

Oracle iStore 11*i*

Concepts and Procedures

Release 11*i*

August 2000

Part No. A86123-01

ORACLE®

Copyright © 2000, Oracle Corporation. All rights reserved.

The Programs (which include both the software and documentation) contain proprietary information of Oracle Corporation; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent, and other intellectual and industrial property laws. Reverse engineering, disassembly, or decompilation of the Programs is prohibited.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. Oracle Corporation does not warrant that this document is error free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of Oracle Corporation.

If the Programs are delivered to the U.S. Government or anyone licensing or using the programs on behalf of the U.S. Government, the following notice is applicable:

Restricted Rights Notice Programs delivered subject to the DOD FAR Supplement are "commercial computer software" and use, duplication, and disclosure of the Programs, including documentation, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement. Otherwise, Programs delivered subject to the Federal Acquisition Regulations are "restricted computer software" and use, duplication, and disclosure of the Programs shall be subject to the restrictions in FAR 52.227-19, Commercial Computer Software - Restricted Rights (June, 1987). Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065.

The Programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be the licensee's responsibility to take all appropriate fail-safe, backup, redundancy, and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and Oracle Corporation disclaims liability for any damages caused by such use of the Programs.

Oracle is a registered trademark, and Oracle iStore 11i is a trademark or registered trademark of Oracle Corporation. Other names may be trademarks of their respective owners.

Contents

Send Us Your Comments	vii
Preface.....	ix
Intended Audience	ix
Structure.....	x
Related Documents.....	x
 Understanding Oracle iStore 11i	
Overview of Oracle iStore 11i	1
Main Components of Oracle iStore 11i Functionality	2
Oracle iStore 11i Business Objects	2
Product Catalog Key Points and Concepts	3
Hierarchy Key Points and Concepts	4
Multimedia and Templates Key Points and Concepts	5
Customer User Interface Key Points and Concepts.....	5
Logical Templates	6
Object and Template Relationships.....	6
Section Templates	6
Product Templates.....	6
Relationship of Oracle iStore 11i to Other Oracle Applications	7
Search Functionality Key Points and Concepts.....	8
Business Flows and Templates for Search	9
Business Flows and Templates for Shopping Cart	10
Business Flows and Templates for Checkout	12

Business Flows and Templates for Account	14
Business Flows and Templates for Customer Information.....	16
Business Flows and Templates for Carts and Lists.....	18
Business Flows and Templates for Orders and Invoices	19
How the Store Displays Multimedia.....	20
Multimedia Called Directly	20
Multimedia Called Indirectly	20
How the Store Displays Templates	21
Templates Called Directly	21
Templates Called Indirectly	22
Understanding JSPs	22
Standard Includes	25
JSP Architecture	25
PageCompileServlet	26
The Request Object	26

Administering Oracle iStore 11i

Setting Up the Store Business Flow	27
Creating Your Store	28
Setting Up the Product Catalog	29
Creating Specialty Stores	32
Cataloging Multimedia	34
Defining Multimedia Components	36
Defining Display Styles.....	37
Creating Templates.....	38
Building the Product Catalog.....	39
Verifying Templates	41
Mapping Templates in the Merchant UI	41
Assigning Templates	42
Assigning a Template to a Category	43
Viewing Categories Assigned to a Template.....	43
Viewing Template Details Assigned to a Category	43
Cataloging Templates.....	44
Modifying Templates	46
Defining Hierarchies	47

Defining Products	51
Defining Category Mappings	54
Defining Product Relationship Rules	55
Setting Up Product Searches	59
Setting Up Search Profile Options.....	62
Customizing Search.....	63
Adding Stopwords to Searches.....	64
Assigning Templates, Multimedia, and Styles to Categories.....	65
Testing The Store	66

A IBE Template Mappings

IBE Template Mappings	A-1
-----------------------------	-----

B JSP Customizing Standards

JSP Standards (Customer Facing)	B-1
Transactions + Database Connections	B-2
Exception Handling	B-4
API Standards	B-4
Passing Values Across Templates	B-5
Passing Parameters/ Cookies	B-6
Changing API Documentation	B-7
Modifying the User Interface.....	B-7
Changing the Text in Bins	B-7
Changing the Layout of Bins	B-7
Using Forms/JavaBeans with JSPs	B-9
Using JavaBeans with JSPs	B-9
Adding Error Pages.....	B-10
Handling Runtime Exceptions.....	B-10

Glossary

Index

Send Us Your Comments

Oracle iStore 11/ Concepts and Procedures, Release 11/

Part No. A86123-01

Oracle Corporation welcomes your comments and suggestions on the quality and usefulness of this document. Your input is an important part of the information used for revision.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where?
- Are the examples correct? Do you need more examples?
- What features did you like most?

If you find any errors or have any other suggestions for improvement, please indicate the document title and part number, and the chapter, section, and page number (if available). You can send comments to us via the postal service.

Oracle Corporation
CRM Content Development Manager
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

If you would like a reply, please give your name, address, telephone number, and (optionally) electronic mail address.

If you have problems with the software, please contact your local Oracle Support Services.

Preface

Welcome to the Oracle Customer Relationship Management, Release 11i, suite of applications.

This Concepts and Procedures provides information and instructions to help you work effectively with Oracle iStore 11i, and primarily covers functions in the admin console.

This preface explains how Concepts and Procedures is organized and introduces other sources of information that can help you.

For instructions on post-installation set up of Oracle iStore 11i and creating a specialty store, see Oracle iStore 11i Implementation Guide.

Intended Audience

This guide is aimed at the following users:

- Web Store Administrators
- System Administrators (SA), Database Administrators (DBA), and others with similar responsibility

This guide assumes you have the following prerequisites:

- Understanding of the company business processes
- Knowledge of products and services as defined by your marketing policies
- Basic understanding of Oracle and Developer / 2000.

Structure

This manual contains the following chapters:

“Understanding Oracle iStore 11i” provides overviews of the application and its components, explanations of key concepts, features, and functions, as well as the application’s relationships to other Oracle or third-party applications.

“Administering Oracle iStore 11i” provides task-based procedures for configuring the application to your business specifications.

Related Documents

List related documents using variables for the book names. Do not build the named character format into the variable. Always apply the character format after you insert the variable. Although including character formats in the variables works well for FrameMaker hardcopy output, this does not work when filtering to HTML.

For more information, see the following manuals:

- *Oracle iStore and iMarketing 11i Implementation Guide* (Part No. A83719-01)
- *Implementing Oracle CRM: ERP Functional Checklist*, available at <http://support.oracle.com>
- *Implementing Oracle CRM: Foundation Functional Checklist*, available at <http://support.oracle.com>

Understanding Oracle iStore 11i

This topic group provides overviews of Oracle iStore 11i and its components, explanations of key concepts, features, and functions, as well as Oracle iStore 11i's relationships to other Oracle or third-party applications.

Overview of Oracle iStore 11i

Oracle iStore 11i is an electronic commerce application that businesses use to sell products directly to customers over the world wide web. Businesses can customize Oracle iStore 11i to set up an online store and integrate the front end order capturing system with the order fulfillment system. Using Oracle iStore 11i, a business can create, manage, and personalize a powerful web-based store in a secure and scalable environment.

Create one or more specialty stores using your own configuration of the following components:

- Templates that the application provides or that you can create
- Media components that the application provides or that you can define
- Multimedia, including text, from any source
- Product information you enter using the application
- Product information supplied in Oracle Inventory
- Logical template names that you map to template files based on specialty store and customer's language
- Logical multimedia names that you map to multimedia files based on product and customer's language
- Registration requirements including approval process for business customers

The integration with other Oracle applications provides powerful functionality for your store that you can turn into an exceptional customer experience.

You can use the templates provided with Oracle iStore 11i to quickly create a fully functioning web store. Oracle iStore 11i provides functional templates, that control the flow of information through the store, as well as display templates.

Note: After a customer places an order in the store, the order cannot be cancelled within Oracle iStore 11i. The customer must contact a customer server representative to cancel the order.

Main Components of Oracle iStore 11i Functionality

Oracle iStore 11i functionality consists of the following components:

- **Business Objects**—The schema for representing and storing customers, orders, product catalog and presentation elements. These entities are persistent, shared across all Oracle applications, and manipulated by Java APIs provided within Oracle iStore 11i's runtime services.
- **Customer Facing UI**—Referred to as templates in Oracle iStore 11i, essentially the UI consists of Java Server Pages (JSP). The application receives an HTTP request, executing the JSP. Accordingly, the JSPs use Oracle iStore 11i's runtime services to reply in HTML.
- **Runtime Services and APIs**—The coupling of certain common services available within all e-commerce applications (Oracle Foundation) and Java based APIs (includes some PL/SQL APIs). This combination queries Oracle iStore 11i's persistent storage of objects and relationships and enables update operations.
- **Store Manager**—The component that enables business users to manage the Oracle iStore 11i through the set up of data without programming knowledge.

Oracle iStore 11i Business Objects

Oracle iStore 11i shares and leverages the Oracle Applications schema for modeling and storing customer names, products, and orders. In addition, Oracle iStore 11i also provides the following additional business objects to enhance presentation capability:

- **Customer Master**—The manager and storage unit for customer information, this business object model is consistent across all Oracle applications.

- **Multimedia**—Oracle iStore 11i organizes and stores the body of media files used for presentation on the storefront. The multimedia object also supports the organization of foreign language and speciality store media files. You can sort and name these media files using a logical name, which the application calls according to the Oracle iStore 11i user's language preference and chosen speciality store.
- **Order System**—The manager and storage unit for customer order and their statuses, this business object model is consistent across all Oracle applications.
- **Product Catalog**—The product catalog consists of the following components:
 - Item Master in the Oracle Inventory
 - Hierarchy created and managed within Oracle iStore 11i
 - Relationships and rules between differing items

The product catalog is complemented by a set of dependency rules created in Oracle Configurator that provides configuration and validation abilities.

- **Speciality Stores**—This feature of Oracle iStore 11i allows multiple store fronts operated and managed in one central instance. The speciality stores may differ in appearance, products, or processing flow.
- **Templates**—Essentially the JSP pages, templates are similar to multimedia files in that Oracle iStore 11i supports the mapping of multiple JSP files to one logically named template.
- **User Information**—The manager and storage unit for user roles and privileges, this business object model is consistent across all Oracle applications.

Product Catalog Key Points and Concepts

The product catalog in Oracle iStore 11i is created against one inventory organization of the ERP application which you need to identify during the setup operation. You typically use the Master Inventory Organization. Business users create items using Oracle Inventory form screens, and choose only items marked as WEB_PUBLISHABLE for use with Oracle iStore 11i. After item creation, the business users then use the store manager to perform the following tasks:

- **Provide or update the short and long descriptions.** The short description serves as the name of the product, and the long description provides detail about the product for the customer. You may provide the descriptions while creating the items.

- Assign the product to the hierarchy and relative order in which it should appear, and effective dates.
- Describe the product display specifications for different locations on the customer UI.
- Provide additional content in the form of media files. Users have the option of creating unlimited multimedia components at setup or additionally using the store manager. The user can provide multimedia components in varying sizes, for example, “small image” and “large image.”

You also can assign default values for each of the media components at the category level. These values are picked up at the product level as follows:

When setting up Oracle iStore 11i, the user indicates the primary category set. Once the primary category is identified, if an item belongs to a category in that category set, then it is associated with the category. In this case, the default ID is picked from the settings at category level. However, if an item does not belong to any category in the category set, then defaults are picked up at the product store level. Future functionality for Oracle iStore 11i may support the business user’s capability to indicate the primary category for a product.

- Define or review the related items for different relationships.
- Restrict the product to appear in only certain specialty stores.
- Publish the product.

Hierarchy Key Points and Concepts

Oracle iStore 11i hierarchy facilitates customers’ browsing. The Oracle iStore 11i creator has the option to arrange their product in a hierarchy of sections and children. Each section has a translatable name, a section identifier (section code), and associated content descriptions.

You can classify sections as navigational or your own definable types. For example, section type in Oracle iStore 11i has been seeded with the FEATURED value. The navigational sections serve as the tree that the customers navigate when browsing. The non-navigational sections organize products for highlighting at any level. Each section has two parameters: the template that displays that section, and the style that displays products in that section.

If the creator wants to indicate if a particular section should appear in a specialty store, then they utilize the root section id. Normally, customers in a specialty store

see only the store and products starting from the root id of the section, unless a section has been excluded from the specialty store.

To provide richer information to customers or facilitate the cross-sell or up-sell functionality, the creator can specify different types of relationships between products. Oracle iStore 11i specialty store creators can determine their own relationships (substitute, cross-sell, up-sell), and then specify the related items for each relationship. Creators can also establish rules relating all items in a section to all items in another section or category.

Certain programmatic advanced features exist for the Oracle iStore 11i creator. Instead of manually assigning products to a hierarchy, developers may write an auto-placement rule (WHEREBY clause of a SQL query) to automatically assign items to sections. Whenever an item is created or changes in the underlying inventory tables, Oracle iStore 11i executes the placement rules that indicate the location of the added item. Oracle iStore 11i also supports the addition of a SQL query that determines the relationship of an item to a set of related items.

Multimedia and Templates Key Points and Concepts

All multimedia and template files possess the following features:

- Name
- Programmatic access name
- Set of source files for foreign language and specialty stores
- Description and keywords

Template developers can access the multimedia file directly by referencing its programmatic access name, or by referring to the multimedia file associated with a product or section. In the second scenario, a template developer would ask for the particular multimedia component of a given product or section, for example the “Section Logo” multimedia of the “Book” section.

For detailed information, see [How the Store Displays Multimedia](#).

Customer User Interface Key Points and Concepts

Oracle iStore 11i ships with templates that render the customer UI and process flow. Templates essentially are JSPs that the template developer can use to call JavaBeans to get dynamic data and present that returned data by embedding it within HTML tags. Oracle iStore 11i's out of the box templates generate only HTML, but output

can be generated into other formats. In addition to calling Oracle iStore 11i's APIs, template developers can call any other components to get data for inclusion in the final HTML page that is returned to the browser.

Logical Templates

Oracle iStore 11i keeps a mapping from logical templates to the exact JSP page for that template based on language and specialty store. While writing the JSP pages, the template developer does not have to write multiple versions of the JSP to compensate for the usage of that JSP page in differing languages or speciality stores.

Note: Referencing a JSP page directly without a logical name makes maintaining and editing templates more difficult and susceptible to error.

For a list of seeded template filenames, with programmatic access names and descriptions, see IBE Template Mappings. For detailed information, see [How the Store Displays Templates](#).

Object and Template Relationships

When setting up Oracle iStore 11i, you can indicate template affiliations for presenting sections as well as displaying products.

Section Templates

Display a section by using the section's own template. If different sections require different layouts, then indicate the template for use at the section level. Also at the section level, specify a display style for the products for displaying products in that section. Template developers do not need to hard code the display style in the each template for the section. This feature and method allow template developers to focus on presentation rather than coding logic into the JSP pages.

Product Templates

Typically products are displayed differently depending on their location and type. Accordingly, Oracle iStore 11i provides functionality that supports a merchant definable set of display styles for store products. For each display style, the store

creator can indicate which template to use to display a particular product for that display style.

Oracle iStore 11i provides a default mechanism similar to the functionality surrounding multimedia components. For each display style, specify a default at the category level. At the product level, indicate use of the default level value or a different template. For example, “Product Details” is one of the display styles seeded in Oracle iStore 11i. A television and a computer may require different templates to display the product in a detailed page—they may refer to different flex fields or have a different look and feel. For this example the template developer only needs to indicate the display style for a particular product, and the application determines which appropriate template to use. Alternatively, the template developer would need to calculate the product type and to code the presentation, resulting in too much logic for the page and intensifying the modification effort or incorporation of the product category.

Relationship of Oracle iStore 11i to Other Oracle Applications

Oracle iStore 11i depends upon the following modules:

- Oracle Order Capture to capture the customer’s order from the shopping cart
- Oracle Inventory to provide product information
- Oracle Receivables to provide customer and invoice information and to register customers
- Oracle Order Management to process the order and provide post-order views
- Integration with other Oracle products extends the functionality of your store.

Oracle iStore 11i integrates with the following applications:

- Oracle iPayment to accept customer’s credit card
- Oracle Alert to send e-mail confirmations to customers
- Oracle iMarketing to define promotions and discounts
- Oracle Configurator to enable customer configured products and provide guided selling as well as to perform some of the validations of the shopping cart
- Oracle Material Requirements Planning to provide product availability information

- Oracle iSupport to provide return authorizations and knowledge base integration
- Oracle Telephony Manager to process call-me-back requests
- Oracle CRM Business Intelligence to assess the performance of the store

Search Functionality Key Points and Concepts

The Oracle iStore 11i application has expanded search functionality. The new functionality for search relates both to the Oracle iStore 11i user and merchant.

