SAP NetWeaver®
BI and Portals 2006
LAS VEGAS • MARCH 27 – 29
Produced and conducted by WIS, publisher of SAP Insider, with permission from SAP AG

Converting a JSR-168 Portlet to an SAP iView

Martin Snyder
Wingspan Technology

WIS
Wellesley Information Services

What We'll Cover …

- Introduction
- Hello World Example
- JSP-based Example
- Personalization Example
- Advanced Topics
- Wrap-up

What We’ll Cover …

- Introduction
- Hello World Example
- JSP-based Example
- Personalization Example
- Advanced Topics
- Wrap-up

Introduction

- JSR-168 (Java Specification Request) defines a standard that allows portlets written in Java to run in multiple portal frameworks
- SAP Enterprise Portal 6.0 does not support the JSR-168 standard, it only supports the SAP proprietary API as part of the SAP Portal Developer's Kit
- Understanding both APIs enables developers to operate in today's environment, but also keep an eye on the future

In This Session ...

- Understand the similarities and differences associated with the two portal APIs
- Examine a series of examples migrating JSR-168 portlets to SAP iViews
  - Portlet API
  - Deployment archive format
  - Deployment descriptor format
- Advanced topics discussion
- Identify additional resources
- Recap key points

What We'll Cover …

- Introduction
- Hello World Example
- JSP-based Example
- Personalization Example
- Advanced Topics
- Wrap-up

Goal

- Convert a simple JSR-168 portlet to run as an SAP iView

JSR-168 Hello World Portlet

```
package com.wingspan.example.jsr168;

import javax.portlet.*;
import java.io.*;

public class HelloWorldPortlet extends GenericPortlet
{
    public void doView(RenderRequest request, RenderResponse
   response)
        throws PortletException, IOException
   {
        response.setContentType("text/html");

        PrintWriter writer = response.getWriter();
        writer.println("<h3>Hello World Portlet Content</h3>");
        writer.flush();
  }
}
```

SAP Hello World iView

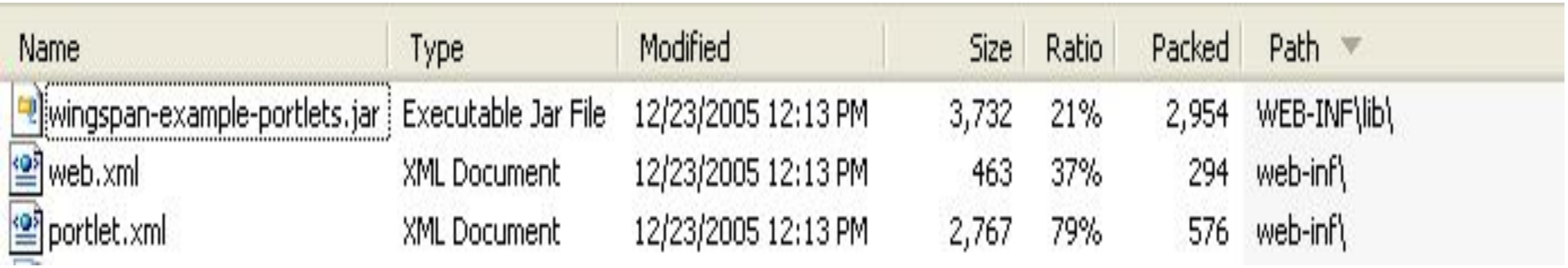
```
package com.wingspan.example.sap;

import com.sapportals.portal.prt.component.*;

public class HelloWorldIView extends AbstractPortalComponent
{
    public void doContent(IPortalComponentRequest request,
   IPortalComponentResponse response)
    {
        response.write("<h3>Hello World Portlet Content</h3>");
    }
}
```

That Was Too Easy

- One-to-one correlation of classes, methods
- Mechanics are only slightly different
- We’re not done yet …

