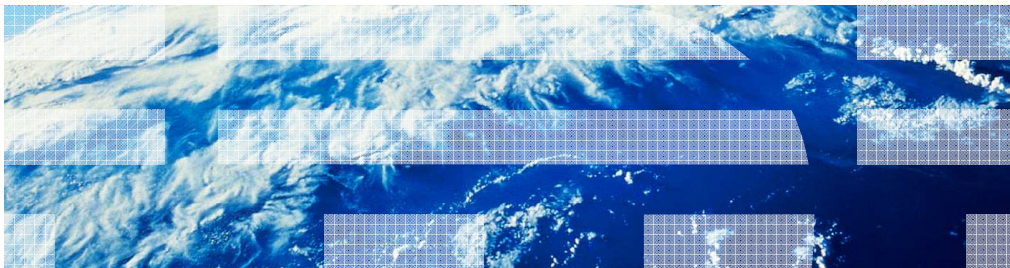


WebSphere Commerce V7 Feature Pack 5

New Aurora store including widgets and page layouts



© 2012 IBM Corporation

Feature Pack 5 introduces a new Aurora store that includes widgets and page layouts.

Table of contents

- Overview
- Page layouts and widgets
- Widget example
- Widget communication
- New Aurora starter store

This presentation begins by introducing the store capabilities added in Feature Pack 5. It then looks at each capability in more detail beginning with the concepts of page layouts and widgets. There is then a widget example and a section on how the widgets communicate. Finally, the presentation will cover the new Aurora starter store.

Feature pack 5 store enhancements overview

- New Aurora storefront
- Modular page design
 - Identify various UI elements, features, or components which are self contained and independent
- Page layouts that help define the sections in the page along with the styling
- Business user tool support
 - Empower business users to select the layout used for various store pages

Feature Pack 5 includes a new starter store called the Aurora store. The Aurora store demonstrates a new, modular page design where UI components are self contained and can be easily added to, removed from, or repositioned on the page. For example, the product compare zone, mini shopping cart, product listing view, and wish list are all examples of components. The introduction of independent, reusable components allows for the creation of page layout templates that define the sections in the page and provide overall styling. Management Center support is added to allow business users to select which page layout templates to use for various pages within the store.



Section

Widgets and Page Layouts

4

New Aurora store

© 2012 IBM Corporation

This section covers Widgets and Page Layouts.

Widget

- Independent UI module providing a specific feature or function
- Can be embedded and moved around in store pages
- Capable of retrieving its own data while providing multiple ways of rendering
- Comprised of four parts including two JSP fragments:
 - Data provider (JSP page)
 - Java Script provider
 - User interface provider (JSP page)
 - CSS provider

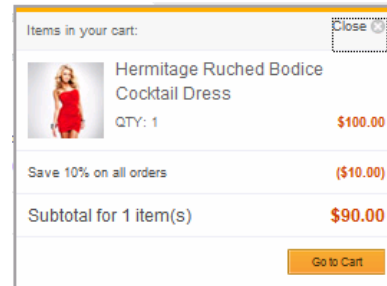
A widget is an independent UI module providing a specific feature or function that can be embedded and moved around in store pages. Each widget is capable of retrieving the data it requires while providing multiple ways of rendering the data and defining user interactions associated with the data. A few examples of widgets are the header widget, footer widget, e-Marketing Spot widget, and product list widget.

Each widget is composed of four parts. The first part, called the data provider, handles the data retrieval aspects for a widget. This includes fetching the data required to render from the WebSphere Commerce server. Each widget can have at most one data provider which is modeled as an independent JSP page. The second part is the Java Script provider. This defines the set of user actions that are supported by the widget. The provider is modeled as a Java script file. For example, the product list widget might support adding the product to shop cart, viewing quick info about the product, and moving the product to the compare zone. The third part is the User Interface provider. This provides the user interface for rendering the data handled by the widget. A widget supports multiple user interface providers, allowing the widget data to be displayed in different ways. The user interface providers handles the sizing and positioning of the data within the widget. Data in the widget is rendered based on the UI provider chosen. For example: A product list component has multiple UI providers that support rendering the product list in a detailed view, thumbnail view and so on. Another example is the Merchandising widget, it can display the list of product associations in a scrollable widget or as a flat view with pagination support. The fourth part is the CSS provider. The CSS provider assists in the positioning of the basic HTML UI elements within the component's user interfaces. Additional styling information comes from the page template that includes the components in the form of widgets. This way the same widget can be used in different stores and the widgets will nicely blend with the overall theme of the store.