The Oracle iStore 11i user has the following search capabilities:

- Quick search by category or across all store products
- Quick search by entering keywords
- Advanced search by category or across all store products via the following options:
 - Keyword search
 - Boolean searching using **and/or** qualifiers
 - Excluded keyword search

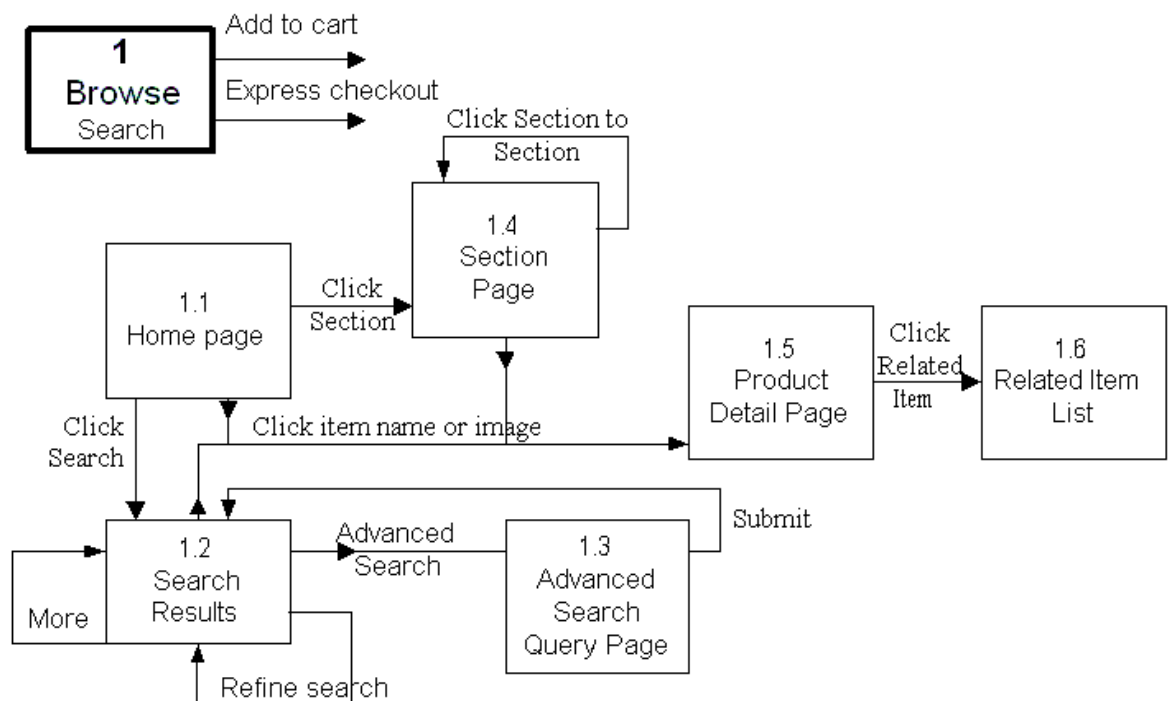
The Oracle iStore 11i merchant has the following configurable options for search functionality:

- Context keyword default—Set up the system so that the Quick search keyword is entered by default. With this functionality the user does not need to enter a contextual value; the application populates the field according to the defaults by category. The search engine uses the following user information to return the products that match the search criteria:
 - Name of the product
 - Different descriptions of the product
 - Attributes of the product
- Product display information—Customize the template to decide which fields of product information to display. By default the product name, id, key flex fields, and pricing information are displayed.
- Search result display number—Limit the number of results that display for a search. You determine the number of search results in total and the number to display per page.

- Search result prioritization—Divide the search results by priority, top matches versus the complete set.

Business Flows and Templates for Search

The customer enters the store at the home page and searches for information or browses through pages that display section and product information. The following diagram shows the required flows your templates must follow in this section of the store.



Customers enter the store at the home page (1.1).

Customers conduct a simple search (1.2) to find items of interest.

Customers can also conduct an advanced search (1.3). Add to cart and express checkout are not available on this page.

Customers view groupings of products called sections (1.4). In the sample store, one template is used to display all sections.

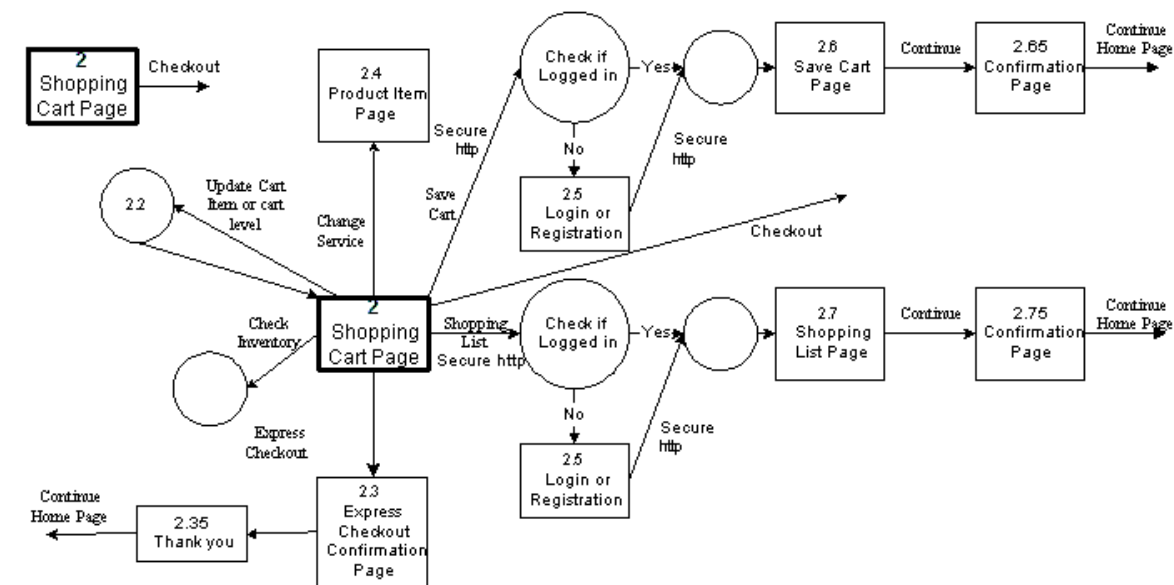
Customers review detailed product information (1.5). Any page that displays a product also links to the product detail page. In the sample store, one display style is used for all product detail pages.

While viewing a product, customers can choose to look at products that relate to the product being viewed (1.6).

Customers can add to cart and perform the express checkout from any page.

Business Flows and Templates for Shopping Cart

While browsing the store customers add products to their shopping carts, at which time the shopping cart page appears. Customers can review their selected items and perform other functions such as saving the cart or saving it as a shopping list. The following diagram shows the required flows your templates must follow in this section of the store.



Customers can save the shopping cart (2.6) for future use. Login or registration (2.5) is required before the cart can be saved, and the store changes to secure mode.

Customers receive confirmation (2.65) when the cart is saved. Only one shopping cart can be active at any time. Items cannot be combined from another shopping cart. Customers can then choose to continue shopping and are returned to the home page (1.1).

Customers can also choose to save the shopping cart as a shopping list (2.7). Shopping lists can be combined into a single order. Login and registration are required (2.5) before saving a shopping list, and the store changes to secure mode. Customers receive confirmation (2.75) when the shopping list is saved. Customers can then choose to continue shopping and are returned to the home page (1.1).

Customers can click an item description to view detailed item information on the product item page (2.4). They then return to the shopping cart page.

Customers can click Customize to go to the configuration page (2.8) for products that can be customized.

Customers can choose to change or upgrade service and go to the product item page (2.4).

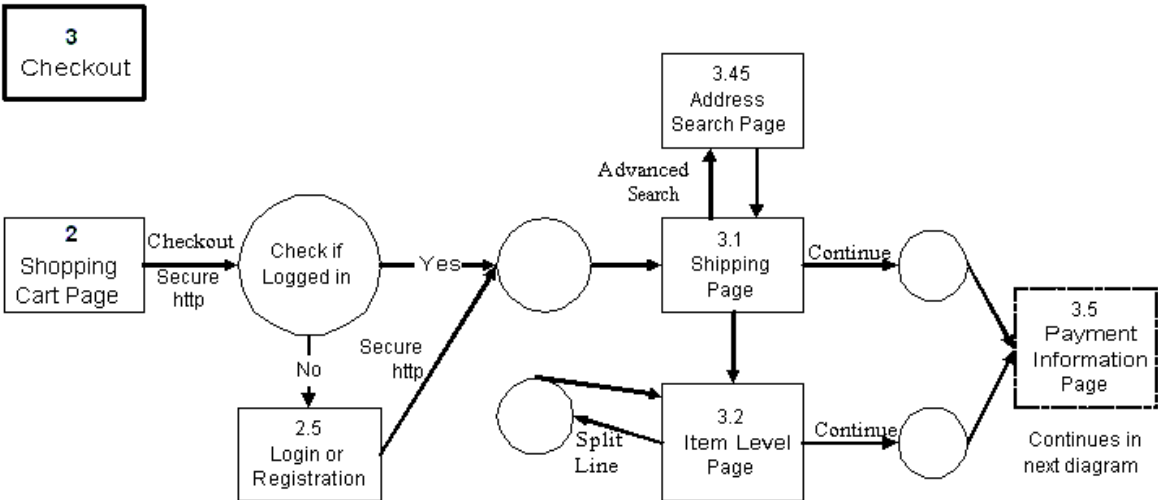
Customers can choose to remove an item, recalculate, or make other changes to the cart. The update is processed (2.2).

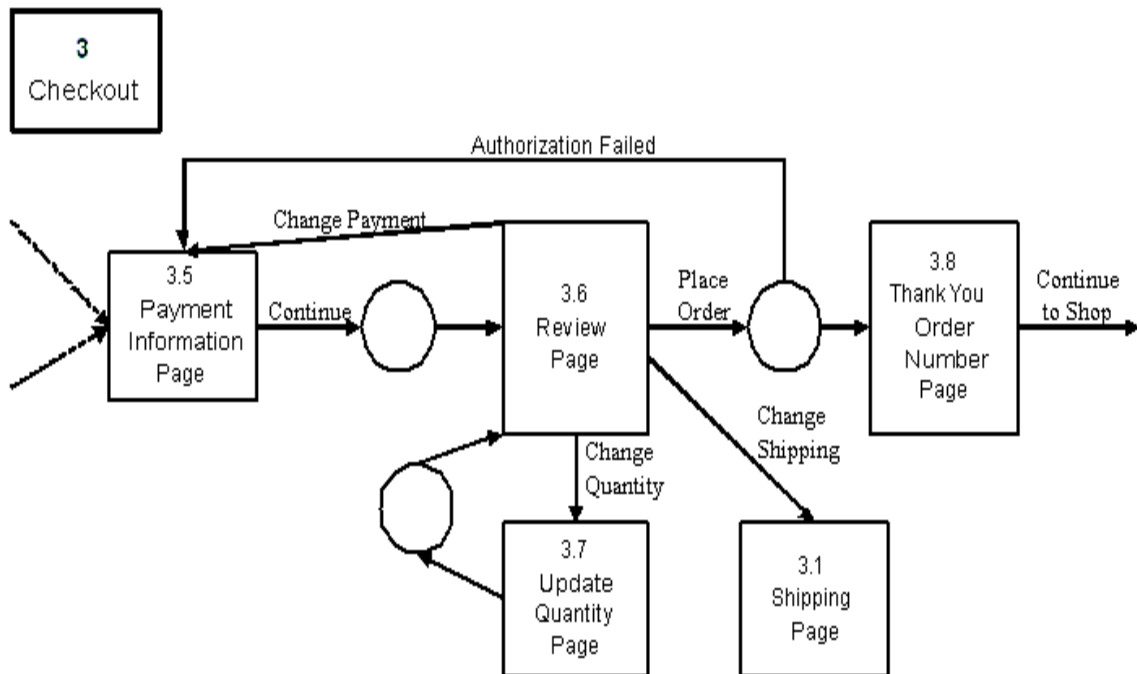
Customers can check inventory for available to promise (2.1) information. They then return to the shopping cart page.

Customers can either go to checkout (3) or perform an express checkout. They receive confirmation of the express checkout (2.3) and a thank-you page (2.35). Then return to the home page.

Business Flows and Templates for Checkout

Customers ready to place their orders go to the checkout area of the store where they enter shipping and payment information. The following diagram shows the required flows your templates must follow in this section of the store.





When a customer chooses to check out, Oracle iStore 11i changes to secure mode and checks to see if the customer is logged in. If not, then the customer goes to log in or register (2.5).

The shipping page appears (3.1). The customer can choose an existing address or add a new address and choose the shipping method and shipping instructions.

Customers can choose to set shipping at the item level (3.2). They can select an address, choose the search method (3.4), or add an address to their address books (3.3).

Continue takes customers to the payment information page (3.5).

Customers can choose to attach a file for one or more items (3.9). Continue returns the customer to the payment information page.

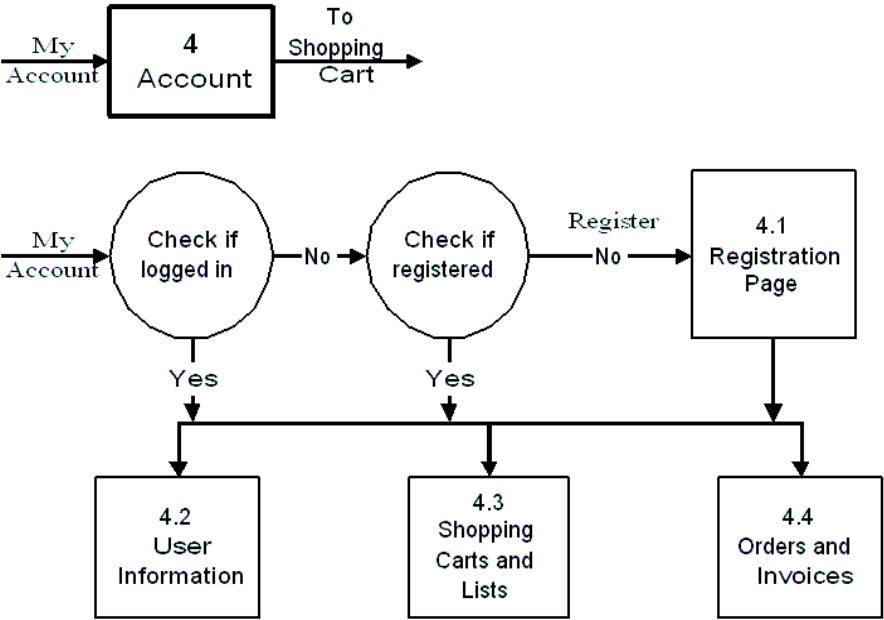
Continue starts the payment process and takes the customers to the review order details page (3.6).

From the review page, customers can update quantities (3.7), change payment information (3.5), and change shipping information (3.1).

Customers click Place Order on the review order details page to process the order and go to an order thank-you page (3.8). Customers can then choose to continue to shop and return to the home page.

Business Flows and Templates for Account

Customers can click My Account from any page to update their user information, check on orders and invoices, work with shopping carts and shopping lists, or directly enter an item to purchase. The following diagram shows the required flows your templates must follow in this section of the store.



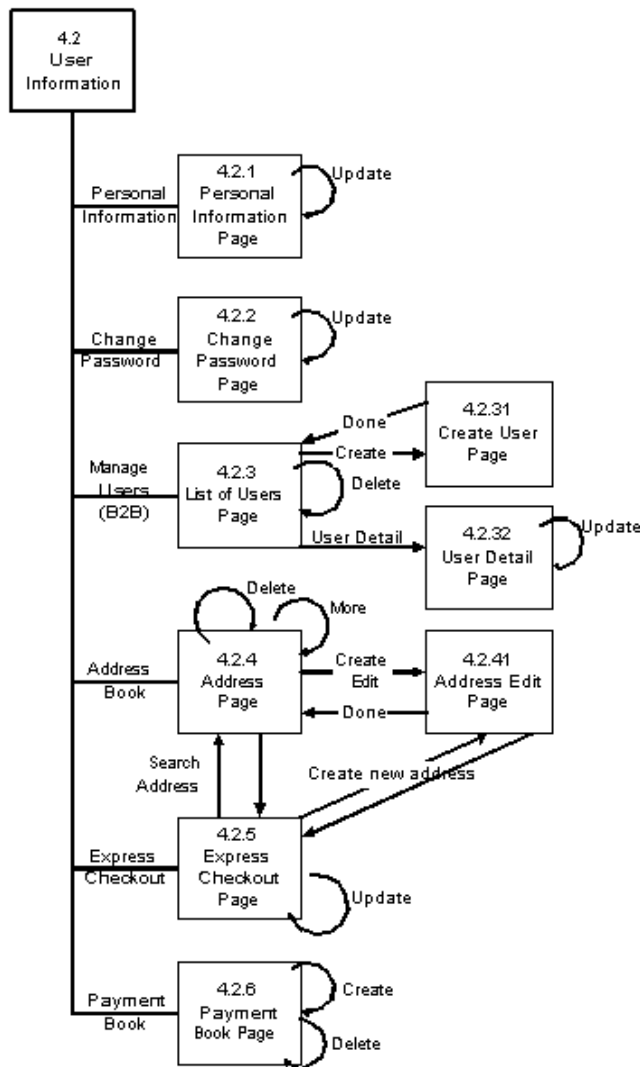
Clicking My Account takes customers to user information (4.2), the personal information page. See [Business Flows and Templates for Customer Information](#) to view the expanded diagram.

Customers choose Carts and Lists (4.3) to use their shopping lists or saved carts. Customers are first taken to the saved carts page. See [Business Flows and Templates for Carts and Lists](#) to view the expanded diagram.

Customers choose Orders and Invoices (4.4) to go to the order status page, and from there view invoice and payment information. See [Business Flows and Templates for Orders and Invoices](#) to view the expanded diagram.

Business Flows and Templates for Customer Information

Customers manage their own personal, address, and payment information. The following diagram shows the required flows your templates must follow in this section of the store.



When customers click My Account, the personal information page (4.2.1) appears.

Customers click Change Password to access the change password page (4.2.2). The password must be at least six characters long.

Customers click Addresses to review their billing and shipping addresses (4.2.4).

They click Edit or Add Address to access the address edit page (4.2.41).