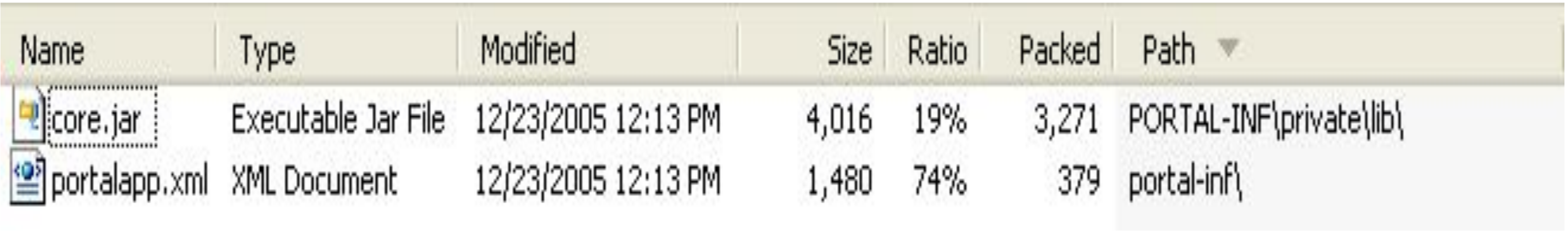
Deployment Archives

JSR-168 Portlets (WAR)



Name	Type	Modified	Size	Ratio	Packed	Path
wingspan-example-portlets.jar	Executable Jar File	12/23/2005 12:13 PM	3,732	21%	2,954	WEB-INF\lib\
web.xml	XML Document	12/23/2005 12:13 PM	463	37%	294	web-inf\
portlet.xml	XML Document	12/23/2005 12:13 PM	2,767	79%	576	web-inf\

SAP iViews (PAR)



Name	Type	Modified	Size	Ratio	Packed	Path
core.jar	Executable Jar File	12/23/2005 12:13 PM	4,016	19%	3,271	PORTAL-INF\private\lib\
portalapp.xml	XML Document	12/23/2005 12:13 PM	1,480	74%	379	portal-inf\

JSR-168 Deployment Descriptors – web.xml

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE web-app PUBLIC "-//Sun Microsystems,
  Inc.//DTD Web Application 2.3//EN"
  "http://java.sun.com/dtd/web-app_2_3.dtd">
<web-app>
   <display-name>Wingspan JSR-168 Example Portlets
  Web Application</display-name>
   <description>These portlets cannot be accessed outside
  of a JSR-168 Portlet container</description>
   <welcome-file-list>
      <welcome-file>index.html</welcome-file>
   </welcome-file-list>
</web-app>
```

JSR-168 Deployment Descriptors – portlet.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<portlet-app version="1.0"
   xmlns="http://java.sun.com/xml/ns/portlet/portlet-app_1_0.xsd"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://java.sun.com/xml/ns/portlet/portlet-app_1_0.xsd http://java.sun.com/xml/ns/portlet/portlet-app_1_0.xsd">
  <portlet>
      <description xml:lang="EN">Wingspan Hello World JSR-168 Portlet</description>
      <portlet-name>WingspanHelloWorld</portlet-name>
      <display-name xml:lang="EN">Wingspan Hello World JSR-168 Portlet</display-name>
      <portlet-class>com.wingspan.example.jsr168.HelloWorldPortlet</portlet-class>
```

JSR-168 Deployment Descriptors – portlet.xml (cont.)

```
        <expiration-cache>0</expiration-cache>
        <supports>
            <mime-type>text/html</mime-type>
            <portlet-mode>view</portlet-mode>
        </supports>
        <supported-locale>en</supported-locale>
        <portlet-info>
            <title>Wingspan Hello World JSR-168 Portlet</title>
            <short-title>Hello World Portlet</short-title>
            <keywords>Example Portlet</keywords>
        </portlet-info>
    </portlet>
</portlet-app>
```

SAP iView Deployment Descriptor – portalapp.xml

```xml
<?xml version="1.0" encoding="ISO-8859-1"?>
<application>
  <application-config>
    <property name="Vendor" value="wingspan.com" />
    <property name="SecurityArea" value="Wingspan" />
  </application-config>
  <components>
    <component name="HelloWorld">
      <component-config>
        <property name="ClassName"
     value="com.wingspan.example.sap.HelloWorldIView" />
        <property name="SafetyLevel" value="no_safety" />
      </component-config>
```

SAP iView Deployment Descriptor – portalapp.xml (cont.)

