



# Portlet Development Strategies Birds of a Feather

**Cris J. Holdorph (holdorph@unicon.net)**

**Unicon, Inc.**

**<http://www.unicon.net/>**

## Introduction

- JSR-168 Defines the Java Portlet Standard
- Developing JSR-168 Compliant Portlets is still relatively new
  - Most portals still allow other ways to code, e.g., Channels in uPortal
- Portlet Development Frameworks are still in their infancy
- Portlets **SHOULD** be treated differently than Servlets

## Best Practices

- Action / Render phase distinction
  - Portlets should put any logic that changes the state of the portlet in the processAction method
  - Render should ONLY display the Portlet's current state (data)
  - Suggestion: renderURLs are generally used incorrectly. Play it safe and do not use them
- Portlet URLs
  - Only the portal container can create Portlet URLs

## Best Practices

- Render Parameters
  - Useful, but must be exercised with caution
  - If processAction is called render parameters will not be passed along unless processAction puts them back into the request object
- Portlet Preferences
  - Extremely lightweight persistence for Portlets
  - Should not be used as a database replacement
  - uPortal bug (UP-1040)
    - Multiple instances of the same portlet

## Best Practices

- PortletSession
  - Make all objects put in the PortletSession Serializable
  - APPLICATION\_SCOPE may be used to pass data between two portlets in the same portlet web application, however, not two portlets in different web applications
  - Tomcat 5.5.16 and above fixes session replication in a clustered environment (and tomcat 5.5.10-5.5.15 contains a memory leak with uPortal)

## Best Practices

- Make sure your portlet is standalone
  - Does not depend on the Portal services
- Try not to have jars in common/shared
  - Common/shared directories are not universally implemented across different application server vendors

# Frameworks

- Spring Portlet MVC
- Spring Web Flow
- Struts (using Apache Portals-Bridge)
- JavaServer Faces (use MyFaces)
- Tapestry
- WebWork

# Planning for the Future

- uPortal 3.0
- JSR-286 (Portlet 2.0 specification)

## uPortal 3.0

- Use the standalone projects for PersonDirectory, Groups and Permissions (GAP)

## JSR-286

- Expected to be finished May 2007
  - Some parts expected to be available for review as early as July 2006
- Features for JSR-286
  - Portlet to Portlet communication through events
  - Sharing of Portlet Session data ACROSS portlet web applications
  - Sharing of Render Parameters
  - Better support for development frameworks (especially JavaServer Faces)
  - AJAX support



## Conversation Starters

- What Development Framework (Spring Portlet MVC, Struts, JSF) are you using?
- What View Technology (JSP, XSLT, Velocity) are use using?

## Conversation Starters

- What is one problem you've had to work around in your Portal environment?

## Conversation Starters

- What are you most looking forward to from the next Portlet Standard?
- Why?
- TALK TO CHUCK

## Conversation Starters

- If you had to give a new uPortal Portlet Developer one tip, what would you say?

## Conversation Starters

- What Portlet problem do you currently have that you do not have a solution to?