



IBM Software Group

Cooperative Portlets

WebSphere Portal Server v5.1

Lotus software



ON DEMAND BUSINESS™

IBM Proof of Technology

IBM Product Introduction + Exploration

© 2005 IBM Corporation

IBM Software Group | Lotus software

© 2005 IBM Corporation

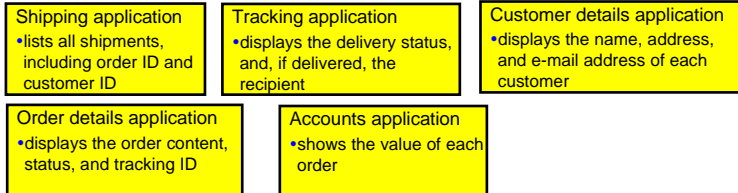


Session Objectives

- After completing this session, you should be able to:
 - ▶ Understand property, action, and wire concepts
 - ▶ Develop portlets which are capable of cooperating using Click-to-Action and Wires
 - ▶ Exploit portlet cooperation in standard portlets
 - ▶ Be able to implement advanced cooperation scenarios

Value of Cooperative Portlets - Shipping Department

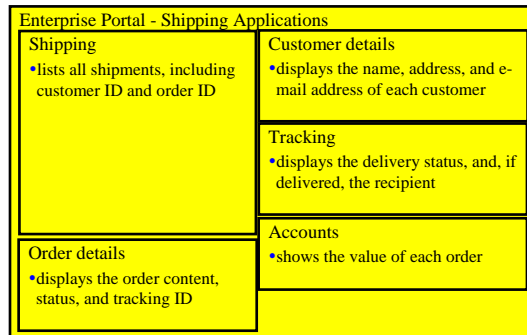
- The shipping department at a major company is responsible for researching and resolving shipping problems.
- This process relies on several applications:



- Customer problem: employee productivity
 - log into each application separately
 - manually switch between applications throughout the task
 - repeatedly re-key or cut & paste information from one window into another
 - training new employees

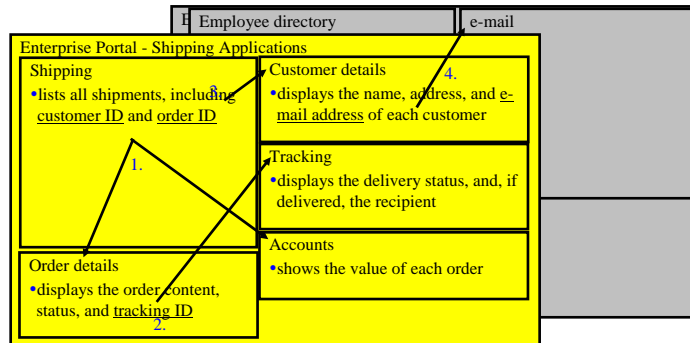
Shipping Solution with WebSphere Portal (without "Click-to-Action")

- Using WebSphere Portal, all the applications used for researching and resolving shipping problems can be grouped on a single page, for immediate access with a single logon
- The Portal Administrator can
 - Select the applications that each employee needs
 - Determine the layout of the applications on the page



Shipping Solution with WebSphere Portal and "Click-to-Action"

- Using "Click-to-Action", shipping employees can, for example,
 - click on an order ID to see the order details or the value of the order
 - click on a tracking ID to see the delivery status
 - click on a customer ID to see the e-mail address of the customer
 - click on the e-mail address to compose an e-mail to the customer
 - Broadcast information from one portlet to others ("page resync")



Features of Cooperative Portlets

- User Friendly Interface
 - "Click and Select" Alternative to a Text Entry Field
- Property Transfer Methods
 - Click-to-Action Menu
 - Wires – Wiring Tool or Control + Click-to-Action Menu [wire created]
 - Programmatic [no transfer unless wired!]
- Output Property Transfer Options
 - Single Portlet
 - Broadcast
 - Scatter
- Chained Propagation
- Dynamic Association Between Source and Target Portlets

Property Broker

- Component of WebSphere Portal introduced in WP 4.2 and enhanced in 5.0 and 5.1
- Maintains a registry in the portal database of:
 - Properties, actions and parameters of portlets
 - Wires
- Processes portlets at deployment time to register their property/action information
- Examines registered property/action/wire information at runtime to facilitate information exchange between portlets