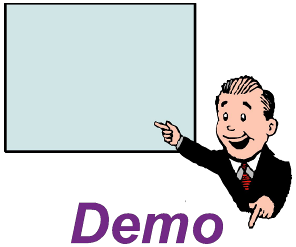
```
<component-profile/>
    </component>
  </components>
<services/>
</application>
```

Summary – Conversion Steps

- Convert GenericPortlet to AbstractPortalComponent
- Convert doView method to doContent
- Update signature on init method
- Convert portlet.xml to portalapp.xml

Deployment Demonstration

- Liferay Portal Professional 3.6.1
  - Open source portal (MIT license)
  - http://www.liferay.com
  - Console deployment
    - ► Download build.xml from www.liferay.com
    - ► Edit build.xml for your server runtime
    - ► Deploy
- SAP Enterprise Portal 6.0 SP 9
  - Web-based deployment
  - Navigate to: System Administration
    - ► Support
      - ▪ *Portal Runtime*
        - ➢ Administration Console



Demo

What We'll Cover …

- Introduction
- Hello World Example
- JSP-based Example
- Personalization Example
- Advanced Topics
- Wrap-up

Goal

- Examine a JSR-168 portlet that dispatches rendering to a JSP page
- Implement a similar dispatching mechanism in an SAP iView

JSR-168 JSP Portlet

```
package com.wingspan.example.jsr168;

import java.io.*;
import javax.portlet.*;

public class JSPPortlet extends GenericPortlet
{
   private String _portletPath = null;

   public void init(PortletConfig config) throws PortletException
   {
      super.init(config);

       _portletPath = config.getInitParameter("portlet.jsp.path");
       if(null == _portletPath)
         throw new UnavailableException("portlet.jsp.path init-param not
    set in portlet.xml for " + config.getPortletName());
   }
```

JSR-168 JSP Portlet (cont.)

```
  public void doView(RenderRequest request, RenderResponse
   response) throws PortletException, IOException
  {
     include( request, response, "view.jsp" );
  }

  protected void include(RenderRequest request, RenderResponse
   response, String page) throws PortletException, IOException
  {
     response.setContentType( "text/html" );

     PortletRequestDispatcher rd =
   getPortletContext().getRequestDispatcher( _portletPath + page );
     rd.include( request, response );
  }
}
```

SAP JSP iView

```
package com.wingspan.example.sap;

import com.sapportals.portal.prt.component.*;
import com.sapportals.portal.prt.resource.*;

public class JSPIView extends AbstractPortalComponent
{
    private String _portletPath = null;

    public void init( IPortalComponentConfig config ) throws
     PortalComponentException
    {
        super.init(config);
        _portletPath = config.getProperty( "portlet.jsp.path" );
        if(null == _portletPath)
            throw new PortalComponentException( "portlet.jsp.path property
     not set in portalapp.xml" );
    }
```

SAP JSP iView (cont.)

