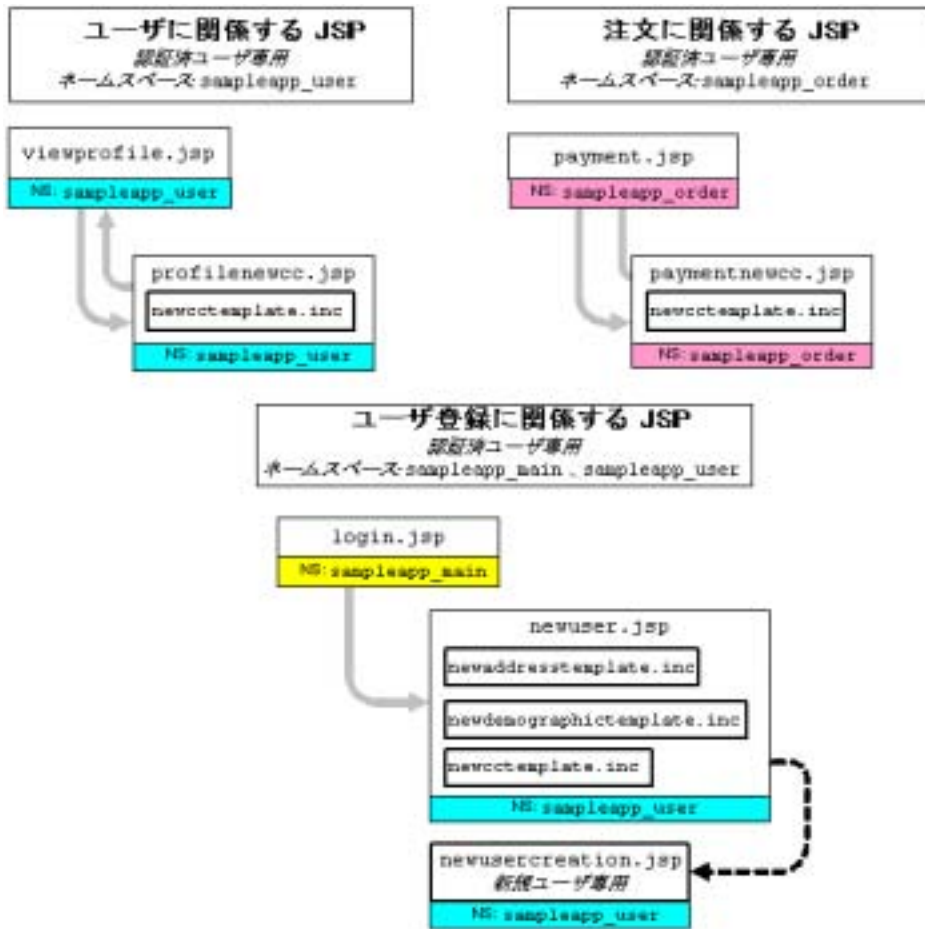
WebLogic Portal の JSP テンプレートでは、入力フィールドなどの情報を複数の  
ファイルに重複して定義するのではなく、再利用される JSP タグと HTML 要素  
を一群のインクルード ファイルに割り当て、今度はこれらのファイルが Web サ  
イトの開発に使える JSP テンプレートになります。たとえば、

paymentnewcc.jsp (先ほどの節で説明したもの)、profilenewcc.jsp、および  
newuser.jsp は、newcctemplate.jsp を子ページとしてインクルードし一貫性  
のあるクレジットカード情報入力フィールドを提供する親 JSP ページです。

newcctemplate.jsp は、作成対象の他の JSP ファイルにインクルードするこ  
とができます。

Webflow では、これら 3 つの親 JSP ページを使って、ユーザのコンテキストを決  
定し、ユーザが新しいクレジットカード情報を入力したあとのどのページを表示  
すべきかを決定します。たとえば、[図 3-16](#) は、ユーザが paymentnewcc.jsp、  
profilenewcc.jsp、newuser.jsp に必要な情報を入力すると、Webflow がそれ  
ぞれ payment.jsp、viewprofile.jsp、login.jsp を表示することを示してい  
ます。

図3-16 テンプレート インクルード ファイル



ツアーを続行するには

1. クレジットカードの情報を入力します。
2. 残りの必須フィールドに情報を入力し、[Save] をクリックします。



## ステップ 8: クレジット カードの選択

payment.jsp で [Save] をクリックすると、WebLogic Portal は新たに追加されたクレジットカード情報を payment.jsp に表示します ( 図 3-17 を参照 )。

### payment.jsp のその他の特徴

図 3-17 クレジット カードの選択



payment.jsp の以下の特徴に注目してください。

- User Management JSP タグによって、ユーザ プロファイルに基づきクレジットカード リストが生成される。以前初めて payment.jsp にアクセスしたときには ( 図 3-14 を参照 ) User Management JSP タグによってユーザ プロファイルから返されるクレジットカード定義は存在しませんでした。

ツアーを続行するには

[Use] をクリックします。

## ステップ 9: 商品の購入

payment.jsp で [Use] をクリックすると、WebLogic Portal は checkout.jsp を表示し、ユーザはこのページで税金に関する情報をレビューして購入を完了します。

### checkout.jsp の特徴

図 3-18 商品の購入

ようこそ  
鈴木一郎 様  
[プロフィールを見る](#)  
[ログアウト](#)

履歴を見る  
[ユーザ](#)  
[支払い](#)

**Don't Forget**  
Extension Cords!  
100% click here

Catalog data provided courtesy of [IPN Register](#), where supply meets demand.

### 最終チェックアウトの確認

**お届け先**  
鈴木一郎  
〒105-0001  
東京都 港区虎ノ門 2-10-1  
新日協ビル東棟14F  
Japan

**分割発送オプション**  
一括発送

**特別な指示**

**配送方法**  
宅急便

**支払い方法**  
クレジットカード: xxxxxxxxxxxx1111

オーダ				
ID	詳細	数量	実価格	小計
9-27205	drill-9-27205	1	¥ 13,200	¥ 13,200
			送料および手数料	¥ 495.00
			消費税	¥ 718
				<b>お支払総額 ¥ 15,097</b>

Taxation サービスから受け取った税金情報を示す。

[< Back](#) [Complete purchase >](#)

Payment サービスと Taxation サービスを呼び出す。

checkout.jsp の以下の特徴に注目してください。

- このファイルの絶対パス名は、以下のとおり。  
<BEA\_HOME>/weblogic700/samples/portal/wlcsDomain/beaApps/wlcsApp/wlcs/commerce/order/  
checkout.jsp (UNIX)
- JSP タグを使って、**Taxation サービス**で計算される税金情報を始めとする情報を Pipeline から収集し表示する。
- **[Complete purchase]** をクリックすると、**Payment サービス**が呼び出される。これは、WebLogic Portal におけるビジネス ロジックの実装の 1 つで、支払を処理するものです。

ツアーを続行するには

**[Complete purchase]** をクリックします。

## ステップ 10: 注文確認のレビュー

**[Complete purchase]** をクリックすると、WebLogic Portal は購入処理を完了し、confirmorder.jsp に確認番号を表示します ( [図 3-19](#) 参照 )。

## confirmorder.jsp の特徴

図 3-19 注文確認のレビュー

The screenshot shows a web page for order confirmation. On the left is a sidebar with navigation links like 'ようこそ', 'プロフィールを見る', and '注文を見る'. The main content area is titled 'オーダの確認' and includes a confirmation number '#4004', a card number, and shipping address. Below this is a table of items with columns for ID, 詳細, 数量, 単価, and 小計. A 'Continue shopping' button is at the bottom.

ID	詳細	数量	単価	小計
9-27205	drill-9-27205	1	¥ 13,200	¥ 13,200
送料および手数料				¥ 495.00
消費税				¥ 718
請求金額				¥ 15,097

confirmorder.jsp の以下の特徴に注目してください。

- このファイルの絶対パス名は、以下のとおり。  
`<BEA_HOME>/weblogic700/samples/portal/wlcsDomain/beaApps/wlcsApp/wlcs/commerce/order/confirmorder.jsp`  
 (UNIX)
- Pipeline メカニズムによって確認番号が生成される。
- Pipeline JSP タグによって、このページの情報が収集され表示される。

## ツアーを続行するには

新規ユーザ向けのツアーはこれで終わりです。JSP テンプレートのツアーを続けるには、ナビゲーションバーの[ **ホーム** ]をクリックして、次のセクションである第 4 章「登録ユーザによる商品の購入」に進みます。



---

## 第4章 登録ユーザによる商品の購入

この e コマース Web サイトにアクセスした際に、品揃えのよさと値段の安さに感心したので、再びサイトにアクセスして、自宅の修繕計画を開始しようとしている友人のためにドリルを買うことにします。今回のツアーでは、この Web サイト内を以下の順に進んでいきます。

### ■ 登録ユーザ ロードマップ

**注意：** 登録ユーザ ロードマップを別のブラウザ ウィンドウに開いておくとういでしょう。ツアー中にマップの内容を参照したほうがよい場合が多いからです。

- ステップ 1: カタログのブラウズと商品の選択
- ステップ 2: ショッピング カートのレビューとチェックアウト
- 寄り道 : 認証済ユーザ用ホーム ページの復習
- ステップ 3: 出荷オプションの選択
- ステップ 4: 新しい届け先住所の入力と選択
- ステップ 5: クレジット カードの選択
- ステップ 6: 商品の購入
- ステップ 7: 注文確認のレビュー