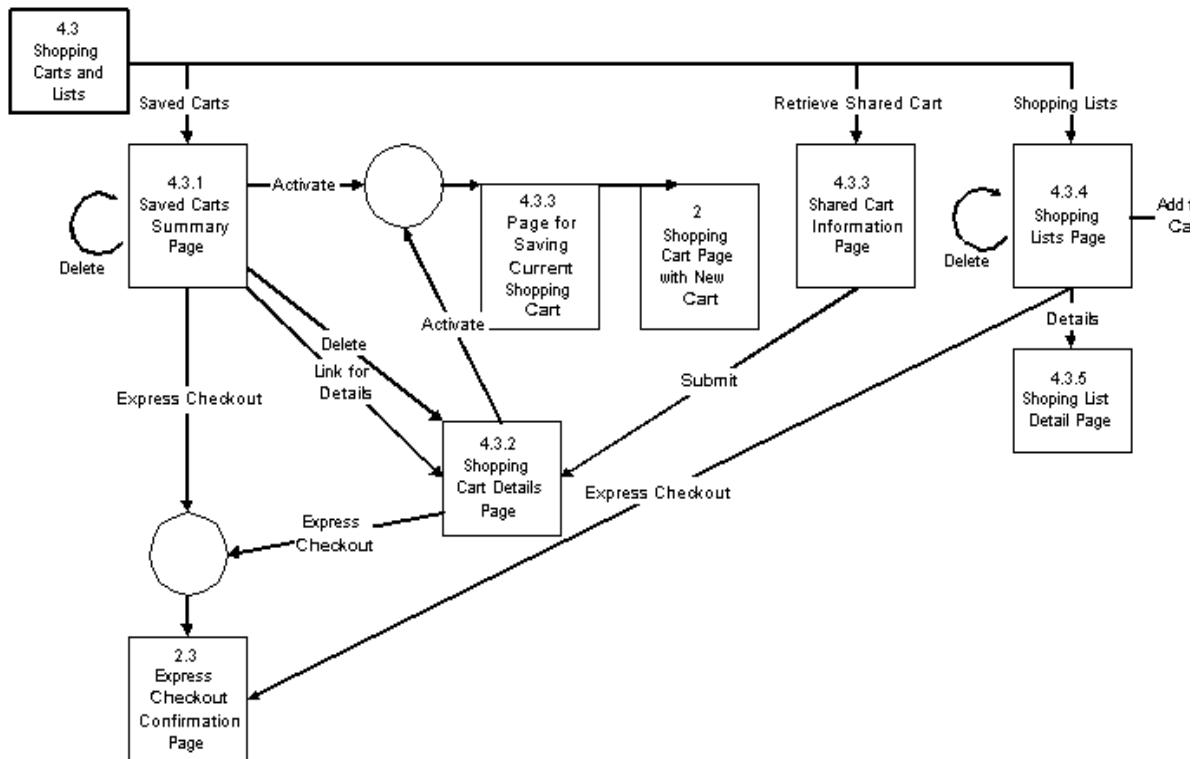
Customers click Payment Books to change payment methods or add to their payment options (4.2.6).

Customers click Express Checkout Preferences (4.2.5) to edit their address (4.2.41), payment, and shipping information to use for express checkouts.

A business customer can set up individual users to be able to use your store. They click User Management which takes them to a list of their users (4.2.3). They can then add new users (4.2.31) and edit the details for a user (4.2.32).

Business Flows and Templates for Carts and Lists

Customers use the Carts and Lists section of the store to share carts with others, retrieve saved carts, and use shopping lists. The following diagram shows the required flows your templates must follow in this section of the store.



Customers choose Carts and Lists and the saved carts summary page (4.3.1) appears.

Customers can activate a saved cart, in which case they are asked to save their currently active cart (4.3.3) and are taken to the active shopping cart page (2).

Customers can also click a shopping cart link on the summary page to view the detail information (4.3.2). They can also activate the cart from the detail page.

Customers can choose a cart and click Express Checkout, which takes them to the express checkout confirmation page (2.3).

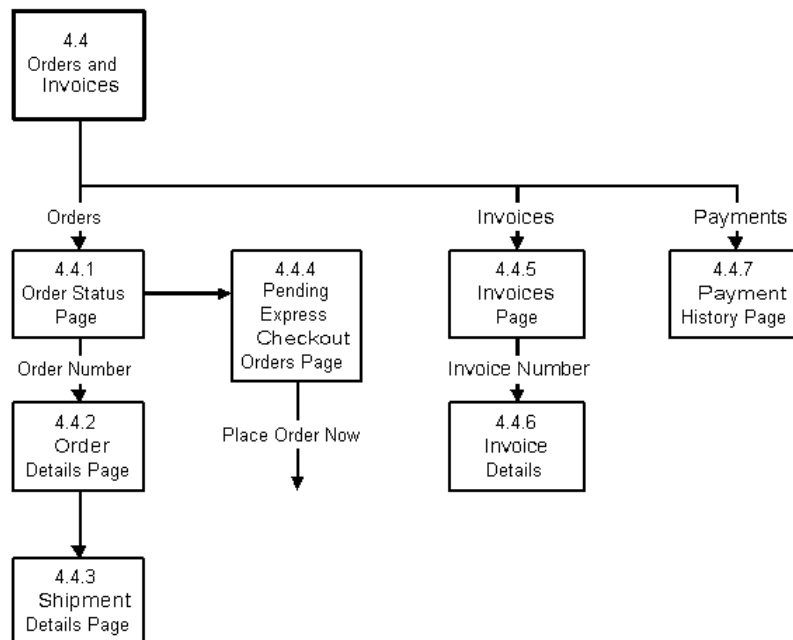
A shopping cart can be shared by one customer with another customer. The receiving customer clicks Retrieve Shared Cart and the retrieve cart information page appears (4.3.3). The customer must enter the cart name and password. The password must be at least six characters long.

Customers choose Shopping Lists to view existing shopping lists (4.3.4).

From the shopping list customers can view the details (4.3.5) of a particular shopping list. One or more shopping lists can be added to the active shopping cart or sent directly to express checkout to become an order.

Business Flows and Templates for Orders and Invoices

Customers can review their orders, invoices, and shipping status. The following diagram shows the required flows your templates must follow in this section of the store.



Customers choose Orders and Invoices and the order status page (4.4.1) displays status information for their outstanding orders.

Customers click an order number to view the order details (4.4.2).

Customers click shipment information to view shipment details (4.4.3).

Customers choose Pending Express Checkout Orders to view a list of express checkout orders that they have not yet placed (4.4.4). They can choose to place a selected order.

Customers choose Invoices to view a list of their invoices (4.4.5) with payment and order information.

They can click an invoice number anywhere to view the details for an invoice (4.4.6).

Customers choose Payments to view a history of all payments (4.4.7) and how payments have been applied to invoices.

How the Store Displays Multimedia

Multimedia consist of files used to present content on a web page to your customer, such as graphics, text, audio, and video. Your web page can call for multimedia to display in two ways:

- Directly, by using the multimedia name
- Indirectly, by using a multimedia component

Multimedia Called Directly

You assign a multimedia name and a programmatic access name to a multimedia object using the Multimedia tab. You then assign one or more physical media files to combinations of specialty stores and languages. The result is that when the customer looks at your page, Oracle iStore 11i looks at the customer's language and displays the files that are assigned to the customer's language for the specialty store the customer is in. If no file is specified for the language and specialty store, then the default file for all specialty stores and languages is displayed. See [Cataloging Multimedia](#) for information about how to define multimedia relationships to specialty stores and languages.

Multimedia Called Indirectly

You can show multimedia files associated with a product in or section by APIs to show the multimedia file for a specific multimedia component. See [Defining Multimedia Components](#) for information about creating multimedia component

names, their related programmatic access names, and assigning a default multimedia name to each multimedia component. Oracle iStore 11i determines what multimedia file to display for a multimedia component.

In this case, you associate the multimedia object with a given multimedia component associated for products or sections. Then, when that product section is displayed (using defaulting rules that are described below), Oracle iStore 11i gets the multimedia object that is associated with the multimedia component for the product/section according to the following process.

Example 1: The Product is Displayed

1. If you specified a multimedia object for the multimedia component, Oracle iStore 11i retrieves the correct file based on the language of the specialty store.
2. If you have not specified or specified that the default is used then Oracle iStore 11i gets the default for the multimedia component at the product's primary display category.
3. If no value is specified, then Oracle iStore 11i gets the setting for the component from the store level. If no setting is at the store level, then Oracle iStore 11i gets the setting from the profile value.

Example 2: The Section is Displayed

1. If you specified a multimedia object for the multimedia component, Oracle iStore 11i retrieves the correct file based on the language of the specialty store.
2. If no value is specified, then Oracle iStore 11i gets the setting for the component from the store level. If no setting is at the store level, then Oracle iStore 11i gets the setting from the profile value.

How the Store Displays Templates

Your web page can call for templates to display in two ways:

- Directly by using the template name
- Indirectly by using the display style

Templates Called Directly

You assign an a template name and a programmatic access name to a template using the Template tab. You then assign one or more physical template files to combinations of specialty stores and languages. The result is that when the

customer enters the page, Oracle iStore 11i looks at the customer's language and displays the files that are assigned to the customer's language for the specialty store the customer is in. If no file is specified for the language and specialty store, then the default file for all specialty stores and languages is displayed.

Templates Called Indirectly

You can indicate that the template associated with a given display will be used when displaying a product. See *Defining Display Styles* for information on creating your own display styles. You can also indicate at the section level the display style to use for displaying products that belong to that section. Oracle iStore 11i uses the following process to determine which template to use for a specific display style and product.

1. Oracle iStore 11i will use the template that you associated for the product display style.
2. If you do not have a template associated for a product display style, Oracle iStore 11i gets the associated template for the display style at the product's primary display category.
3. If you do not have a template associated for a product display style, or the primary display category could not be found, then Oracle iStore 11i gets the template from the store level defaults.

Understanding JSPs

A Java Server Page (JSP) is basically a dynamic HTML web page that embeds Java language methods in the HTML content to generate dynamic content on the web page. The JSP file contains HTML, Java, JavaScript, forms, and so on.

JSP files consist of the following basic components:

Directives

- Page Directive

```
<%@ page import="hello.NameHandler" %>
<%@ page info="a hello world example" %>
```
- Include Directive

```
<%@ include file="banner.html" %>
```

Declarations

```
<%! ...%>
<%! int a, b; double c; %>
```

Expressions

```
<%= ...%>
<%= a + b + c %>
```

Scriptlets

```
<%...%>
<% String name=null;
if ( request.getParameter("name") == ) { %>
```

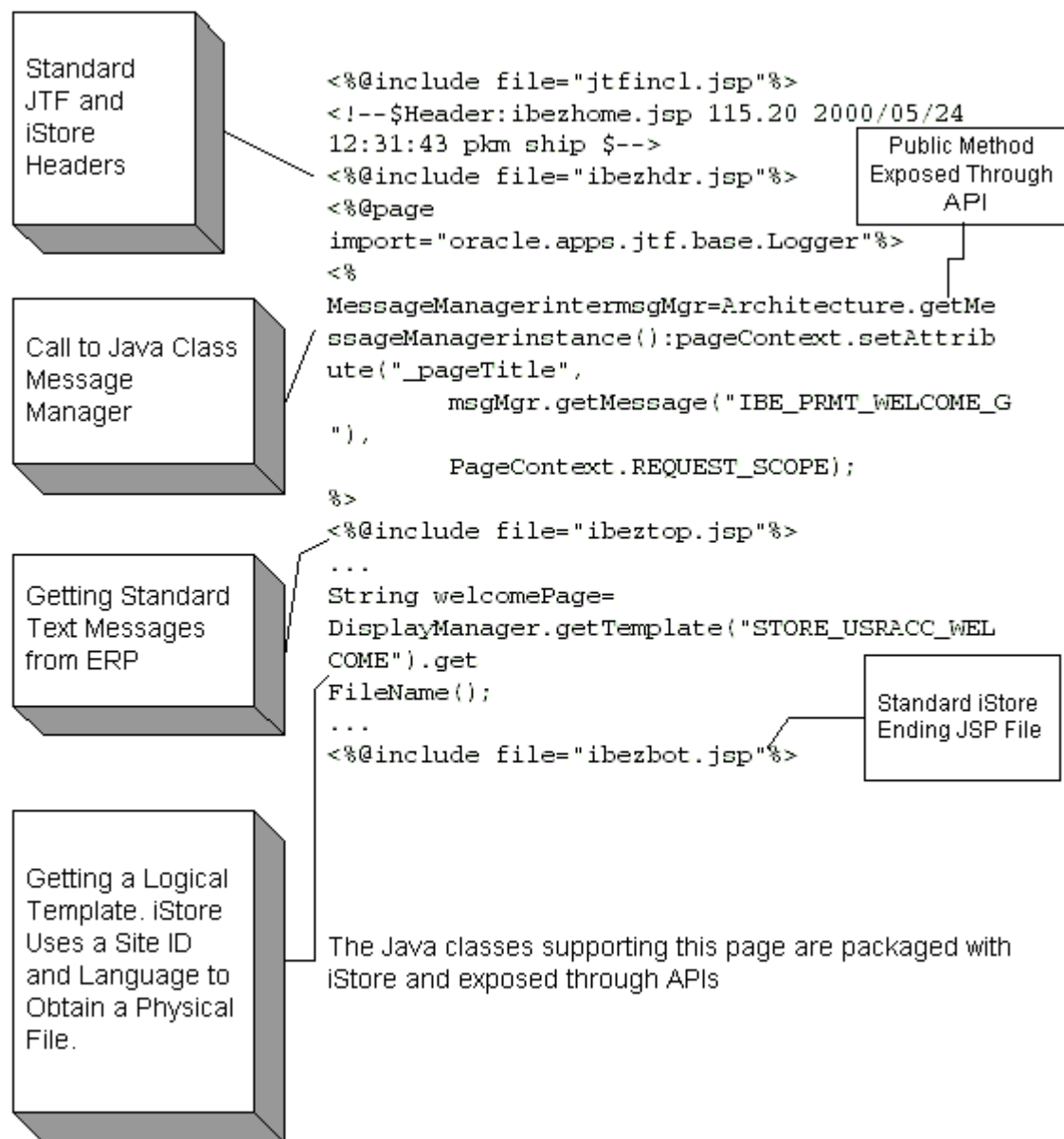
Example 1 foo.jsp

```
<%@ include file="jtfincl.jsp" %>
<%@ include file="ibezhdr.jsp" %>
<%@pageContext.setAttribute ("_pageTitle", "Test", pageContext.REQUEST_SCOPE);
%>
<%@ include file="ibezhdr.jsp" %>
Hello World
<%@ include file="ibezbot.jsp" %>
```

foo.jsp is accessed from a browser as
`http://some.domain.com/html/foo.jsp?minisite=10120`

The minisite id is a different number for different installations. After the first session, the minisite parameter is passed along in the cookie.

For an example of a JSP layout, see the following illustration.



Standard Includes

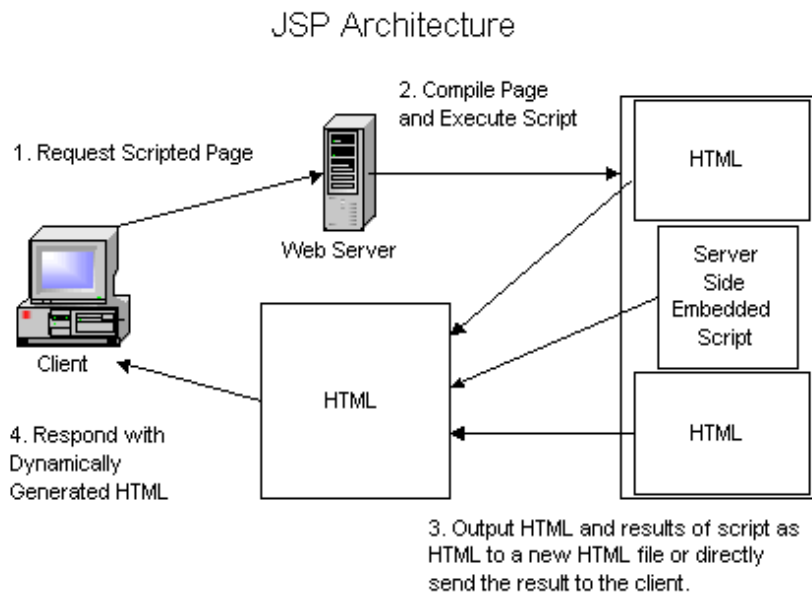
The Oracle iStore 11i standard includes are listed in the following table.

Standard Includes

JSP	Description
jtfinel.jsp	CRM Foundation standard header
ibezhdr.jsp	iStore standard header
ibeztop.jsp	Prints the top of the page to be generated: <head>, <title>, and so on
ibezmenu.jsp	Prints the top links, tabs, subtabs and the search bar
ibezbot.jsp	Standard footer

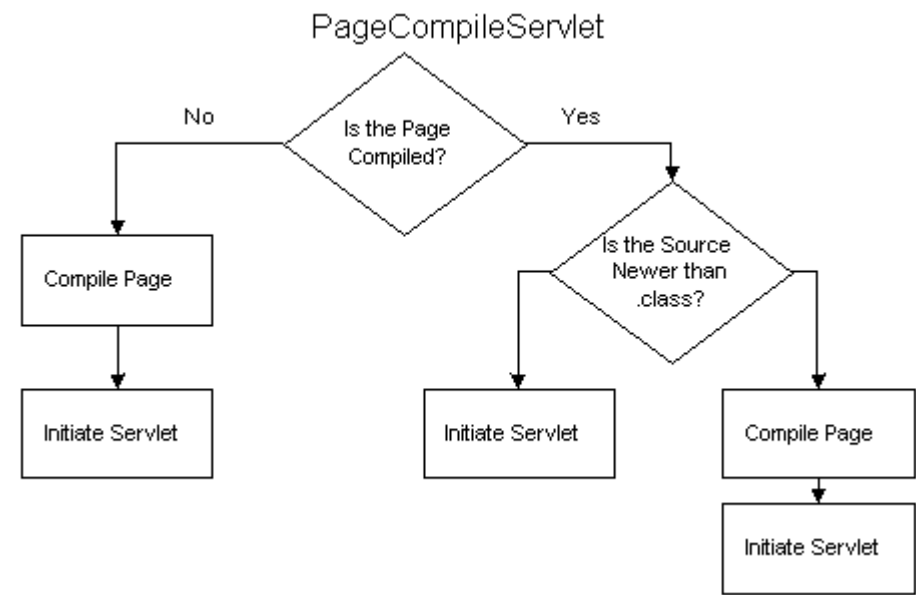
JSP Architecture

The Java Server Page architecture is described in the following drawing.