Property Broker

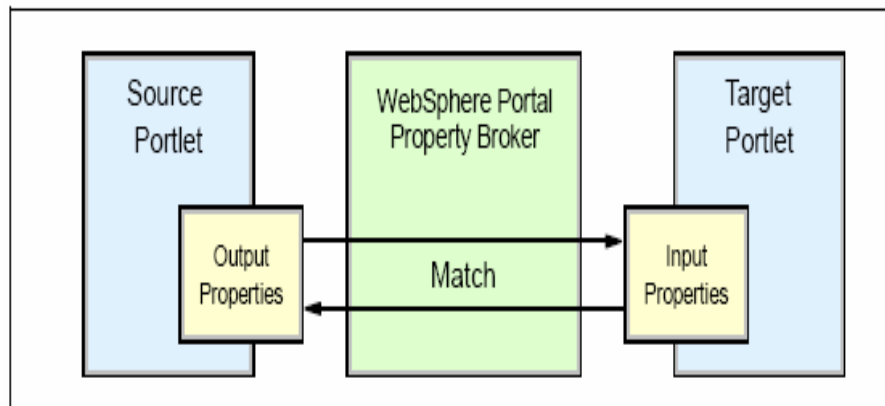


Figure 12-2 The property broker matches input and output properties

Property

- Cooperative Portlets are associated with “Properties” and “Actions”
- Property
 - ▶ an exchangeable data item, which the portlet can produce or consume
 - ▶ a part of the portlet’s data model, suitable for exposure to other portlets
 - ▶ associated with type information, which is used to determine exchangeability
 - ▶ can be tied to a visual control on the screen, or produced when an action is executed

Action

- Action
 - ▶ logic which can be executed to process a Property
 - ▶ or, logic which after execution produces Properties
 - ▶ or, both at the same time
 - ▶ associated with Parameters
- Parameter
 - ▶ Maps an Action to a Property
 - ▶ Encapsulates information about how the Action produces or consumes the Property (this is called a “binding”)
- Properties and Actions are intrinsic to a portlet definition (template), and are defined by the portlet developer. All instances of the definition share the same Action/Property set

Cooperation Paradigm: Click-to-Action

- A portlet can place a clickable control in its markup which is associated with a Property
- At runtime, the Property Broker computes Actions which can process the Property
- A pop-up menu is associated with the clickable control, listing the actions which can be triggered
- The user may choose an action from the list and trigger it
- The pop-up menu can be associated with an arbitrary clickable control

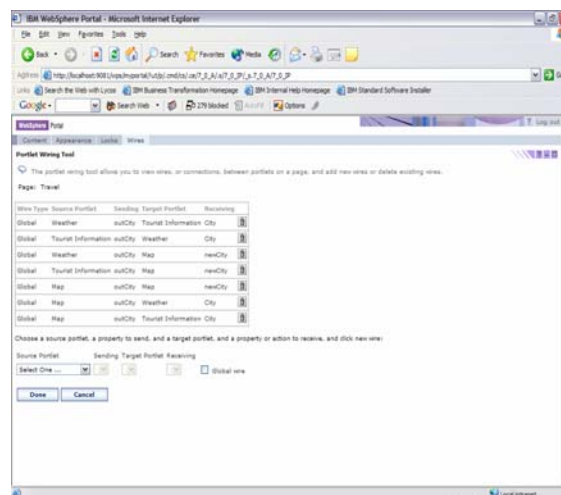
Cooperation Paradigm: Wired Portlets

- Wires are established as a post-development configuration step
- When the wired Property value is produced at runtime, the wired Action is triggered and passed the Property value
- An Administrator may create "public wires" to set up cooperative behaviors involving multiple portlets for all users
- A Privileged User may create "personal wires" whose effects apply only to the creator

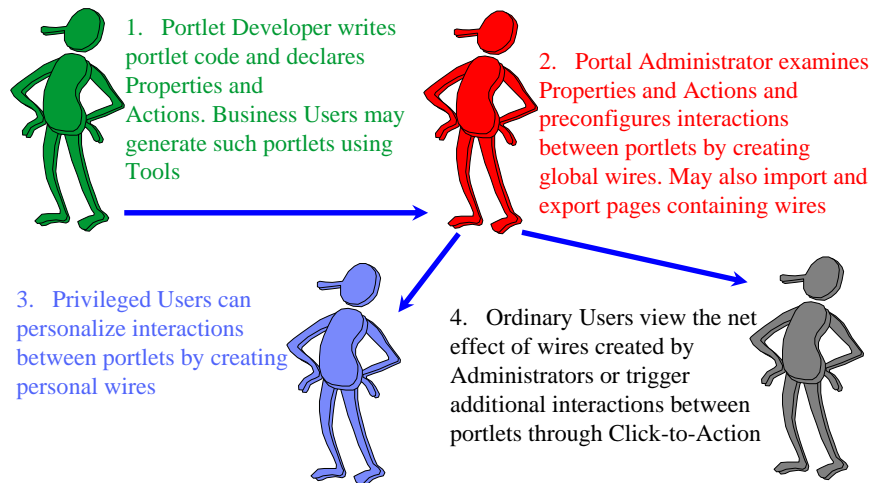
Wired vs. Click-to-Action: Related Paradigms

