



【IBM SOA講堂】
Lotus動員協作：
Portal 6.1 開發及 Web
2.0 應用

鄭志傑
IBM 資深資訊工程師
chengcc@tw.ibm.com

IBM Certified System Administrator
WebSphere Portal

WebSphere Portlet

- WebSphere Portal v6.1 針對Web 2.0 新增功能
- 如何設計 Ajax 應用程式
- WebSphere Portlet Factory 6.1 新增功能



WebSphere Portal 6.1

Deliver an irresistible user experience

Greatly enhanced User Experience

- Improved response times – AJAX, REST, client-side aggregation, C2A
- Enhanced flexibility – easy end user customization through drag & drop, Google gadgets, etc.
- Smart markup and dynamic menus through semantic tags

Improve time to value, time to market

- Business user situational application development through the Portlet Palette, drag & drop page composition, and composite application templates
- Business user contribution of WCM content
- More assets (REST, ATOM, gadgets, widgets) available to choose from.



Portal V6.0 has many Web 2.0 Features,...

- **Social Software with Lotus Connections portlets**
- **Blogs and wikis, and forums in Quickr**
- **Allows User Contribution** to portal sites (e.g. through WCM)
- **Enables Situational Development** through the Portlet Palette, Drag and Drop page composition, and Composite Application Templates
- **AJAX, REST, ATOM, DOJO,...**



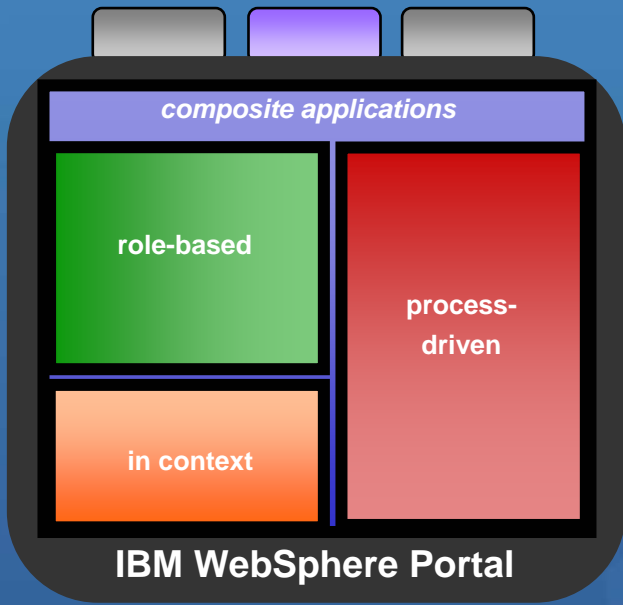
IBM WebSphere Portal and Google Gadget Integration

Seamless Integration With Thousands of Web-based Solutions Further Extends IBM's Market Leadership

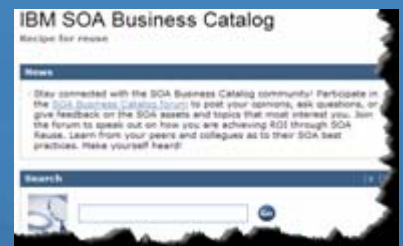
IBM Portlet for Google Gadgets

Easily Configure Google Gadgets to Portal applications:

- Search and select any Google Gadget
- Customize to portlet
- Leverages Portal's Web 2.0 APIs support
- Supports inter-portlet communication between Portal portlets and Google Gadget portlet instances.



Enterprise and internet data and services



- Community SOA Wiki
- Business Catalog forum
- > 3600 Assets in IBM SOA Business Catalog



- Google Gadget Services
- > 4000 Applications and Utilities

Business Benefit:
Extends reach of rich internet services to role-based business centric composite applications for improved flexibility and responsiveness.



Catalog Support for Google Gadgets

- **Only available on the Catalog (<http://catalog.lotus.com/wps/portal/portal>)**
 - Does NOT ship with Portal but a download link is provided (there will be version for WebSphere Portal 6.1).
- **Integrates Google Gadgets into portal pages**
 - Google Gadgets integrated in WebSphere Portal will behave like local portlets
 - Viewable and customizable like any local portlet
 - Why? Large amount of user contributed content available via Google Gadgets instantly available from one portlet
- **Administrators can pre-configure Gadget Portlets for the portlet palette**
 - Generic Gadget Portlet is pre-configured by the admin to connect it to a certain gadget,
e.g. an admin could create a “Map Portlet” by creating a Gadget Portlet and connecting it to the Google Maps Gadget
 - Users can select such pre-configured Gadget Portlets and drag them onto their pages like any local portlet
- **If allowed by admin, users can drag unconfigured Gadget Portlets on their pages and select Gadgets from the Google Gadget Catalog**
 - An unconfigured Gadget Portlet initially lets user select Gadget to display from the Gadget Catalog
 - The Gadget Portlet then displays the selected Gadget
 - User can view and customize the selected gadget like any local portlet



..and even more Web 2.0 in V 6.1

- **Live Text** (fka Semantic Tags) provides smart markup value add by portal, e.g. dynamic menus
- **REST Services** open up portal for mashup applications
server persistence, portlet settings, user profile access simplify Web 2.0 applications
- **AJAX Portlet Programming Model Extensions** based on Dojo+IBM Extensions
- **Client Side Aggregation and Customization** using *REST Services* for better UX and improved performance
- **AJAX Client Side Feed Consumption** for highly efficient integration of information through feeds (Atom and RSS)
- **Client Side C2A / Property Broker and Drag & Drop** based on *Semantic Tags* integrated with server side property broker and C2A support to enable cross-portlet interaction locally in the browser as well as with server side code
- **Sample AJAX Portlets** with source showcasing new capabilities and demonstrate how to exploit all the above capabilities.



Portal REST Services

- All REST Services use the ATOM format and APP (Atom Publishing Protocol) for Write access.
- Navigation/Content Model
 - Pages, site structure
 - Corresponds to navigation loop JSP tag in the theme JSPs
- Layout Model
 - Layout of the page (rows, columns)
 - Corresponds to the UnlayeredContainer-H.jsp and UnlayeredContainer-V.jsp in the skin.
- Portlet Fragment
 - Rendering individual portlets
- Portlet Model
 - Gives access to the portlet definition, portlet preferences
 - Preferred way to access is the Client-Side Programming Model
- User Profile
 - Access user information