PageCompileServlet

The decision process for PageCompileServlet is described in the following diagram.



The Request Object

After a user enters data, the data is stored in the request object, which usually implements javax.servlet.HttpServletRequest. You can access the request object from within a scriplet.

Request Objects

Method	Defined In	Job Performed
getRequest	Javax.servlet.jsp.PageContext	Returns the current request object.
getParameterNames	Javax.servlet.ServletRequest	Returns the names of the parameters that the request currently contains.
getParameterValues	Javax.seervlet.ServletRequest	Returns the values of the parameters that the request currently contains.
getParameter	Javax.servlet.ServletRequest	Returns the value of the parameter if you provide the name.

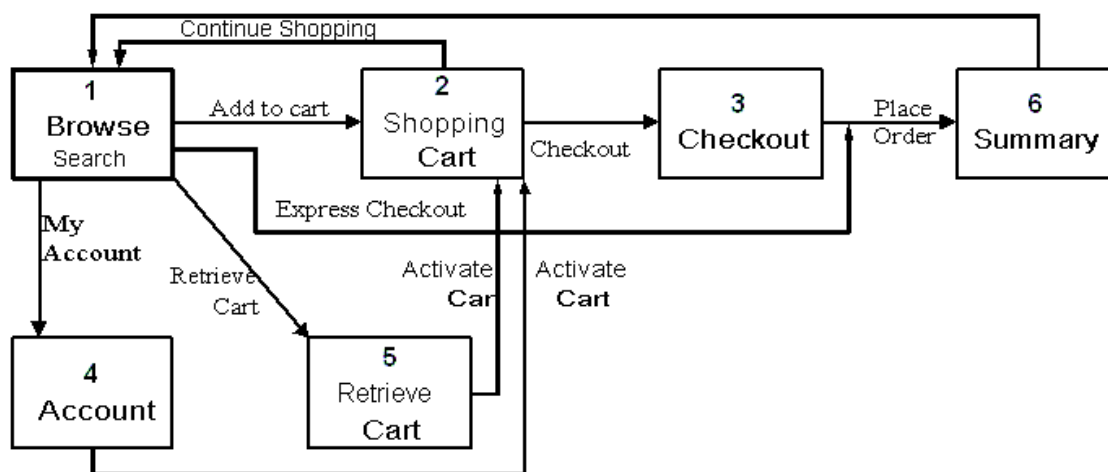
Administering Oracle iStore 11i

This chapter describes the administration of Oracle iStore 11i after you have verified your installation and dependency setup.

Refer to Oracle iStore 11i Implementation Guide and Oracle ERP Checklist.

Setting Up the Store Business Flow

The flow through your store must follow a certain path, as shown in the following top level diagram and related detail diagrams. Oracle iStore 11i provides templates you can use for the pages illustrated in the diagrams.



Customers enter the store at (1) and browse through the store pages. See [Business Flows and Templates for Search](#) to view the expanded diagram.

If the customers want to edit user information, view orders, or use saved shopping carts and lists, then they click My Account (4). A customer can reach the account page from any other page. See [Business Flows and Templates for Account](#) to view the expanded diagram.

While shopping, customers choose items to add to the shopping cart (2), at which time the shopping cart page appears. See [Business Flows and Templates for Shopping Cart](#) to view the expanded diagram. Continue Shopping takes the customer back to the home page.

Customers who are browsing the store can retrieve a previously saved shopping cart (5), which takes them to the shopping cart page.

Customers ready to purchase the items in the shopping cart click Checkout and go to the checkout page (3). See [Business Flows and Templates for Checkout](#) to view the expanded diagram.

The checkout process takes customers to the summary page (6). Customers click Express Checkout to omit the shopping cart and checkout process and go directly to placing the order.

Creating Your Store

Use the following procedure as an example of the sequence of steps you can use to create your store.

Prerequisites

Identify the ways in which you will display products in the store. Plan your page designs and divide the designs into common components that you can make into templates. Invent a name for each possible template to facilitate planning and communication of designs. Oracle iStore 11i ships with templates that allow you to develop the initial store. You can customize templates now, if you need to, as part of the design or as a revision to the templates later in the setup cycle.

Steps

1. In the Merchant UI, set up at least one specialty store by choosing **Specialty Stores** in the Setup tab.

2. Define the types of media you will use when building your store by choosing **Multi Media Components** in the Setup tab. Examples of media types include small gif, large gif, short description, and 10 second audio file.
3. Enter default source files for your multimedia components using **Multi Media Components** in the Setup tab.
4. Define each type of display template you identified above by choosing **Display Styles** in the Setup tab.
5. Enter default source files for your display styles using Display Styles in the Setup tab.
6. Create templates for pages and for blocks within pages using your chosen web authoring application.
7. Record and organize your template files using the Template tab. (Do not yet assign categories to the template.)
8. Create one overall hierarchy for your products, including grouping them into sections, using the Hierarchy tab
9. Choose the section from your hierarchy to be the root node for your specialty store by choosing **Specialty Stores > Store Flags** in the Setup tab.
10. Create media files outside of Oracle iStore 11i. Media include graphics files (.gif), text files, audio files, video files, and so on.
11. Assign templates and multimedia to product categories in the Category tab.
12. Build your product catalog in the Product tab and assign templates.
13. Record and organize your media files using the Multimedia tab.
14. Build product relationships for merchandising using the Relationship tab.

The new specialty store is saved.

Setting Up the Product Catalog

Setting up the product catalog involves the following considerations.

- Designing screen flow and navigation, also called developing hierarchy structures, the principles and philosophy behind the navigation paths.
- Determining items to be sold, their display features, and configuration options, also called detailing fields for inventory and configurator.

- Determining types of data required, for example, books = title, author, publisher; jacket = size, color, men's, women's. Use the flexfield structures in Inventory to store and sort data accordingly.
- Designing the display appearance for different product types. This process determines the number and type of product templates required. For example, perhaps all music products list the artist first and then provide a link to an audio clip, but all clothing products list the clothing type, for example jacket, first followed by a static graphic of the item. Oracle iStore 11i ships with the assumption that all product types appear the same on the Customer User Interface (UI).
- Creating template text for product types. Text imbedded in a template makes that template specific to the given product type. For example, the word "Artist" in front of the flexfield where a performer's name is to appear can only be used for the compact disc product category. Imbedded text must be manually translated and saved in the required multiple languages as additional template types. Oracle iStore 11i does not translate template text. Alternatively, templates using generic terminology can be more easily applied across product types. For example, using the term "Lead Performer(s)" as a flexfield label could apply to both compact disc and videotape product categories. Providing no flexfield labels in a template allows templates to be most broadly applied across product types.

Prerequisites

- Products must be loaded into inventory before they can be imported into Oracle iStore 11i. The publish / unpublish flag identifies the store-bound products.
- Profile options must have been set.
- Shipping options must have been seeded into Oracle Shipping.
- Payment options and setup must have been seeded into iPayment.
- Store layout must have been determined with the following considerations:
 - Site appearance must have been decided.
 - Hierarchy of products, sections and specialty stores must have been identified. If the site has to be deployed in multiple languages and currencies, you may create a specialty store for each language.
 - Templates associated with each product, section and specialty store must have been identified.
 - Templates and sub-templates must have been identified.

- At the implementation level, the mapping between templates and source files must have been decided.
- Source files (physical templates) must have been created by the UI implementation team with stubs for the dynamic elements, along with the multimedia components to be displayed on the site.
- Templates must have been populated with the dynamic JSP elements calling Oracle iStore 11i, using the templates shipped with Oracle iStore 11i as a model.

Steps

Following is the suggested sequence of steps for setting up Oracle iStore 11i. Alternatively, you can use the Merchant UI navigational sequence of creating the Specialty Store first.

1. Define JTF profiles.
2. Define IBE Profiles.
3. Run Concurrent jobs.
4. Set up Shipping options:
5. Set up Payment options.
6. Set up Product Search.
7. Create Multimedia.
8. Create / modify custom templates.
9. Identify templates.
10. Define display styles.
11. Create content.
12. Define Multimedia components.
13. Create store sections (Hierarchy).
14. Create Store Hierarchy.
15. Create Specialty Store.
16. Create Product Relationships.

Creating Specialty Stores

A specialty store is any web store. You can create multiple stores, for example a main store, a store for one large customer, a holiday specials store, and a store that requires registered users. You must create at least one store. Use the following procedure to create a store.

Prerequisites

Check that you have set up the following prerequisites:

- Business units in Oracle General Ledger
- Price lists and currencies in Oracle Order Management
- Language in AOL

Note: If the information for these prerequisites is unknown, you can continue with the setup now and revise this information later.

After you build your product hierarchy you can return to this tab and specify the root section from your hierarchy for this specialty store in the Store Flags page.

A default language is set for the store. However, a customer can then choose a preferred language from those offered. Once a registered customer selects a preferred language, the store defaults to that preferred language each time the customer enters.

You can choose to make the store active at a later time.

You must select a price list for each type of customer on the Currencies and Price Lists page. Only maximum order limit is optional.

To create a new speciality store, click Create or click the name of the specialty store that you want to modify. You can reach this section from the **Merchant UI > Setup > Specialty Stores** section.

Note: Each Specialty Store must have a root node, which has not yet been created. As a work around, leave the root ID blank and proceed to the step “Choose root node,” and then return.

Steps

1. In the Setup tab, choose **Specialty Stores**.

The Specialty Store page lists existing store names and descriptions.

2. Click **Create**.

The wizard for creating a specialty store starts. Enter the basic information for the store, including name and default language, and click **Continue**.

The Supported Business Units page displays available business units from Oracle General Ledger.

3. Select a default business unit, optionally select other business units, and click **Continue**.

The Specialty Store Flags page appears.

This section is used to define the profile that determines whether the specialty store has:

- Walk-in customers allowed
- ATP Enabled for inventory
- Root Section - Name of the root store (required).

4. Choose to:

- Designate the store active or inactive
- Provide inventory information to the customer
- Allow walk-in customers who do not log in

5. Click **Continue**.

The Currencies and Price Lists page displays available currencies.

6. Select currencies and price lists for each selected currency.

Assign the Price lists for Walk-in customer, Registered customer and Business customer. This screen is also used to set the currency for the specialty store and the item orderable limit. You can select multiple currencies and languages for a

specialty store. A user’s preference defines the currency and language to use. If the user preference is not set, the default language and currency settings take effect. Set the default language and currency before you set the store flags.

- 7. Enter a maximum order limit for each selected currency.
- 8. Click **Continue**.

The new specialty store is saved.

Cataloging Multimedia

Multimedia consist of files used to present content on a web page to your customer, such as graphics, text, audio, and video. Use this procedure to add multimedia names and to catalog available media files mapped to the multimedia name. See [How the Store Displays Media](#) for information about how the information you enter here is used by the store.

Naming Multimedia

The multimedia name is the catalog name that is easy to communicate and use when planning your page designs. An example is *CompanyLogo*.

Every multimedia name is given a unique programmatic access name that is shorter and less descriptive than the multimedia name. The programmatic access name is used to display that multimedia file in a web page, if you want to refer to it directly. It is not translated. An example is *clogo*.

The multimedia name and programmatic access name represent several source files. You assign each source file to combinations of specialty stores and languages. The following table lists example file names for the example multimedia name *CompanyLogo*.

Sample Media File Names

File	Specialty Store	Language
clog1f.gif	specialty store 1	French
clog1e.gif	specialty store 1	English
clog2f.gif	specialty store 2	French
clog2e.gif	specialty store 2	English

The result is that if a French customer enters specialty store 1, the store displays the French logo file `clog1f.gif`. If an English customer enters the same specialty store, the store displays English `clog1e.gif`.

You can search for multimedia more easily if you enter keywords for the multimedia.

Prerequisites

- The default language must have already been defined.
- At least one speciality store must have already been created.

Steps

1. In the Multimedia tab, conduct a search for media that is already cataloged and available to use in your store.

The Multimedia page lists the multimedia that match your search criteria and the access names assigned to the multimedia, keywords, descriptions, and the default source files to use for all specialty stores and languages.

2. Click **Create**.

The Multimedia Details page appears.

3. Enter the media detail information.
4. Define the programmatic access name, which is the name by which the media will be accessed from the template.
5. Define a common name to which the media can be referred.
6. Optionally, define a media description.
7. Define the default source file which contains the media content. Define the location of the file relative to the `OA_Media` set up for example, `/OA_MEDIA/product.gif`. Oracle iStore 11i uses this default source file unless a specific mapping for a speciality store or non-default store language is preferred by the customer. If only one language or specialty store is defined or if no specialty store has been created, use the defaults to continue with the Oracle iStore 11i setup.
8. Optionally, click **Add** to provide files for the same multimedia object in different languages and specialty stores.

The Source File Details page appears.

9. Enter the name of a media source (physical) file, such as a graphic file, that you want to display on a web page for the media name that you are creating, for example, /OA_MEDIA/video.jpg.

10. Add each specialty store and language where you want the new source file to appear, then click **Update**.

The relationship between the new media name, source files, specialty stores, and languages is saved.

11. Repeat from step 8 for each source file you want to add.

12. In the Multimedia tab, choose **View All Mappings**.

The View All Mappings page displays each source file name and its relationship to specialty stores and languages.

Defining Multimedia Components

Multimedia components define the types of media objects available to display on a web page, such as a certain size picture, short text description, or audio of a certain length. One multimedia component dynamically displays the correct media file in your store according to the product, category, or section that the customer chooses to view. Use this procedure to define multimedia components that you want to assign to sections, categories, or products. See [How the Store Displays Multimedia](#) for more information about how store uses the information that you enter here.

Prerequisites

You can select a default multimedia in step 4 of the following procedure only after you have cataloged multimedia. If the default information for step 4 is unavailable, you may proceed with the setup process and complete step 4 later. However, if a multimedia association is requested for any product or section with that multimedia component, and an association is not specified between the product or section and the multimedia component, Oracle iStore 11i uses the default multimedia defined at the store level (profile value). To avoid the error, you can also use the multimedia components seeded values:

- STORE_PRODUCT_LARGE_IMAGE
- STORE_PRODUCT_SMALL_IMAGE
- STORE_SECTION_SMALL_IMAGE

Steps

1. In the Setup tab, choose **Multimedia Components**.

The Multimedia Components page displays a list of existing media components and default source files that the store uses if no file is assigned to the product or category.

2. Click **Create**.

The Multimedia Component Details page appears.

3. Assign names and descriptions to your multimedia component.
4. Optionally, enter a multimedia name that was created in the Multimedia tab in the Default Source File field or click **Go** to select a file.
5. Click **Update**.

The multimedia component information is saved. The multimedia components appear in the Product and Category tabs where you choose a multimedia name to correspond with the component for the product or category. If no multimedia name is selected for both product and category, then the default source file you entered in step 4 is used on any store web page for that product or category.

Defining Display Styles

When you design your pages, you can specify in a display style how to present products. For example, you can use one display style to display product A on a special sale page containing multiple products, and a different display style to display product A on a page detailing product information.

Use this procedure to define display styles. See [How the Store Displays Templates](#) for information about how the information you enter here is used by the store.

Prerequisites

You can select a default template in step 4 of the following procedure only after you have cataloged templates. If the information for step 4 is unavailable, you may proceed with the set up process and complete step 4 at a later time. However, if a template association is requested for any product or section with that display style and an association is not specified between the product or section and the display style, Oracle iStore 11i will use the default template defined at the store level (profile value).

To avoid the error, you can also use the seeded values for display styles. For example:

Display Styles

- Go to Setup Display Styles.
- Review the seeded values. Optionally, create more.

Seeded Values

STORE_FEATURED_PRODUCT

STORE_PRODUCT_DETAIL

STORE_PRODUCT_SMALL_DESCR

Steps

1. In the Setup tab, choose **Display Styles**.

The Display Styles page displays a list of existing display styles.

2. Click **Create**.

The Display Style Details page appears.

3. Assign names and descriptions to the display style.
4. Optionally, click **Go** to select a default template for this display style.
5. Click **Update**.

The display style information is saved. The display styles appear in the Product and Category tabs where you choose a template name to correspond with the display style for the product or category. If no template name is selected for both product and category, then the default source file you entered in step 4 is used on any store web page for that product or category.

Creating Templates

You can create new templates to replace or add to the templates already available in the base product. Creating templates involves adding the name and description of these new templates and also specifying the different physical JSP templates to be used at run-time based on language and the specialty store.

Major skills required to modify the templates are HTML and JSP (Java Server Pages). The templates are in the HTML directory base. All ibem*.jsp templates are for Merchant UI, all ibez*.jsp templates are the included templates for customer UI. Users can create new templates in this directory.

The Oracle iStore 11i seeded templates are developed in JSP, which is basically a dynamic HTML web page. JSP embeds Java language methods in the HTML content to generate dynamic content on the web page. The structure of a JSP page is demonstrated in the following HTML example.

Note: Never change an original JSP. To modify a JSP, make a copy of the original JSP and modify only the copy. If a bug occurs, compare the JSP copy to the JSP original.

<HTML>

<% import="oracle.apps.ibe.util.*" %>

....

....

<P> Name: <% = customer.getName(12334) %> Where customer is a Java class on the server and getName is a public method in the class to retrieve the customer Name.

<P> Picture: <IMG SRC = "<%= customer.getPict(12334) %>"> This step can retrieve the image file name from the customer Java class on the server.

After making the changes to the templates, you can pre-compile the template to check for compilation errors and to increase the speed of the initial loading.

Follow the store-specific standards for modifying templates.

See [JSP Customizing Standards](#).