Widget library

- Breakdown each store page into store widgets
- Create widget library
 - Header
 - Footer
 - Search field
 - Product page widgets
 - Product image
 - Product name and description
 - Product price
 - Product inventory
 - Product merchandising association with scrollable view and flat view
 - Product attachments
 - Different flavors of e-marketing spots
 - Mini shop cart
 - Facet navigation
 - Header department listing
 - Bread crumb trail
 - Compare

Mini shopping cart

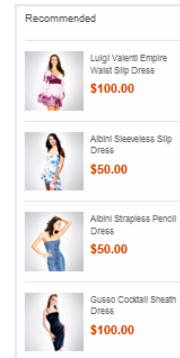


The Aurora store provides modular UI widgets that can be plugged into store pages. These UI widgets are independent, self-contained entities. The following widgets are included with the Aurora store.

Widget examples

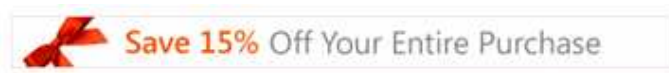
- **ProductRecommendation widget**

- Type: Displays product recommendations with a vertical UI alignment
- Pages used on: Content page, Product page



- **ContentAreaESpot widget**

- Type: Displays content horizontally across screen
- Pages used on: Almost all pages



This slide shows two different widget examples, the product recommendation widget and the content area e-spot widget. The product recommendation widget displays the product recommendations with a vertical UI alignment, you can see this widget in the Aurora store by looking at a content page or product page. The Content area e-spot widget displays content horizontally across the screen and within the Aurora store you can see this widget within almost all pages.

Page layout

- A page layout is a store page created for a particular layout type
- Has widgets associated with it and values for the properties of those widgets
- Communication between different widgets on the page is handled using Dojo publish/subscribe event handlers
- Associated with an object in WebSphere Commerce rather than a store page
- Uses its own CSS for styling of page and styling for widgets

A page layout is a store page created for a particular layout type such as a category page, product page, or home page. A page layout has widgets associated with it and each widget has a set of configurable properties that can be provided to allow finer configuration of the widget in a layout. These configurable properties are called widget properties. The widget properties are of two types. The first type are properties that control the display of UI elements in the widget. These properties provide an alternate to flex flow in places where flex flow tags were previously used to control the display of UI elements. The second type of properties are properties that provide general configuration in the widget. For example, the size for the mini shop cart component, or the e-marketing spot name in the e-spot component. Each widget property can have either a predefined list of property values or is open to allow you to provide the value. In addition, each widget property has a default value that is used when you do not provide any value for the property. When associating a widget to an area in the layout, you can choose to provide values for the widget properties to customize the widget's usage within the layout. The UI provider makes use of these properties to handle the rendering of data within the component.

The store pages in the Aurora store are dynamic and there are a handful of store pages that display different content based on the object identified through its query parameters. Thus, a layout is associated with an object in WebSphere commerce rather than a store page. A layout is also indirectly associated with the page. For example, Layout1 applies to all categories; Layout2 applies only to the furniture category. Communication between different widgets on the pages are handled using dojo publish/subscribe event handlers. A page layout uses its own CSS for styling the page and some aspects of widgets on the pages such as font size and color.

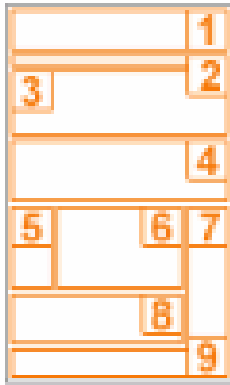
Page layouts

- A page layout is a JSP written by a store developer
- It draws the outline of the page and it includes the widgets that it needs

- Different Page Layouts
 - Home Page
 - Department Page
 - Category Page
 - Product Page
 - Item Page
 - Bundle Page
 - Kit Page
 - Dynamic Kit Page
 - Search Landing Page
 - Compare Product Page

Each store page is associated with a page layout. A page layout consists of layout areas that specify the widgets and widget properties to show in each of the areas. The page layout itself is a JSP page that is written by a store developer. This slide lists the different types of page layouts defined for the Aurora store.