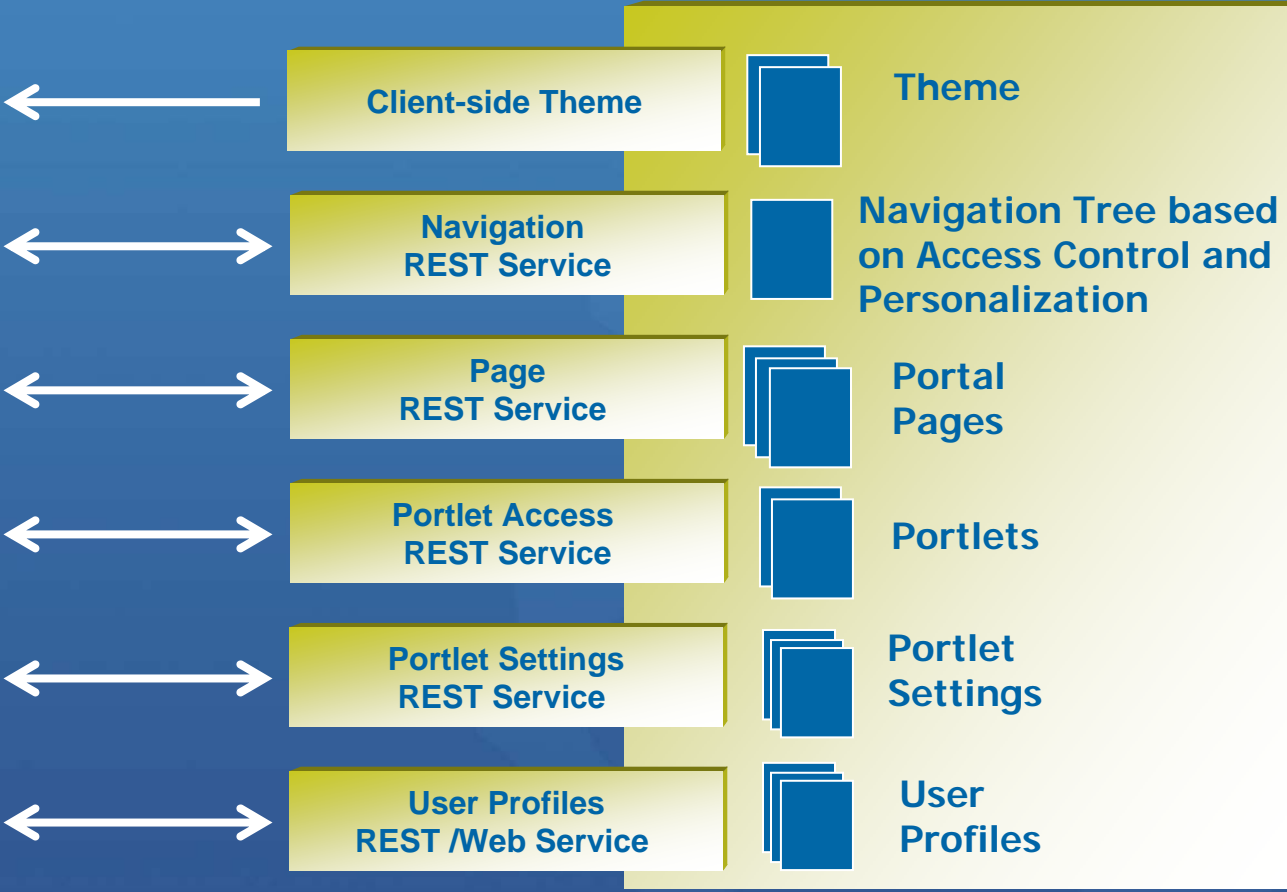


Client Side Aggregation

Customer and Third Party Applications or Mashups



Client and Server Side aggregation



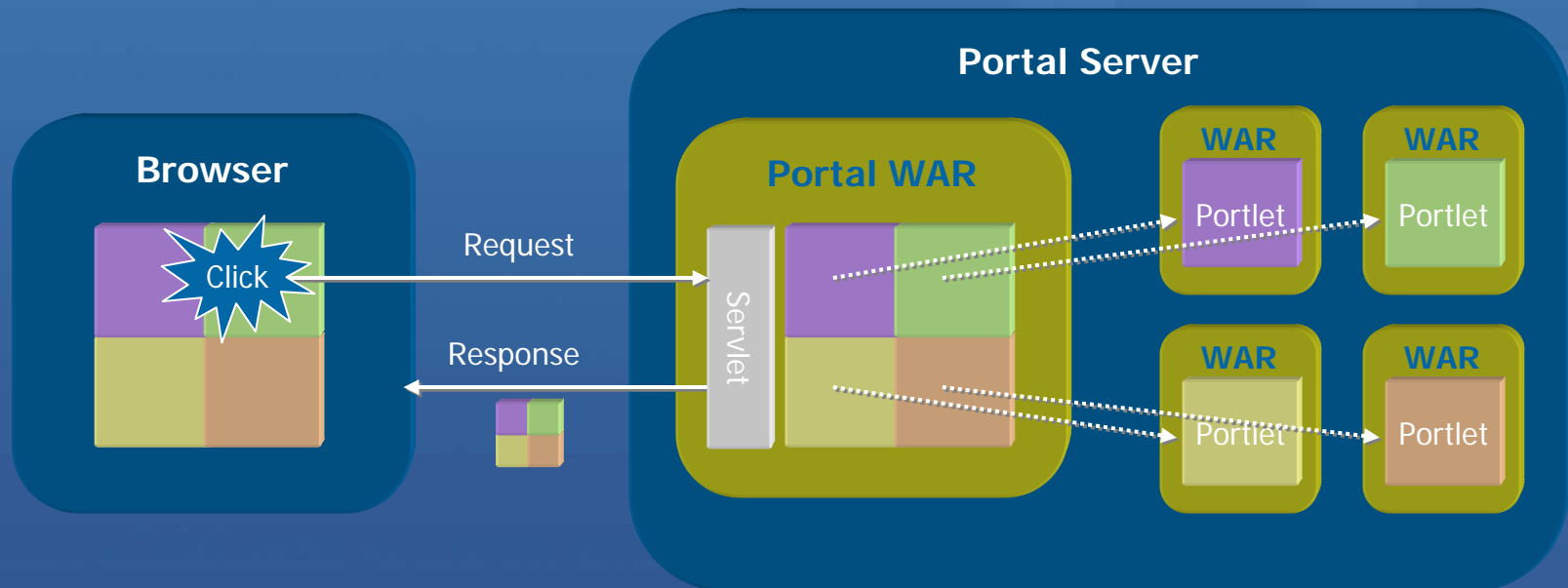
Client Side Aggregation

- ① **Sample Mash-up Application using the Portal REST Services**
- ① **Browser-side Page Aggregation, Navigation and Customization**
- ① **Superior user experience**
 - Highly reactive and direct user interface
 - Many actions possible without server roundtrips
 - Avoids page flickering
- ① **Accesses and manipulates portal information through REST services**
- ① **Renders XML obtained from the server on the browser side**
- ① **Implemented using AJAX, XSLT, and JavaScript**
- ① **Improved performance and scalability through**
 - Reduced server side processing – offloads rendering to browser
 - Reduced bandwidth requirements between server and browser
 - Reduced client-side processing – mostly fragment reloads, few page reloads
 - Improved cachability – all artifacts can be cached independently



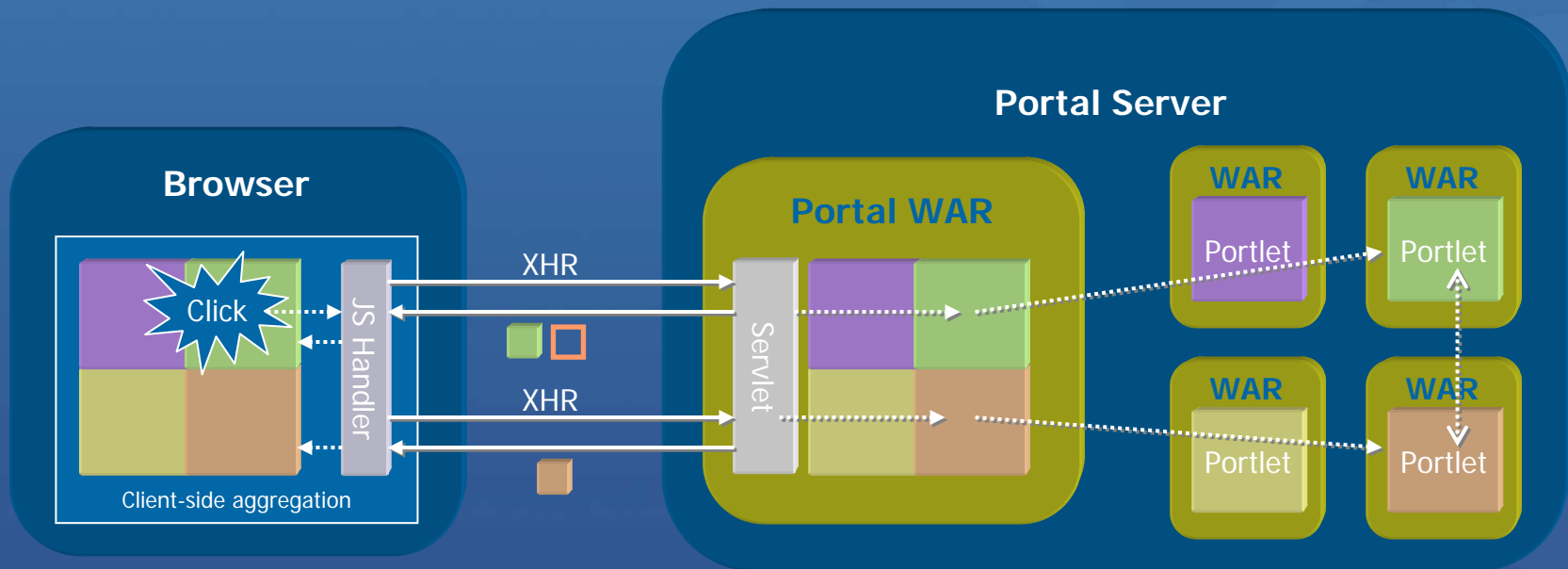
The Classic Portal Interaction Model

- Each request goes to the main portal servlet
- Each interaction causes a full page refresh
 - Even if the portlet does not communicate with other portlets



The Client Side Aggregation Portal Interaction Model

- **AJAX handling implemented entirely by the portal**
 - AJAX-enabled portal refreshes only those portlets affected by the user interaction



Client Side Aggregation (CSA) Features

- Only the “changed” portion of the page is updated as a result of a user interaction with the server
 - The traditional model requires the entire page to be re-rendered when a user interaction requires an interaction with the server (full server roundtrip)
 - Only the updated portlet has to render, instead of every portlet on the page re-rendering (especially useful if one portlet is particularly slow).
 - Results in a snappier feel to the user
- Navigational State is maintained on the client
 - Back button can still be supported in most cases (80/20 rule)
 - Bookmarkability is supported via a “Create Link to this Page” link (similar to how Google Maps works)
 - ✓ Current navigational state has to be serialized into a URL that can be entered into the browser address bar
- Enhanced Skin
 - Inline mode support
 - ✓ Make changes in Edit mode and see the results immediately
- Client-side Drag and Drop
 - No full page refresh when dropping a portlet on to the page or re-arranging portlets on a page.



CSA in Action

The screenshot shows a web browser window displaying the IBA WebSphere Portal. The browser's address bar shows a local host URL. The portal's navigation bar includes links for Home, Administration, Applications, Search Center, and More. The user is logged in as 'wpaadmin'. The main content area is titled 'CSA' and contains several portlets:

- JSR 286 Supplier Detail Portlet:** A table listing suppliers with columns for 'Supplier id' and 'description'.

Supplier id	description
883-2217	Quality Components Supply Inc.
735-4967	QualTec Wholesale Inc.
478-4035	Microparts Wholesale Inc.
303-7885	Tech Supply Inc.
- JSR 286 Reorder Portlet:** A form for reordering parts. It includes a 'supplier information' section with a dropdown menu, 'email' and 'phone' input fields, and a table for 'part id', 'material', 'price per piece', and 'quantity'. A 'finish order' button is located at the bottom of the form.
- JSR 286 Part Detail Portlet:** A table listing parts with columns for 'Part_ID' and 'description'.

Part_ID	description
TR-123002	SOME DESCRIPTION
AC-288399	SOME DESCRIPTION
UV-442892	SOME DESCRIPTION
88-019008	SOME DESCRIPTION
OR-001834	SOME DESCRIPTION
AB-299011	SOME DESCRIPTION
BB-128933	SOME DESCRIPTION

A large 'Loading...' indicator is visible in the top-left area of the main content, suggesting that some data or components are still being fetched.

Enhanced Web 2.0 Skin in Action

'Actions' toggles Portlet Mode and Delete actions

Portlet Title

Personalize **Configure** Edit Settings Delete

- Lorem ipsum dolor sit amet, consectetur adipiscing elit.
- Pellentesque urna. Integer mi.
- Pellentesque eget ante. Proin non neque. Cras cursus
- Etiam dictum augue vel felis. Ut mauris.
- Morbi orci. Proin turpis lectus, pulvinar imperdiet, pretium aliquet, placerat a, risus.
- Etiam hendrerit justo eget diam.
- Sed tristique, odio dapibus vehicula cursus, nisi lorem

Selecting a mode will render that mode inline

Portlet Title

Personalize **Configure** Edit Settings Delete

To edit the content displayed in this portlet, choose a new feed from the list of popular feeds -or- if you know the URL of the feed you wish to display, type it in the text field below.

of feed items displayed in portlet: 10

Feed Title: CNN Save

Feed URL: http://www.cnn.com Save

Popular Feeds: None Save

- Lorem ipsum dolor sit amet, consectetur adipiscing elit.
- Pellentesque urna. Integer mi.
- Pellentesque eget ante. Proin non neque. Cras cursus
- Etiam dictum augue vel felis. Ut mauris.
- Morbi orci. Proin turpis lectus, pulvinar imperdiet, pretium aliquet, placerat a, risus.
- Etiam hendrerit justo eget diam.
- Sed tristique, odio dapibus vehicula cursus, nisi lorem

Note that view mode is still displayed



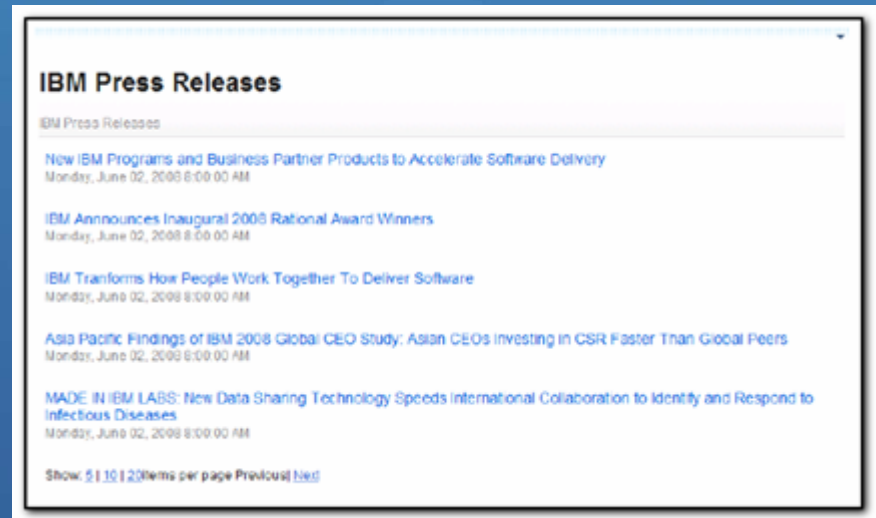
Client Side Aggregation (CSA) Limitations