- Both paradigms restrict matches based on Property type information
- Both are supported by the same runtime
- Click-to-Action controls can (but need not) be wired
- Wires can (but need not) be triggered through Click-to-Action controls
- Wires are used to pre-configure integrated behaviors across a set of portlets
- Click-to-Action is used to allow portal users to trigger cross-portlet interaction from a dynamic list of choices
- Users can change from Click-to-Action to wired mode if they favor a particular mode of interaction

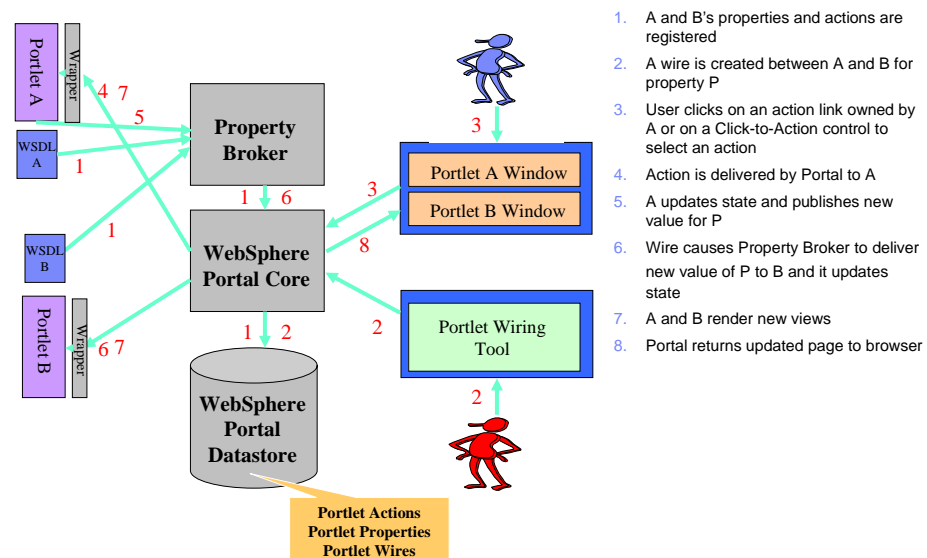
Wiring Tool



Roles in Establishing Portlet Cooperation



Cooperative Portlets Architecture



Development Steps

- Registering Properties and Actions
- Publishing Property values
- Receiving Property values
- Deployment changes

Registering Actions and Properties

- Two main options
- Declaratively, using a WSDL file
 - ▶ Usually preferred
 - ▶ Only option available for JSR-compliant portlets
- Programmatically, using Java APIs
 - ▶ Results are equivalent to using the WSDL approach
 - ▶ Can be used if providing the WSDL file is difficult, e.g. in builder generated portlets

Using WSDL for Registration (IBM and JSR)

- WSDL supports the definition of abstract operations which produce and consume typed data items
- The base language lends itself well to defining Properties and Actions
- Custom extensions are used in the Binding section which map the Properties and Actions to the portlet implementation
- WSAD 5.1.2 + Portal Toolkit 5.0.2.2 provides a wizard interface for generating the WSDL

Publishing Property Values

- Through Click-to-Action menus
 - JSP tag library
 - Java APIs
- Through portlet wires
 - Action Binding
 - Through `actionPerformed` method in `ActionListener` interface
 - Additional Java APIs
 - Through `changedProperties` method in `PropertyBrokerService` interface

Click-to-Action JSP Tag (IBM Only)

```
<td>
  <%= object.getOrderId() %>
  <C2A:encodeProperty name="orderId"
    namespace="http://www.myCo.com/myTypes" type="OrderIDType"
    value="<%= object.getOrderId() %>" />
</td>
```

- Tag attributes
 - ▶ "name" identifies property
 - ▶ "type" and "namespace" are used for matching to targets
 - ▶ "value" provides the value of the property
- `encodeProperties` tag for scattering a set of properties to all matching actions