Page layout home page example



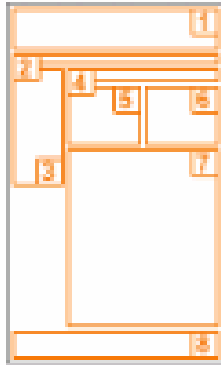
Home Page Layout

1. Header
2. Double E-Marketing Spot for Contents
3. E-Marketing Spot for Contents
4. E-Marketing Spot for Catalog entries
5. E-Marketing Spot for Contents
6. E-Marketing Spot for Contents
7. E-Marketing Spot for Contents
8. E-Marketing Spot for Contents
9. Footer



This slide is a visual complete representation of a store page, it contains all the information required to layout widgets in the proper sequence on the page. The layout shown is the Home page layout.

Page layout category example



Category Page Layout

1. Header
2. E-Marketing Spot for Contents
3. Faceted Navigation
4. Category Name
5. E-Marketing Spot for Contents
6. E-Marketing Spot for Contents
7. Product listing
8. Footer



Here is the Category Page Layout with the list of widgets incorporated into the layout.

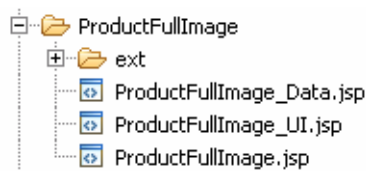
Section

Widget example

This section goes over a Widget example.

Product full image widget

- Widget files created
 - ProductFullImage.jsp
 - ProductFullImage_Data.jsp (Data provider)
 - ProductFullImage_UI.jsp (User interface provider)
 - ProductFullImage.js (JavaScript provider)
 - common1_1.css (CSS provider)



The example that the next few slides will cover is the Product full image widget. The slide lists the widget files that are created for this widget. The next few slides will go over each of these files in more detail.

Product full image widget

- **ProductFullImage.jsp**

```
<%@ include file= "../Common/JSTLEnvironmentSetup.jspf" %>
<%@ include file="ProductFullImage_Data.jsp" %>
<%@ include file="ProductFullImage_UI.jsp" %>
```

- **ProductFullImage_Data.jsp** – fetches data from the WebSphere Commerce server

```
...
<wcf:getData
  type="com.ibm.commerce.catalog.facade.datatypes.CatalogNavigation
  ViewType" var="catalogNavigationView"
expressionBuilder="getCatalogEntryViewAllByID"
varShowVerb="showCatalogNavigationView" maxItems="1"
recordSetStartNumber="0">
<wcf:param name="UniqueID" value="{productId}"/>
<wcf:contextData name="storeId" data="{storeId}"/>
<wcf:contextData name="catalogId" data="{catalogId}"/>
</wcf:getData>
...
// then set var="fullImage"
```

The first thing to note is the naming convention of `WidgetName.jsp` and `WidgetName_Data.jsp`. The `ProductFullImage.jsp` is a JSP that includes the data provider `WidgetName_Data.jsp` (in this case `ProductFullImage_Data.jsp`) and the UI provider `WidgetName_UI.jsp` (in this case `ProductFullImage_UI.jsp`). The `ProductFullImage_Data.jsp` will retrieve data from the database using `wcf:getData` tags. It also does all computations. If there is more than one UI layout for a widget, for example the product list widget has three different UI layouts, all will use the same data retrieval code. The include tag is used in the `ProductFullImage.jsp` instead of import so that the presentation code in the UI JSP will have access to the data retrieved by the data JSP.

Product full image widget

- **ProductFullImage_UI.jsp** – provides the user interface for rendering the data within the widget

```
...  
<div class="image_container">  
"  
title="<c:out value="{fullImageAltDescription}"  
escapeXml="false"/>" class="product_main_image"/>  
<div class="clear_float"></div>  
<div class="hover_text">  
<span style="display:none;"><fmt:message  
key="FI_HOVER_OVER_IMAGE_TO_ZOOM" /></span>  
</div>  
</div>  
...
```

This slide shows a snippet of the ProductFullImage_UI.jsp which provides the user interface for rendering the data handled by the widget. The user interface provider handles the sizing and positioning aspects for the data within the widget.