- Client Side Aggregation is really a bleeding edge technology and pushes the browser to the limit. As with any bleeding edge technology, there are some limitations. This is especially true given the scope of what we support in Portal.
- Not every portlet works in CSA
- CSA supports a forced SSA (Server-Side Aggregation) mode for cases that do not work
 - Can be applied on a page or a portlet level
 - ✓ Page Level: Assign a page parameter (a.k.a. Meta Data attribute) to a particular page
 - ✓ Portlet Level: Add the portlet application id to a javascript file in the theme OR add a config level portlet preference
- Back button only works until the next full page refresh
 - Uses Dojo's backbutton support so subject to the same limitations as Dojo
 - Waiting on browser/standards support to catch up with the needs of the web
- Different interaction paradigm
 - Partial page update is normal, as opposed to a full page refresh for every portlet interaction
 - Dojo widget parser must be invoked manually
 - Can't rely on a page refresh to clear out global javascript variables



AJAX-based RSS /Atom Feed Consumption

- Allow simple consumption and display of **Feeds in portal pages**
 - Atom feeds
 - RSS feeds
- Implemented using AJAX, Dojo and JavaScript
- Gets settings defining the feed to display from portal
- Retrieves feeds from origin servers via **AJAX proxy**
- **Renders feeds in the browser** rather than causing server load



WebSphere Portlet

- WebSphere Portal v6.1 針對Web 2.0 新增功能
- 如何設計 **Ajax** 應用程式
- WebSphere Portlet Factory 6.1 新增功能



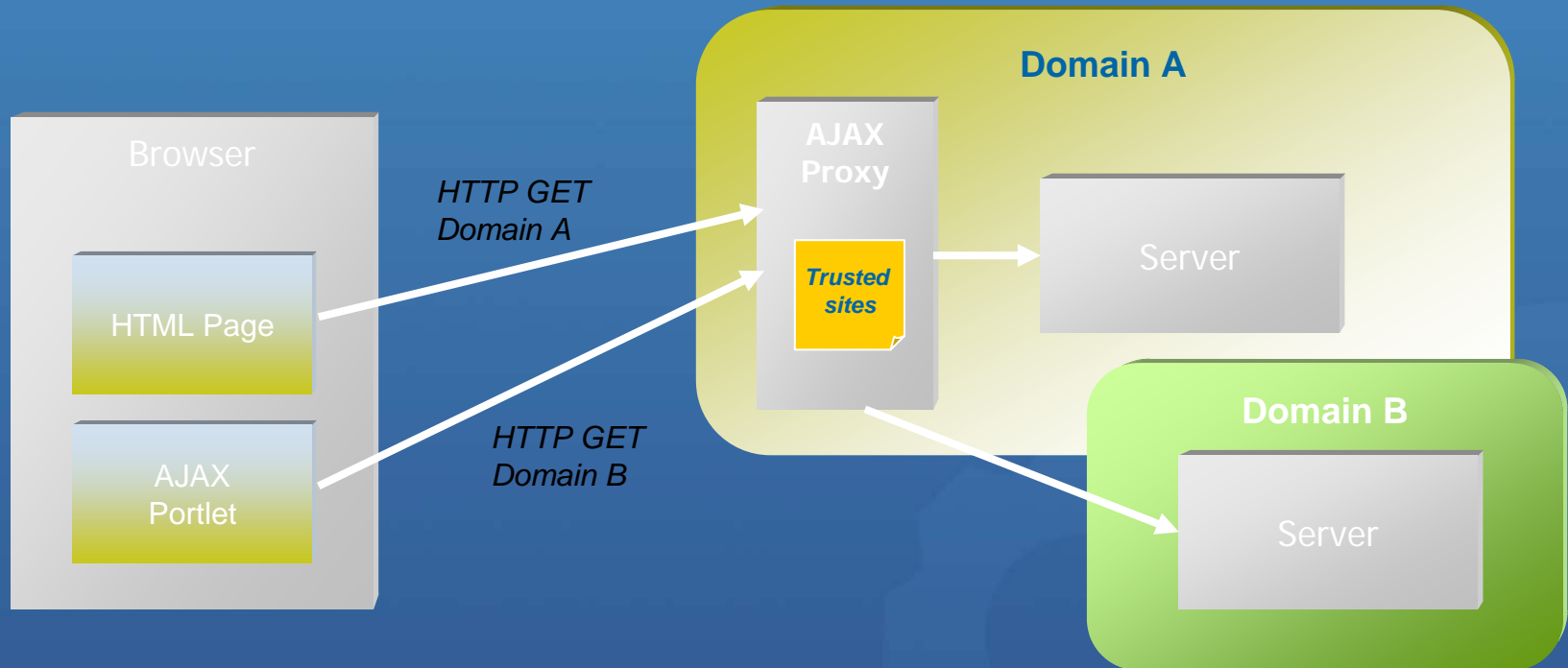
Ajax Portlet Development

- Portal 6.1 introduces several new capabilities to make AJAX development in portlets much easier.
 - AJAX Proxy
 - ✓ Supports cross-domain requests
 - ✓ Easily administered
 - JSR 286 Portlet support
 - ✓ Adds ResourceRequests for retrieving Portlet resources without the full Portal aggregation
 - ✓ Navigational State can not be altered
 - Client Side Programming Model support
 - ✓ Adds easy to use APIs for accessing portlet preferences, user profile information, and maintaining navigational state on the client



AJAX Proxy

AJAX proxy: Central security component to manage access to other domains



- Today's browsers restrict the functionality of asynchronous requests to the same domain because of security reasons.
 - Example: Your portlet is served from `www.mycompany.com` but your AJAX application tries to load a feed from `cnn.com`. This would be blocked from the browser



JSR 286 Portlet Support

- **Java Portlet Specification 1.0 (JSR 168) aimed at a 60% target**
 - Support most common use cases
 - Agree on basic functionality quickly
 - Defer advanced functionality to later standard versions
- **More functionality in Java Portlet Specification 2.0 (JSR 286)**
 - <http://jcp.org/en/jsr/detail?id=286>
 - Equivalent standards update for remote portlets: WSRP 2.0
 - <http://docs.oasis-open.org/wsrp/v2/wsrp-2.0-spec.html>
- **Key capabilities introduced with JSR 286**
 - Events
 - ✓ Enable portlets to communicate with each other through sending and receiving events
 - Shared render parameters
 - ✓ Enable portlets to specify which render parameters they can share with other portlets
 - **Resource Serving**
 - ✓ Enables portlets to serve resources within the portlet context using resource requests

Key AJAX Feature



Client Side Programming Model support

Client-side portlet programming model

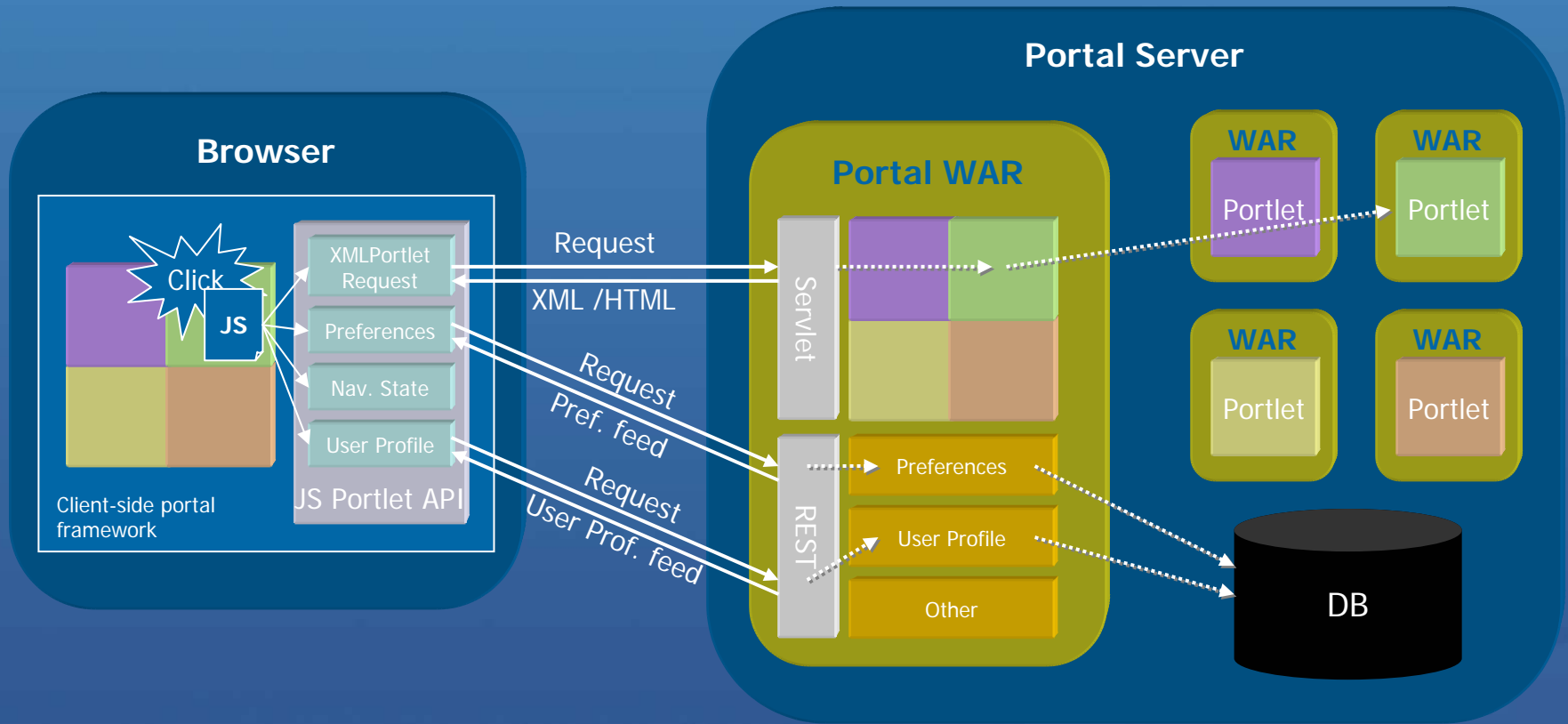
- Convenience JavaScript APIs simplifying AJAX portlet development
- Client-side API similar to the Java Portlet API
- Coordinates AJAX calls with the portal
 - ✓ Consistent behavior after a full page refresh
 - ✓ Navigational state changes

Functionality

- XMLPortletRequest
- Read & write navigational state (mode, window state, render parameters)
- Read & write portlet preferences
- Read user profile information
- Report errors to the portal framework



Client Side Programming Model support (part 2)



LiveText

- Add semantic keywords to markup to simplify information discovery, retrieval, and navigation
- Semantic Web vs. Collaborative Tagging (del.icio.us, flickr)
- **Microformats (<http://microformats.org/>)**
 - Set of simple data formats
 - Built upon existing standards (XHTML)
 - Available for many areas /domains
 - ✓ People and organizations
 - ✓ Calendar events
 - ✓ Rating and opinions
 - ✓ Social networking
- Allows for Client Side Click to Action

```
<div class="vevent">  
  <span class="summary">WebSphere Portal  
  Technical Conference 2007</span>  
  <abbr class="dtstart" title="2007-09-  
10">September 10</abbr>-  
  <abbr class="dtend" title="2007-09-  
12">12</abbr>  
  <span class="location">Hilton Munich</span>  
</div>
```



LiveText in Action

Sample People Tag Portlet

This is a demo portlet showing the people tag capabilities.
Click the names below for further options

[Thomas Schaeck](#)

[Stephan Hesmer](#) [found online]

IBM
wor
Bo
Ge
+4
+4



Stephan Hesmer

stephan.hesmer@de.ibm.com

WebSphere Portal - Performance Chief

Developer - WAS Technical Focalpoint

49-7031-16-3275

Show Profile

Send E-mail

Go to Blog...