Building the Product Catalog

Use this procedure to build your product catalog.

Prerequisites

- Products must exist in Oracle Inventory.
- Display styles must exist before you can assign them to a product.
- Products must have a status of “Published” before they can be sold in the store.

Steps

1. In the Product tab, choose **Products**.

The Products page appears.

2. Enter search information to find products that you want to include in your catalog.

The Products page lists products in Oracle Inventory that match your search criteria and displays existing product catalog information for those products.

3. Double-click the product name that you want to edit.

The Basic Description page displays the inventory description and any descriptions you have already added to the product catalog.

4. If you want to make the product available to be sold in your store, then change the posting status to **Published** and click **Update**.

5. Optionally, enter or modify the short and long descriptions and click **Update**.

The descriptions are saved and are available to display in your store.

6. In the Products menu, choose **Hierarchy Paths**.

The Hierarchy Paths page displays the hierarchy of products and sections that have been set up for the store.

7. Optionally, you can remove or add parent sections for the part and edit the dates and display order.

8. In the Products menu, choose **Category and Display Styles**.

The Category and Display Styles page displays the product and category you are editing and lists all display styles and any template names already assigned to the product.

9. Choose a template name for each display style that you want to use for the product. Answer **Yes** if you want the category level template used as the default rather than the product-level template for the display style. Click **Update**.

When the display style is used for a store page that relates to the product, then the template name retrieves the correct template file according the specialty store the customer is in and the customer's language.

10. In the Products menu, choose **Multimedia Components**.

The Multimedia Components page lists all multimedia components you defined in the Setup tab.

11. Optionally, assign multimedia names to multimedia components and click **Update**.

When the multimedia component is used in a store page that relates to the product, then the multimedia name retrieves the correct multimedia file according the specialty store the customer is in and the customer's language.

12. In the Products menu, choose **Relationships**.

The Relationships page displays existing relationships between the product and other products or sections and the rules for those relationships, such as the product to show for an upsell or cross sell.

13. If you want to add a relationship, then click **Add Related Products**.

14. In the Product menu, choose **Specialty Store**.

The Specialty Store page lists the specialty stores where the product will be displayed.

15. Either choose **Default** and then **Update**, or click **Select** and select the specialty stores that can display the product and then click **Add**.

The included specialty stores appear.

Verifying Templates

To verify templates, click on Templates, and search for STORE_HOME. It should be set to ibezhome.jsp.

Mapping Templates in the Merchant UI

Once you have created customized templates, map the templates in the Merchant UI. The display manager screens in Merchant UI are used to give a logical name to a template.

The Display Manager is the class that implements Oracle iStore 11i's Template Manager. The template Manager maintains a mapping from a logical name or access name of a media object to a physical name on the file system. For example, STORE_HOME (logical) maps to ibezhome.jsp (physical).

Note: Leave this blank if you are running Oracle iStore 11i out of the box.

The first screen in the Templates tab lets the user enter the basic details of the template. Once these details are entered, the template then can be assigned to a

category / section / item and can be mapped based on the display style and language settings, on subsequent screens. To display text in multiple language, use AOL's message manager.

Assigning Templates

When an item is displayed, Oracle iStore 11i process searches for the template assignment on item level. If there is no item-specific template defined at this level, it searches for the Category level template. If there is no template at this level, it searches for the Section level template. If the section level template is not defined, the store level template is used.

- **Item** - To display a particular item differently based upon a display style, like a Featured Product
- **Category** - For a specific Display Style
- **Section** - For example, the Database section can have a different template than the CRM section

After the process defined above has retrieved the logical name for the template, the source filename of the template is determined by the Logical Name + Specialty Store +Language mapping.

Template Assignments

Template Name	Programmatic Access Name	Keywords	Description	Displays	Default Source File
ADDRESS SELECT	ST_ADDR_SL	Address	Select a particular address	OTHERS	ibeaads1.jsp
AGREEMENT	ST_AGGR	Agreement	Agreement for consulting	FEATURE	ibeagreem.jsp

To set the mapping for the template language and specialty store combination choose **View All Mappings > Specialty Store and Language Mappings >Template Name**.

Template Mapping for Language and Specialty Store

Specialty Store	Languages
Oracle Store External	ALL

Assigning a Template to a Category

To assign a template to a category, use the following procedure.

Steps

1. Choose the Category tab > category name.
The Templates Assigned page opens.
2. Click **Add**.
3. Choose the template that you want to add. You can search for the template by name.
4. Choose **Add** > **Done**.

Viewing Categories Assigned to a Template

You can view the categories assigned to a template by using the following procedure.

Steps

1. Choose the Template tab.
2. Click a template Name.
The Template Details page opens.
3. Click Categories.
The Templates - Assigned Categories page opens, listing all categories assigned to the template.

Viewing Template Details Assigned to a Category

You can view the template details that are assigned to a category by using the following procedure.

Steps

1. Choose the Category tab.
The Categories page opens.
2. Click the category Name.

The Templates Assigned page opens, listing the template names, the default source files for all sites and languages for each template, and a description of each template.

Cataloging Templates

You assemble templates to create a web page. When you design your pages, you use designs that consist of common components that you make into templates. Use this procedure to create names and programmatic access names for templates that you design and assign to specialty stores and languages.

This step involves identifying the JSP templates that are needed to run Oracle iStore 11i. The out-of-the-box application comes packaged with a complete set of templates needed to run the store. If you want to expand the functionality or wish to customize the pre-packaged templates, then you need to identify the flow of the application and identify the JSP templates needed to implement the flow.

There are two steps required for setting the templates for the store. The first step is to create or modify the templates. The second step is to set up the mapping through the display / template manager functionality that is provided in the Merchant UI.

This section describes the set up of the templates, which are available as standard installation or are created new. This step involves adding new templates to replace or add to the templates that are already available in Oracle iStore 11i. It involves adding the name and description for these new templates and also specifying the different source (physical) JSP templates that will be used at run-time based on language and the specialty store.

The template name is the catalog name that is easy to communicate and use when planning your page designs. An example of a template name is *ProductHome*.

Every template name is given a programmatic access name that is unique, and shorter and not as descriptive as the template name. The programmatic access name is inserted into your web page or template. The programmatic access name is not translated. An example of a programmatic access name is *phome*.

The template name and programmatic access name represent several physical files. You assign each physical file to combinations of specialty stores and languages. The following table shows an example of files for *ProductHome*.

Example Template File Names

File	specialty store	Language
hom1f.jsp	specialty store 1	French
hom1e.jsp	specialty store 1	English
hom2f.jsp	specialty store 2	French
hom2e.jsp	specialty store 2	English

The result is that if a French customer enters your specialty store 1, the store displays the home page file hom1f.jsp. An English customer in the same specialty store 1 sees hom1e.jsp instead.

You can find templates more easily in a search if you enter keywords for the template.

Steps

1. In the Templates tab, conduct a search for templates that are already cataloged and available to use in your store.

The Templates page lists the templates that match your search criteria along with the programmatic access names assigned to the templates, keywords, descriptions, and the default source files to use for all specialty stores and languages.

2. Click **Create**.

The Template Details page appears.

3. Enter the template detail information.

4. Click **Add**.

The Source File Details page appears.

5. Enter the name of a template source file that you want to display on a web page that calls for the template name you are creating.
6. Add each store specialty store and language where you want the new physical file to appear, then click **Update**.

The relationship between the new template name, source files, specialty stores, and languages is saved.

7. Repeat from step 4 for each physical file you want to add.

8. In the Templates tab, choose **View All Mappings**.

The View All Mappings page displays each physical file name and its relationship to specialty stores and languages.

Modifying Templates

To modify templates, use the following procedures.

Note: Never modify an original JSP. Make a copy of the original JSP and modify only the copy. If a bug occurs, compare the JSP copy to the JSP original.

Prerequisites

- At least one specialty store must have already been created.
- At least one language must have already been defined.

Steps

1. In the Templates tab, conduct a search for templates that are already cataloged and available to use in your store.

The Templates page lists the templates that match your search criteria along with the programmatic access names assigned to the templates, keywords, descriptions, and the default source files to use for all specialty stores and languages.

2. Click **Create**.

The Template Details page appears.

3. Enter the template detail information.

- a. **Name:** The name by which the template is referred to from the design, the common name.
- b. **Programmatic Access Name:** The name by which the template is referred to in the JSP.
- c. **Description:** Optionally, enter a description and key words.
- d. **Default source file:** The JSP to be used as the default if no non-default language or specialty store mapping is defined. See Guidelines below for details.

- e. Define whether the template is to be used to display a section category or other items. If the template will display a product, choose Category from the drop down list.

- 4. Optionally, click **Add** to provide files for the same template in different languages and specialty stores. See the Guidelines below for details.

The Source File Details page appears.

- 5. Enter the name of a JSP source (physical) file that you want to use for the template name you are creating.
- 6. Add each store specialty store and language where you want the new physical file to be used for this template, then click **Update**.

The relationship between the new template name, source files, specialty stores, and languages is saved.

- 7. Repeat from step 4 for each physical file that you want to add.

- 8. In the Templates tab, choose **View All Mappings**.

The View All Mappings page displays each physical file name and its relationship to specialty stores and languages. This step is highly recommended, but not required to continue the store set up.

- 9. Optionally, if the template you created is meant to display product categories, you may optionally assign it to categories at this point by going to the Category tab to assign it.
- 10. Restart Apache to save your changes.

Defining Hierarchies

The hierarchy determines the browsing experience of the customer and what products are featured at different levels in the store. When users come to a specialty store, they see the hierarchy starting from the root node of the store. You can choose not to show a particular section in a specialty store even though the given specialty store might be pointing to an ancestor of the store. Your template design determines how to manifest the hierarchy for the user. In the out-of-box store templates, the top level appears as tabs while the lower level appears as browse bins on the store pages.

To define hierarchies, create one overall hierarchy that contains products from Oracle Inventory that you group into sections. Associate your specialty stores to a portion of the overall hierarchy or to the whole hierarchy itself by setting up their

root section to point to a section. Set up the top level sections first. For each top level section, create as many subsections or children as you wish. The levels of sections are driven by the design. You can create and revise new templates at any time.

You can assign products that belong to the hierarchy in this procedure or when you work on products. Similarly, you can create groups of featured products at any level in the hierarchy by creating a subsection of type featured in that section. Then in the template for the section you can explicitly show the products in that section. The products in a section are shown by using the display style that you specify for that section.

Out of the box, if a section contains other subsections, show the featured products in the middle and show the subsection on the left browse bin. If the section contains only products, show the list of the products in the middle.

The section code is the name used in the coded web page or template to access the section information.

The display style you choose for products in this section is the style used to display products on a section page for this section.

You can choose Default for included specialty stores that ties the section to its ancestors. When a specialty store is added to or deleted from the ancestor, then the same change applies to the section that is tied to the ancestor.

The Relationships page displays existing relationships for a section.

You can enter a SQL rule in the Advanced Settings page that will result in a new product in the inventory to be automatically added to a selected section.

Prerequisites

Profile set up for IBE:Item Validation Organization (see definition for IBE_ITEM_VALIDATION_ORGANIZATION profile)

Optionally, you may also have to set up IBE: Default Section Template. See step 2.

Steps

1. In the hierarchy, select the node below the node that you want to create, and choose **Create**. (If you want to modify existing information, click on the node.)
2. You will see Basic Information Screen. Fill in the details as follows and choose **Continue**.
 - a. Name: The name of the section.

- b. Section Code: If you want to refer to the section in templates explicitly by name, use the section code for the section.
 - c. Section Type: Navigational or Featured.
 - d. Basic And Long Description: If you want to show content about the section in the store, fill in this field.
 - e. Keywords: For search purpose (not currently used).
 - f. Display style for displaying the products in the section.
 - g. Template for displaying this section. Several sections can share the template. If you do not provide a value, then Oracle iStore 11*i* uses the value for the IBE:Default Section Template.
 - h. Start Date Active and End Date Active: Specify the start time and end time for the section to be active.
3. Provide information about multimedia components specific for this section. This will be used if you want to show / associate content with sections too. For a given component, hit go to search the multimedia catalog and if not found then go and create one and then associate with the section.
4. Hitting continue after the next steps will lead to a screen where you can decide if this section is going to be restricted from appearing in some specialty store(s). You can also specify default behavior for new specialty stores you might create: Whether the section should only appear in the specialty stores you have selected here or should it appear in other specialty stores you might create in future if those specialty store's root points to an ancestor of the current section.
5. Choosing Continue leads you to review and create relationship rules for this section. See Building Relationship.
6. After this step you can optionally specify if the section is going to be automatically populated with products based on certain SQL clause as well as specify the order. The auto-placement rule is not currently used. You can also specify Order By clause to specify how the product for a section should be ordered when displayed in the Customer UI. The value for this field could be just one column name from MTL_SYSTEM_ITEMS or it could be comma-separated columns of the same table. Click **Continue**.
7. Now you have created sections and can go and can add products or subsections to this section. Please note that a section can only have a subsection or products as children. You can also add products to a section when you are working on the product (Building product catalog). To add product now, search products by clicking on Add Product that will lead to a pop-up page and then selecting one

or more from the result and hitting Add. When done, choose **Done** to close the pop-up and return to the main store manager window.

Guidelines

The section code is the name used in the template to directly access the section information.

The display style you choose for products in this section is the style used to display products on a section page for this section.

You can choose “**Include in all future specialty stores if the stores root section is ancestor of this section**” for the Included Specialty Stores screen which ties the section to its ancestors. When a specialty store is added to or deleted from the ancestor, the same change applies to all descendant sections.

The Relationships page displays existing relationships for a section.

Example:

Store Hierarchy

1. Go to the Hierarchy tab.
2. Highlight Home, and click **Create**.
3. Enter Featured Products as the name as well as the code.
4. Select **Featured** as the section type. Everything else is optional.
5. Click **Continue**.
6. On the Multimedia Components screen, everything is optional. Choose **Continue**.
7. Accept the defaults on the Specialty Store and Navigation Relationships screens, and choose **Continue**.
8. The auto-placement rule in the Advanced Settings screen is currently not supported. If required, enter Order By clause. Click **Finish**.
9. In the left frame, click **Refresh**. Expand the Home node, which should have the newly-created section under it.
10. Highlight **Home**, and click **Create**. Repeat the above steps to create another section. Call it Books, section type Navigational.
11. Repeat again for Music, Electronics, Computers, and so on.

Adding Products To Sections

1. Click on a section name (for example, Featured Products) to add products to the section.
2. Click **Add Product** on the children page for the section.
3. Search for a product to add (for example, Sentinel Multimedia).
4. Check the selected products, and click **Add**.
5. Click on the product name, and change its posting status to Published. (See note about WEB_STATUS above.)
6. Repeat the above steps to add more items.

Defining Products

Use this procedure to build your product catalog.

Prerequisites

- Products must exist in Oracle Inventory.
- Display styles must exist before you can assign them to a product.
- Hierarchy must be defined before you can assign products to sections. See Defining Product Hierarchy section.

Steps

1. In the Product tab, choose **Products**.

The Products page appears.

2. Enter search information to find products you want to include in your catalog.

The Products page lists products in Oracle Inventory that match your search criteria and displays existing product catalog information for those products.

The search criteria are:

- Name
- Part number
- Belongs to category
- Created after date
- Created before date

- Status - values should be published or unpublished
 - New - products created in last x days where x is the profile value of - IBE: Number of Days for New Items
 - % can be used for a wild character search
3. Click the product name that you want to edit.

The Basic Description page displays the inventory description and any descriptions you have already added to the product catalog.

4. If you want to make the product available to be sold in your store, then change the posting status to **Published** and click **Update**. The product is published immediately.

If the product is not published, you will see the Publish button in the Product Listing page that will take you through the steps required to publish. Alternatively, you can click on **Name** and follow the steps below. If the product is published, then you can modify information as noted below. Note that the information is immediately published. To make changes, first unpublish the product and then republish the product after making the changes.

If no products are showing up on this screen, it is likely because the WEB_STATUS flag in the MTL_SYSTEM_ITEMS table is NULL. In order to see items in the Merchant UI, the items should have a web status of either UNPUBLISHED or PUBLISHED. To make an item show up in the Merchant UI, log in to Oracle Forms, choose the Inventory, Vision Operations (USA) responsibility, and go to Items > Master Items > Vision Operations. Choose F11 to enter a search query, then choose Ctrl-F11 to execute it. Use the Web Option tab to publish the desired item(s).

At the time of this writing, a bug in Oracle Inventory is preventing the Web Option tab in the Master Items form from working properly. As a temporary work around, log into the database via SQL*Plus as apps/apps, and update MTL_SYSTEM_ITEMS directly as follows:

```
UPDATE MTL_SYSTEM_ITEMS
SET WEB_STATUS='PUBLISHED'
WHERE INVENTORY_ITEM_ID=<ID number of the item to be published>
AND ORGANIZATION_ID=<Inventory Organization ID>;
```

The value for Inventory Organization ID should be the same as IBE_ITEM_VALIDATION_ORGANIZATION profile.

Note: Products in the published status are visible in the store. Be careful about changing information, as the changes go to the production system immediately. Oracle recommends that you unpublish the product, make changes, and then republish the product.

5. Optionally, enter or modify the short and long descriptions and click **Update**.
The descriptions are saved and are available to display in your store.
6. In the Products menu, choose **Hierarchy Paths**.
The Hierarchy Paths page displays the hierarchy of sections that have been set up for the store.
7. Optionally, you can remove or add parent sections for the part and edit the dates and display order.
8. In the Products menu, choose **Category and Display Styles**.
The Category and Display Styles page displays the category to which the product belongs and lists all display styles and any template names already assigned to the product.
9. Choose a template name for each display style that you want to use for the product. Answer **Yes** if you want the category level template used as the default rather than the product-level template for the display style. Click **Update**.
10. In the Products menu, choose **Multimedia Components**.
The Multimedia Components page lists all multimedia components you defined in the Setup tab.
11. Optionally, assign multimedia names to multimedia components and click **Update**.
12. In the Products menu, choose **Relationships**.
The Relationships page displays existing relationships between the product and other products or sections and the rules for those relationships, such as the product to show for an upsell or cross sell.
13. In the Product menu, choose **Specialty Store**.