Product image widget JavaScript provider and CSS

- **ProductFullImage.js** – defines user actions supported by the widget

```
/** Stores the identifier of the image currently used for a product */
var currentAngleImgId="productAngleLi0";

function changeThumbNail(angleImgId,imsrc){
  if (currentAngleImgId != "") {
    document.getElementById(currentAngleImgId).className = '';
    currentAngleImgId = angleImgId;
    document.getElementById(angleImgId).className = 'selected';
    document.getElementById("productMainImage").src = imsrc;
  }
}
```

- **common1_1.css** – assists in the positioning of the basic HTML UI elements within the widget

```
// 4.3.1 Product image Viewer Widget
div#widget_product_image_viewer {
  width: 424px;
}
div#widget_product_image_viewer h1{
  font-size: 13px;
  margin-bottom: 5px;
}
```

Here you see the ProductFullImage.js file, this is the Java Script Provider which defines the set of user actions that are supported by the widget. For the product full image widget, this includes the functions to manipulate the product images and for switching the tabs used on the product pages. The CSS Provider, common1_1.css assists in the positioning of the basic HTML UI elements such as buttons, text boxes, and radio buttons within the widget's user interfaces.

Product full image widget in the product page layout

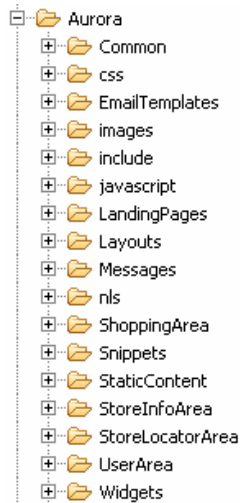


- **ProductDisplay.jsp** – Product display page layout
 - Contains the product image widget

```
<script type="text/javascript"
src="${jsAssetsDir}javascript/Widgets/ProductFullImage/ProductFullImage.js"></script>
...
<!-- Widget Product Image Viewer -->
<div class="widget_product_image_viewer_position">
<%out.flush();%>
<c:import url =
  "${env_jspStoreDir}Widgets/ProductFullImage/Produc
tFullImage.jsp" />
<%out.flush();%>
</div>
<!-- End Widget Product Image Viewer ...
```

Here is the product page layout JSP file. This layout incorporates different widgets and sets the widget properties for the widgets. The snippet shows the incorporation of the product full image widget in the product page layout.

New folder structure



- <WC_installdir>\Stores\WebContent\Aurora
- **Common** contains JSPF files
- **css** contains stylesheet information
- **images** contains images
- **javaScript** contains the widget JS files
- **Layouts** contains the page layout JSP files
- **Widgets** contains all the Widget JavaServer Pages files

Within the Aurora store is a new folder structure for the widgets and page layouts. The Common folder contains JSPF files that retrieve and prepare a lot of information used by all the store JSP files. These JSPF files contain JSTL variables which can just be referenced by the store JSP files that include it. The css folder contains stylesheet information for the store pages and widgets. The Images folder contains images including the page layout images used in Management Center. The JavaScript folder contains the widget JavaScript files. The Layouts folder contains the Page Layout JSP files. The Widgets folder contains all the Widget JSP files for all the defined widgets. The rest of the folders are the same as seen within Madisons.

New store widget

- No registration needed for the widget
- Create JSP files
 - Use same naming standards, directory structure as existing widgets
 - Add widget to a page layout JSP
- Naming:
 - **<WidgetName>.jsp** – JSP includes the WidgetName_Data.jsp and WidgetName_UI.jsp to display the data
 - **<WidgetName>_Data.jsp** (Data Provider)
 - **<WidgetName>_UI.jsp** (User Interface Provider)
 - **<WidgetName>.js** (JavaScript Provider)
 - **<CSSName>.css** (CSS Provider)

If you plan on creating a new store widget there is no registration needed. You just create the JavaServer Pages files using the naming standards and directory structure that were shown on the pervious slide. When your widget is complete, you can add it to a page layout JSP. The naming convention for files is listed here.