Send Instant Message...

Google Map...



C2A for Standard Portlets – Live Text

HTML Markup is enriched by tagging interesting areas in the Markup, these parts become like (live) objects a user can interact with.

1. An Order Summary Portlet lists a set of active orders for the current month. (Each Order Id is represented as a “live object”)
2. The user clicks on the icon next to an Order Id.
3. A **menu appears** with a header that displays information about the item ordered and a photo of it.
 - ✓ Below the header are menu items which represent actions on the Order Id. The list of menu actions depend on which target Portlets are on the same page.
4. The **user clicks on a menu item** to execute an action.
5. **Another Portlet** on the page updates to display the result of that action.

The screenshot shows a web application interface. On the left, there is a table titled "Standard Orders" with columns "Order_ID", "Customer_ID", and "Status". The table lists three orders. The first order (ID: 112001000000) is "DELAYED". The second order (ID: 112001000001) is "ON_TIME". The third order (ID: 112001000002) is selected, and a menu overlay is shown for it. The menu header displays a gear icon, the text "Small Gear (112001000001)", and "Availability: in stock". Below the header are two menu items: "Show Account Details" and "Show Order Details". To the right of the table is a "Standard Order Detail" portlet showing the selected order's details: Order_ID: 112001000001, SKU: AB-299011, Quantity: 200, Status: ON_TIME. It includes an "Enter order id:" input field and a "Submit" button. Below this is an "Account Detail" portlet with an input field and a "Submit" button.

```
<div class="c2a:source someotherclass">
  <span class="c2a:typename"
    style="display:none">http://...datatype#email822</span>
  <b class="c2a:value">johndoe@acme.com</b>
  <p class="c2a:header" style="display:none;">
    <b><c>This is a sample C2A source</c></b><br>
    <b><c>You can add an optional header </c></b>
  </p>
Source
```

```
</div>
<FORM class="c2a:target" onsubmit="doSomething(this);return
false" action="javascript:void(0)" style="display:none">
  <span class="c2a:typename">
    http://www.ibm.com/xmlns/prod/datatype#email822</span>
  <p class="c2a:action-label">Show inbox</p>
  <input type="text" class="c2a:paramname"></input>
</FORM>
Target
```

Client Side Click-to-Action

1. Hover UI shows when user moves mouse over C2A source.

The screenshot shows two portlets: 'ColorsSource portlet' and 'ColorsTarget portlet'. The 'ColorsSource portlet' contains a list of color links: blue, silver, maroon, white, cyan, red, gold, green, black. A tooltip box labeled 'Click for Actions' is positioned over the 'cyan' link. The 'ColorsTarget portlet' displays the message 'The color value has not been set.' and a list of radio button options: 'Enable all targets' (selected), 'Disable all targets', 'Enable server-side action only', and 'Enable client-side action only'. An 'Apply' button is at the bottom.

2. C2A menu appears when user clicks on the Hover UI.

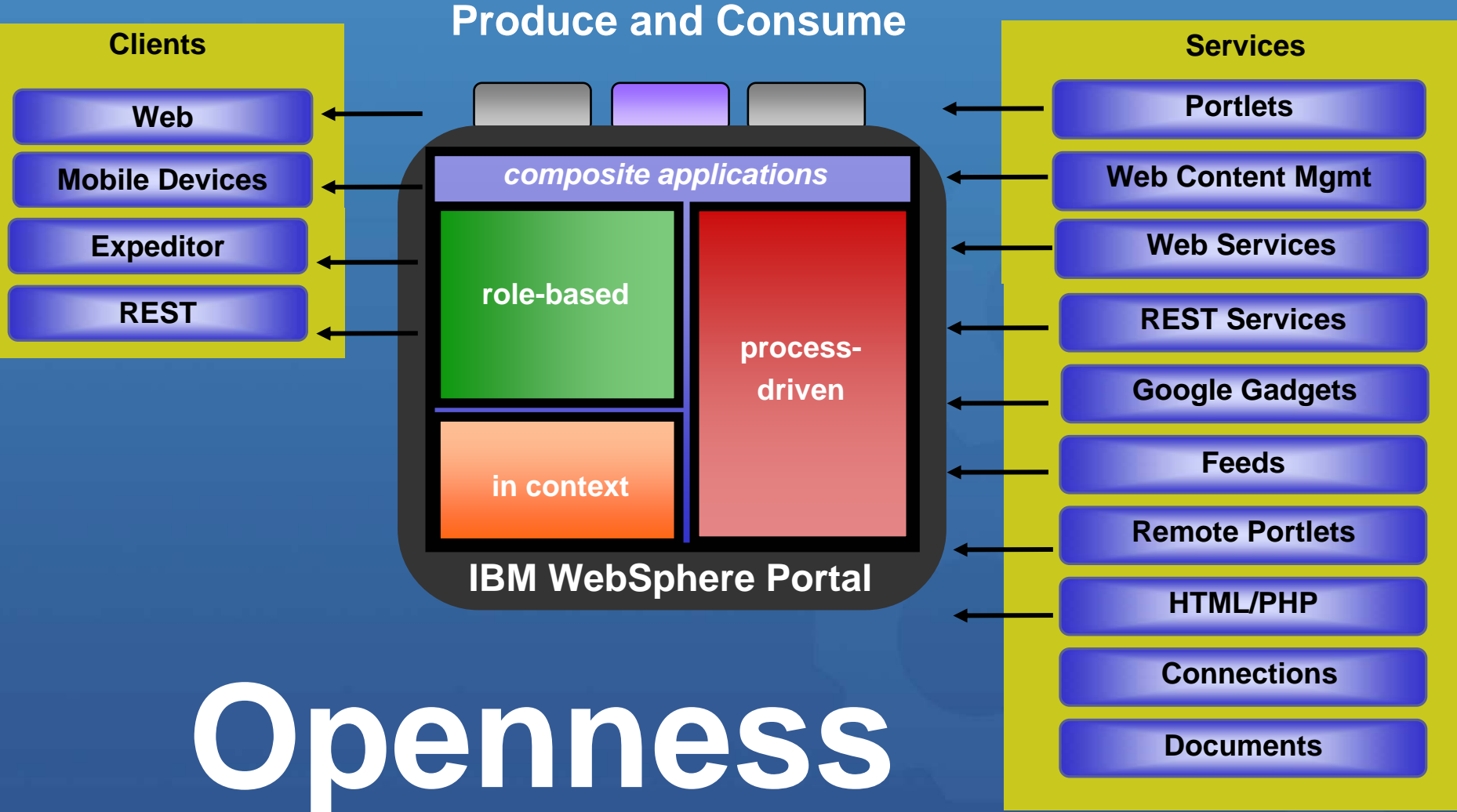
This screenshot shows a tooltip menu over the 'cyan' link. The menu contains the following items: 'name: cyan', 'RGB value: rgb(0,255,255)', and 'what does it look like?' (with a cyan color swatch). Below these are three action options: 'Change color using CLIENT-side C2A action', 'Change color using SERVER-side C2A action', and 'Invoke all Actions'. The 'ColorsTarget portlet' on the right shows the message 'The color value has not been set.' and the same radio button options as in the previous screenshot.

3. Result of C2A action shown in target portlet

The screenshot shows the 'ColorsSource portlet' with the 'cyan' link selected. The 'ColorsTarget portlet' now displays the message 'This color (ie, cyan) was applied by client-side C2A - no server request was issued.' and a large cyan color swatch. The radio button options and 'Apply' button remain the same as in the previous screenshots.



WebSphere Portal 6.1除了Web 2.0外的新增網站功能



Openness

Standards based access to integration and innovation



Site Construction and Customization

Improved control for site owners by new Portal capabilities.

Enhanced URL management and improve readability of URLs

- Navigation Link:
www.ibm.com/wps/myportal/MyPlaces/Welcome
- Action Link:
www.ibm.com/wps/myportal/MyPlaces/Welcome/!st/04_SB8K8xLLS..

Enhanced Virtual Portal Support

- Allow virtual portal identification by the hostname or a path segment
 - ✓ www.myVP1.ibm.com, www.myVP2.ibm.com
- Configurable per VP if identification is encoded as part of the hostname or the path

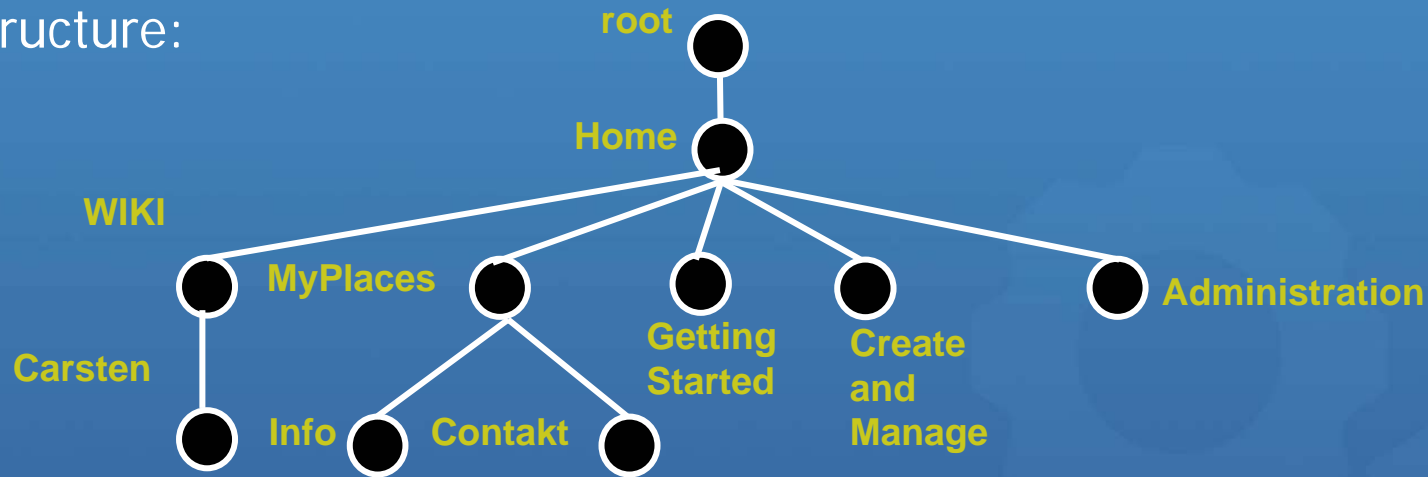
HTML Page Aggregation

- Deploy HTML pages to Portal (incl. all needed resources) and use it in your Portal site...
 - ✓ ...protected by Portal entitlements...
 - ✓ ...as full Page...or...decorated with banner and navigation...
- ...and use Portlets on your HTML pages if you want.