**注意：** この章では、製品名を WebLogic Portal と呼ぶことにします。BEA Campaign サービスのライセンスを取得している場合、この製品パッケージには Commerce サービス が同梱されています。

## 登録ユーザ ロードマップ

ツアーのこのセクションを開始する前に、JSP テンプレートの処理フローを以下のロードマップ (図 4-1、図 4-2、および図 4-3) で確認しておいてください。これらのロードマップに示されている矢印は、Webflow が提供するナビゲーション上の選択肢を表しています。そのうち、濃い矢印は、ツアーのこのセクションで説明する経路を示しています。

図 4-1 カタログをブラウズするための JSP

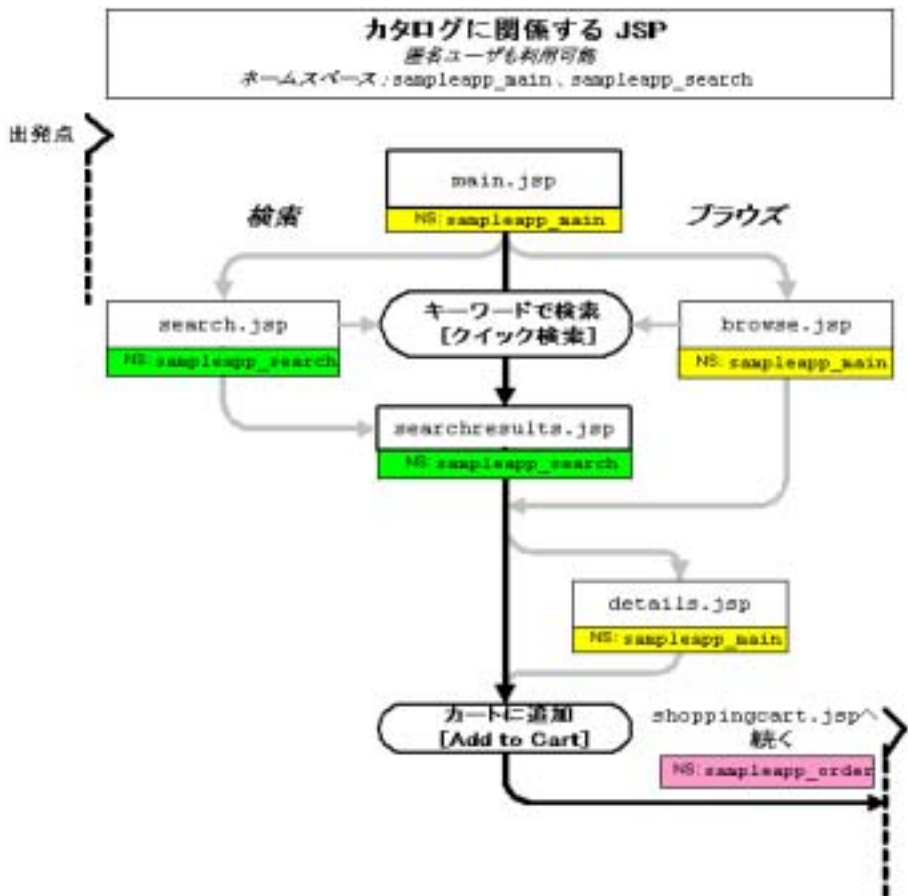




図 4-2 ショッピングカートとユーザログインのための JSP

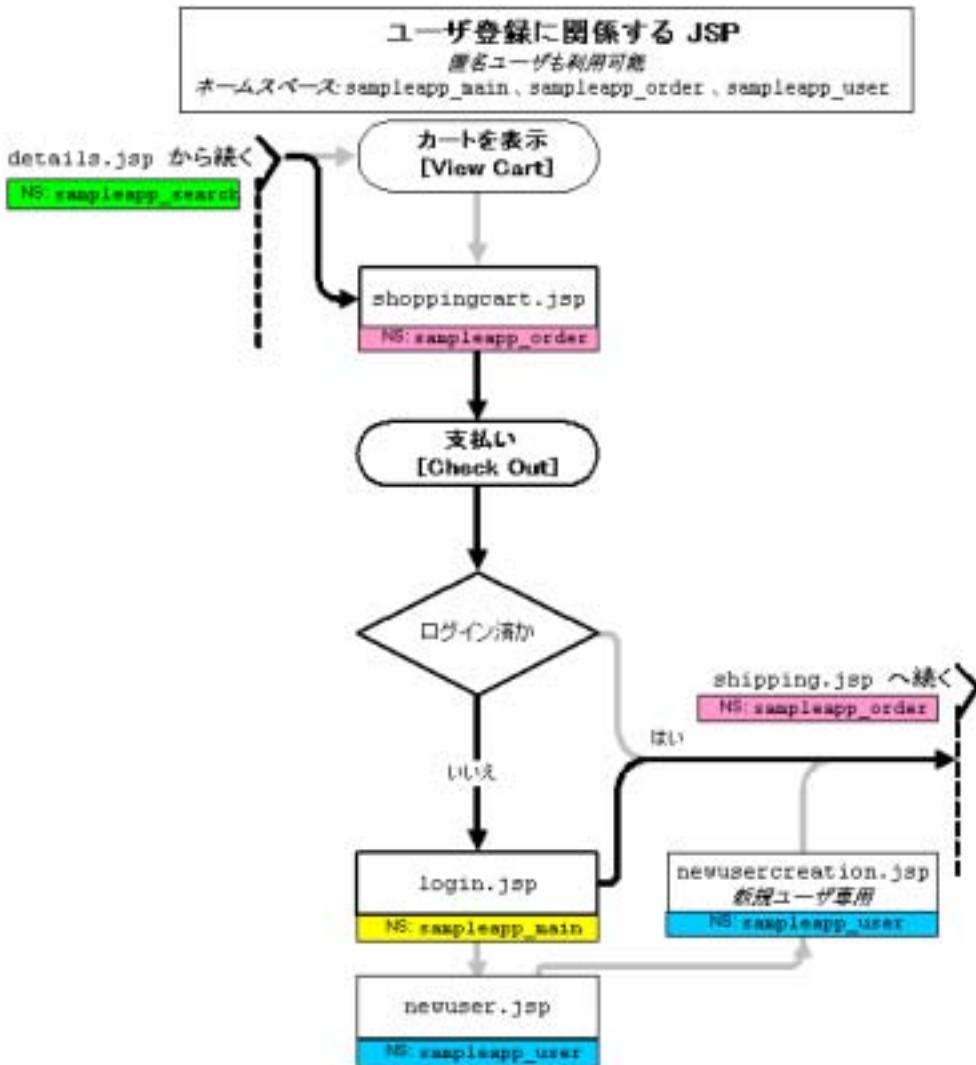
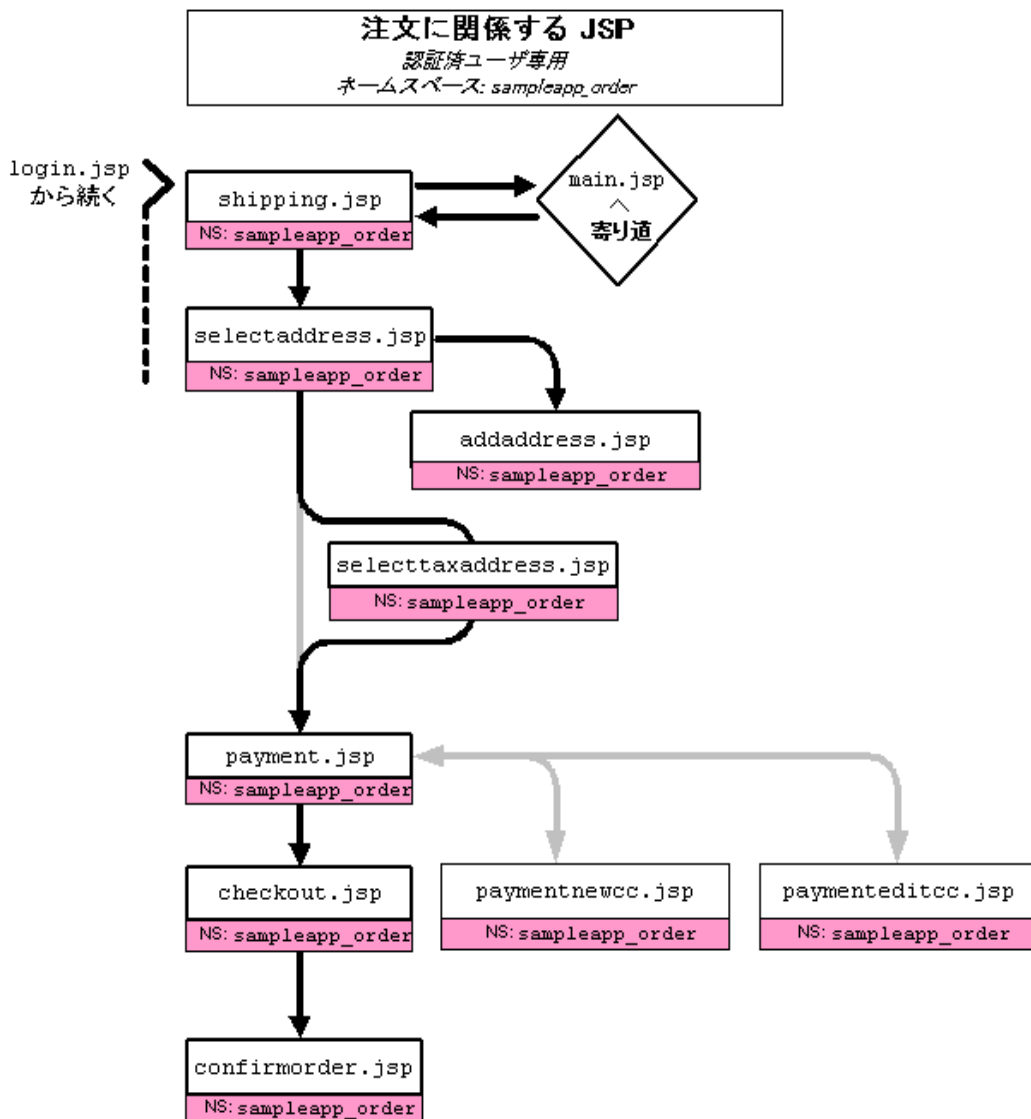