Widget communication

Communication between different widgets on a page is handled using dojo publish/subscribe event handlers. This section will cover an example of this widget communication.

Dojo publish/subscribe

- Dojo and Web 2.0 behavior are the same in Aurora and Madisons storefront
- Aurora uses Dojo's publish/subscribe to do communication
- Example: Select color/size and inventory changes

The screenshot displays a product page for the 'Luigi Valenti Empire Waist Slip Dress'. The product name is 'Luigi Valenti Empire Waist Slip Dress' with SKU: MadisonsWMDRE-039 and a price of \$100.00. The description is 'Empire waist slip dress with a V-neck and long flared sleeves'. Two red boxes highlight key features: the left box shows color selection (Red, Blue, Purple) and size selection (XS, S, M, L, XL) swatches; the right box shows inventory status (Available Online: In-Stock, Available In-Store: Calgary Circle Mall, Backordered) and a 'Change Store' link. Below the swatches is a quantity input field set to 1, and buttons for 'Add to Cart' and 'Add to Wish List'.

21

New Aurora store

© 2012 IBM Corporation

Dojo and Web 2.0 behavior are the same in the Aurora and Madisons storefronts. The Aurora store uses Dojo's publish/subscribe event handlers to communicate between the different widgets. The example that the next few slides will walk through is the communication needed between the product description widget and inventory widget. In the product description widget, a shopper selects a color and size combination and the inventory widget needs to know these attribute changes so that the inventory can be correctly displayed. Notice the selection of the color and size attributes are shown as swatches. This is a new feature within the Aurora store and will be covered later in the presentation.

Inventory status and product description widgets

- Inventory status widget subscribes to the “Attribute Change” event during initialization
- Product description widget publishes an “Attribute Change” event when modified
- Shopper selects a swatch within the Product description widget
 - An “Attribute Change” event is published and contains the attribute that was selected
 - Inventory status widget receives the event notification and reacts with updated inventory

When a shopper selects a swatch attribute within the Product Description widget, the attribute change is communicated through a Dojo event so that the inventory status widget will know what attributes are selected for the product. The widget can then correctly display inventory information for that product with the selected attributes. When the inventory status widget gets initialized, it uses Dojo to subscribe to the attribute change event. The Product description widget uses Dojo publish to send an attribute change event to pass the parameters such as the attribute that the shopper selects. When a shopper selects a swatch within the Product description widget, an Attribute Change event is published and includes the attribute that was selected. The Inventory status widget is listening for the attribute change event. It receives the updated attribute information and reacts by updating the inventory for that item's color and size.

Subscribe

- InventoryStatus_UI.jsp

```
<script>
    InventoryStatusJS_${param.productId} = new InventoryStatusJS({storeParams}, ${catEntryParams},
    [${physicalStores}], '${param.productId}');
</script>
...
```

- InventoryStatus.js

```
...
    dojo.subscribe("attributesChanged_"+this.productId, this, this.setCatEntryAttributes);
...
```

Here is the Dojo subscribe code for the Inventory status widget.

Publish

- ProductDescription_UI.jsp

```
shoppingActionsJS.notifyAttributeChange("${catalogEntryID}");
```

- ShoppingAction.js

```
//Notify the change in attribute to other components that are  
//subscribed to attributesChanged event.  
notifyAttributeChange: function(catalogEntryID){  
    this.baseCatalogEntryId = catalogEntryID;  
    var selectedAttributes =  
    this.selectedAttributesList["entitledItem_" + catalogEntryID];  
    dojo.publish('attributesChanged_'+catalogEntryID,  
    [dojo.toJson(selectedAttributes)]);  
    dojo.publish('attributesChanged'  
    [dojo.toJson(selectedAttributes)]);  
},
```

Here is the dojo publish notifyAttributeChange code for the Product Description widget.

New Aurora starter store

This section covers the new Aurora starter store.