Friendly URLs - Example

Site Structure:



- By clicking on MyPlaces (link in the navigation), for portal pages* the URL in the browser will be:

`www.somecompany.com/wps/portal/Home/MyPlaces`

- By adding /Info in the browser address bar the URL in browser would be:

`www.somecompany.com/lotus/quickr/Home/MyPlaces/Info`

- By clicking links/actions in the portlets, the URL in the browser is readable up until the rich URL part, and could be e.g.:

`www.somecompany.com/lotus/quickr/Home/MyPlaces/Info/!ut/p/04_SB8K8xLLM9..`

* special cases for static pages and disambiguation scenarios will use URLs with appended state.

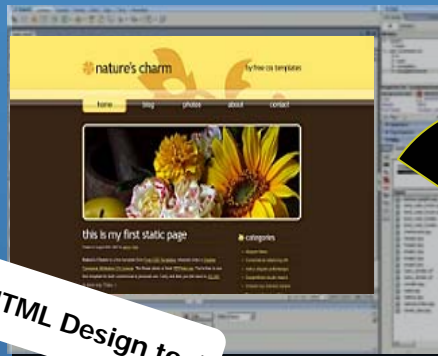


HTML Page Support

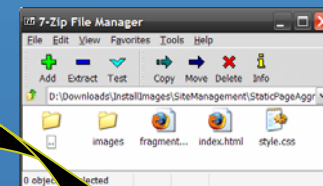
Simplify creation of portal pages

Use HTML design tools and publish to WebSphere Portal

Enable your designers to create smart HTML pages using live text in WebSphere Portal



HTML Design tool



WebSphere Portal

Unleash the power of your web design teams



Render HTML Pages within WebSphere Portal

*Create HTML pages in your preferred tool and upload them into Portal...
...optionally use Portal Navigation, Portlets, and more on your HTML Pages.*



g.

Use of the <HTML> tag within the HTML pages defines the rendering mode.

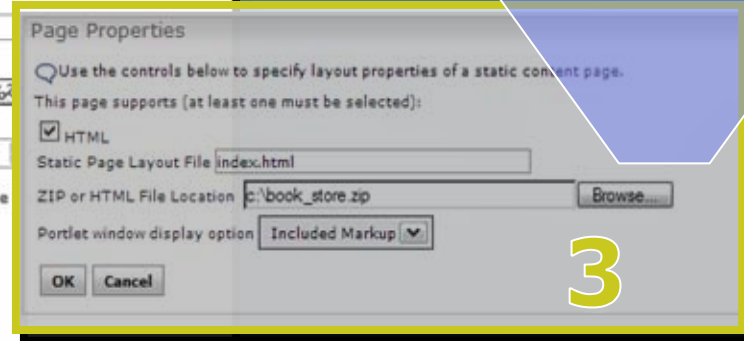
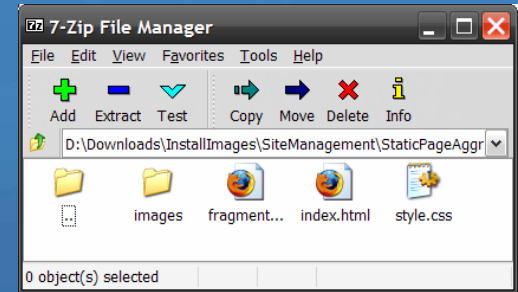
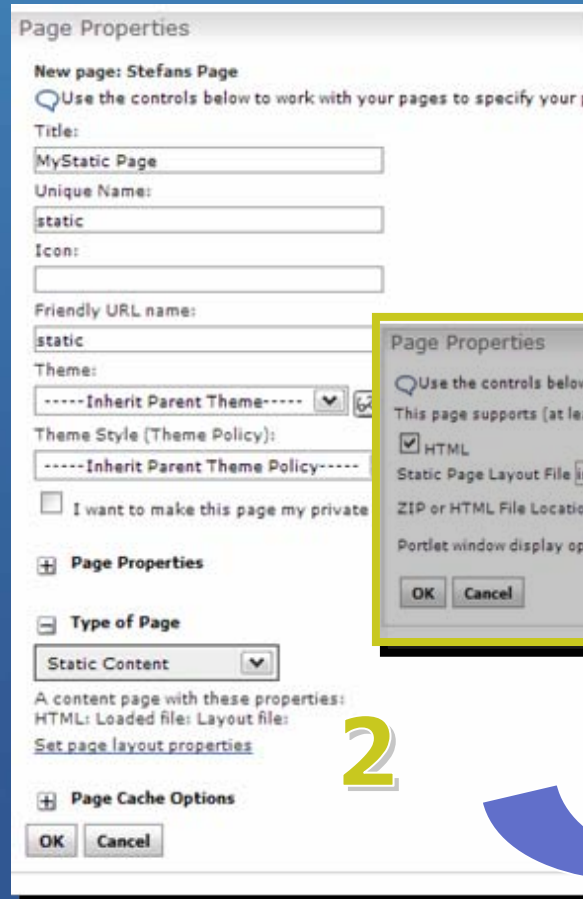
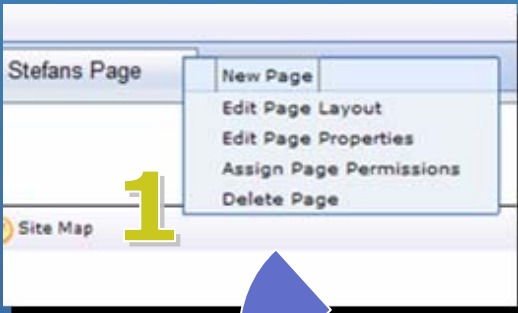
Technical Capabilities:

- ✓ Any HTML page can be imported into Portal
- ✓ HTML pages (full page /embedded page) can be used side-by-side with “portal layout pages”
- ✓ Different modes of portlet rendering can be specified (Server Side, AJAX or iFrames)
- ✓ Support for multiple languages or markup versions of the page
- ✓ Include any web resources (images, java script files, style-sheets) used by the page.



Defining a Portal Page using HTML

Import your packaged HTML Page and select the HTML page to be rendered by WebSphere Portal.



Add dynamic elements to the HTML page

- Portlets are referenced via a microformat in the HTML source code
- The server parses the HTML and replaces the microformat with the portlet markup
 - directly on the server
 - in a way that JS in the page can reload portlet fragments (AJAX)
 - as an IFrame
- WP 6.1 ships a navigation portlet and a breadcrumb trail portlet out of the box

The screenshot shows a Mozilla Firefox browser window displaying a web page titled 'supplementary by free css templates'. The page features a search bar, a navigation menu, and a portlet titled 'ibm press releases'. The portlet content includes two news items:

ibm press releases

ibm teams with michigan technological university to prepare students for jobs of the future
 Donnerstag, 5. Juni 2008 14:00:00 ARMONK, NY (MARKET WIRE) IBM (NYSE: IBM) and Michigan Technological University today announced several initiatives to help students develop the multi-disciplinary skills required in a growing number of jobs and professions around the world.

intermec selects ibm cognos 8 controller for financial consolidation & corporate reporting
 Mittwoch, 4. Juni 2008 14:00:00 Cognos, an IBM company and the world leader in business intelligence and performance management solutions, today announced that Intermec has selected IBM Cognos 8 Controller to automate and manage its financial consolidation and corporate reporting processes. A developer and manufacturer of technologies that identify, track, and manage supply chain assets, Intermec is using IBM Cognos 8 Controller to consolidate, certify and report on financial data from its 25 subsidiaries.

The word 'Portlet' is written in large white text on the right side of the image, with a red box highlighting the portlet content area.

Reference static resources

- Images, CSS and JS are referenced via relative URLs in the static HTML
- These images are either served out of the ZIP file of the page ...
- ... or out of the portal theme, which allows to share the resources across pages
 - important e.g. for large JS libraries such as DOJO

- URLs to resources do not require to use any special API!**

Shared DOJO copy from the theme

JS and CSS files packaged with the static page

```
<script type="text/javascript" src="dojo/dojo.js"
  djConfig="parseOnLoad: true">
</script>

<script type="text/javascript" src="js/init.js">
</script>

<link rel="stylesheet" media="screen" href="style.css" />
```

Reference other pages (navigation links)

- Direct URLs if the friendly name of the target page is known
- POC URLs if the unique name of the target page is known
- By referencing the „navigation link portlet“ for a completely dynamic navigation inside a static page

The navigation portlet renders a „navigation microformat“ that can be styled via the CSS of the static page

The image shows a screenshot of a web browser displaying a search results page for 'supplementary by free css templates'. The browser window title is 'Supplementary by Free Css Templates - Mozilla Firefox'. The search results show 'ibm press releases' with several articles listed. A navigation portlet is overlaid on the page, showing a list of links:

- navigation
- Getting Started
- Web 2.0 Introduction
- Feeds
- Gadgets
- CTF Test Page
- CTF Legacy Test Page
- Static Page 1
- Static Page 2
- Static Page 3
- Dynamic Page