Receiving Property Values

- The same options for receiving apply on the target side whether the property value is being transferred through a Click-to-Action event or a wire being triggered
 - ▶ Action Binding Approach
 - Through `actionPerformed` method from `ActionListener` interface
 - ▶ Additional Java APIs
 - Through `setProperties` method in `com.ibm.wps.pb.portlet.PropertyListener` interface

Deployment Considerations

- Specify a special wrapper class as the servlet class in web.xml (IBM only)
 - Provide the application class using a special parameter
 - Include a jar file containing the wrapper class in the war file
- Specify the location of the WSDL file in portlet.xml (IBM and JSR)
- Specify the name of the resource file for translated captions for Actions and Properties in portlet.xml

Example Cooperative Portlet “Non-Wired”

Choose an action – e.g., Broadcast

The new location (Denver) is broadcast to all the portlets

City Name: my city selected
 City: Denver
 State: Colorado
 Population: 2,099,194
 Alt: 1,609 meters above sea level one of the highest cities in the world. The home of the Denver Broncos.

City Name: my city selected
 City: Denver
 State: Colorado
 Population: 2,099,194
 Alt: 1,609 meters above sea level one of the highest cities in the world. The home of the Denver Broncos.

City Name: my city selected
 City: Denver
 State: Colorado
 Population: 2,099,194
 Alt: 1,609 meters above sea level one of the highest cities in the world. The home of the Denver Broncos.

City Name: my city selected
 City: Denver
 State: Colorado
 Population: 2,099,194
 Alt: 1,609 meters above sea level one of the highest cities in the world. The home of the Denver Broncos.

Example Cooperative Portlet “Wiring”

“Ctrl” +
Click

The screenshots show two portlets. The left portlet, titled 'Denver', displays city information and a map of the United States with a red dot in Colorado. The right portlet, titled 'Atlanta', displays weather information and a map of the United States with a red dot in Georgia. A red text box on the right explains that the new location (Atlanta) is transmitted to the 'Weather too' Portlet, and that in the future, the user will not get the option to choose the action (unless he breaks the wire).

The new location
(Atlanta) is
transmitted to the
“Weather too”
Portlet. In the
future, the user
will Not get the
option to choose
the action (unless
he breaks the
wire)

Session Summary

- Now that you have completed this session, you should be able to:
 - ▶ Describe the advantages of cooperative portlets over direct messaging
 - ▶ Describe property, action, and wire concepts
 - ▶ Describe how the Wired Portlets and Click-to-Action paradigms are similar and different
 - ▶ Describe the main development steps required to use cooperative portlets

References (1 of 2)

- WP 5.0 and 5.1 InfoCenter (under Developing Portlets -> Cooperative Portlets)
 - ▶ <http://publib.boulder.ibm.com/pvc/wp/500/index.html>
- WP 5.0 and 5.1 samples and Javadoc
 - ▶ Samples in installableApps subdirectory – source code included
 - ▶ Javadoc in doc/Javadoc/WPS subdirectory
- Portlet Wiring Tool (download from Portlet Catalog in 5.0, integrated install in 5.1)
 - ▶ https://www-3.ibm.com/services/cwi/portal/.cmd/ActionDispatcher/_pagr/105/_pa.105/112/.st/X/.piid/139/.ciid/170/PC_139_NAVCODE/1WP10004E#170

References (2 of 2)

- Article on Click-to-Action in WP 4 (concepts still apply in WP 5)
 - ▶ http://www7b.software.ibm.com/wsdd/library/techarticles/0212_roy/roy.html
- Article on Cooperative Portlets in WP 5
 - ▶ http://www7b.software.ibm.com/wsdd/library/techarticles/0310_roy/roy.html
- Article on Portlet Wiring
 - ▶ http://www-106.ibm.com/developerworks/websphere/library/techarticles/0404_manekar/0404_manekar.html
- Article on passing Objects between Cooperative Portlets
 - ▶ http://www-106.ibm.com/developerworks/websphere/library/techarticles/0404_ramamoorthy/0404_ramamoorthy.html
- Article on wiring JSR 168 compliant portlets
 - ▶ http://www-106.ibm.com/developerworks/websphere/library/techarticles/0412_roy/0412_roy.htm

Lab 2.3 Exercise: Cooperative Portlets lab Overview

- Part One – Installing the Click-to-Action application
- Part Two – Creating a new page and adding the portlets
- Part Three -- Investigating Click-to-Action