Aurora starter store highlights

- For shoppers
 - Visual refresh of store design and sample data
 - Improved usability
 - Faceted navigation enhancements
 - Bazaarvoice integration for ratings and reviews
- For developers
 - Define discreet store functions as self-contained widgets
 - Combine widgets to build a store page layout
 - Make layouts available to business users through the Page Layout tool
 - WebSphere Commerce search built in
- New store archives
 - Aurora.sar
 - ExtendedSitesV2.sar

The Aurora starter store demonstrates best practices in store development with WebSphere Commerce, and can act as a base for implementing custom storefronts. The store showcases an updated page design and high quality sample data including product color swatches. Enhancements to faceted navigation, such as multi value selection, allows shoppers to quickly and easily find the products they are looking for. Client side integration of Bazaarvoice provides a rich set of widgets for viewing and creating product ratings and reviews. For store developers, the Aurora store facilitates customization and reuse by providing discreet store functions as self-contained widgets. This design allows widgets such as product images and e-Marketing Spots to be easily added to or removed from store pages with minimal impact to the overall page layout. Widgets can be combined to form a series of page layouts which can be made available to business users through the Management Center Page Layout tool.

The Aurora starter store simplifies the store publish process by removing the need to publish add-on store archive files in most cases. WebSphere Commerce search is built in and all Aurora features are contained in a single store archive file.

Store design and sample data

The image displays two screenshots of the Aurora e-commerce store interface. The left screenshot shows a category page for 'Dresses'. It features a top navigation bar with the Aurora logo, account links, and a shopping cart showing a subtotal of \$0.00. Below the navigation, there are promotional banners for 'Save 15% Off Your Entire Purchase' and 'Free Shipping On orders over \$100'. The main content area includes a 'Filter by' section with options for Material (Cotton blend, 100% cotton, Neoprene, Sandstone, Silk, Satin, Wool), Color (25, 14, 13, 8, 6, 4, 3, 2, 1), Brand (Alibi, Chouca, Luigi Valent, Versati, Hermitage Collection), and Price (Less than \$100, Between \$100 and \$300). The product grid displays items like 'Hermitage Ruched Bodice Cocktail Dress' for \$250.00 (originally \$100.00), 'Versati Empire Waist Sundress' for \$60.00, 'Luigi Valent Empire Waist Slip Dress' for \$100.00, and 'Hermitage Panel Dress' for \$200.00. The right screenshot shows a product page for the 'Hermitage Ruched Bodice Cocktail Dress'. It features a large product image, a color swatch for 'Red', and a price of \$250.00 (originally \$100.00). The page includes a description, 'Other Customers Also Purchased' section, and a 'Recommended' section with items like 'Luigi Valent Empire Waist Sundress' for \$100.00 and 'Alibi Sleeveless Slip Dress' for \$50.00. The footer of the screenshots includes '27 New Aurora store' and '© 2012 IBM Corporation'.

Aurora better supports evolving online business models. Aurora has new HTML and CSS code that updates layout and design patterns for efficiency. While retail focused, Aurora showcases a wide scope of e-commerce scenarios and storefront features available in Feature Pack 5. These screen captures show a new category and product page in the Aurora store. In the category page on the left, you can see the new faceted navigation bar with multi value selection and swatch facet values. In the product page on the right, you can see the enhanced product images and color swatch attribute. Selecting a color refreshes the product image on the page.

Swatches

- To set up swatches, provide an image path for each attribute value in Management Center

Attribute values	Sequence (United States English)	* Type	Store	* Value (United States English)	Image path 1 (United States English)
	1.0		Extended Sites Catalog Asset Store	Red	images/catalog/apparel/women/Swatches/swatch_red.png
	2.0		Extended Sites Catalog Asset Store	Teal	images/catalog/apparel/women/Swatches/swatch_teal.png

- If the attribute value has an image path, it is displayed as a swatch icon in the storefront
- If no image is associated with the attribute values, a drop down with text values is shown in the storefront

Swatches are just attribute values that have a value set for the image path. In Management Center, each attribute value has an image path and if the image path value is set then WebSphere Commerce will treat it as a swatch icon. For example, you can set up an attribute called swatchcolor with values such as red and blue and specify an image path for each attribute value. On the storefront, instead of a drop down you will see swatch image icons. When a shopper in the storefront selects the red swatch image icon, WebSphere Commerce will pull in the product image for that resulting item and change the product display image. If no images are associated with the attribute values, the store design uses the standard drop down style and the product image does not change when an attribute value is selected.