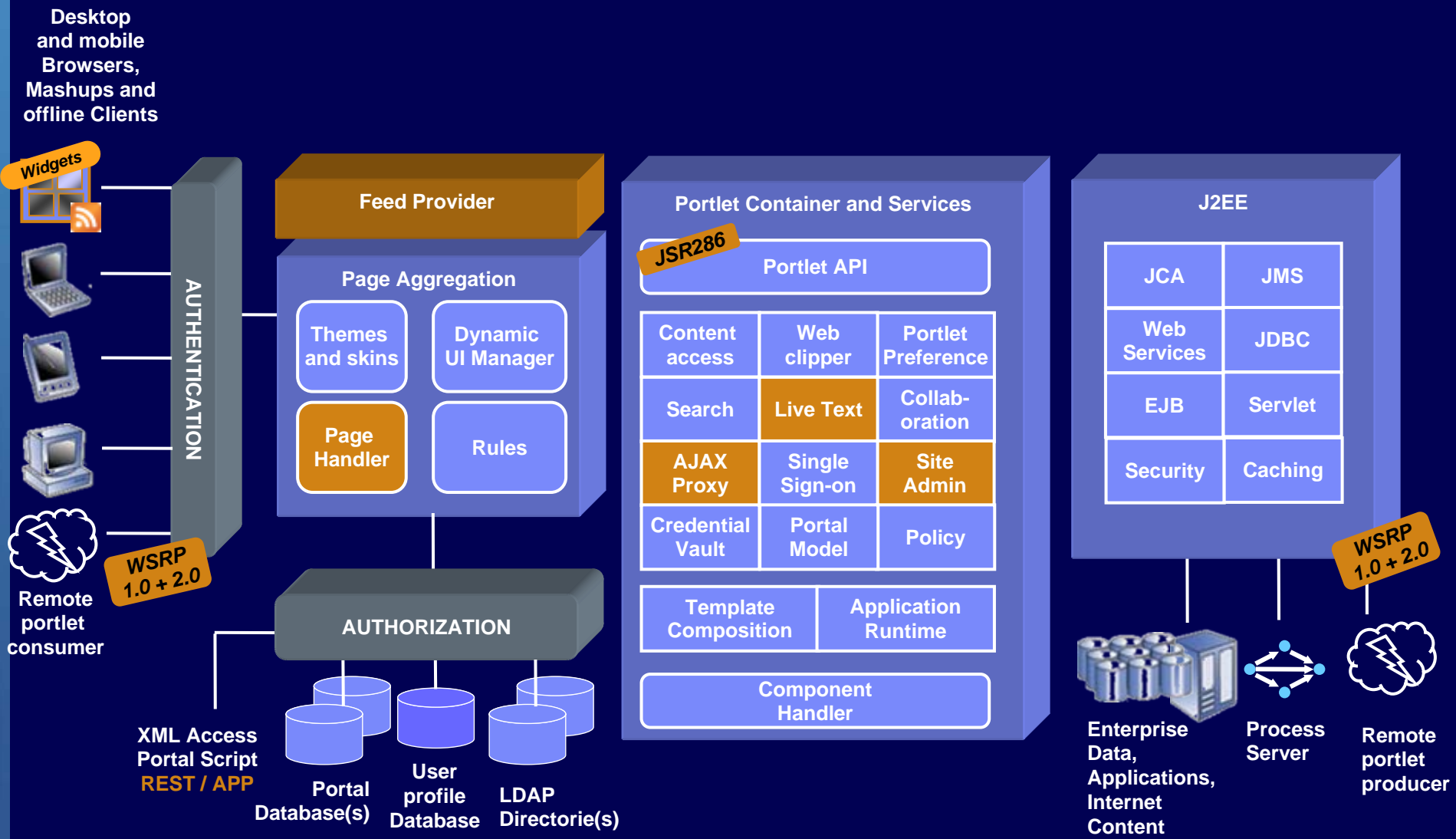
The portlet also includes a 'quick links' section and a 'home' link.

Operational Aspects

- All resources for a static page are packaged and deployed via a simple ZIP file
- The ZIP file is stored in the portal DB
 - Automatic backup/restore
 - Tight integration into staging-to-production and migration
 - ZIP is part of XMLAccess exports and SiteManagement ATOM feeds
- **Storing the ZIP in the DB limits its size to 1MB per page**



Architecture Overview WebSphere Portal V6.1 Beta



Resource Serving for Portlets

- **New callback interface triggered by “Resource URLs”**
 - `serveResource(ResourceRequest, ResourceResponse)`
- **Allow portlets to render stand-alone content**
 - No portal page aggregation, no theme content
 - Portlet has full control of the entire output, and access to all APIs and state data (e.g. portlet session)
 - Generate binary content (PDF, SVG etc.)
 - Pop-up windows
- **Allows for limited AJAX support inside portlets**
 - Portlets now have a way to return XML, JSON, HTML fragments or other content
 - Better programming model than packaging a servlet in the same WAR file
 - Can not modify the navigational state or send events!



Resource Serving in the Portal Context

- The “Resource Addressability Framework” allows to expose portal managed data as REST services
 - Custom servlets cannot make use of portal APIs or support virtual portals
 - JSR286 resource serving requires a portlet
- Resources are identified via URIs, the REST services can be addressed using well defined URLs or via JAVA APIs
- Customers can deploy their own feeds using the Eclipse plugin model
- Example: WP content model URI = `cm:oid:wp.content.root`
- <http://host:port/wps/poc?uri=cm:oid:wp.content.root&mode=download>
- <http://host:port/wps/poc/cm/oid:wp.content.root?mode=download>