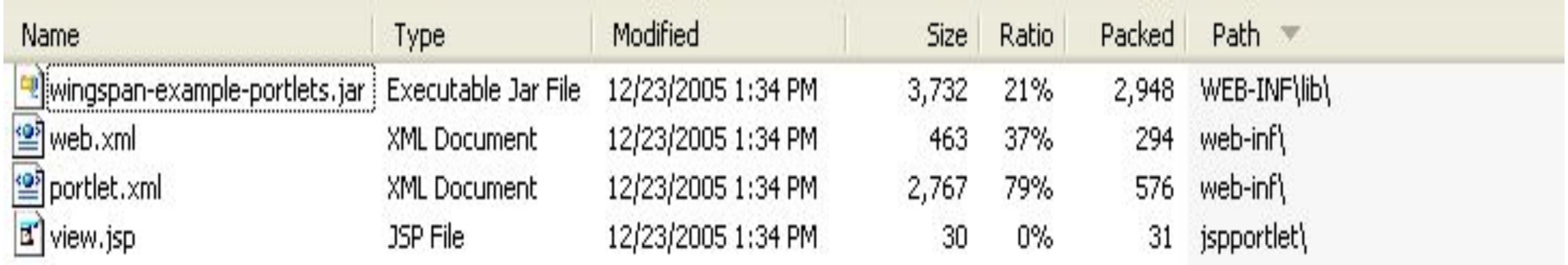
```
    public void doContent(IPortalComponentRequest request,
     IPortalComponentResponse response)
    {
        include(request, response, "view.jsp");
    }

    protected void include(IPortalComponentRequest request,
     IPortalComponentResponse response, String page)
    {
        request.getServletRequest().setAttribute("sap.request", request);
        request.getServletRequest().setAttribute("sap.response",
     response);

        IResource res = request.getResource(IResource.JSP, "/jsp" +
     _portletPath + page);
        if (res.isAvailable())
            response.include(request, res);
    }
}
```

view.jsp

```
<h3>JSP Portlet Content</h3>
```

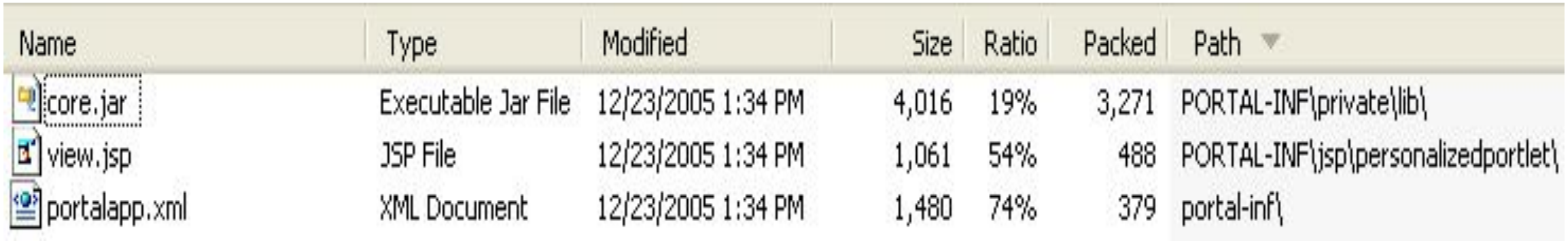
Deployment Archives

JSR-168 Portlets (WAR)



Name	Type	Modified	Size	Ratio	Packed	Path
wingspan-example-portlets.jar	Executable Jar File	12/23/2005 1:34 PM	3,732	21%	2,948	WEB-INF\lib\
web.xml	XML Document	12/23/2005 1:34 PM	463	37%	294	web-inf\
portlet.xml	XML Document	12/23/2005 1:34 PM	2,767	79%	576	web-inf\
view.jsp	JSP File	12/23/2005 1:34 PM	30	0%	31	jspportlet\

SAP iViews (PAR)



Name	Type	Modified	Size	Ratio	Packed	Path
core.jar	Executable Jar File	12/23/2005 1:34 PM	4,016	19%	3,271	PORTAL-INF\private\lib\
view.jsp	JSP File	12/23/2005 1:34 PM	1,061	54%	488	PORTAL-INF\jsp\personalizedportlet\
portalapp.xml	XML Document	12/23/2005 1:34 PM	1,480	74%	379	portal-inf\