図 4-3 注文完了のための JSP



## ステップ 1: カタログのブラウズと商品の選択

ツアーにログインしていない場合は、再びログインします。再度ショッピングを行うためのホーム ページにアクセスしたあと、カタログに載っているドリルをブラウズすることにします。

ツアーを続行するには

[パワー ツール] をクリックします。

[パワー ツール] をクリックすると、WebLogic Portal は「パワー ツール」カテゴリの内容を `browse.jsp` に表示します ( 図 4-4 を参照)。

### browse.jsp の特徴 (前半)

図 4-4 カタログの第 2 レベル



browse.jsp の以下の特徴に注目してください。

- このファイルの絶対パス名は、以下のとおり。  
`<BEA_HOME>/weblogic700/samples/portal/wlcsDomain/beaApps/wlcsApp/wlcs/commerce/catalog/browse.jsp (UNIX)`
- Pipeline JSP タグによってプロダクト カタログにクエリが発行され、兄弟カテゴリ（同レベルのカテゴリ）の一覧が取得される。現在のカテゴリの名前とその兄弟カテゴリへのリンクがカテゴリの名前の上に表示されます。
- 中央のカラムでは、以下の JSP タグによって、現在のカテゴリからカタログのトップ レベルまでのツリー ビューが作成される。  
`<jsp:include page="/commerce/catalog/includes/navigation.jsp" flush="true"/>`
- [コードリスト 4-1](#) は、Catalog によって生成される JSP ファイルが browse.jsp にインクルードされ、現在のカテゴリの内容が記述されている様子を示している。

### コード リスト 4-1 現在のカテゴリを表示する Catalog タグ

---

```
<!-- メイン コンテンツ -->
<!-- 現在のカテゴリから集計 JSP を取得する --%>
<catalog:getProperty object="<%= category %>"
propertyName="Jsp"
id="displayJsp"
returnType="com.beasys.commerce.ebusiness.catalog.JspInfo"/>

<!-- このカテゴリのカテゴリ レイアウト用 JSP をインクルードする --%>
<jsp:include page="<%= displayJsp.getUrl() %>" flush="true"/>
```

---

ツアーを続行するには

[Drills] をクリックします。

[Drills] をクリックすると、WebLogic Portal は「Drills」カテゴリの内容を browse.jsp に表示します（図 4-5 を参照）。

## browse.jsp のその他の特徴（後半）

図 4-5 カタログの第 3 レベル



browse.jsp のさらに以下の特徴にも注目してください。

- これは 図 4-4 に示すのと同じソース ファイルであるが、ユーザのコンテキストが異なるのでコンテンツも異なる。
- Pipeline JSP タグによってプロダクト カタログにクエリが発行され、兄弟カテゴリの一覧が取得される。今回このページでは 図 4-4 に示したカテゴリとは異なるカテゴリが表示されるので、一覧も変わります。
- 中央のカラムでは、JSP タグによって、現在のカテゴリを反映するようツリー ビューが変更される。

- 中央のカラムの 2 行目では、[図 4-4](#) に示したものと異なる JSP ファイルが Catalog によって生成されるので、[図 4-5](#) の内容とレイアウトも変わる。
- ドリル カテゴリ内の全商品はカテゴリ JSP と集計 JSP を使って表示される。各ドリルには、詳細な製品情報へのリンクが付いています。

### ツアーを続行するには

ドリルの名前をクリックします。

ドリルの名前をクリックすると、WebLogic Portal は製品説明を `details.jsp` に表示します ([図 4-6](#) を参照)。

## details.jsp の特徴

図 4-6 商品の詳細



`details.jsp` の以下の特徴に注目してください。

- このファイルの絶対パス名は、以下のとおり。  
`<BEA_HOME>/weblogic700/samples/portal/wlcsDomain/beaApps/wlcsApp/wlcs/commerce/catalog/details.jsp (UNIX)`

- このページには、`browse.jsp` が `searchresults.jsp` のいずれかからアクセスすることができる ( [図 4-1](#) 参照 )。
- ユーザが `browse.jsp` からこのページにアクセスした**場合にのみ** ツリービューを作成する JSP ファイル (`navigation2.jsp`) がインクルードされている。ユーザが `searchresults.jsp` からこのページにアクセスしたと Pipeline メカニズムによって判定された場合には、このインクルード JSP ファイルはツリー ビューを作成しません。
- 表示される商品の要約用 JSP ファイルがインクルードされている。この要約用 JSP ファイルは、プロダクト カタログに記載されている商品情報に基づきプロダクト カタログによって生成されます。

ツアーを続行するには

[Add to cart] をクリックします。

## ステップ 2: ショッピング カートのレビューとチェック アウト

[Add to cart] をクリックすると、WebLogic Portal は、ユーザが選択した商品を `shoppingcart.jsp` に表示します ( [図 4-7](#) を参照 )。

図4-7 ショッピングカート



shoppingcart.jsp については、8 ページの「shoppingcart.jsp の特徴」を参照してください。

ツアーを続行するには

[Check out] をクリックします。

[Check out] をクリックすると、WebLogic Portal はログインするようユーザに促します (図 4-8 を参照)。第 3 章「新規ユーザによる商品の購入」ですでにユーザプロフィールを作成したので、今回は登録ユーザです。



図 4-8 ログイン



login.jsp については、11 ページの「[login.jsp の特徴](#)」を参照してください。

## テクニカル ノート セキュリティ モデル

ユーザがリンクをクリックしたときにログインが必要かどうかを、WebLogic Portal はどのように判断しているのでしょうか。WebLogic Portal では、Webflow を使ってセキュリティを実装するのではなく、J2EE のセキュリティ仕様に従っています。この仕様では、どのファイル セットが認証済ユーザからしかアクセスできないのかを Web サイト管理者が決定することになっています。デフォルトでは、order ディレクトリと user ディレクトリの中の JSP テンプレートはすべて、認証済ユーザからしかアクセスできません。

匿名ユーザがこれらのファイルのいずれかにアクセスしようとする、WebLogic Portal は login.jsp を表示し、このページからユーザは、ログインするか、あるいは新しいユーザ アカウントを作成することができます。認証が済むと、Webflow によってそのあとのページの表示シーケンスが決定されます。

### ツアーを続行するには

10 ページの「[ステップ 4: 新規ユーザ プロファイルの作成](#)」でセットアップしたユーザ名とパスワードを入力したあと、[ **ログイン** ] をクリックします。

[ **ログイン** ] をクリックすると、WebLogic Portal は出荷オプションを `shipping.jsp` に表示します ( [図 4-9](#) を参照 )。

**図 4-9 出荷ページでの寄り道**



`shipping.jsp` については、19 ページの「[shipping.jsp の特徴](#)」を参照してください。

### 寄り道を少々

ツアーを続ける前に、少し時間を割いてホーム ページについて復習しておきましょう。すでにログインしている ( 認証済み ) ので、ホーム ページには、認証済ユーザだけが利用可能な選択肢が表示されています。

寄り道をするには、[ **ホーム** ] をクリックします ( [図 4-9](#) を参照 )。

# 寄り道：認証済ユーザ用ホーム ページの復習

[ホーム] をクリックすると、WebLogic Portal は main.jsp を表示します ( 図 4-10 を参照 )。

## 認証済ユーザ用の main.jsp の特徴

図 4-10 認証済ユーザ用のホーム ページ



main.jsp の以下の特徴に注目してください。

- 認証が済んでいるので、main.jsp には、ユーザ名と、注文および支払の履歴へのリンクが表示される。ユーザ名が Pipeline セッションに関連付けられている場合には、leftside.inc ファイル内の User Management JSP タグでデータベースにクエリを発行してユーザ名を検索し、その結果を表示します。

ツアーを続行するには

1. [ **カートの中を見る** ] をクリックします。
2. 「ショッピングカート」のページで [ **Check out** ] をクリックします。

## ステップ 3: 出荷オプションの選択

`shoppingcart.jsp` で [ **Check out** ] をクリックすると、WebLogic Portal は出荷オプションを `shipping.jsp` に表示します ( [図 4-9](#) を参照 )。

`shipping.jsp` については、[19 ページ](#)の「[shipping.jsp の特徴](#)」を参照してください。

ツアーを続行するには

`shipping.jsp` で [ **Continue** ] をクリックします。

## ステップ 4: 新しい届け先住所の入力と選択

`shipping.jsp` で [ **Continue** ] をクリックすると、WebLogic Portal は届け先住所を `selectaddress.jsp` に表示します。