The Specialty Store page lists the specialty stores where the product will be displayed. By default the product appears in those specialty stores to which the product **parent section** belongs.

14. Select the specialty stores that can display the product and then click **Update**.

Guidelines

Only products with a status of Published can be sold in your store.

Defining Category Mappings

Every product is assigned to a category in Oracle Inventory. Use this procedure to set up defaults for products belonging to the category. Note that in release 11.5.2 you can only specify defaults for categories belonging to a **primary display category** set - the value of IBE:category set profile. For other categories, select store-level defaults.

Prerequisites

- Products must be assigned to categories in Oracle Inventory in order to be returned upon a search of those categories.
- Multimedia, templates, and styles must exist before you can assign them to a category.
- IBE: Category set profile must be set

Steps

1. In the Category tab, conduct a search for categories.

The Categories page displays a list of item categories from Oracle Inventory which belong to the category set specified in the profile IBE: Category Set along with related templates, display styles, and multimedia components. A

2. Double-click the category name that you want to update.

The Templates Assigned page lists all template names and default source template files for the chosen category.

3. If you want to add a template, then perform the following steps:

- a. Click **Go**.

A list of available template names appears.

- b. Select a template.

- c. Click **Add**. The pop-up box closes when you select Done.
4. In the Category menu, choose **Display Styles**.
The Display Styles page lists all display styles you defined in the Setup tab. For each display style you can choose a template from the templates assigned.
5. Select **Update**.
Oracle iStore 11i selects the appropriate template based upon the following criteria:
 - Product-specific template for the given display style
 - If the product belongs to a category in **Primary Display Category Set**, then the defaults specified here are used. If you do not specify a mapping at the category level, then Oracle iStore 11i uses the store-level default.
 - If the product does not belong to any category in the above category set, then the store level defaults are used.
6. In the Category menu, choose **Multimedia Components**.
The Multimedia Components page lists all multimedia components you defined in the Setup tab.
7. Optionally, assign multimedia names to multimedia components and click **Update**.
When the multimedia component is used in a store page that relates to the product category, then the multimedia name retrieves the correct multimedia file according to the specialty store the customer is in and the customer's language. A product level multimedia component can override the category level multimedia component. The resolution rule is similar to those in display styles.
8. Choose the Category tab. The category set should be Inv.Items.

Defining Product Relationship Rules

Relationships are used for merchandising, for example, to offer a substitute product for a product that is out of stock. To speed your rule building process, use this procedure to relate products, categories, and sections to other products, categories, and sections. One relationship type can contain either rules created using the rule builder or one SQL rule. It cannot contain both.

Using SQL rules to define relationships by querying the database on particular fields is a tactic primarily used by consulting or other highly technical personal. Most store managers will use the mapping rules. One relationship name can contain either rules created using the rule builder or a SQL rule. It cannot contain both. To enter a SQL rule, choose SQL instead of Mapping in step 2 of the procedure and enter the rule.

The mapping rules define relationships in a from-to form. The type of from and to objects can be Item Categories (defined in Oracle Inventory), Sections or Hierarchies (defined in Oracle iStore 11i), or Items (defined in Oracle Inventory). For example, if you choose 3 objects in the from list and 2 objects in the to list, a total of six rules are built.

The application evaluates each mapping rule and inserts rows in a table that maintains the evaluated relationships. For example, if you have a category in your From list that has two products assigned, and have a section in your To list that has four products, then Oracle iStore 11i creates a total of eight product relationships.

Business needs drive the creation of the relationship rules. Oracle iStore 11i ships with the following seeded relationships:

- RELATED
- SUBSTITUTE
- CROSS_SELL
- UP_SELL
- SERVICE
- PREREQUISITE
- COLLATERAL
- SUPERSEDED
- COMPLIMENTARY
- IMPACT
- CONFLICT
- MANDATORY_CHARGE
- OPTIONAL_CHARGE
- PROMOTIONAL_UPGRADE

These seeded relationship types are also seeded in Oracle Inventory for Item Relationships. Unlike in Oracle iStore 11i, the user cannot add more relationship types in Inventory. If you use Oracle iStore 11i's Java API to retrieve related items given an item ID and a seeded relationship type, you will get related items defined in Oracle iStore 11i plus the ones defined in Inventory.

Prerequisites

Products must exist in Oracle Inventory.

Steps

1. Go to the Relationship tab; review the seeded relationships (note that only SUBSTITUTE is used by the store now).
2. Click SUBSTITUTE to create a relationship between items (for example, item B may be a substitute for item A if item A is out of stock).
3. Click **Create Rule**.
4. In the middle frame, search for the base product, and click the right arrow to add it to the From List.
5. Search for the related product, and click [à] to add it to the To List.
6. Click **Done** to save the relationship.
7. In the Relationship tab, choose **Create**.
The Create Relationship page appears.
8. Enter relationship information and choose **Create**.
Relationship detail page appears and you can choose to specify the pairs of related items by SQL query or by mapping rules. Choose an option and select **Create Rule**.
The Create Rules page appears.
9. If you chose **Create a SQL Rule**, go to step 10. If you chose **Create Mapping Rules** in step 2, conduct a search to view products, categories, or sections in the center table.
The search results appear in the table.
10. Select the items in your search results that you want to be in the *from* side of your rule, and click the left arrow.
The selected items appear in the From List.

11. Conduct a search to view products, categories, or sections in the center table.
This text appears only if you choose mapping rules option in step 2.

The search results appear in the table.

12. Select the items in your search results that you want to be in the *to* side of your rule, and click the right arrow.

The selected items appear in the To List.

13. Repeat as needed to complete your *from* and *to* lists for this rule.

14. Click **Done** to submit the relationship rule creation. The **Relationship Detail** page appears. The application generates a rule from every object in the *from* list to every object in the *to* list.

Or select **Preview Rules** to validate or exclude the relationship rules to be added. The Preview Rules Page appears. At this point the rules have not yet been added to the system. You may exclude any rules not needed. When finished click **Done** to see the **Relationship Detail** page.

15. From the **Relationship Detail** page, you may select a link for each rule to view all of the product to product relationships generated by that rule.

Or click **View All Results** to view all of the product to product relationships generated by all of the rules in this relationship type.

From either of the previous two options, the **Rule Results** page displays the product-level relationship results.

If you don't want to include one or more of the generated rules, then select **Exclude** and click **Update**.

The excluded product-level relationships can be re-included.

16. The following incomplete SQL displays:

```
Select msi.inventory_item_id  
From mtl_system_items msi  
Where
```

The SQL should return only the column `inventory_item_id` in `mtl_system_items` table. You can add as many tables as you want in the From list and add any conditions in the Where clause.

Example: Pricing Setup (Defining Simple Freight Charges)

This section describes an example of how to set up simple fixed-amount freight charges in the store.

1. Go to Oracle Forms, log in as SYSADMIN / SYSADMIN, choose the **Oracle Pricing Manager** responsibility.
2. Go to **Modifiers > Define Modifier**. (If you do not have the Oracle Pricing Manager responsibility, grant it to the SYSADMIN user first.)
3. Choose **Type**.
4. Click **Freight** and **Special Charge List**.
5. Enter an identifier for the modifier in the Number field (for example, IBEFR01).
6. Enter a description in the Name field (for example, IBE Freight Modifier).
7. Click **Modifiers Summary Modifier No**, and enter a number (for example, 1).
8. Click **Level**, choose modifier level (for example, Line).
9. Click **Modifier Type**, and choose **Freight/Special Charge**.
10. Enter a **Start Date** and an optional **End Date**.
11. Scroll right. Click in **Pricing Phases**, and choose **Line Charges**.
12. Click the Discounts / Charges tab.
13. Click in **Charge Name**, and choose **Freight Costs**.
14. Click **Application Method**, and choose **Amount**. Enter a per-line freight cost dollar amount (for example, 3.00).
15. Save the form.

Setting Up Product Searches

Oracle iStore 11i's product search feature provides merchants with ability to search for products they want to sell. Oracle InterMedia is used to implement the product search feature. Product information (description and long description) is first loaded into an Oracle iStore 11i table via a concurrent program. Perform this step after inventory setup is complete. Once the data is loaded, any changes to the product information in Oracle Inventory is updated in the Oracle iStore 11i table through a database trigger on the inventory table. This keeps product information current in the search database. Once the data is moved into the search table, an iMedia index is created to facilitate keyword search.

Note: You must ensure that both Oracle Inventory and Oracle InterMedia are installed and configured properly before setting up store search. Refer to the Oracle InterMedia documentation for details on how to set up and configure InterMedia.

What Information Do You Store in the Search Table?

The `ibe_ct_imedia_search` is basically a denormalized table of `mtl_system_items_tl` and `mtl_item_categories`.

The core text on which you search for is stored in a clob called `indexed_search`. Currently it stores a concatenation of name and description of products. The table also stores `inventory_item_id`, `organizationId`, `category_id`, `category_set_id` and the web status field from `mtl_system_items_b` table.

What Attributes of a Product Can I Search for?

Name and description of a product. They are stored as `description` and `long description` columns in `mtl_system_items_tl` table.

Does Search Have any Dependencies?

Product search needs version 8.1.6 of the database with the `intermedia` option correctly installed. It also requires the 11i version of the inventory schema.

First the merchant sets up the inventory, under a common master org id. Points to remember while setting up inventory include:

- Give products unique names
- Do not leave category names (concatenated segments) blank or non-unique, it can be null or non-unique in the database but will show as blank / multiple times in the categories list of values in your customer home page

Make sure the products have the web enabled flag under web options set to `PUBLISHED`. You can also query this field by examining the `web_status` column of the item.

The next major step is to run the concurrent program to populate the search table with data and to create the intermedia text index.

Populating the Oracle iStore 11i Search Table

To populate the search table, use the following procedure.

Steps

1. Log on to Oracle Forms as SYSADMIN / SYSADMIN.
2. Select the **Oracle iStore 11i Concurrent Programs Manager** responsibility (as always, if you don't have this responsibility, use the System Administrator responsibility to grant it to yourself).
3. In the pop-up window, choose Single Request, and click OK.
4. Click the list of values in the Name field, and select **iStore Search Insert**.
5. Click **Submit** to start the concurrent request. Note the request ID.

You can monitor the progress of your request by looking at the request log and output files in \$COMMON_TOP/admin/log/l<request ID>.log and \$COMMON_TOP/admin/out/o<request ID>.out, respectively.

You can also view the request status by selecting View Requests, and searching by the request ID.

Note: You will only be able to search for products whose WEB_STATUS is 'PUBLISHED'.

This process can take substantial amount of time (depending on number of items you have). As an estimate for about 300 thousand items in inventory, this program can take about forty-minutes minutes to run.

The concurrent manager will call IBEVCSMV.sql script, which will move the product data from the inventory table to Oracle iStore 11i's search table. When this job is running, the search tables are purged and product search will not work correctly on the store. Since this batch job deletes data from the search table, the rollback segment should be large enough for the process to complete.

Once the request is complete, you will be able to search for products based on name and description. If additional product attributes are to be added in the search, this SQL script needs to be modified to add the extra search column.

Setting Up Search Profile Options

Search also needs 2 profile options to be setup:

Profile Option	Description
Search Lines Per Page	This option sets the number of lines to be displayed per page. If this profile option is not set, the code default of 20 lines will be used.
Total Search Results	This option sets the maximum cap on the search results. For example, if the user searches for a very common keyword (not in the stop words list), then the search process will stop after the max cap set as per this profile option. If this profile option is not set, then the code default of 500 results will be used.

Creating Search Index Tables

In order to be able to run external procedures to create a search index table, please ensure that ENV\$ is included in your SID_DESC part of listener.ora as follows:

```
Go to 8.1.6 ORACLE_HOME:

cd /u02/visappl
ksh
. ./APPSORA.env
cd $ORACLE_HOME/./8.1.6/network/admin
```

This directory should contain listener.ora. Verify that listener.ora contains the following:

```
(SID_DESC =
  (SID_NAME = PLSExtProc)
  (ORACLE_HOME = /u04/visora/8.1.6)
  (ENV$ = LD_LIBRARY_PATH=/u04/visora/8.1.6/ctx/lib)
  (PROGRAM = extproc)
)
```

Before Creating the search index table, please make sure that the Oracle ConText server is up. Use the following command to check:

```
$ ps -ef | grep ctxsrv
```

If it doesn't exist, start the Oracle ConText server as follows:

```
Go to 8.1.6 ORACLE_HOME:

cd /u02/visappl
```

```
ksh
. ./APPSORA.env
cd $ORACLE_HOME/./8.1.6
. ./VIS.env
```

This will setup 8.1.6 ORACLE_HOME env. Run the following command:

```
ctxsrv -user ctxsys/ctxsys&
```

PL/SQL and Java/ JSPs Used in Search

The following program units are used in product search process and can be modified to add other attributes in product search.

- java/catalog/Search.java (main java program that executes the query)
- ibeckr3.jp (Search result jsp)
- ibezmenu.jsp (Main homepage)
- ibeckrf.jsp (Search result jsp)
- IBEVCSMV.sql (one time load of product descriptions through concurrent manager)
- IBEVCSKS.pls (package specification)
- IBEVCSKB.pls (package body)
- IBEVIDTS.pls (package specification for the main database trigger)
- IBEVIDTB.pls (package body for the main database trigger)
- java/catalog/PrdRec.java (definition of search result object)

Customizing Search

If you need to add more attributes of the item to search for, you will need to modify IBEVCSMV.sql for the initial load and the PL/SQL triggers mentioned above to make sure that updates to these attributes get propagated to the search table.

1. Modify the search package (IBEVCSKS.pls and IBEVCSKB.pls) for adding the additional product search attributes. By default, only product name (description column) and the product description (long description column) are included in the search.

If additional attributes are to be added in the product search, the parameters for the package specification and body will have to be changed accordingly, with

the new attributes. This package moves the subsequent changes in the product information, to Oracle iStore 11i's search table. Any insert / delete / update on MTL_ITEM_CATEGORIES, any delete / update on MTL_SYSTEM_ITEMS_B and any delete / update / insert on MTL_SYSTEM_ITEMS_TL table will move the change to the search table through this procedure. This procedure is called from the main database trigger procedures, as explained in the next step.

2. If new parameters are added to the search package in step 1 above, the call to the search package will have to be modified in the main database trigger package IBEVIDTB.plb. This package body calls all the iStore's ERP related db trigger procedures, including the search package procedures. The call to search package will have to be modified to include the new parameters added to the search move procedures.
3. The database trigger on the product tables will call the main db trigger package, as described in the step 2, to move the product data change to the iStore search table.

Step 2, however, will not recreate the iMedia index to include the changed information in the search table. Administrators must rebuild the iMedia search index every time a new product is added or an existing product is changed / deleted. This can be achieved by recreating the iMedia index IBE_CT_IMEDIA_SEARCH_IM through OEM utility or through executing the command "ALTER INDEX IBE_CT_IMEDIA_SEARCH_IM REBUILD ONLINE PARAMETERS ('sync');" in SQL*Plus. Note that you should have privileges to alter the iMedia index. After this step, the modified product information will be visible in the product search process of iStore.

Adding Stopwords to Searches

There are many search words like "and," "oracle," "if," "then," etc. which are very common and will return numerous search results. Search results may not be relevant to the user's query if such common search keywords are used. In addition, using common keywords in searches uses processing resources and will slow down performance. These common keywords can be excluded from the search by using the "Stop Words" utility in iMedia.

Log into OEM as CTXSYS to see the stop words in the Stop List. Additional stop keywords can be added to the stop list.

This topic group provides process-oriented, task-based procedures for using the application to perform essential business tasks.

Assigning Templates, Multimedia, and Styles to Categories

Every product is assigned to a category in Oracle Inventory. Use this procedure to assign objects that you want to automatically display for your store customers when they link to particular categories.

When the multimedia component is used in a store page that relates to the product category, then the multimedia name retrieves the correct multimedia file according the specialty store the customer is in and the customer's language. A product level multimedia component can override the category level multimedia component.

Prerequisites

- Products must be assigned categories in Oracle Inventory.
- Multimedia, templates, and styles must exist before you can assign them to a category.

Steps

1. In the Category tab, conduct a search for categories.

The Categories page displays a list of item categories from Oracle Inventory along with related templates, display styles, and multimedia components.

2. Double-click the category name you want to update.

The Templates Assigned page lists all template names and default source template files for the chosen category.

3. If you want to add a template, then perform the following steps:

- a. Click **Go**.

A list of available template names appears.

- b. Select a template.

The Templates Assigned page displays the file for all specialty stores and languages and the description for the selected template name.

- c. Click **Add**.

4. In the Category menu, choose **Display Styles**.

The Display Styles page lists all display styles you defined in the Setup tab.

5. Optionally, assign template names to display styles and click **Update**.

When the display style is used for a store page that relates to the product category, then the template name retrieves the correct template file according to the specialty store the customer is in and the customer's language. A product level display style can override the category display style.

6. In the Category menu, choose **Multimedia Components**.

The Multimedia Components page lists all multimedia components that you defined in the Setup tab.

7. Optionally, assign multimedia names to multimedia components and click **Update**.

Testing The Store

To test the storefront, apply your site's path values to the following URL:

`http://<host>:<apache port>/html/ibezhome.jsp?minisite=<minisite ID>`

Where <minisite ID> is the ID of the specialty store created above. Try browsing the catalog and placing an order.

IBE Template Mappings

The following table is a list of seeded template filenames, with programmatic access names and descriptions.