Site level features and enhancements

- Faster and easier customization of store pages
 - New CSS enables easier styling and customization of the site
- Updated header and footer
- Site Map that dynamically populates categories and footer pages
- Language and Currency selection
- Scalable department menu tiers
- Enhanced search
- Mini cart interaction and functionality
- Social network integration
- Quick Info dialog
- Wish List creation and added product confirmation
- Unstructured content support for unique pages
- Email templates designs
- Store compliant with Web Content Accessibility Guidelines (WCAG) 2.0 set out by the W3C
- Paging controls that follow updated design patterns
- Store navigation

The Aurora starter store features various site enhancements that provide an improved user experience. This slide lists the various site enhancements and features within the Aurora store. There are more details within the information center on each of these features.

Category and product features and enhancements

- Category level features/enhancements
 - Product compare
 - Category product list view
 - Search facets
 - Comprehensive category list view (grid view and list view)
- Product level features/enhancements
 - Item stock availability behavior
 - Product recommendations
 - Page designs for Package and Bundles
 - Apparel focused "customers also purchased" product tab using Coremetrics Intelligent Offer
 - Recently viewed
 - Product image sizes

This slide lists the various category and product features and enhancements within the Aurora store. There are more details within the information center on each of these features.

Categories that show off different features

- Apparel
 - 3-Tier Department Menu
 - Multiple Product Views (SKU: AuroraWMDRS-22, SKU: AuroraWMDRS-25, AuroraWMDRS-31)
 - Swatches (Woman -> Dresses only)
 - Sizes
 - Apparel cross sells and intelligent offers
 - Package and Bundle (SKU: MAC013_1333, SKU: BCL014_1417)
- Grocery, Health
 - Recurring orders
- Magazines, Newsletters
 - Subscriptions
- Electronics
 - Dynamic Kits
- Home and Furnishings
 - Kits (SKU: HTA029_2932, SKU: HAP027_2744)
 - Special Shipments for large items / Delivery Options (for example: stoves)
- Search term "coffee" or "dress", articles and videos results
 - Unstructured Content includes: e-spots, apparel PDF, coffee maker user manual and rich media videos
- Search term "coffeemaker", "women's dresses", "women dresses"
 - Search Landing Page: includes e-spots and rich media video

To quickly see the new features within the Aurora store follow this guideline when browsing the storefront.

Limitations

- Elite, Brazil, and MayUJoy have not been updated
- If you publish a store in a workspace environment, deactivate and then reactivate all the sample web activities
- Mini shopping cart displays a maximum of 50 items by default
- Madisons smart phone and tablet add-on SAR files can be published on Aurora
 - Some limitations exist
 - Aurora sample data is not fully optimized for mobile stores
 - Madisons mobile stores are not updated for new Aurora features

<http://publib.boulder.ibm.com/infocenter/wchelp/v7r0m0/index.jsp?topic=/com.ibm.commerce.starterstores.doc/tasks/tsmmopub.htm>

This slide lists some of the limitations within Feature Pack 5 related to starter stores. The first thing to note is the Elite, Brazil, and MayUJoy starter stores have not been updated. Next, if you publish a store in a workspace environment, be sure to deactivate and then reactivate all the sample web activities so that they display correctly. The mini shopping cart displays a maximum of 50 items by default. That is, if a shopper reaches 50 items and adds more items to the shopping cart, the mini shopping cart does not display the new additions. You must increase the default value in the mini cart JSP file fragment when expecting typical shopping cart sizes to exceed 50 items.

A mobile-equivalent of the Aurora starter store is not available in Feature Pack 5. However, the Madisons mobile starter store is available using the Aurora sample data. The Madisons smart phone and tablet add-on SAR files can be published on Aurora. To use the Aurora starter store as a mobile starter store requires some customization to overcome limitations. The Aurora sample data is not fully optimized for mobile stores. There is no mobile-specific category images or e-Marketing Spot ads. The Madisons mobile stores are not updated for new Aurora features, so there is no swatch selection or 3-tier category support.