自分自身以外の人に商品を発送するには、新しい住所を入力しなければなりません。

`selectaddress.jsp` については、[20 ページ](#)の「[selectaddress.jsp の特徴](#)」を参照してください。

ツアーを続行するには

[ **Add address** ] をクリックします。

[ **Add address** ] をクリックすると、WebLogic Portal は `addaddress.jsp` を開いて、新しい届け先住所を入力するようユーザに促します ( [図 4-11](#) を参照 )。

図 4-11 新しい住所の入力



ツアーを続行するには

住所のフィールドに以下の情報を入力します。

- 郵便番号 :105-0001
- 都道府県 : 東京都
- 市区町村 : 港区虎ノ門
- 番地 : 2-10-1
- アパート・マンション : 新日鉱ビル 14F
- 国 : Japan

[Save] をクリックします。

[Save] をクリックすると、WebLogic Portal は、ユーザ プロファイルに記載されている住所を `selectaddress.jsp` に表示します (図 4-12 を参照)。

図 4-12 新しい住所の選択



`selectaddress.jsp` については、20 ページの「[selectaddress.jsp の特徴](#)」を参照してください。

ツアーを続行するには

先ほど作成した「東京都港区虎ノ門」の住所の右側にある [Use] をクリックします。

[Use] をクリックすると、自動的に消費税が計算されます。

## ステップ 5: クレジットカードの選択

[Use] をクリックすると、WebLogic Portal は `payment.jsp` を表示し、その中に、ユーザ プロファイルに記載されているクレジットカードが列挙されます ( [図 4-13](#) を参照 )。

図 4-13 クレジットカードの選択



payment.jsp については、23 ページの「payment.jsp の特徴」を参照してください。

ツアーを続行するには

[Use] をクリックします。

## ステップ 6: 商品の購入

payment.jsp で [Use] をクリックすると、WebLogic Portal は、購入に関する最終情報を checkout.jsp に表示します (図 4-14 を参照)。





## ステップ 7: 注文確認のレビュー

[Complete purchase] をクリックすると、WebLogic Portal は購入処理を完了し、確認番号を `confirmorder.jsp` に表示します ( 図 4-15 を参照 )。

図 4-15 注文確認のレビュー

ようこそ  
録木一郎様  
[プロフィールを見る](#)  
[ログアウト](#)

履歴を見る  
[コード](#)  
[支払い](#)

クイック検索:  
キーワードを入力

See Our  
Large  
Selection  
of Saws  
Here!

Catalog data  
provided courtesy of  
[IPH Requester](#), where  
supply meets  
demand.

### オーダの確認

このページを印刷して、大切に保存しておいてください。

BEA WebLogic Commerce Server 3.5 でお買い上げいただきありがとうございます。またのお越しをお待ちしております。

#### オーダの確認 #4005

ご利用のカード:  
XXXXXXXXXXXX1111

お届け先:  
録木一郎  
〒105-0001  
東京都港区虎ノ門 2-10-1  
新日館ビル東棟14F  
Japan

発送オプション:  
定価値  
一括発送

ID	詳細	数量	単価	小計
9-10505	drill-9-10505	1	¥ 10,100	¥ 10,100
送料および手数料				¥ 495.00
消費税				¥ 556
請求金額				¥ 11,680

\*\*は特定のアイテムIDに関連する割引、もしくは調整を示しています。

[Continue shopping](#)

`confirmorder.jsp` については、30 ページの「[confirmorder.jsp の特徴](#)」を参照してください。

### ツアーを続行するには

登録ユーザによる商品の購入を想定したツアーはこれで終わりです。

JSP テンプレートのツアーを続けるには、ヘッダー部の [[プロフィールを見る](#)] をクリックして、第 5 章「[登録ユーザによるクレジットカードの追加](#)」に進みます。



---

## 第5章 登録ユーザによるクレジット カードの追加

ツアーのこのセクションでは、Web サイトにアクセスして、自分のユーザ プロファイルへのクレジット カード情報の追加と、最後に行った注文のステータス 確認を行いたいと仮定します（これはおそらく、顧客にとって現実的なシナリオ ではないでしょうが、JSP テンプレートで利用できる付加的な機能を浮き彫りに する上で役に立ちます）。ツアーのこのセクションでは、以下の順に話を進めて いきます。

- ユーザ プロファイル ロードマップ

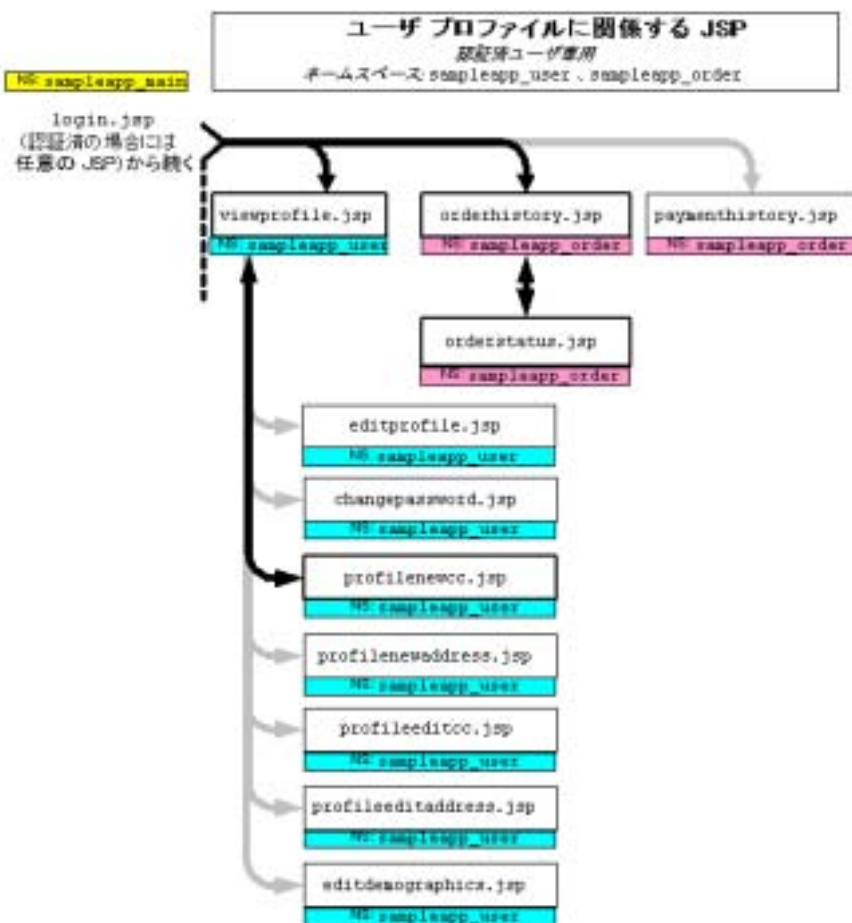
**注意：** ユーザ プロファイル ロードマップを別のブラウザ ウィンドウに開いて おくとよいでしょう。ツアー中にマップの内容を参照したほうがよい場 合が多いからです。

- ステップ 1: ユーザ プロファイルへのアクセス
- ステップ 2: ユーザ プロファイルのレビュー
- ステップ 3: クレジット カードの追加
- ステップ 4: 注文の履歴とステータスの表示
- ステップ 5: ログアウト

# ユーザ プロファイル ロードマップ

ツアーのこのセクションを開始する前に、[図 5-1](#) に目を通しておいてください。これが、ユーザ プロファイル関係の JSP テンプレートのロードマップになります。このロードマップに示されている矢印は、Webflow が提供するナビゲーション上の選択肢を表しています。そのうち、濃い矢印は、ツアーのこのセクションで説明する経路を示しています。

図 5-1 ユーザ プロファイルを変更するための JSP



## ステップ 1: ユーザ プロファイルへのアクセス

ユーザは、画面の左側にある [ プロファイルを見る ] をクリックすることで、Web サイトのどのページからでも自分のユーザ プロファイルにアクセスすることができます。各ページのこの部分はインクルード ファイル `leftside.inc` から読み出されます。ユーザが匿名で Web サイトにアクセスしている場合には、WebLogic Portal はプロフィールを表示する前に、ログインするようユーザに促します。

### ツアーを続行するには

WebLogic Portal によってログイン ページが表示されたらログインします。

WebLogic Portal のセキュリティについては、[11 ページの「テクニカル ノート セキュリティ モデル」](#)を参照してください。

## ステップ 2: ユーザ プロファイルのレビュー

認証が済んでから [ プロファイルを見る ] をクリックするか、`login.jsp` で [ ログイン ] をクリックすると、WebLogic Portal は `viewprofile.jsp` を表示します ( [図 5-2](#) を参照 )。

## viewprofile.jsp の特徴

図 5-2 ユーザプロフィールの内容



viewprofile.jsp の以下の特徴に注目してください。

- このファイルの絶対パス名は、以下のとおり。  
`<BEA_HOME>/weblogic700/samples/portal/wlcsDomain/beanApps/wlcsApp/wlcs/commerce/user/viewprofile.jsp` (UNIX)
- HTTPS セッションを 7502 番ポートに作成して、暗号化されたデータを SSL を通じて送信する。