IBE Template Mappings

Access Name	JSP Page	Description
STORE_SCART_GET_SHAREE_P	ibescsr.jsp	Retrieve sharer shopping cart process page
STORE_SCART_SAVE	ibescsch.jsp	Save shopping cart
STORE_SCART_SAVE_P	ibescscr.jsp	Save shopping cart process page
STORE_SCART_SHARE_MERCHANT	ibescsmh.jsp	Share cart with merchant
STORE_SCART_SHARE_MERCHANT_P	ibescsmr.jsp	Share cart with merchant process page
STORE_SCART_SAVE_SHAREE	ibescssh.jsp	Save sharer shopping cart
STORE_SCART_SAVE_SHAREE_P	ibescssr.jsp	Save sharer shopping cart process page
STORE_REGISTER_USER	ibescuih.jsp	Register new user
STORE_SCART_VIEW	ibescvch.jsp	View shopping cart
STORE_SCART_VIEW_P	ibescvcr.jsp	View shopping cart process page
STORE_SCART_WARN1	ibescwha.jsp	Shopping cart warning page
STORE_SCART_WARN2	ibescwhb.jsp	Shopping cart warning page
STORE_SCART_BLURB	ibesciah.jsp	Shopping cart blurb

STORE_SCART_ADD_ONE_ITEM	ibescabh.jsp	Add Item to Cart Page for one item
STORE_SCART_ADD_ITEM	ibescalh.jsp	Add Item to Cart Page
STORE_SRCH_RESULT	ibecskr3.jsp	Show search results
STORE_SRCH_REFINE	ibecskrf.jsp	Refine search criteria
STORE_SHLIST_ERROR	ibeslwar.jsp	Show shopping list errors
STORE_SCART_DIRECT_ENTRY	ibescedh.jsp	Enter customer inventory items to cart - direct entry
STORE_SCART_DIRECT_ENTRY_P	ibescedp.jsp	Process direct entries
STORE_SHLIST_DISPLAY_LIST	ibesldlh.jsp	Display shopping lists
STORE_SHLIST_DISPLAY_LIST_P	ibesldhp.jsp	Process shopping lists
STORE_SHLIST_DISPLAY	ibesldld.jsp	Display shopping list
STORE_SHLIST_DISPLAY_P	ibesldlp.jsp	Process shopping list
STORE_SHLIST_ADD_ITEM	ibeslald.jsp	Add items to a list
STORE_SHLIST_ADD_ITEM_P	ibeslalp.jsp	Process addition of items to list
STORE_SHLIST_SAVE	ibeslcla.jsp	Save a shopping list
STORE_SHLIST_SAVE_P	ibeslclp.jsp	Process saving of shopping list
STORE_USR_CC_CREATE	ibeaccr.jsp	Create user credit card
STORE_USR_CC_DELETE	ibeaccdl.jsp	Delete user credit card
STORE_USR_CC_EDIT	ibeacced.jsp	Edit user credit card
STORE_USR_CC_EXPIRATION	ibeaccex.jsp	User credit card expiration
STORE_USR_CC_PRIMARY	ibeaccpr.jsp	User credit card primary
STORE_USR_CC_DISPLAY	ibeaccv1.jsp	User credit card display
STORE_USR_ACC_HOME	ibeahome.jsp	User account home
STORE_USR_ACC_LOGIN	ibealgn1.jsp	User account login
STORE_USR_ACC_LOGIN_AUTH	ibealgn2.jsp	User account authentication
STORE_USR_ACC_SELECT	ibealgn3.jsp	User account selection
STORE_USR_ACC_SELECTED	ibealgn4.jsp	User account selected
STORE_USR_PMTBK_DISPLAY	ibeapmtb.jsp	Displays user payment books
STORE_USR_PMTBK_SAVE	ibeapbsv.jsp	Save user payment books

STORE_USR_SIGNIN	ibeasgnn.jsp	Store user sign in
STORE_CONFIG_BRANCH	ibefbrch.jsp	Process a configurator request
STORE_CONFIG_LAUNCH	ibefcnfg.jsp	Launch the configurator
STORE_CONFIG_REDIRECT	ibefredr.jsp	Redirect request to the configurator
STORE_CONFIG_FRAME_SRCS	ibeffrms.jsp	Provide frame specification to the configurator
STORE_CONFIG_DISPLAY	ibefdisp.jsp	Include page to display a configured item
STORE_ORDER_XPR_SETTINGS_DISPLAY	ibeo1set.jsp	Display express checkout settings
STORE_ORDER_XPR_SETTINGS_SAVE	ibeo1sav.jsp	Save express checkout settings
STORE_ORDER_XPR_TAG_AREA	ibeo1tag.jsp	Display the area around the checkout tag
STORE_ORDER_XPR_PROCESS	ibeo1dob.jsp	Process the express checkout request
STORE_ORDER_XPR_VIEW	ibeo1vew.jsp	View an express checkout order
STORE_ORDER_XPR_CANCEL	ibeo1ccl.jsp	Cancel an express checkout order
STORE_ORDER_XPR_ERROR	ibeo1err.jsp	Error during express checkout
STORE_ORDER_PAYMENT_OPTS_INCLUDE	ibeopmto.jsp	Include page for displaying payment options
STORE_PSI_ORDER_DETAIL	ibeiodtl.jsp	Show order detail
STORE_PSI_ORDER_SUMMARY	ibeiohdr.jsp	Order search results page
STORE_PSI_ITEM_AVAILABILITY	ibeicatp.jsp	Item availability report
STORE_PSI_SHIPMENT_DETAIL	ibeiship.jsp	Shipment details
STORE_PSI_INVOICE_SUMMARY	ibeiinvh.jsp	Invoice summary
STORE_PSI_INVOICE_DETAIL	ibeiinvd.jsp	Invoice details
STORE_PSI_PAYMENT_SUMMARY	ibeipmth.jsp	Payment summary
STORE_PSI_PAYMENT_DETAIL	ibeipmtd.jsp	Payment summary
STORE_PSI_INVOICE_PAYMENT_DETAIL	ibeiivpd.jsp	Invoice payment details

STORE_PSI_SIDE_NAVBAR	ibeiside.jsp	Post sales navigation bar
STORE_PSI_ATP_HEADER	ibeiatph.jsp	Display ATP headers
STORE_PSI_ATP_REDIRECT	ibeiatpr.jsp	ATP Redirect page
STORE_PSI_ADDRESS_DISPLAY	ibeiatdr.jsp	Address display page
STORE_SCART_LIST1	ibesccar.jsp	Convert Cart to List 1
STORE_SCART_LIST1_P	ibesccbr.jsp	Convert Cart to List 1 Process Page
STORE_SCART_LIST2	ibesccla.jsp	Convert Cart to List 2
STORE_SCART_LIST2_P	ibescclb.jsp	Convert cart to list 2 process page
STORE_SCART_MODIFY	ibescdch.jsp	Modify shopping cart
STORE_SCART_MODIFY_P	ibescdcr.jsp	Modify shopping cart process page
STORE_SCART_VIEW_LIST	ibescdlh.jsp	View shopping cart lists
STORE_SCART_VIEW_LIST_P	ibescdlr.jsp	View shopping cart lists process page
STORE_SCART_VIEW_DISCOUNT	ibescpch.jsp	View shopping cart discount
STORE_SCART_GET_SHAREE	ibescrsh.jsp	Retrieve sharer shopping cart
STORE_USR_ACC_LOGOUT	ibealgot.jsp	User account logout
STORE_USR_ACC_SUBMENU	ibeamenu.jsp	User account submenu
STORE_USR_CREATE	ibebcrus.jsp	User creation
STORE_USR_EDIT	ibebmous.jsp	User edit
STORE_USR_ACC_CREATE	ibebcrac.jsp	User account creation
STORE_USR_ACC_EDIT	ibebmoac.jsp	User account edit
STORE_USR_ADMIN_ROLE	ibebrole.jsp	Role Management
STORE_USR_ADMIN_PERM	ibebperm.jsp	Permission Management
STORE_USR_ADMIN_USER	ibebuser.jsp	User Management
STORE_USR_ADMIN_ACC	ibebusac.jsp	User Account Management
STORE_USR_ADMIN_ADD_ACC	ibebadac.jsp	Add User Account
STORE_USR_ACC_PASSWORD	ibeapwd.jsp	User Password

STORE_USR_ACC_WELCOME	ibeawlcm.jsp	User welcome bin
STORE_USR_ADMIN_ROLE_W	ibebwrol.jsp	Role Management
STORE_USR_ADMIN_PERM_W	ibebwprm.jsp	Permission Management
STORE_USR_ADMIN_USER_W	ibebwusr.jsp	User Management
STORE_USR_ADMIN_ACC_W	ibebwusa.jsp	User Account Management
STORE_USR_ADMIN_ADD_ACC_W	ibebwada.jsp	Add User Account
STORE_CTLG_BUY_ROUTE	ibecbrte.jsp	Catalog - Buy Routing
STORE_CTLG_BROWSE	ibecbrws.jsp	Catalog - Hierarchy Browse
STORE_CTLG_DISPLAY_ROUTE	ibecdrte.jsp	Catalog Display Routing
STORE_CTLG_DISPLAY_ROUTE_INCLUDED	ibecdrti.jsp	Catalog - display routing page included within another page
STORE_CTLG_ITEM_RELATION	ibecirel.jsp	Catalog - Item Relationship
STORE_CTLG_ITEM_ROUTE	ibecirte.jsp	Catalog - Item Routing
STORE_CTLG_PATH	ibecpath.jsp	Catalog - Path
STORE_CTLG_FEATURED_SECTION	ibecfsct.jsp	Catalog - featured section
STORE_CTLG_LEAF_SECTION_MULTIPLE	ibeclfsm.jsp	Catalog - leaf section allowing to add multiple products to the shopping cart
STORE_CTLG_LEAF_SECTION_SINGLE	ibeclfss.jsp	Catalog - leaf section allowing addition of single product to the shopping cart
STORE_CTLG_DRILL_DOWN_LEAF	ibecddlf.jsp	Catalog - drill down to the leaf sections
STORE_CTLG_FEATURED_SECTION_INCLUDED	ibecfsti.jsp	Catalog - featured section included within another page
STORE_CTLG_LEAF_SECT_MULTIPLE_INCLUDED	ibecslmi.jsp	Catalog - leaf section allowing addition of multiple products to the shopping cart included in another page
STORE_CTLG_LEAF_SECT_SINGLE_INCLUDED	ibecslsi.jsp	Catalog - leaf section allowing addition of single product to the shopping cart included in another page

STORE_CTLG_DRILL_DOWN_LEAF_INCLUDED	ibecdli.jsp	Catalog - drill down to the leaf sections included in another page
STORE_CTLG_FEATURED_ITEM	ibecfitm.jsp	Catalog - featured product
STORE_CTLG_LEAF_ITEM	ibeclitm.jsp	Catalog - leaf product allowing addition of single product to the shopping cart
STORE_CTLG_ITEM_DETAIL	ibecitmd.jsp	Catalog - product detail
STORE_ADDR_BOOK	ibeaadbk.jsp	Show address book entries
STORE_ADDR_CREATE	ibeaadcr.jsp	Create address book entry
STORE_ADDR_DELETE	ibeaaddl.jsp	Delete an address book entry
STORE_ADDR_EDIT	ibeaaded.jsp	Edit an address book entry
STORE_ADDR_PRIMARY	ibeaadpr.jsp	Display the primary address of the customer
STORE_ADDR_DISPLAY	ibeaadv1.jsp	Display the address book
STORE_ADDR_DISPLAY_ALT	ibeaadv2.jsp	Display the address book
STORE_ADDR_SELECT	ibeaads1.jsp	Select a particular address
STORE_ORDER_SHIP_HEADER	ibeoshdr.jsp	Display the ship header
STORE_ORDER_INVOICE_HEADER	ibeobhdr.jsp	Display invoice header
STORE_ORDER_CONFIRM	ibeocnfm.jsp	Confirm an order
STORE_ORDER_SHIP_LINE	ibeoslne.jsp	Display order ship line
STORE_ORDER_EDIT	ibeoedit.jsp	Edit an order
STORE_ORDER_EDIT_P	ibeoedtd.jsp	Edit an order process page
STORE_ORDER_SHIP_LINE_P	ibeoslnl.jsp	Process a ship line
STORE_ORDER_SHIP_HEADER_P	ibeoshdd.jsp	Process a ship header
STORE_ORDER_INVOICE_HEADER_P	ibeobhdd.jsp	Process an invoice header
STORE_ORDER_SUMMARY	ibeosumm.jsp	Display order summary
STORE_ORDER_CONFIRM_P	ibeocnfd.jsp	Process an order confirmation

JSP Customizing Standards

JSP Standards (Customer Facing)

Internal Oracle documentation for the store coding standards is listed below.

Note: Never modify an original JSP. Make a copy of the original JSP and modify only the copy. If a bug occurs, compare the JSP copy to the JSP original.

----- requested jsp, forwarded jsp -----

Top of JSP Page

```
<!-- $Header: standards.html 115.3 2000/06/22 19:04:06 jnath noship $ -->
<% int pc = PageContext.REQUEST_SCOPE; %>
<% pageContext.setAttribute("_permission", "<permission>", pc); //WA %>
<% pageContext.setAttribute("_guestNotAllowed", "true", pc); //WA %>
<% pageContext.setAttribute("_B2CNotAllowed", "true", pc); //WA %>
<% pageContext.setAttribute("_sensitivePage", "true", pc); //WA %>
<%@include file="ibezhdr.jsp" %>
<% pageContext.setAttribute("_pageTitle",
AOLMessageManager.getMessageSt("<prompt>"), pc); %>
<%@include file="ibeztop.jsp" %>
<% // preferably do all processing before ibezmenu %>
<%@include file="ibezmenu.jsp" %>
<% // display rest of the page %>
```

WA: Wherever Applicable

Bottom of JSP Page

(Wherever execution ends)

```
<%@include file="ibezbot.jsp" %>
```

----- included jsp -----

Top of JSP Page

```
<!-- $Header: standards.html 115.3 2000/06/22 19:04:06 jnath noship $ -->
```

Bottom of JSP Page

```
<!-- ibexyyyy.jsp end -->
```

Hyperlink

```
<%= DisplayManager.getURL(<accessName>) %> OR  
<%= DisplayManager.getURL(<accessName>, <query string>) %> OR  
<%= RequestCtx.getURL(<fileName>) %> OR  
<%= RequestCtx.getURL(<fileName>, <query string>) %>
```

File Name (for JSP Include, JSP Forward, Form Action)

```
<%= DisplayManager.getTemplate(<accessName>).getFileName() %>
```

Form

```
<%= RequestCtx.getSessionInfoAsHiddenParam() %>
```

Transactions + Database Connections

Transactional APIs

```
try {  
    TransactionScope.begin(Thread.currentThread());  
    transactionalApi();  
    readonlyApi();  
    TransactionScope.end(Thread.currentThread());  
}  
catch (CustomerException e) { // catch exceptions that can be handled  
    TransactionScope.setRollbackOnly();  
    TransactionScope.end(Thread.currentThread());  
    // handle exception (possibly a jsp:forward)  
}
```

```

    public static void transactionalApi()
        throws FrameworkException, CustomerException, SQLException
    {
        OracleConnection conn = (OracleConnection)
TransactionScope.getConnection();
        Statement          stmt = ... ;
        ...
        try {
            stmt.execute();
            ...
        }
        finally {
            if (stmt != null) stmt.close();
        }
    }

```

for readonly APIs: [red font - jsp page; blue font - java api]

```

    try {
        readonlyApi();
        readonlyApi();
    }
    catch (CustomerException e) { // catch exceptions that can be handled
        // handle exception
    }

    public static void readonlyApi()
        throws FrameworkException, SQLException
    {
        OracleConnection conn = (OracleConnection)
TransactionScope.getConnection();
        Statement          stmt = ... ;
        ...
        try {
            stmt.execute();
            ...
        }
        finally {
            if (stmt != null) stmt.close();
            TransactionScope.releaseConnection(conn);
        }
    }

```

Exception Handling

User Errors

Raise a subclass of `FrameworkException` in the API and handle it in the calling jsp page.

Fatal Errors

Raise `FrameworkException` or let the original exception (for example, `SQLException`) propagate to `ibezerr.jsp`.

API Standards

Package Naming

Customer Side

`oracle.apps.ibe.<sub-component>`

or

`oracle.apps.ibe.<sub-component>.<sub-sub-component>`

Merchant Side

`oracle.apps.ibe.setup.<sub-component>`

Load Method(s)

Each class will have static load method(s) given a `recordId`. It gets the record from the database / cache, the API returns the record object, caches the record object (for certain classes), deep load or shallow load (for certain classes).

For example,

```
Product prd = Product.load(productId, "DEEP");  
Account acc = Account.load(accountId);
```

Accessing Member Variables

When a record object is loaded, the member variables can be accessed directly. for example, `acc.accountId` or they can be accessed using get methods. For example, `prd.getProductId()`.

Get Methods

Get methods are not mandatory, except for translatable attributes. It will return the value in appropriate language, for example, `prd.getDescription()`.

Standard Parameters

Standard parameters (userid, lang code, etc.) will be available statically from `RequestCtx`, for example, `RequestCtx.getUserId()`.