JSR-168 Deployment Descriptor – portlet.xml

```
<portlet>
    <description xml:lang="EN">Wingspan JSP JSR-168 Portlet</description>
    <portlet-name>WingspanJSP</portlet-name>
    <display-name xml:lang="EN">Wingspan JSP JSR-168 Portlet</display-name>
    <portlet-class>com.wingspan.example.jsr168.JSPPortlet</portlet-class>
    <init-param>
        <name>portlet.jsp.path</name>
        <value>/jspportlet/</value>
    </init-param>
    <expiration-cache>0</expiration-cache>
    <supports>
        <mime-type>text/html</mime-type>
        <portlet-mode>view</portlet-mode>
    </supports>
```

JSR-168 Deployment Descriptor – portlet.xml (cont.)

```
    <supported-locale>en</supported-locale>
    <portlet-info>
        <title>Wingspan JSP JSR-168 Portlet</title>
        <short-title>JSP Portlet</short-title>
        <keywords>Example Portlet</keywords>
    </portlet-info>
</portlet>
```

SAP iView Deployment Descriptor – portalapp.xml

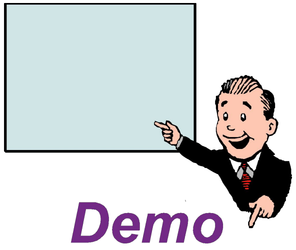
```
<component name="JSP">
  <component-config>
    <property name="ClassName" value="com.wingspan.example.sap.JSPIView" />
    <property name="SafetyLevel" value="no_safety" />
    <property name="portlet.jsp.path" value="/jspportlet/" />
  </component-config>
  <component-profile/>
 </component>
```

Summary – Conversion Steps

- Store JSP filename in portlet descriptor
- Read JSP filename in portlet init
- Dispatch rendering to JSP file in doView/DoContent
- Store API objects as attributes on the request object

Portlet Demonstration

- Liferay Portal Professional 3.6.1
  - Create new page
  - Add portlets to page
- SAP Enterprise Portal 6.0 SP 9
  - Create iView instances
  - Create new page
  - Link iViews to page



What We'll Cover …

- Introduction
- Hello World Example
- JSP-based Example
- Personalization Example
- Advanced Topics
- Wrap-up

Goal

- Convert per-user settings storage in a JSR-168 portlet to an SAP iView

JSR-168 view.jsp

```
<%@ page import=" javax.portlet.*" %>
<%
    RenderRequest portletRenderRequest =
   (RenderRequest)request.getAttribute("javax.portlet.request");
    RenderResponse portletRenderResponse =
   (RenderResponse)request.getAttribute("javax.portlet.response");
    PortletConfig portletConfig =
   (PortletConfig)request.getAttribute("javax.portlet.config");

    String thingToRemember =
   portletRenderRequest.getPreferences().getValue("forgetmenot", null);
    if (null == thingToRemember || "".equals(thingToRemember))
        thingToRemember = "... to remember something.";
%>
<h3>Don't forget: <%= thingToRemember %></h3>
```

SAP view.jsp

```
<%@ page import=" com.sapportals.portal.prt.component.*" %>
<%
    IPortalComponentRequest sapRequest =
  (IPortalComponentRequest)request.getAttribute("sap.request");
    IPortalComponentResponse sapResponse =
  (IPortalComponentResponse)request.getAttribute("sap.response");
    IPortalComponentProfile profile =
  sapRequest.getComponentContext().getProfile();

    String thingToRemember = (String)profile.getProperty("forgetmenot");
    if (null == thingToRemember || "".equals(thingToRemember))
        thingToRemember = "... to remember something.";
%>
<h3>Don't forget: <%= thingToRemember %></h3>
```

JSR-168 Personalized Portlet

```
package com.wingspan.example.jsr168;

import java.io.*;
import javax.portlet.*;

public class PersonalizedPortlet extends JSPPortlet
{
    public void doEdit(RenderRequest request, RenderResponse
     response) throws PortletException, IOException
    {
      include( request, response, "edit.jsp" );
    }
```