- これまでのショッピング セッションで追加された届け先住所を表示する。最初の住所は、届け先住所が問い合わせ先住所と同じであるとユーザが指定したときに、デフォルトで追加されたものです（10 ページの「[ステップ 4: 新規ユーザ プロファイルの作成](#)」を参照）。
- 最初のショッピング セッションで入力されたクレジットカード情報を表示する。

ツアーを続行するには

[Add card] をクリックします。

## ステップ 3: クレジット カードの追加

[Add card] をクリックすると、WebLogic Portal は `profilenewcc.jsp` を開いて、クレジットカード情報を入力するようユーザに促します（[図 5-3](#) を参照）。

## profilenewcc.jsp の特徴

図 5-3 クレジット カードの追加



profilenewcc.jsp の以下の特徴に注目してください。

- このファイルの絶対パス名は、以下のとおり。  
`<BEA_HOME>/weblogic700/samples/portal/wlcsDomain/beaApps/wlcsApp/wlcs/commerce/user/profilenewcc.jsp` (UNIX の場合)
- 以下の JSP タグによって、データ入力フィールドを定義するファイルがインクルードされる。  
`<%@ include file="/commerce/includes/newcctemplate.jsp" %>`



テンプレート インクルード ファイルについては、[25 ページの「テクニカルノート テンプレート内にインクルードされるテンプレート」](#)を参照してください。

ツアーを続行するには

1. クレジット カードの情報を入力します。
2. 残りの必須フィールドに情報を入力したあと、[Save] をクリックします。  
ツアーの次のステップでは、注文の履歴とステータスが示されますが、これらには認証済ユーザ用のホーム ページからアクセスします。
3. ヘッダー内の [ホーム] をクリックします。

## ステップ 4: 注文の履歴とステータスの表示

[ホーム] をクリックすると、WebLogic Portal は、Web サイトのホーム ページである `main.jsp` を表示します( [図 5-4](#) を参照 )。 `main.jsp` では 7501 番ポートを通じて HTTP プロトコルを使う点に注意してください。つまり、WebLogic Portal は、このホーム ページには SSL を使わないのです。

図 5-4 [オーダ]のクリック



認証済ユーザ用の main.jsp の詳細については、13 ページの「認証済ユーザ用の main.jsp の特徴」を参照してください。

ツアーを続行するには

[オーダ] をクリックすると、WebLogic Portal は orderhistory.jsp を表示します (図 5-5 を参照)。

## orderhistory.jsp の特徴

図 5-5 注文の履歴



orderhistory.jsp の以下の特徴に注目してください。

- このファイルの絶対パス名は、以下のとおり。  
`<BEA_HOME>/weblogic700/samples/portal/wlcsDomain/beaApps/wlcsApp/wlcs/commerce/order/orderhistory.jsp (UNIX)`
- 以下の Pipeline JSP タグによって履歴が取得される。

```
<!-- スクロール可能なモデルの抽出 -->
```

```
<webflow:getProperty id="orderHistory"
property="<%=PipelineSessionConstants.SCROLLABLE_MODEL%>"
type="com.beasys.commerce.ebusiness.util.ScrollableModel"
scope="session" namespace="sampleapp_main" />
```

- WebLogic Portal は、データベース管理者がデータをアーカイブ化するまで、この履歴を WLCS\_ORDER データベース テーブルに格納します。

ツアーを続行するには

[View] をクリックして、注文のステータスを確認します。

[View] をクリックすると、WebLogic Portal は `orderstatus.jsp` を表示します (図 5-6 を参照)。

## orderstatus.jsp の特徴

図 5-6 注文のステータス

The screenshot shows a web page for 'Commerce Templates' with a navigation bar and a main content area. The main content area is titled 'オーダー状況' (Order Status) and displays the following information:

- Order ID: 1
- Order Status: Authorized
- Order Date: 2003-07-05
- Order Type: 一括発注 (Bulk Order)
- Order Description: 特別な指示 (Special Instructions)
- Order Location: 東京都港区北門2-10-1, 新日産ビル4F, Japan

注文のステータスは、その注文に関する他のデータと一緒に WLCSS\_ORDER テーブルに登録されている。

ID	品名	数量	小計
1-62250	603-V-62250	1	¥ 39,100
71-UF4200	3000 71-UF4200	1	¥ 5,150
送料および手数料			¥ 495.00
消費税			¥ 1,981
お支払総額			¥ 41,613

`orderhistory.jsp` の以下の特徴に注目してください。

- このファイルの絶対パス名は、以下のとおり。  
<BEA\_HOME>/weblogic700/samples/portal/wlcsDomain/beaApps/wlcsApp/wlcs/commerce/order/orderstatus.jsp (UNIX)

- 以下の Pipeline JSP タグによって、注文のプロパティが収集される。

```
<!-- CustomerValue を取得する Bean の定義 -->
<webflow:getProperty id="orderValue"
property="<%=PipelineSessionConstants.SELECTED_ORDER%>"
type="com.beasys.commerce.ebusiness.order.OrderValue"
scope="session" namespace="sampleapp_main" />
```

- 注文データベース テーブルには注文のプロパティが格納されており、その中に注文のステータスがあります。

**注意:** Pipeline によって注文に割り当てられる状態は、New か Submitted だけです。それ以外の状態を追加したい場合には、Pipeline を変更しなければなりません。

ツアーを続行するには

[ ホーム ] をクリックします。

## ステップ 5: ログアウト

認証済セッションを終了するには、[ ログアウト ] をクリックします。ログアウトすると、Pipeline セッションが終了し、ショッピングカートの中身が空になり、(再びログインするまで) ユーザ名が Web サイト上でそのあとの出来事とは無関係になります。

ここから先の話

JSP テンプレートと Webflow の変更方法の例については、[第 6 章「テンプレートのカスタマイズ例」](#)を参照してください。



---

## 第6章 テンプレートのカスタマイズ例

ツアーのこれまでのセクションでは、コマース テンプレートの [Add to cart] リンクをクリックすると、WebLogic Portal が `shoppingcart.jsp` を表示して、チェックアウトするようユーザに促していました。しかし、ユーザが [Add to cart] をクリックするとプロダクト カタログに戻るようにすることで、顧客にショッピングを続けるよう仕向けるビジネス モデルもあり得ます。

wlcs Web アプリケーションに関連付けられた Webflow を修正すれば、[Add to cart] リンクの動作を変更できますが、Webflow を修正する前に、いくらか背景的知识が必要です。ツアーのこのセクションでは、以下の順に話を進めていきます。

- [ステップ 1: \[Add to cart\] リンクを理解する](#)
- [ステップ 2: \[Add to cart\] リンクの変更](#)
- [ステップ 3: 変更の検証](#)

## ステップ 1: [Add to cart] リンクを理解する

WebLogic Portal におけるリンク アンカでは、静的なファイル名を URL として指定するのではなく、Webflow メカニズムを呼び出して遷移先リソースを決定します。[Add to cart] リンクのリンク アンカ ( [コード リスト 6-1](#) に示す ) は、インクルード ファイル

<BEA\_HOME>/weblogic700/samples/portal/wlcsDomain/beaApps/wlcsApps/wlcs/commerce/catalog/includes/itemdetails.jsp に記述されています。

### コード リスト 6-1 [Add to cart] アンカ

---

```
<!-- Add the 'Put in Cart' link -->
<%
allParams = HttpRequestConstants.CATALOG_ITEM_SKU + "=" +
productItem.getKey().getIdentifier();
%>
<td align="right" valign="top" width="5%"><a
href="<webflow:createWebflowURL event="link.add"
namespace="sampleapp_order" extraParams="<%= allParams %>" />">
  " border="0"
alt="Add to Shopping Cart"></a>
</td>
```

---

ツアーのこのステップでは次の動作について説明します。

- [リンク アンカが Webflow を呼び出す仕組み](#)
- [Webflow がリンクを解決する仕組み](#)

## リンク アンカが Webflow を呼び出す仕組み

アンカは <webflow:createWebflowURL> JSP タグを使って、データを Webflow に渡します。



<webflow:createWebflowURL> JSP タグは、以下の 3 つのデータ項目を Webflow に渡します。

- イベントのタイプと名前。[Add to cart] リンクの場合には、イベントのタイプは link で、名前は add です。
- リンク ソース（遷移元）が記述されているファイルが存在する、ネームスペースの名前。[Add to cart] リンクの場合には、このネームスペース名は sampleapp\_order です。
- インクルードが必要なその他のすべてのパラメータ。

こうしたデータを任意の数の Pipeline を通じて送ったあと、Webflow はリンクを解決する URL を返します。

## Webflow がリンクを解決する仕組み

<webflow:createWebflowURL> JSP タグがデータを Webflow に送ると、Webflow は[コード リスト 6-2](#) に示すように、ワイルドカード表示ノードのイベント リストから link.add イベントの sampleapp\_order.wf を検索します。

### コード リスト 6-2 sampleapp\_order.wf での [Add to cart] リンクの指定

---

```
<wildcard-presentation-origin node-type="jsp">
  <event-list>
    ...
    <event event-name="link.add">
      <destination namespace="sampleapp_order"
        node-name="addProductItemToShoppingCartIP"
      >
    </event>
    ...
  </event-list>
</wildcard-presentation-origin>
```

---

この場合 link.add イベントの送り先として示されているのは、sampleapp\_order にある addProductItemToShoppingCartIP 入力プロセッサの呼び出しです。そのため、Webflow はコードリスト 6-3 に示すように、プロセッサ ノード元のリストから addProductItemToShoppingCartIP 入力プロセッサを検索します。