Passing Values Across Templates

Use the following guidelines for passing values across templates.

pageContext.setAttribute

```
pageContext.setAttribute("_pageTitle",
    "iStore Framework",
    PageContext.REQUEST_SCOPE);
```

pageContext.getAttribute

```
pageContext.getAttribute("_pageTitle",
    PageContext.REQUEST_SCOPE);
```

Example Parameters

```
<%@ include file="jtfincl.jsp" %>

<% int pc = PageContext.REQUEST_SCOPE; %>
<% pageContext.setAttribute("_guestNotAllowed", "true", pc); // #1
<% pageContext.setAttribute("_B2CNotAllowed", "true", pc); // #2
<% pageContext.setAttribute("_permission", "<permission>", pc); // #3
<% pageContext.setAttribute("_sensitivePage", "true", pc); // #4
<%@ include file="ibezhdr.jsp" %>

<% pageContext.setAttribute("_pageTitle", "iStore Framework", pc); %>
<%@ include file="ibeztop.jsp" %>

pageContext.setAttribute("selectedTab", "0", pc); // #5
pageContext.setAttribute("selectedMenu", "0", pc); // #6
pageContext.setAttribute("selectedSide", "0", pc); // #7
<%@ include file="ibezmenu.jsp" %>

Hello World
```

```
<%@ include file="ibezbot.jsp" %>
```

Example 2 Notes

Note	Description
# 1	- Ensures that only logged in users can access the page and see the “Hello World” message. - Anonymous users are asked to sign in or register.
# 2	Ensures that only registered business customers can access the page.
# 3	- To check permissions, for example, if foo.jsp is used to create an address, then <permission> is CREATE_ADDRESS. - Oracle iStore 11i ships with predefined permissions that you can use to restrict the access of B2B users to a page. For B2C users, Oracle iStore 11i does not check permissions, and instead checks the _B2CNotAllowed flag.
# 4	- Controls access to sensitive pages, for example, users’ personal information such as email address, credit card number, and so on. - The product catalog and view shopping cart pages are non-sensitive pages. When users go from a non-sensitive page to a sensitive page, they are re-authenticated. - Sensitive pages should be SSL enabled. In general, all _guestNotAllowed pages are sensitive pages.
# 5, # 6, # 7	Specifies which tab, subtab, and side menu to highlight, starting from 0. For example, see ibezmenu.jsp.

Passing Parameters/Cookies

To pass parameters /cookies, use the following guidelines.

See [Example 2](#), and replace “Hello World” with the following line:

```
<ahref=<%=DisplayManager.getURL("STORE_HOME")%>>Go to iStore Home</a>
```

Display Manager then calls RequestCtx.getURL(<jspFileName>) to append the cookie string to the jsp file name if necessary.

To pass more parameters, use

```
DisplayManager.getURL (<accessName>, <query strong>) or
```

```
RequestCtx.getURL (<jspFileName>, <query string>).
```

Where the query string is of the form: “foo 1=bar1&foo2=bar2&foo3=bar3”.

In a form, the cookie string is passed as a hidden field, for example,

```
RequestCtx.getSessionInfoAsHiddenParam()
```

To access a physical file without a cookie, use

```
DisplayManager.getTemplate (<access name>) .getFileName()
```

Changing API Documentation

To make changes to the UI or page displays, you must have complete knowledge of the APIs being called from the JSP page. These APIs are the key for displaying data on the store pages. These are the application objects/beans. Customers and users cannot modify these class files.

Modifying the User Interface

The Oracle iStore 11i page is sectioned into various information containers, referred to as *bins* or place holders. These bins hold the content-specific information and display it logically on the page. To change the look and feel of the UI, you can modify the text in the bin and the layout of the bins.

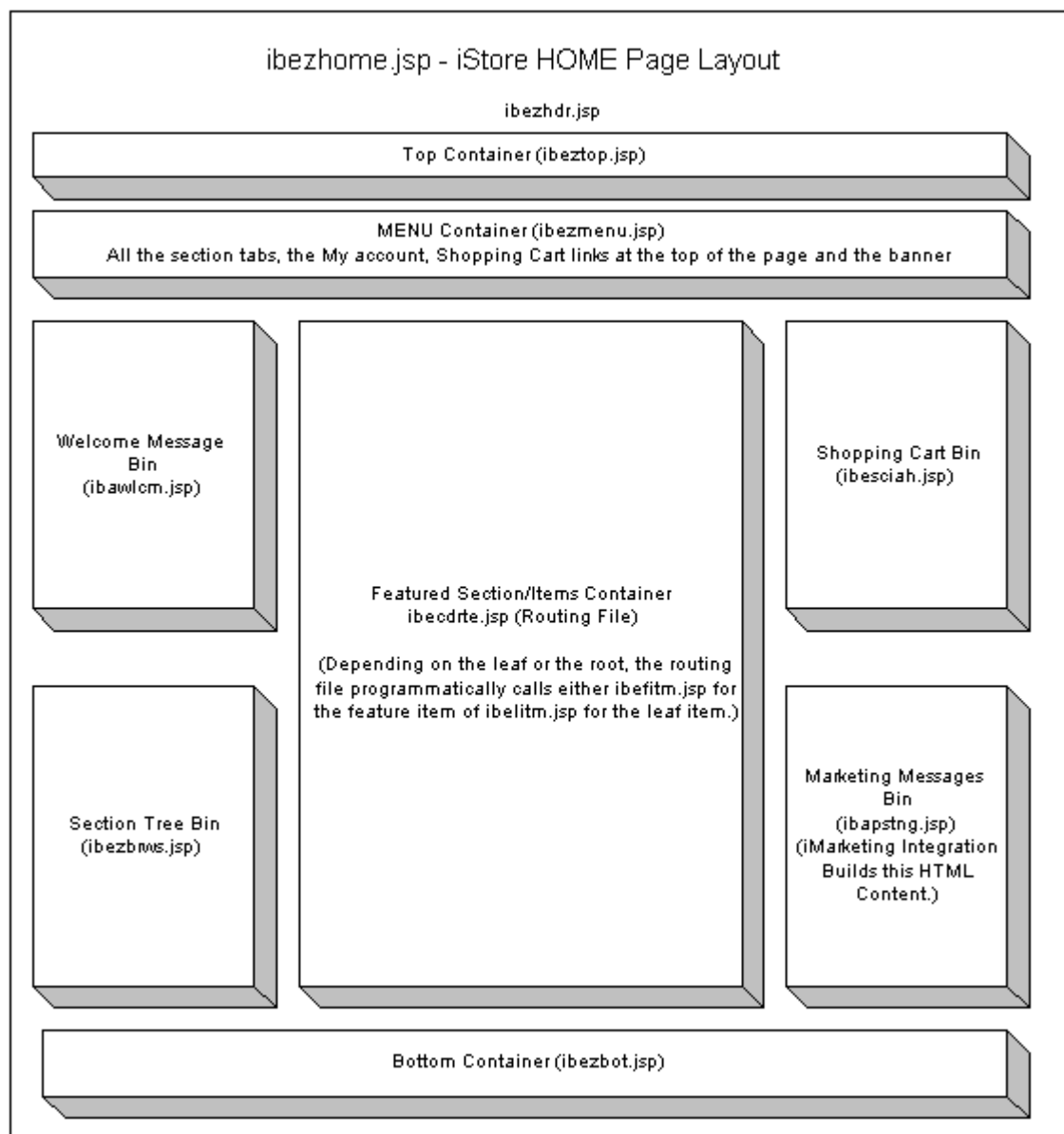
Changing the Text in Bins

The text of the bins (for example, Welcome Message Bin, Shopping Cart Bin, Section Tree Bin) comes from the Messages stored in ERP. To change the text, log in to ERP Forms as Application Developer responsibility, choose the Messages menu option, and search for ‘IBE%’ messages. You can modify these messages to change the text in the bin. You can find a specific message name by viewing the respective bin JSP file.

Changing the Layout of Bins

You can either change the bin placement or remove a bin from the UI by modifying the home page ibezhome.jsp and other corresponding JSP files. Take special care when making these changes, which affect the HTML.

The following diagram shows a layout of the bins on the store page. The layout is the default or standard layout that comes with the plain install of the product.



Using Forms/JavaBeans with JSPs

To use forms / JavaBeans with Java Server Pages, use the following procedures.

Prerequisites

None.

Steps

1. Create an HTML form in a JSP template to process user input.
2. Perform basic checks on the input that the user enters, for example, using JavaScript.
3. Pass the data to a bean that implements the business logic.

The bean processes the input (it can maintain a persistent state), and returns any results back to the user as HTML.

Using JavaBeans with JSPs

To use JavaBeans with Java Server Pages, use the following guidelines.

<jsp:useBean>

Instantiate or locate the JavaBean instance

```
<jsp:useBean id="mybean" scope="session"
class="hello.nameHandler" />
```

<jsp:setProperty>

Set property values in the JavaBean.

```
<jsp:setProperty name="mybean" property="*" />
```

<jsp:getProperty>

Get property values from the JavaBean.

```
<jsp:getProperty name="mybean" property="username" />
```

Adding Error Pages

To add error pages, use the following procedure.

1. Write the JavaBean (or servlet or other component) so that it throws certain exceptions under certain conditions.
2. In the JSP file, use a page directive with `errorPage` set to the name of a JSP file that will display a message to the user when the exception occurs.
3. Write an error page file using a page directive with `isErrorPage="true"`. For example, see `ibezerr.jsp`.
4. In the error page file, use the exception object to get information about the exception.
5. Give informative messages in the error file or the JSP file on the error.

Handling Runtime Exceptions

To handle runtime exceptions, use the following guidelines.

User Errors

To handle user errors, raise a subclass of `FrameworkException` in the API and handle it in the calling jsp page.

Fatal Errors

To handle fatal errors, raise `FrameworkException` or let the original exception (for example, `SQLException`) propagate to `ibezerr.jsp`.

Glossary

B2B user

A user who registers as Business user. A B2B user gets additional privileges to manage other users.

Blueprint phase

A phase of a project in which the business plan around the solution is developed, the solution is defined, business and system integration requirements are modeled, and the partitions of tasks are defined.

Call-Me-Back link

Allows customers to request telephone assistance from a sales representative.

Checkout

Checkout takes customers to the first in a sequence of checkout pages. Moving through the process, customers enter their shipping address, choose a shipping method, enter payment information and billing address, then review and place their order.

Complimentary sell

Complementary sells persuade customers to buy a product that complements a product in which they are interested.

Construction phase

The construction phase of a project is where the solution is built. It involves refining the data and functional models, the physical application modules, and the system interfaces until they meet the business requirements.

ConText

See Intermedia.

Context-sensitive advertising

Context-sensitive advertising means that advertisements are targeted to appear along with certain content on the web site.

Cookie

Cookies are general mechanisms which server side connections use to both store and retrieve information on the client side of the connection.

Cross-sell

See Complimentary sell.

Demilitarized zone (DMZ)

Derived from the term demilitarized zone, it is the area between two firewalls which is isolated from both un-trusted and trusted networks.

Deployment system

The basis of the online store Web pages, deployment system is the store that is made accessible to the outside world.

Display Style

A way of displaying products. For a given product and a given display style the rendering is implemented in the template associated for that display style to the product.

End-to-end customizing

End-to-end customizing affects every aspect of the system. New pages and new fields within existing pages must be created on the front end, new code implementing the logic behind those new pages has to be written on the back end.

Firewall

A firewall controls the type of traffic entering a site. Internet traffic should be limited to connections originating from browser-based HTTP sessions, that is, from Telnet and FTP.

Flex fields

Flex fields allow merchants to store additional information in the Oracle iStore 11i database according to their specific needs.

Front-end customizing

Front-end customizing involves adding new functionality to the presentation layer of Oracle iStore 11i, but without altering the underlying data model, also known as the persistence layer, or involving any third-party systems.

Intermedia

Formerly named ConText. Intermedia is an Oracle server option which enables text queries to be performed through SQL and PL/SQL.

Invoice

The bill for an order that is generated by AR when items are shipped.

Launch phase

During this phase, the project team prepares for and executes the launch of the store including production installation, validation, documentation, and training.

During this time, the implementation team may perform additional incremental development to provide more functionality or address non-functional requirements, such as enhancing performance and reliability.

Multimedia component

A content attribute that merchants may associate with a product or a section

Multimedia content

An audio, image, text, video, or binary file. Content or “media content” is the term used by Oracle iStore 11i to broadly refer to any one of the following file types of files: audio, image, text, video, and binary.

Net Perceptions

Net Perceptions makes recommendations based on ratings contained in its database.

Object caching

Object caching allows you to specify the number and type of objects that you want to cache.

Order status

Order Status allows customers to check the status of their current order. Account customers may also review past orders.

Payments

Payments made by a customer against invoices.

Primary display category

Primary display category of a product determines the default values for multimedia components and display styles. If no values are provided (for a multimedia component or display style) then iStore uses the value specified for the primary display category in which the product belongs. To determine primary display category iStore checks if any of the categories to which the product belongs to is part of the primary display category set (see Primary display category set). If yes, then that category is the primary display category. If the product does not belong to a category in the primary display category set, then there is no primary display category for the product. In such situations, Oracle iStore 11i picks defaults (for multimedia components and display styles) from defaults specified at store level.

Primary display category set

The profile value which tells which of the categories to which product belongs to is the primary display category.

Product item/Product

Product item represents an individual item for sale.

Rating

Rating is a numeric representation of how much a user would like to see more items like the one being rated. Ratings may be explicit or implicit. Explicit ratings are user actions that directly specify the level of like or dislike of an item.

Registration

Allows walk-in customers to register with the store and become account users.

Sections

A node in the hierarchy. A section can contain other sections or other products.

Secure Electronic Transaction

Secure Electronic Transaction (SET) 1.0 protocol is an open standard developed jointly by Visa and MasterCard to ensure the privacy and security of credit card transactions over open networks such as the Internet.

Settling transactions

Settling transactions includes capturing authorized transactions, processing voids and returns, and batch administration.

Smart Linking

Smart Linking is a mechanism for affiliated sites to link directly into a location within Oracle iStore 11i, bypassing the storefront.

Soft launch

The launch of the storefront, where a select group of individuals is allowed to use the site, or the client places the site on the Internet but does not market it.

Specialty store

Multiple, specialized store fronts operated and managed in one central instance. Specialty stores have a specialized look and feel or set of products intended for specific types of customers. Speciality stores may differ in appearance, products, or processing flow. An example of a specialty store is a web site for children's clothing within a retail clothing web site.

Store attributes

Store attributes, such as RefSet objects, are common to all Oracle iStore 11i stores. Unlike product attributes, which are defined on a per-product basis, store attributes are defined for the store as a whole.

Store Manager

Store Manager is an HTML-based interface that provides a comprehensive set of features for building and maintaining an Internet store.

Store search

Store Search brings up a query form that allowing users to search for products in the store.

Templates

Templates contain presentation logic. A template has a logical name and may be manifested in one or more JSP pages for different specialty store / language

combinations. A merchant may have two JSP pages: home_regualr.jsp and home_partner.jsp for the home_page. Oracle iStore 11i displays home_regular when showing the home page to regular users and displays home_partner.jsp in the partner specialty store.

Transactional data

Transactional data refers to information that is the result of customer interaction with the store.

Upsell

Upsells aim to persuade customers to buy more expensive types of the same product.

Walk-in users

Customers who do not have a store account.

Index

A

Account business flows, 14
APIs, B-2, B-4, B-7
ATP, 33

B

Boolean searching, 8
Business flow, setting up, 27
Business objects, 2

C

Carts and lists, 18
Categories, 65
Category Mappings, 54
Checkout, 12
Concurrent Programs Manager, 61
Cookies, B-6
Customer addresses, 17
Customer information business flow, 16
Customer master business object, 2
Customer passwords, 17

D

Display style, 37, 38, 48, 50

E

Error pages, B-10
Exceptions, B-4, B-10
Express Checkout Preferences, 17

F

Forms, B-9

H

Hierarchies, 50
 assigning products to, 48
 defining, 47
 key points and concepts, 4

I

Inventory, 29, 30, 33, 48, 52, 57, 59, 60

J

JavaBeans, B-9
JSP architecture, 25
JSP structure, 39
JSPs, 22, B-1, B-7, B-9

K

Keywords, 8, 45

L

Layout, changing in JSPs, B-7
Logical templates, 6

M

Multimedia, 5, 20, 65
 business object, 3

- cataloging, 34
- components, 36
- My Account, 17

O

- Oracle applications' relationship to iStore, 7
- Oracle ConText, 62
- Oracle InterMedia, 60
- Oracle Inventory, 47, 56, 60
- Order system business object, 3
- Orders and invoices, 19

P

- PageCompileServlet, 26
- Passwords, 17
- Payment Books, 17
- Pricing setup, 59
- Product catalog
 - building, 39, 51
 - business object, 3
 - key points and concepts, 3
 - setting up, 29
- Products
 - adding to sections, 51
 - assigning to categories, 54
 - defining, 51
- Programmatic access name, 44

R

- Relationship rules, 55
- Relationships page, 50
- Request object, 26
- Root section, 33

S

- Search attributes, 60
- Search dependencies, 60
- Search index tables, 62
- Search process program units, 63
- Search profile options, 62
- Search table, 60

- Searches
 - adding stopwords, 64
 - customizing, 63
 - key points and concepts, 8
 - setting up, 59
- Section code, 48, 50
- Seeded relationships, 56, 57
- Seeded values
 - for multimedia components, 36
- Seeded values for display styles, 38
- Shopping cart, 10
- Specialty store
 - business object, 3
 - creating, 28, 32
- SQL, 48, 52, 58, 63
- SQL rules, 56
- Standard includes, 25
- Stopwords, 64
- Styles, 65

T

- Template names, 44
- Templates, 21
 - account, 14
 - adding to categories, 65
 - assigning, 42
 - assigning to categories, 43
 - business flows, 9
 - business object, 3
 - carts and lists, 18
 - cataloging, 44
 - checkout, 12
 - creating, 38
 - creating text for, 30
 - for searching, 9
 - logical, 6
 - mapping in merchant UI, 41
 - modifying, 46
 - orders and invoices, 19
 - passing values, B-5
 - product, 6
 - relationships with objects, 6
 - section, 6
 - shopping cart, 10

user information, 16
Text, changing in JSPs, B-7

U

User information
 business flows, 16
 business object, 3
User interface
 changing layout of, B-7
 changing text in, B-7
 key points and concepts, 5
User Management, 17

V

Values, passing across templates, B-5
Viewing categories, 43
Viewing template details, 43

W

Walk-in customers, 33
Web Option tab bug workaround, 52