JSR-168 Personalized Portlet (cont.)

```
    public void processAction( ActionRequest request, ActionResponse
     response ) throws PortletException, IOException
    {
      String thingToRemember = request.getParameter("forgetmenot");
      if (null != thingToRemember)
      {
            PortletPreferences prefs = request.getPreferences();
            prefs.setValue("forgetmenot", thingToRemember);
            prefs.store();

            response.setWindowState(WindowState.NORMAL);
            response.setPortletMode(PortletMode.VIEW);
      }
    }
}
```

SAP Personalized iView

```
package com.wingspan.example.sap;

import com.sapportals.portal.prt.component.*;

public class PersonalizedIView extends JSPIView
{
    public void doEdit(IPortalComponentRequest request,
   IPortalComponentResponse response)
    {
        IPortalComponentProfile profile =
   request.getComponentContext().getProfile();
```

SAP Personalized iView (cont.)

```
          String postedValue = request.getParameter("forgetmenot");
          if (null != postedValue)
          {
               profile.setProperty("forgetmenot", postedValue);
               profile.store();

               response.write("<script> \n");
               response.write("\twindow.opener.document.location.reload();
    \n");
               response.write("\twindow.close();");
               response.write("</script> \n");
          }
          else
          {
               // Generate edit.jsp output
          }
     }
}
```

JSR-168 edit.jsp

```
<%@ page import=" javax.portlet.*" %>
<%
  RenderRequest portletRenderRequest = (RenderRequest)request.getAttribute("javax.portlet.request");
  RenderResponse portletRenderResponse = (RenderResponse)request.getAttribute("javax.portlet.response");
  PortletConfig portletConfig = (PortletConfig)request.getAttribute("javax.portlet.config");

  String thingToRemember = portletRenderRequest.getPreferences().getValue("forgetmenot", "");
%>
```

JSR-168 edit.jsp (cont.)

```
<form method="POST" action="<%=
   portletRenderResponse.createActionURL() %>" onsubmit="return
   validate()">
    <table style="width: 100%">
        <tr>
            <td style="width: 1%; white-space: nowrap">Remember
   this:</td>
            <td style="width: 99%">
                <input type="text" name="forgetmenot" id="forgetmenot"
   style="width: 100%"
                    value="<%= thingToRemember %>" />
            </td>
        </tr>
    </table>
    <input type="submit" value="Remember!" />
</form>
```

JSR-168 edit.jsp (cont.)

```
<script>
    function validate()
    {
        var oInput = document.getElementById('forgetmenot');
        if (null != oInput && oInput.value.length > 0)
            return true;

        alert('You cannot remember the empty string.  What\'s the point?');
        return false;
    }
</script>
```

SAP edit.jsp

- Our include technique does not work properly on Enterprise Portal 6.0
  - Users see a blank page instead of our generated HTML
- Output is rendered directly by doEdit method

```
String thingToRemember = (String)profile.getProperty("forgetmenot");

response.write("<form method=\"POST\" action=\"\" onsubmit=\"return validate()\">
    \n");
response.write("\t<table style=\"width: 100%\"> \n");
response.write("\t\t<tr> \n");
response.write("\t\t\t<td style=\"width: 1%; white-space: nowrap\">Remember
    this:</td> \n");
response.write("\t\t\t<td style=\"width: 99%\"><input type=\"text\"
    name=\"forgetmenot\" id=\"forgetmenot\" style=\"width: 100%\" value=\"" +
    thingToRemember + "\" /></td> \n");
```

SAP edit.jsp (cont.)