Deployment of Custom REST feeds

Plugin.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<?eclipse version="3.0"?>
<plugin id="com.ibm.wps.resolver"
  name="My Implementation" version="1.0.0"
  provider-name="IBM">
  <extension
    point="com.ibm.content.operations.registry.locationTypeContribution">
    <contentLocationType
      class="com.ibm.portal.resolver.helper.cor.DefaultContentLocationFactory"
      id="com.ibm.portal.MyLocation" match.uri.scheme="MyScheme" />
    </extension>
  <extension
    point="com.ibm.content.operations.registry.locationServiceHandler">
    <serviceHandler
      class="MyClassName"
      locationTypeId="com.ibm.portal.MyLocation"
      id="com.ibm.portal.resolver.data.DataSourceFactoryEx" />
    </extension>
  </plugin>
```

URI scheme

Type of Service



What the framework does for you

- Simple deployment model for services based on the Eclipse Registry
 - Extensions can e.g. be packaged in WAR files and deployed dynamically ..
 - ... or via a shared library
- Handles the serialization of XML- , character- and multipart streams automatically (*if desired*)
 - integrates with the efficient SAX based ATOM APIs
 - support for multipart streams and server side XML mashups, etc...
- Takes care of HTTP details
 - computes cache headers, handles conditional GETs, ...
 - avoids that the vary header breaks browser caching!
 - implements support for content disposition, byte range requests, efficient transfer encodings etc
- **This framework is the basis for all of the WP 6.1 Web2.0 REST feeds – designed for efficiency and reusability!**



WebSphere Portlet

- WebSphere Portal v6.1 針對Web 2.0 新增功能
- 如何設計 Ajax 應用程式
- WebSphere Portlet Factory 6.1 新增功能



Improved Developer User Experience



Develop Applications Visually

- View and edit user interface components visually
- Provide suggested builders intelligently based on context

The screenshot shows the Eclipse IDE interface for a WebSphere Portlet Factory project. The 'Application Tree' on the left shows a hierarchy of pages: 'allEmployeesPage', 'getEmployeePage', and 'allEmployeesDataPage'. The 'Outline' view at the bottom left lists various components like 'Getting the records', 'main', 'EmployeeData', and 'allEmployeesDataPage'. The main editor displays a visual design of a page with a table and a 'Create' button. A context menu is open over the 'allEmployeesDataPage' component, showing options like 'Toggle Breakpoint', 'Disable Breakpoint', and 'Add Builder Call'. A sub-menu is also visible, listing builders such as 'Data Field Modifier', 'Data Hierarchy Modifier', 'Data Column Modifier', 'Dynamic Validation', and 'Form Layout'. Three green callout boxes point to these features: 'View Pages' points to the Application Tree, 'Edit Pages Visually' points to the main design editor, and 'Suggest Builders Based on Page Selection' points to the context menu.

View Pages

Edit Pages Visually

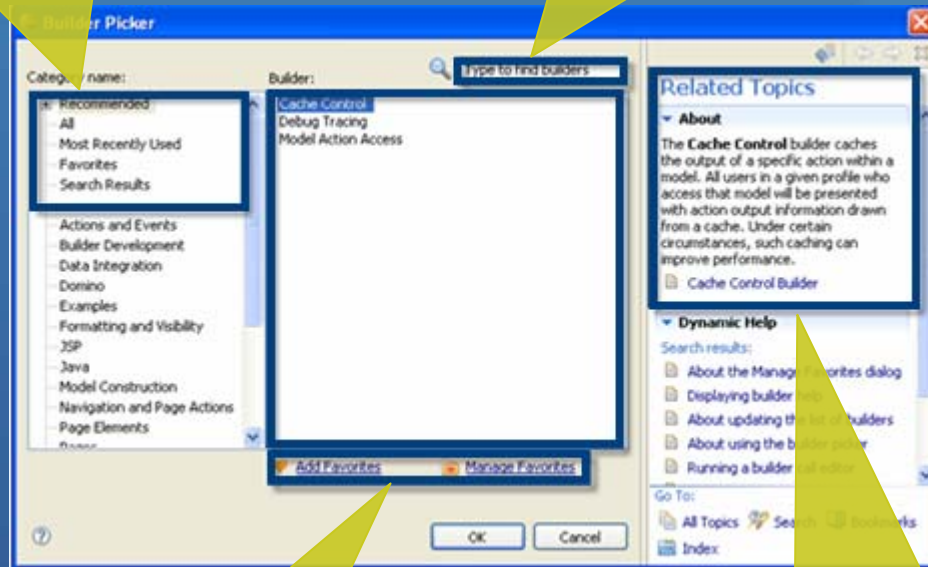
Suggest Builders Based on Page Selection

Create Applications Faster

- Add application functionality using streamlined builder picker

Display Recommended Builders

Search for Builders



Create and Manage Favorite Builder List

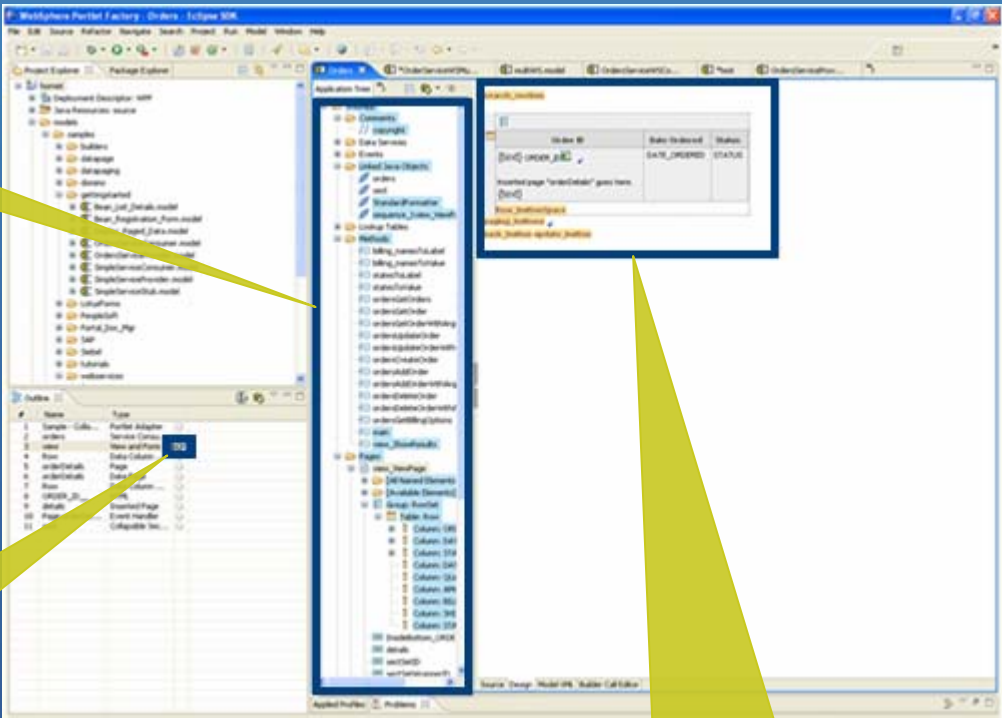
Display Documentation Dynamically Based on Builder Selection



Navigate Intuitively

Click the builder call link icon and automatically display the design or code view

Display WebApp Tree

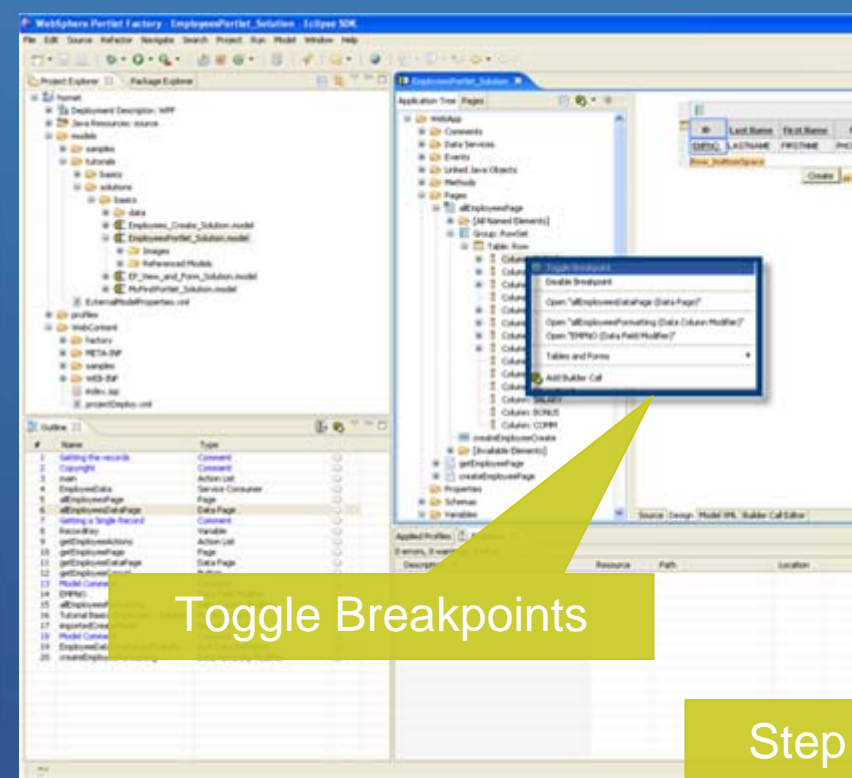


Click to Display Design View and Generated Code

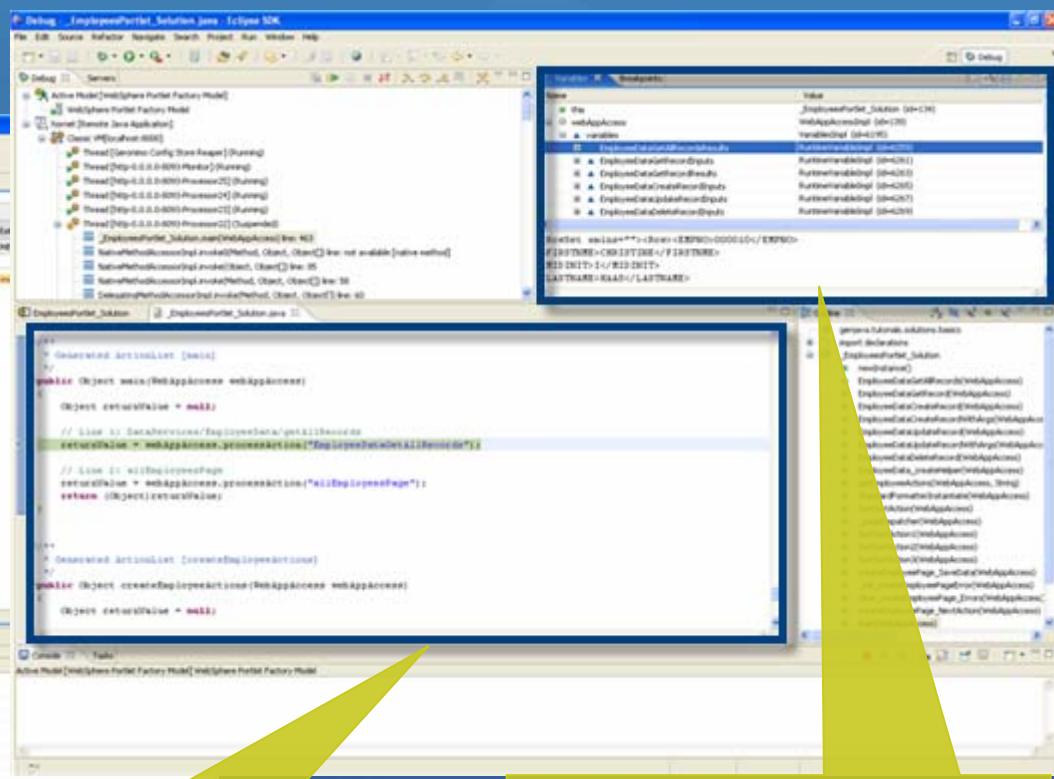
Display Design or Generated Code View

Debug Applications Easily

- Configure WAS CE in debug mode by default
- Set breakpoints on methods, pages, event handlers and data service operations
- Step through code
- Watch variables during execution



Toggle Breakpoints



View Variable Values

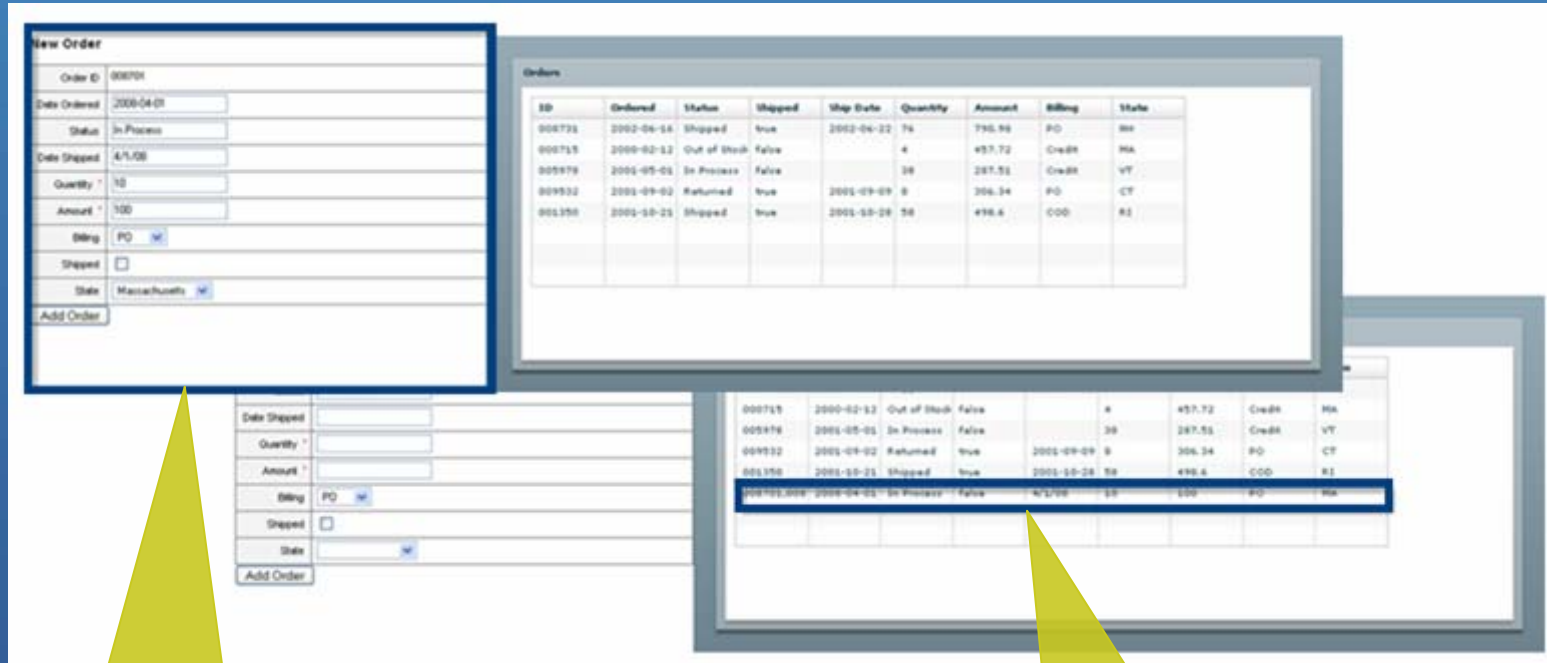
Step Through Code

Added New Builders



Integrate Flash in WPF Applications

- Embed Flash components in WPF applications
- Combine WPF and Flash application functionality



Enter Order in WPF Application

Update Order List in Flash Application



Expand and Collapse Data Details

- Display and hide detail information

Expand Selection

Collapse Detail

Order ID	Date Ordered	Status
<input checked="" type="checkbox"/> 008731	2002-06-16	Shipped
<input type="checkbox"/> 000715	2000-02-12	Out of Stock
<input type="checkbox"/> 005978	2001-05-01	In Process
<input type="checkbox"/> 009532	2001-09-02	Returned
<input type="checkbox"/> 001350	2001-10-21	Shipped

Order ID	Date Ordered	Status
<input checked="" type="checkbox"/> 008731	2002-06-16	Shipped
Status		Shipped
Date Shipped		2002-06-22
Quantity		76
Amount		790.98
Billing		PO
Shipped		true
State		New Hampshire
<input type="checkbox"/> 000715	2000-02-12	Out of Stock
<input type="checkbox"/> 005978	2001-05-01	In Process
<input type="checkbox"/> 009532	2001-09-02	Returned
<input type="checkbox"/> 001350	2001-10-21	Shipped

View Detail