### コード リスト 6-3 sampleapp\_order.wf での addProductItemToShoppingCartIP 入力プロセッサの検索

---

```
<processor-origin node-name="addProductItemToShoppingCartIP"
  node-type="inputprocessor">
  <node-processor-info
class-name="examples.wlcs.sampleapp.catalog.webflow.GetProductItemIP"/>
  <event-list>
    <event event-name="success">
      <destination namespace="sampleapp_order"
        node-name="addProductItemToShoppingCart"
node-type="pipeline"/>
    </event>
  </event-list>
  ...
</processor-origin>
```

---

addProductItemToShoppingCartIP 入力プロセッサが正常に実行されると、Webflow は addProductItemToShoppingCart Pipeline を呼び出します。Webflow はコード リスト 6-4 に示すように、sampleapp\_order を検索して addProductItemToShoppingCart Pipeline を見つけます。

### コード リスト 6-4 sampleapp\_order.wf での addProductItemToShoppingCart Pipeline の呼び出し

---

```
<processor-origin node-name="addProductItemToShoppingCart"
node-type="pipeline">
  <node-processor-info
pipeline-name="addProductItemToShoppingCart"/>
  <event-list>
    <event event-name="success">
      <destination namespace="sampleapp_order"
```

```
        node-name="addProductItemToShoppingCartTracker"  
node-type="pipeline" />  
    </event>  
    </event-list>  
  
    ...  
  
</processor-origin>
```

---

同様に、addProductItemToShoppingCart Pipeline が正常に実行されると、この Pipeline はデータを addProductItemToShoppingCartTracker という名前の別の Pipeline に送ります。addProductItemToShoppingCartTracker Pipeline が正常に実行されると、次に Webflow は refreshSavedList Pipeline を呼び出します。refreshSavedList Pipeline が正常に実行されると、shoppingcart.jsp が再ロードされます。

## ステップ 2: [Add to cart] リンクの変更

ユーザから見た [Add to cart] リンクの動作を決定する Webflow の処理を理解したら、それを変更して、顧客をカタログのトップレベルに導くように指定することができます。

**第 3 章「新規ユーザによる商品の購入」**では、プロダクトカタログのトップレベルが main.jsp に表示されていたことを思い出してください。ただし、refreshSavedList Pipeline の success イベントを変更し送り先として (shoppingcart.jsp ではなく) main.jsp をロードしても、望みどおりの結果は得られません。これは、main.jsp が Web ブラウザに表示可能な標準 HTML タグではなく、JSP タグで記述されているからです。refreshSavedList Pipeline の success イベントの出力先を変更して直接 main.jsp を表示しようとすると、Web ブラウザには空白のページが表示されてしまいます。

そのようなやり方ではなく、getTopCategoriesIP という入力プロセッサを呼び出すのです。この入力プロセッサは sampleapp\_main ネームスペースにあり、プロダクトカタログ内のカテゴリに関するデータを収集して、それを main.jsp に表示します。

## getTopCategoriesIP 入力プロセッサの出力先を設定する

Webflow/Pipeline エディタを使用すれば、refreshSavedList Pipeline の出力を getTopCategoriesIP 入力プロセッサに送ることができます。

エディタを使って sampleapp\_order.wf ファイルを開き、エンドコネクタを shoppingcart.jsp プレゼンテーション ノードから sampleapp\_main\_getTopCategoriesIP プロシキ ノードにドラッグして、refreshSavedList Pipeline ノードの success イベントの送り先を移動します。これで、イベント送り先が正しいネームスペースを指すようになります。

## ステップ 3: 変更の検証

変更内容を sampleapp\_order.wf に保存した後、[ツール | 同期] を選択し、E-Business Control Center を使って変更内容を WebLogic Portal サーバに同期させる必要があります。

同期が完了したら、details.jsp で [Add to cart] リンクをクリックして、変更結果を確かめることができます ( [図 6-1](#) を参照 )。WebLogic Portal サーバを再起動する必要はありません。

図 6-1 details.jsp での [Add to cart] のクリック



[Add to cart] をクリックすると、WebLogic Portal は、プロダクト カタログのトップ レベルを main.jsp に表示します。

ただし、WebLogic Portal が商品をショッピング カートに追加したことを検証するには、main.jsp から以下のことを実行します。

1. ヘッダーで [カートの中を見る] をクリックします。
2. WebLogic Portal からログインの指示があったとき、このツアーの前のいくつかの節で設定したユーザ名とパスワードを入力します。

[カートの中を見る] をクリックすると、WebLogic Portal は shoppingcart.jsp を表示します。この中にカートに追加された商品が入っています。

## ここから先の話

ツアーを続行し、JSP テンプレートで使われるキャンペーン機能とそれを E-Business Control Center で定義する方法を学習するには、[第 7 章「サンプルのキャンペーン」](#)を参照してください。

---

## 第7章 サンプルのキャンペーン

ツアーのこれまでの章では、サンプル Web アプリケーションのコマース関連の機能に焦点を当ててきました。説明では主に、JSP のテンプレートとタグを使用して WebLogic Portal アプリケーションを構築する Java エンタープライズ開発者または企業内開発者を対象に、機能の概略を示すことが目的でした。

ツアーのこの最終章では、キャンペーン関連の機能に焦点を当てます。特に、グラフィカルなクライアントツールの BEA E-Business Control Center を使用するパワー ユーザおよびマーケティング担当者の視点から説明を進めます。

この章では、以下のトピックについて説明します。

- [概論と前提条件](#)
- [キャンペーンの考察 : TourCampaign1](#)

### 概論と前提条件

このツアーでは、サンプルの基本的なキャンペーンを一通り紹介し、サンプルの JSP テンプレートに挿入されている簡単な JSP タグについて解説します。これらの JSP タグは、キャンペーンと関連付けられるコンテンツや、キャンペーンとは関係なく機能するコンテンツのプレースホルダを Web ページ上に提供する働きをします。コンテンツの特性は、E-Business Control Center でキャンペーンを定義する担当者、Web サイトの利用者のプロファイル、および、訪問客が Web サイトを利用している間に発生するイベントによって決定されます。

まず最初に、BEA E-Business Control Center とキャンペーンについての基本的な事項をいくつか挙げます。

- E-Business Control Center を使用して、パワー ユーザ、マーケティング担当者、企業内エンタープライズ開発者を含む幅広いユーザがプロモーション用キャンペーン、キャンペーンでの割引、スタンドアロン割引、顧客セグメント、コンテンツ セレクタ、広告プレースホルダ、カスタム イベントを定義および管理できる。

- キャンペーンは、ビジネスの戦略的目標を達成するために連携して機能するシナリオをひとまとめにして名前を付けたものである。キャンペーンでは、WebLogic Portal のさまざまなサービスを利用して、e コマース Web サイト上でのマーケティング目標を作成および追跡します。
- シナリオはキャンペーンの主要な基礎単位である。シナリオは1つまたは複数のシナリオ アクションで構成され、すべてのアクションは何らかの条件が満たされたときに発生します。たとえば、特定のイベントが発生した場合および顧客が特定の顧客セグメントに該当する場合など。

# キャンペーンの考察 : TourCampaign1

基本的なキャンペーンについて学ぶには、以下の手順のとおりにします。

1. クライアントの Windows システム上で、スタート メニューを使って E-Business Control Center を起動します。たとえば、次の順に選択します。  
[スタート | プログラム | BEA WebLogic Platform | WebLogic Portal 7.0 | E-Business Control Center]  
「初期化中...」というメッセージと共に E-Business Control Center クライアントのスプラッシュ画面が表示された後で、デスクトップ上にウィンドウが開きます。
2. E-Business Control Center のメニューで [ファイル | プロジェクトを開く] を選択します。  
**注意：** E-Business Control Center にアプリケーションをロードしてからでないと、サンプル データを表示することはできません。
3. 下記プロジェクトを開きます。  
[<BEA\_HOME> | weblogic700 | samples | portal | wlcsDomain | beaApps | wlcsApp-project | wlcsApp-project]  
E-Business Control Center のタイトル バーに示されたとおり、wlcsApp サンプル アプリケーションがロードされます。  
このアプリケーションは、WebLogic Portal に同梱されているサンプル エンタープライズ アプリケーションです。ロードされるのは、エンタープライズ アプリケーション自体ではなく、E-Business Control Center の各ツールと関



連付けられた XML ファイルのグループです。これらのファイルを WebLogic Portal サーバと同期させると、エンタープライズアプリケーションの動作を制御するのに役立ちます。

4. [エクスプローラ] ウィンドウ (図 7-1) で [ **ビジネス ロジック** ] タブをクリックし、[ **キャンペーン** ] アイコンをクリックします。

図 7-1 [エクスプローラ] ウィンドウ



5. 「**TourCampaign1**」 ファイルをシングルクリックすると説明が表示されません。この Web アプリケーションでは、顧客セグメント「Affluent Mr/Ms Fix-It」(収入の多い個人客) または「Contractor」(修繕業者) に属するユーザがログインした場合にバナー広告が表示されます。後に、サンプルの JSP テンプレート上でログイン アカウントを試行します。この時点では、TourCampaign1 に関連する定義を見ていくことにします。
6. 「**TourCampaign1**」 をダブルクリックします。しばらくしてから、図 7-2 に示すような編集ウィンドウが表示されます。