Limitations continued

- Features not demonstrated in Aurora
 - Gift registry, Coshopping, and Remote widgets
 - Social Commerce with Smash,
 - Non-Ajax Add to Cart, Non-Ajax Checkout and Non-Ajax my account flows
 - Drag and drop
- Widgets and page layouts used throughout most of the store
 - Excluding shopping cart, checkout and my account pages

This slide covers Aurora store specific limitations. Some features included in the Madisons enhancements store archive are not available in the Aurora store. Features that are not demonstrated in the Aurora starter store such as Gift registry, coshopping and remote widgets can be enabled through customization of your store. Specific social commerce features including social bridging, public-facing user profiles, blogs, photo galleries, and social bookmarks are deprecated in Feature Pack 5. The Facebook integration and ratings and reviews integration with Bazaarvoice are supported in the Aurora store. Non-Ajax features and drag and drop are outdated and not included in the Aurora storefront. Finally, widgets and page layouts are used throughout most of the store, however the shopping cart, checkout and my account pages have not been converted to the new layouts. These details can be found in the WebSphere Commerce information center.

Troubleshooting

- Troubleshooting: Aurora starter store

<http://publib.boulder.ibm.com/infocenter/wchelp/v7r0m0/index.jsp?topic=/com.ibm.commerce.aurora-starterstore.doc/refs/rsmauroratroubleshooting.htm>

- Troubleshooting: Performance considerations for the Aurora starter store

http://publib.boulder.ibm.com/infocenter/wchelp/v7r0m0/index.jsp?topic=/com.ibm.commerce.aurora-starterstore.doc/refs/rsmaurora_tsperformance.htm

This slide lists some useful troubleshooting links including how to change the Aurora store default behaviors to improve performance.

Summary

- Widgets and Page layouts
- Widget example
- Widget communication
- New Aurora store

This presentation began by introducing the store capabilities added in Feature Pack 5. It then looked at each capability in more detail beginning with the concepts of page layouts and widgets. Following that was a widget example and a section on how the widgets communicate. Finally, the presentation covered the new Aurora starter store.

References

- Aurora starter store

<http://publib.boulder.ibm.com/infocenter/wchelp/v7r0m0/index.jsp?topic=/com.ibm.commerce.aurora-starterstore.doc/concepts/csmaurora.htm>

- Aurora starter store page architecture

<http://publib.boulder.ibm.com/infocenter/wchelp/v7r0m0/index.jsp?topic=/com.ibm.commerce.aurora-starterstore.doc/concepts/csmaurorastorepagearch.htm>

- Comparing the Madisons and Aurora starter stores

http://publib.boulder.ibm.com/infocenter/wchelp/v7r0m0/index.jsp?topic=/com.ibm.commerce.aurora-starterstore.doc/concepts/csmaurora_madaur.htm

- Style Guide reference sheet and document provided to enable web designers with quicker customization of the pages

https://www14.software.ibm.com/webapp/iwm/web/preLogin.do?lang=en_US&source=swg-stae

Here are some useful references.



Feedback

Your feedback is valuable

You can help improve the quality of IBM Education Assistant content to better meet your needs by providing feedback.

- Did you find this module useful?
- Did it help you solve a problem or answer a question?
- Do you have suggestions for improvements?

Click to send email feedback:

[mailto:iea@us.ibm.com?subject=Feedback about AuroraStore.ppt](mailto:iea@us.ibm.com?subject=Feedback%20about%20AuroraStore.ppt)

This module is also available in PDF format at: ../AuroraStore.pdf

You can help improve the quality of IBM Education Assistant content by providing feedback.



Trademarks, disclaimer, and copyright information

IBM, the IBM logo, ibm.com, and WebSphere are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of other IBM trademarks is available on the web at "[Copyright and trademark information](http://www.ibm.com/legal/copytrade.shtml)" at <http://www.ibm.com/legal/copytrade.shtml>

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. IN ADDITION, THIS INFORMATION IS BASED ON IBM'S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE. IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION. NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, NOR SHALL HAVE THE EFFECT OF, CREATING ANY WARRANTIES OR REPRESENTATIONS FROM IBM (OR ITS SUPPLIERS OR LICENSORS), OR ALTERING THE TERMS AND CONDITIONS OF ANY AGREEMENT OR LICENSE GOVERNING THE USE OF IBM PRODUCTS OR SOFTWARE.

© Copyright International Business Machines Corporation 2012. All rights reserved.