Improved Performance



Create More Scalable Applications that Execute Faster

- Reduced session memory size
- Increased load capacity
- Enhanced performance statistic reporting to enable targeted tuning
- Added support for DynaCache
- Provided capability to set variable scope to session or request



Enhanced Web Services Functionality



Integrate Better with Web Services

- Expose all Web Service operations defined in a WSDL file with a single builder call
- Support WS-Security LTPA binary and Username tokens

Expose all Web Service Operations

Provide LTPA and Username Tokens

Web Service Multiple Operations
Provides access to all the operations for a web service defined in the specified WSDL file. Creates a new Service Definition with a Service Operation for each operation.

Properties

Name:

WSDL URL:

Create Service Definition:

Operations

Enable Caching of Service Results:

Refresh Intervals (sec):

Operation	Method Description	Soap Action
GetQuotes		http://wsparam.ibm.com/Service/BasicRealTimeQuotes/GetRemainingRights
GetOneQuote		
GetRemainingRights		

Request:

Variable Name	Argument Name	Argument Type
multWS_GetQu...	parameters	SIQuoteInfo

Reply:

Variable Name	Argument Name	Argument Type
multWS_GetQu...	parameters	SISubscriptionInfo

Service Information

Soap URL:

Service Description:

WS-Security

LTPA Binary Token:

Username Token:

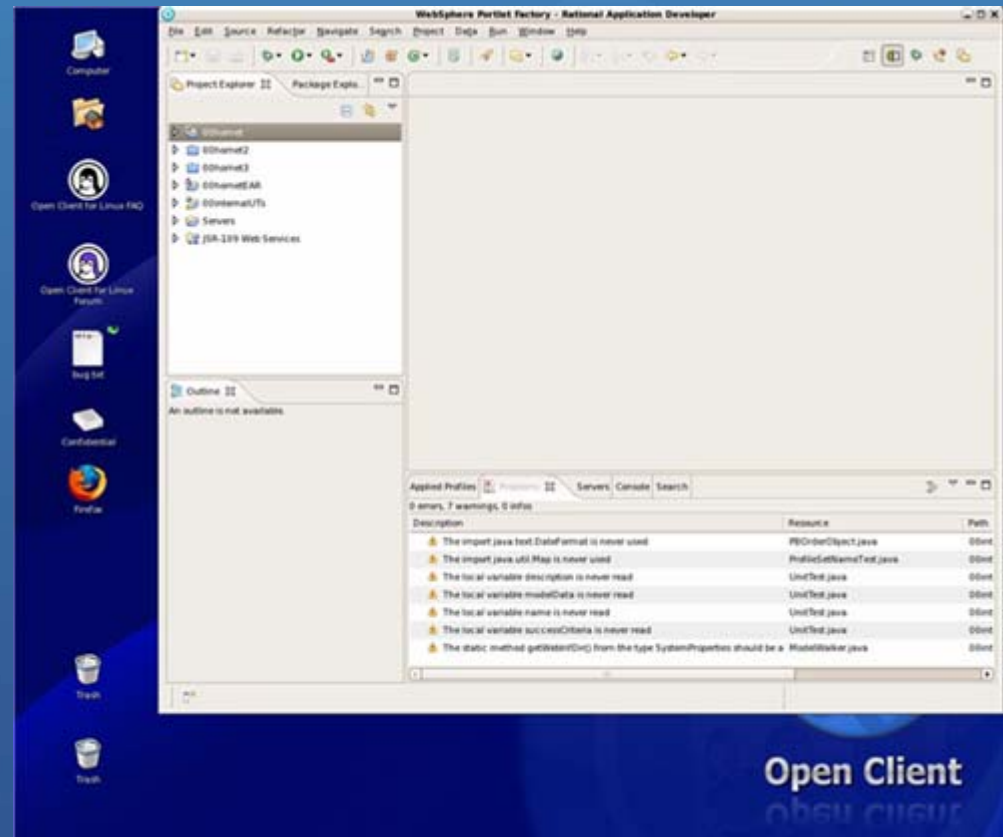


Expanded Designer Platform Support



Develop Applications on Linux Platforms

- Develop applications on SUSE and Red Hat Enterprise Linux
- Plug WPF Designer into Eclipse 3.3*



* Visual application development features not available in Eclipse 3.3



Broadened Portlet Platform Support



Deploy Portlets to More Platforms

- Deploy portlets to WAS JSR 168 container
- Deploy JSR 286 portlets to WebSphere Portal 6.1

Deploy JSR168/286
Portlets to Portal 6.1

The screenshot shows the 'New Deployment Configuration' dialog box. The 'Configuration Name' is 'portals.1'. The 'Server Type' is 'WebSphere Portal 6.1'. The 'Profile Directory' is 'C:/IBM/WebSphere/wp_profile'. Under 'Deployment Configuration Details', the 'Portlet API' is set to 'Java Portlet Standard 2.0'. The 'Auto deploy' checkbox is checked. There are fields for 'JRE Home', 'Admin User', and 'Admin Password', and a 'Test Server Connection' button.

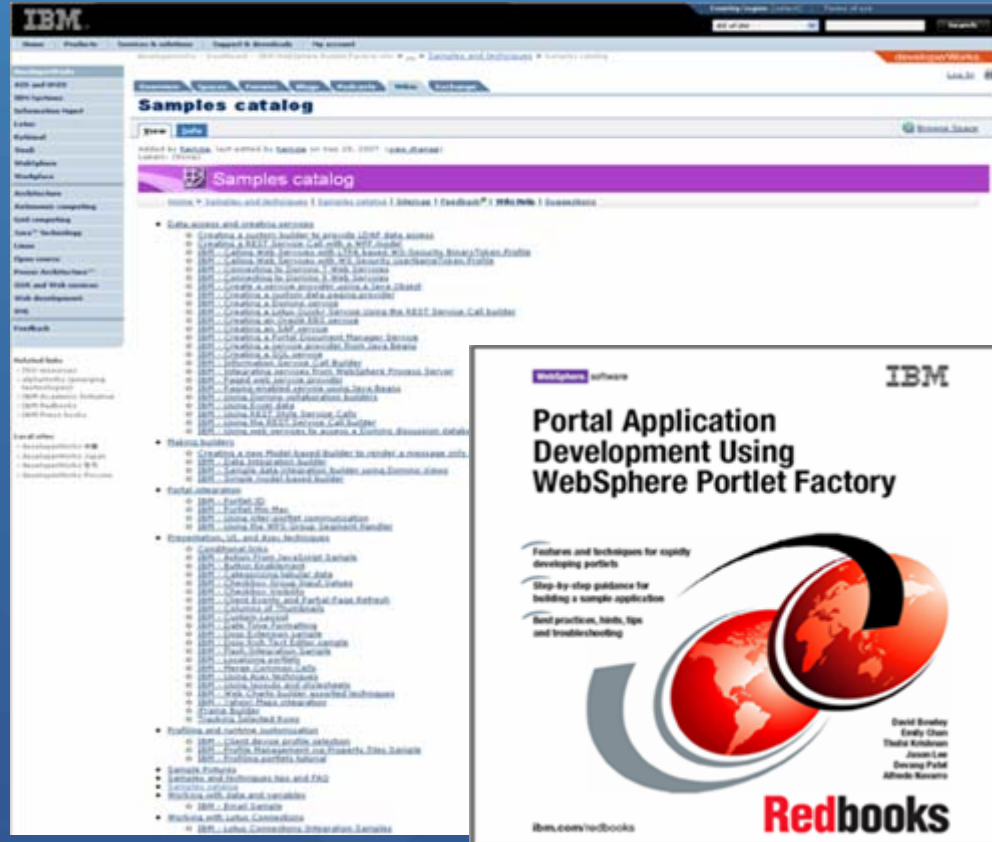
Created New Samples and Documentation



Jumpstart Application Development with Samples and Best Practices

- Published a comprehensive Portlet Factory Redbook
- Added 50+ samples to the Portlet Factory wiki to jumpstart application development

- Creating a rich internet application using Dojo
- Using AJAX techniques in applications
- Integrating with the Connections suite including Dogear, Profiles and Blogs
- Integrating with Quickr using the REST service call builder
- Using the REST call builder and REST style service calls
- Creating a Domino service
- Using Domino collaboration builders
- Localizing portlets to support multiple languages
- Creating a SQL service
- Creating a SAP service
- Using Inter-portlet Communication
- Using Excel Data
- Integrating with Flash
- Leveraging Web Charts to visualize data using graphs and charts



<http://www.redbooks.ibm.com/abstracts/sg247525.html>

<http://www-128.ibm.com/developerworks/wikis/display/PortletFactoryID/Home>

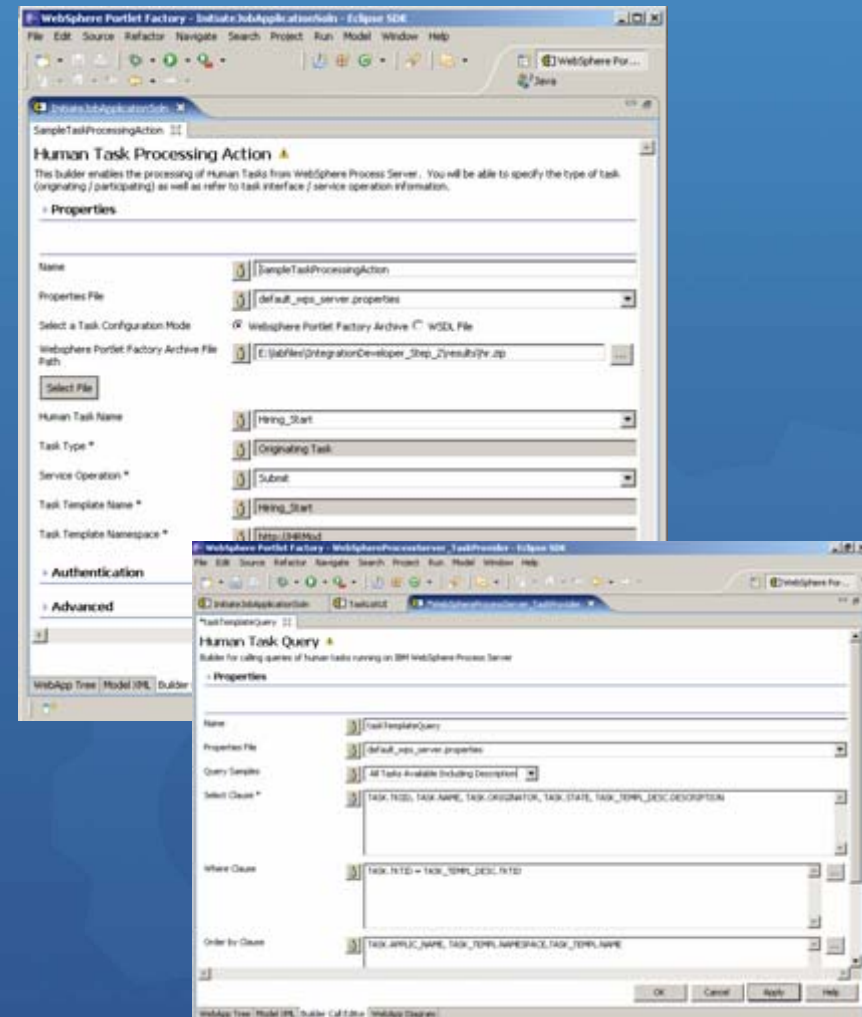


Expanded Builder Catalog Offerings



Add Workflow to Applications by Integrating with WebSphere Process Server

- Initiate or process human tasks
- Query list of human tasks



<http://catalog.lotus.com/wps/portal/soa/details?catalog.label=1W10046T>



Leverage Services Cataloged in IBM Information Server

- Connect to an Information Server instance
- Browse deployed services
- Select a service to use in WPF applications

The image shows two overlapping screenshots. The top screenshot is from the Eclipse IDE, displaying the 'WebSphere Portlet Factory - ExamplePortlet - Eclipse SDK' window. The 'Package Explorer' on the left shows a project structure with folders like 'models', 'factory', and 'portalDemo'. The main editor area shows the configuration for a 'StandardizeAddressService'. A yellow callout box with a pointer highlights the 'Details' button in the 'Load Operators' section. The bottom screenshot is from a Mozilla Firefox browser, displaying a 'Report For Parallel Job' titled 'StandardizeAddress'. The report includes the following text: 'Report generated on 2007-03-10, at 12:52:31 From project portalDemo on server PEETSCH DataStage server version 8.0'. Below the text is a data flow diagram showing 'WSD_Input_B' connected to 'Standardize_17' (a DataStage job icon) and then to 'WSD_Output_1'. A yellow callout box with a pointer highlights the 'Display Service Details' text at the bottom of the browser window.

Browse and Select Service

Display Service Details

<http://catalog.lotus.com/wps/portal/portal/details?catalog.label=1WP1001DS>





Thank You!