図 7-2 キャンペーンの編集ウィンドウ



7. [ **開始 / 停止** ]バーをクリックします。  
[ **開始 / 停止** ]ペインに、キャンペーンの開始日と停止日が表示されます。  
[ **終了条件** ]セクションには、達成されれば、停止日時より前であってもキャンペーンを終わらせる条件となる目標を設定できます。目標は、広告のインプレッション数（訪問者が Web ページ上の広告を見るだけの場合）、またはクリックスルー回数（訪問者が表示された広告を実際にクリックした場合）によって定めます。目標を達成し、キャンペーンを終了するのに必要とされるインプレッションまたはクリックスルーの数を決定してください。  
TourCampaign1 では、この目標を広告インプレッション数 25 に設定します。
8. 目標の設定を詳細に見るには、その目標をクリックしてから、[ **編集** ]ボタンをクリックします。図 7-3 に示すような [ **目標の詳細** ]ウィンドウが表示されます。

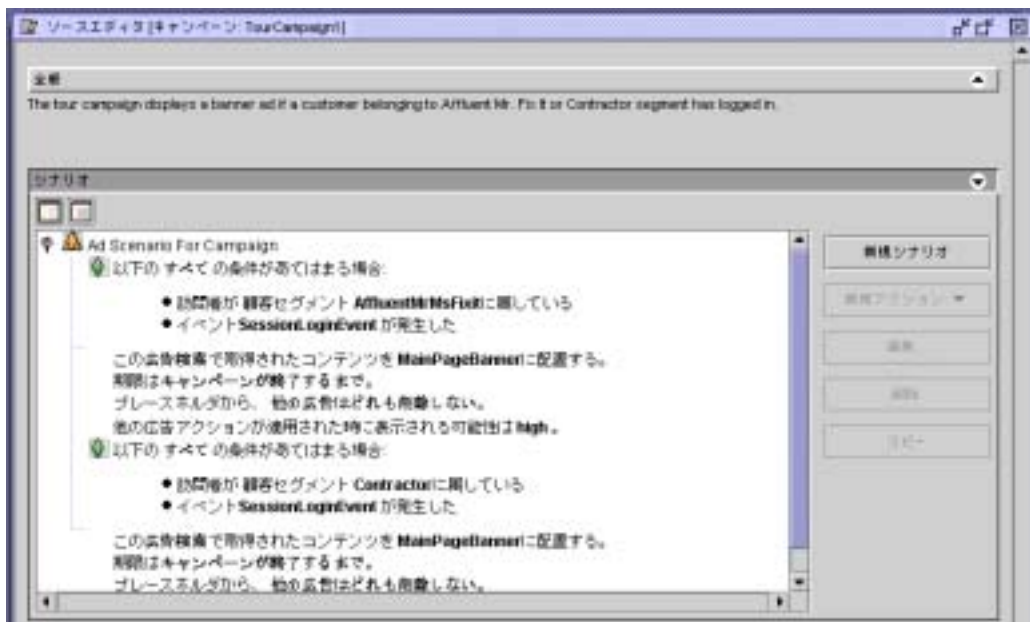
図 7-3 [ 目標の詳細 ] ウィンドウ



このキャンペーンでは、表示用に 2 つのバナー広告画像を指定しています。パワー ユーザまたはマーケティング担当者はこの画面で、キャンペーンの目標数値と特性、目標のスコープ、および目標タイプを設定できます。

9. [ 目標の詳細 ] ウィンドウで、[OK] または [ キャンセル ] をクリックします。TourCampaign1 の編集ウィンドウに戻り、[ 開始 / 停止 ] バーを再びクリックしてカテゴリの表示を最小化します。
10. [ シナリオ ] バーをクリックして、カテゴリの表示を展開します。[Ad Scenario for Campaign] エントリが表示されます。
11. シナリオ名の隣にある拡張ノード ボタンをクリックすると、図 7-4 に示すようにシナリオの詳細が表示されます。

図 7-4 Ad Scenario for Campaign] エントリの詳細表示



繰り返しになりますが、シナリオはキャンペーンの主要な基礎単位です。シナリオは1つまたは複数のシナリオ アクションで構成され、すべてのアクションはその発生条件（特定のイベントが発生した、顧客が特定の顧客セグメントに該当する、など）が満たされたときに発生します。画面に示されているように、このシナリオには2つのアクションが定義されています。シナリオは次の条件で開始します。

- Web サイトにログインした訪問者が顧客セグメント「Affluent Mr/Ms Fixit」に該当する
- または、Web サイトにログインした訪問者が顧客セグメント「Contractor」に該当する

3-14 ページの「newuser.jsp によるデモグラフィック (2/3)」で説明したように、テンプレート newuser.jsp は newdemographictemplate.inc ファイルをインクルードします。サイトの訪問者はこのファイルに含まれるフォーム フィールドを使って Web サイトに登録し、登録時に入力した情報に基づいていずれかのカテゴリに分類されます。図 7-5 は、newuser.jsp 上のデモグラフィック オプションの一部を示したものです。

図 7-5 newuser.jsp テンプレートのデモグラフィック オプション

性別 *	<input type="radio"/> ウィウ <input checked="" type="radio"/> 既婚 <input type="radio"/> 未婚 <input type="radio"/> 離婚
教育レベル *	<input type="radio"/> 高等学校卒 <input checked="" type="radio"/> 短大卒 <input type="radio"/> 大卒卒 <input type="radio"/> 大学院卒 <input type="radio"/> 専門学校卒
年収 *	<input checked="" type="radio"/> ¥ 7,500,000 から ¥ 9,999,999 <input type="radio"/> ¥ 3,500,000 から ¥ 4,999,999 <input type="radio"/> ¥ 0 から ¥ 3,499,999 <input type="radio"/> ¥ 5,000,000 から ¥ 7,499,999 <input type="radio"/> ¥ 10,000,000 から ¥ 1,249,999 <input type="radio"/> ¥ 12,500,000 以上
日曜大工の経験	<input checked="" type="radio"/> 上級者 <input type="radio"/> 不器用 <input type="radio"/> 普通 <input type="radio"/> プロ級

サンプルのキャンペーンでは、次の情報で 2 名の顧客がサイトに登録されています。

- Sue Carpenter: ユーザ名は `suecarpenter`、パスワードは `password`。Web サイトに登録した時点で、Sue のデモグラフィック特性には日曜大工の経験「プロ」(Professional) と年収「12,500,000 円以上」が記録されます。この情報に基づいて、Sue は E-Business Control Center で定義される顧客セグメント「Contractor」に分類されます。
- Bob Smith: ユーザ名は `bobsmith`、パスワードは `password`。Web サイトに登録した時点で、Bob のデモグラフィック特性には日曜大工の経験「上級者」(Do It Yourselfer) と年収「10,000,000 円から 12,499,999 円」が記録されます。この情報に基づいて、Bob は E-Business Control Center で定義される顧客セグメント「Affluent Mr/Ms Fixup」に分類されます。

実行中の Web アプリケーションのテンプレート `main.jsp` で、ユーザがログインする前に表示されるデフォルトのバナー画像は次のとおりです。

- `tourBanner1.gif` (「体験キャンペーン!」のテキストを含むトップバナー)

- tourSide01.gif (「今すぐ登録...」のテキストを含むサイド バナー)  
サンプル Web アプリケーションの初期画面を図 7-6 に示します。

図 7-6 Web アプリケーションの初期画面



tourBanner1.gif と tourSide01.gif はデフォルトの画像です。プレースホルダ JSP タグの内容は次のようにシンプルです。

```
<ph:placeholder name="<%= banner %>" />
```

最初のサンプル キャンペーンを体験し、Web サイトの訪問者に向けて表示される内容を確認するために、ブラウザを使って Web アプリケーションにアクセスします。

**注意：** まだサーバが稼動していない場合には、2-3 ページの「Commerce Server とテンプレートの起動」で説明されているとおりにしてサーバを起動します。

- [ スタート | プログラム | BEA WebLogic Platform 7.0 | WebLogic Portal 7.0 | Portal Examples | Commerce Templates | Start Commerce Templates]

または

- `http://<wl-host>:7501/wlcs` (<wl-host> にはサーバが稼働しているマシンの名前を指定)たとえば、次のように指定します。

`http://yourserver:7501/wlcs`

`main.jsp` のナビゲーション バー領域にある [ **ログイン** ] リンクをクリックします。

`login.jsp` テンプレートが表示されたら、次の情報を入力してログインします。

ユーザ名 : `suecarpenter`

パスワード : `password`

Sue のユーザ名とパスワードを正しく入力したら、`main.jsp` テンプレートに戻ったときに表示内容が変化しています。図 7-7 にサンプルの表示を示します。