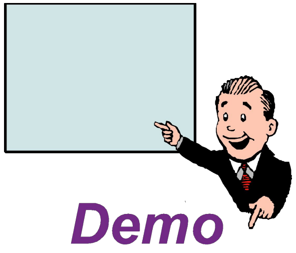
```
response.write("\t\t</tr> \n");
response.write("\t</table> \n");
response.write("\t<input type=\"submit\" value=\"Remember!\" /> \n");
response.write("</form> \n");
response.write("<script> \n");
response.write("\tfunction validate() \n");
response.write("\t{ \n");
response.write("\t\tvar oInput = document.getElementById('forgetmenot'); \n");
response.write("\t\tif (null != oInput && oInput.value.length > 0) \n");
response.write("\t\t\treturn true; \n");
response.write(" \n");
response.write("\t\talert('You cannot remember the empty string.  What\\\'s the point?'); \n");
response.write("\t\treturn false; \n");
response.write("\t} \n");
response.write("</script> \n");
```

Summary – Conversion Steps

- Move POST handling logic from processAction to doEdit
- Do not use JSP files for SAP iView personalization
- Convert PortletPreferences usage to IPortalComponentProfile

Portlet Demonstration

- Liferay Portal Professional 3.6.1
  - Add portlet to page
  - Apply user customization
- SAP Enterprise Portal 6.0 SP 9
  - Create iView instance
  - Link iView to page
  - Apply user customization



What We'll Cover …

- Introduction
- Hello World Example
- JSP-based Example
- Personalization Example
- Advanced Topics
- Wrap-up

Popup Windows

- Separate issues make implementing popup windows a challenge in both environments
  - Many JSR-168 implementations do not properly manage session information between portlets and popups
  - SAP Enterprise Portal requires that popup requests be routed through specialized iViews

Java Servlets

- Servlets are supported in both environments, however
  - They are registered differently
    - ► JSR-168: Registered in web.xml
    - ► SAP iView: Registered in portalapp.xml
  - They are invoked differently
    - ► JSR-168: Accessed via normal web.xml URL
    - ► SAP iView: Accessed via SAP EP-specific URL

Heads Up!

Abstract API

- If you plan on migrating from one API to the other, consider developing an API abstraction
- Start with the following objects
  - Request
  - Response
  - Preferences/Profile
- Instantiate classes implementing your API in classes like JSPPortlet and JSPIView
- Set these objects as attributes on the Java Servlet HTTPServletRequest
- Write JSP files to your abstract APIs

What We'll Cover …

- Introduction
- Hello World Example
- JSP-based Example
- Personalization Example
- Advanced Topics
- Wrap-up

Summary – Conversion Steps

- Convert GenericPortlet to AbstractPortalComponent
- Convert doView method to doContent
- Update signature on init method
- Convert portlet.xml to portalapp.xml
- Store JSP filename in portlet descriptor
- Read JSP filename in portlet init
- Dispatch rendering to JSP file in doView/DoContent
- Store API objects as attributes on the request object

Summary – Conversion Steps (cont.)

- Move POST handling logic from processAction to doEdit
- Do not use JSP files for SAP iView personalization
- Convert PortletPreferences usage to IPortalComponentProfile

Resources

- JSR 168 Specification
  - http://www.jcp.org/en/jsr/detail?id=168
- SAP Portal Development Kit
  - https://www.sdn.sap.com/irj/sdn/developerareas/ep (Downloads section on right)
    - ► Requires login credentials to the SAP Service Marketplace
- Dr. Dobbs Journal
  - "Developing JSR-168 Portlets," Ted O'Connor and Martin Snyder, October 2005 issue (#377)
- Downloadable Examples
  - http://www.wingspan.com/downloads/SAPConference2006.zip

7 Key Points to Take Home

- The JSR-168 specification allows portlets to run within multiple frameworks
- JSR-168 is not supported by SAP Enterprise Portal 6.0
- Migrating between the two is not difficult with the proper guidance
- Implementing (and migrating) popups and servlets is harder than our examples, but it is achievable

7 Key Points to Take Home (cont.)

- Be aware of differences in deployment descriptors and procedures
- Consider building an abstract API layer so JSP pages will work in both (or either) environment
- Download this presentation and example project

Your Turn!

How to contact me:
Martin Snyder
msnyder@wingspan.com