図 7-7 ユーザ Sue Carpenter がログインしたときの main.jsp の表示



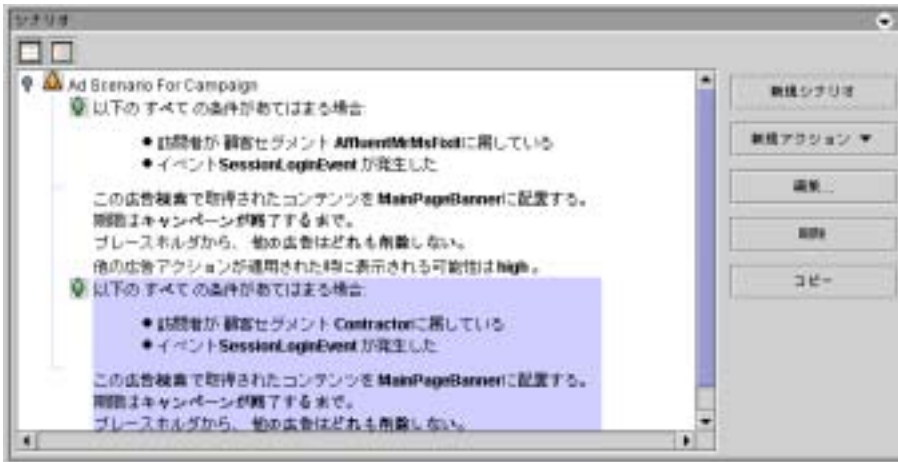
Sue が属する顧客セグメントは「Contractor」です。セグメントに基づいて選択された「Contractor's Journal」の広告がバナーとして使われています。画像ファイルの名前は `tourBannerJournal.gif` です。

E-Business Control Center に戻りましょう。

1. 図 7-8 に示すように、TourCampaign1 の編集ウィンドウの [シナリオ] セクションで、顧客セグメント「Contractor」を参照する 2 つ目のアクションをクリックします。



図 7-8 顧客セグメント「Contractor」を参照するアクション



2. アクションが強調表示された状態で、[ **編集 ...** ] をクリックします。[ 広告アクション ] ウィンドウで、[図 7-9](#) に示すように [ **広告検索** ] リンクをクリックします。

図 7-9 [ 広告アクション ] ウィンドウ上の [ 広告検索 ] リンクに注目



3. [ 広告検索 ] リンクをクリックします。[ 広告検索 ] ウィンドウの内容は図 7-10 のようになります。
4. ログオンを要求されたら、次のように入力します。

ユーザ名 : system

パスワード : weblogic

このパスワードは、WebLogic Server のインストール時に設定されます。正しいパスワードがわからない場合は、システム管理者に問い合わせてください。あなたを WebLogic Portal Administration Tools の Administrators グループに追加してもらおうよう、システム管理者に依頼することもできます。そうすれば、自分の通常のユーザ名とパスワードを使えます。

図 7-10 [広告検索] ウィンドウ



Tour Campaign 1 での定義どおり、顧客セグメント「Contractor」の顧客と、画像 `tourBannerJournal.gif` が関連付けられていることがわかります。必要であれば、[記述子] リストのアイテムを選択してプレビューを表示できます。

次に、Web アプリケーションで、Sue Carpenter をログアウトさせ、Bob Smith として再びログインします（ユーザ名は `bobsmith`、パスワードは `password`）。Bob は顧客セグメント「Affluent Mr/Ms Fixup」に属します。Bob Smith がログインしたときの `main.jsp` の表示は、[図 7-11](#) のようになります。

図7-11 ユーザ Bob Smith がログインしたときの main.jsp の表示



画面からわかるように、Bob Smith 向けに選択されたバナー広告には画像 `tourBannerFixup.gif` が使われています。このことは E-Business Control Center で、顧客セグメント「Affluent Mr/Ms Fixup」を参照するシナリオアクションからリンクをたどることによって確認できます。

ビジネスアナリストまたはマーケティング担当者が E-Business Control Center で新しいプレースホルダを作成した後で、企業内エンタープライズ開発者は広告の表示箇所として設定された位置に特殊なタグ（`<ph:placeholder>` JSP タグ）を作成する必要があります。たとえば、トップバナーに広告を表示するプレースホルダが作成された場合、企業内エンタープライズ開発者はトップバナーに `<ph:placeholder>` タグを追加する必要があります。

プレースホルダの JSP タグを含むページに顧客がアクセスすると、クエリを選択して実行し、ブラウザで広告を表示するために必要な HTML タグを生成するプロセスがタグで開始されます。

Web サイト上のページに企業内エンタープライズ開発者がこのタグを作成した後で、E-Business Control Center を使って、タグで実行可能なクエリを変更することができます。企業内エンタープライズ開発者がタグを修正する必要はありません。



---

# 索引

## A

- [Add to cart] リンク
  - アンカ 6-2
  - 変更 6-1, 6-5
  - 変更結果の確認 6-6

## B

- browse.jsp
  - UNIX の構文 4-6
  - 特徴 4-5, 4-7

## C

- checkout.jsp
  - UNIX の構文 3-29
  - 特徴 3-28
- confirmorder.jsp
  - UNIX の構文 3-30
  - 特徴 3-30

## D

- details.jsp
  - [Add to cart] リンク 6-6
  - UNIX の構文 4-8
  - 特徴 4-8

## J

- JavaServer Pages (JSP)
  - 概要 1-1
  - 商品の検索と選択 3-3
  - 新規ユーザ ロードマップ 3-2
  - 登録ユーザ ロードマップ 4-2
- JSP テンプレート
  - 新規ユーザ ロードマップ 3-2
  - 制限事項 3-16
  - テンプレート内にインクルード 3-25
  - addaddress.jsp 4-14

- browse.jsp 4-5
- checkout.jsp 3-28, 4-17
- confirmorder.jsp 3-29, 4-19
- login.jsp 3-10
- main.jsp 3-6
- newuser.jsp 3-12
- orderhistory.jsp 5-8
- orderstatus.jsp 5-10
- payment.jsp 3-22, 3-27, 4-16
- paymentnewcc.jsp 3-23
- profilenewcc.jsp 5-5
- searchresults.jsp 3-6, 3-7
- selectaddress.jsp 4-15
- shipping.jsp 4-14
- shoppingcart.jsp 3-8, 4-9
- viewprofile.jsp 5-3
- インクルード ファイル 3-25
- カスタマイズ 6-1
- 登録ユーザ ロードマップ 4-2
- ユーザ プロファイル ロードマップ 5-2

## L

- login.jsp
  - UNIX の構文 3-11
  - 特徴 3-11

## M

- main.jsp
  - [Add to cart] リンクの変更 6-5
  - Java パッケージのインポート 2-7
  - UNIX の構文 2-5
  - 商品の検索 3-6
  - タグ ライブラリのインポート 2-7
  - 特徴 2-5, 2-8
  - 認証済ユーザ用 4-13

## N

newuser.jsp  
UNIX の構文 3-12  
特徴 3-12, 3-15

## O

Oracle データベース  
WLS のインストール 1-2  
orderhistory.jsp  
UNIX の構文 5-9  
特徴 5-9  
orderstatus.jsp  
UNIX の構文 5-10  
特徴 5-10

## P

payment.jsp  
UNIX の構文 3-23  
特徴 3-23, 3-27  
paymentnewcc.jsp  
UNIX の構文 3-24  
特徴 3-23  
Pipeline  
コンポーネント 3-9  
コンポーネントの定義 3-9  
セッション 3-9  
セッションの終了 5-11  
セッションの定義 3-9  
入力プロセッサ 3-13  
PointBase データベース  
WLS のインストール 1-2  
PORTAL\_HOME, 説明 1-2  
profilenewcc.jsp  
UNIX の構文 5-6  
特徴 5-6

## S

searchresults.jsp  
UNIX の構文 3-7  
特徴 3-7  
selectaddress.jsp  
Java スクリプトレット 3-21  
UNIX の構文 3-20  
特徴 3-20  
shipping.jsp

UNIX の構文 3-19  
特徴 3-19  
shoppingcart.jsp  
UNIX の構文 3-8  
特徴 3-8  
successfullogin.jsp  
UNIX の構文 3-16  
特徴 3-16

## V

viewprofile.jsp  
UNIX の構文 5-4  
特徴 5-4

## W

Webflow 3-9  
定義 3-9  
Webflow および Pipeline エディタ 6-6  
WebLogic Portal  
プロダクト カタログ 2-9  
WLCS\_ORDER データベース テーブル 5-9

## カ

開始, ツアー 2-3

## キ

キャンペーン  
サンプル ツアー 7-1

## セ

セキュリティ モデル 4-11

## ツ

ツアーの開始 2-3

## テ

テンプレート インクルード ファイル 3-25

## ト

登録ユーザ 4-10



---

## ニ

### 入力プロセッサ

定義 3-13

## ホ

### 方法

JSP テンプレートのカスタマイズ 6-1

新しい届け先住所の入力 4-14

カタログのブラウズ 4-5

クレジットカードの選択 3-27, 4-16

クレジットカードの追加 5-5

支払に関する情報の入力 3-22

出荷オプションの選択 3-18, 4-14

商品の検索 3-6

商品の購入 3-28, 4-17

商品の選択 3-6, 4-5

ショッピング カート内容の表示 3-8

ショッピング カートのレビュー 4-9

新規ユーザの追加 3-12

新規ユーザ プロファイルの作成 3-10

チェックアウト 4-9

注文確認のレビュー 3-29, 4-19

注文のステータスの表示 5-7

注文の履歴の表示 5-7

ユーザ プロファイルのレビュー 5-3

ユーザ プロファイルへのアクセス 5-3

ログアウト 5-11

### ホームページ

main.jsp の特徴 2-5

認証済ユーザ用 4-13

### ボタン

Add address 4-14

Add card 3-23

Add to cart 3-8, 4-9, 6-1

Check out 3-10, 4-10, 3-18

Complete purchase 3-29

Create 3-12

Find 3-6

Use 3-22, 4-16

## ユ

### ユーザ管理 3-27

