

SAP NetWeaver®

BI and

LAS VEGAS • MARCH 27 – 29

Portals 2006

Produced and conducted by WIS, publisher of SAP Insider, with permission from SAP AG

AJAX: What It Is and What It Can Do For Your iViews

Martin Snyder
Wingspan Technology

What We'll Cover ...

- **Introduction**
- **AJAX (Asynchronous JavaScript and XML)**
- **Example iView**
- **Debugging Techniques**
- **Advanced Topics**
- **Wrap-up**

What We'll Cover ...

- **Introduction**
- **AJAX (Asynchronous JavaScript and XML)**
- **Example iView**
- **Debugging Techniques**
- **Advanced Topics**
- **Wrap-up**

Introduction

- **Portal frameworks are primarily designed to deliver multiple simultaneous static views of data from disparate sources**
- **It is often desirable to update an iView's content without refreshing the full portal page**
- **AJAX (Asynchronous JavaScript and XML) can be used to deliver a robust desktop application experience to users of Web applications**

In This Session ...

- **We will ...**
 - **Define AJAX**
 - **Discuss alternate mechanisms that produce similar results**
 - **Examine a working example (available for download)**
 - **Examine debugging challenges and solutions**
 - **Discuss advanced topics**
 - **Identify additional resources**
 - **Recap key points**

What We'll Cover ...

- Introduction
- **AJAX (Asynchronous JavaScript and XML)**
- Example iView
- Debugging Techniques
- Advanced Topics
- Wrap-up

AJAX Defined

- **“AJAX” is a term apparently coined in February 2005 by Jesse James Garrett of Adaptive Path as “... shorthand for (A)synchronous (Ja)vaScript and (X)ML ...”**
 - **A different word to describe Microsoft remote scripting technology released with IE5**
 - **Combination of existing technologies and techniques**
 - ▶ **JavaScript**
 - ▶ **XML**
 - ▶ **Document Object Model (DOM)**
 - ▶ **XMLHttpRequest**
 - **Variety of implementation options**
 - ▶ **Multiple libraries and toolkits**
 - ▶ **No meaningful standards**

Where is AJAX Used?

- **Well-known examples**
 - Google applications (Gmail, Google Maps, etc.)
 - Microsoft Outlook Web Access
- **Broad browser support**
 - Internet Explorer
 - Mozilla Firefox
 - Netscape 7
 - Safari

How is AJAX Implemented?

- **The technique**

- Open a server connection using the XMLHttpRequest object
- Transmit some request data
- Wait for server response
- Change the page contents by manipulating the DOM

- **When to use AJAX**

- Replace existing Java Applets, ActiveX Controls, or Flash
- Multi-step submissions
- Client access to large datasets

Alternative Models

- **XMLHttpRequest is not the only transmission vehicle**
- **Dynamic creation of HTML tags can achieve similar behavior**
 - `<script src="server.js"></script>`
 - `<iframe src="server.jsp"><iframe>`
- **Mechanics are significantly different**
- **Browser interactions are significantly different**

What We'll Cover ...

- Introduction
- AJAX (Asynchronous JavaScript and XML)
- Example iView
- Debugging Techniques
- Advanced Topics
- Wrap-up

Example Scenario

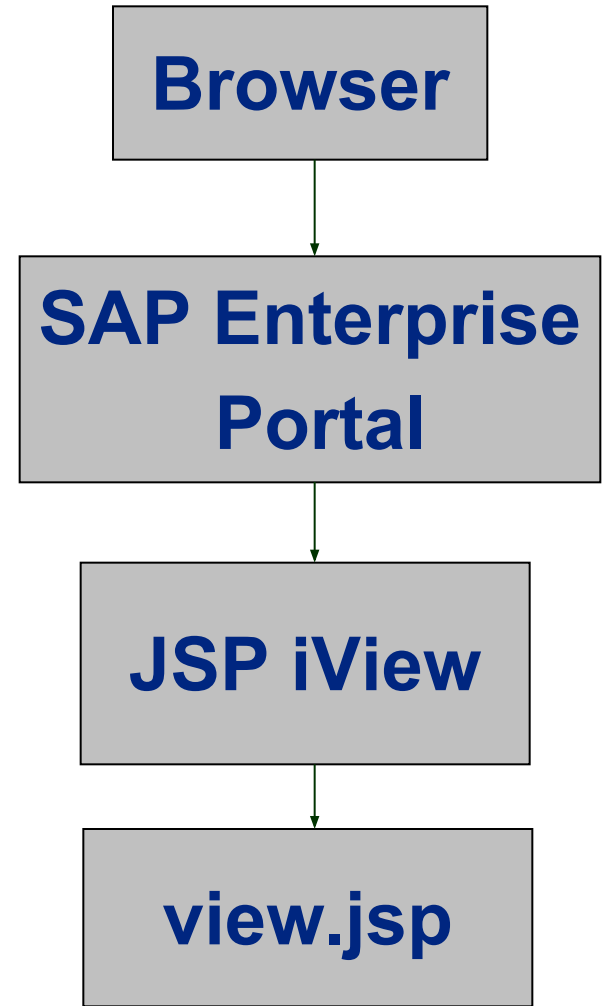
- **Task List iView**
 - Add/remove tasks
 - Mark tasks as complete
 - Separate lists for each user
- **Shortcuts**
 - In-memory persistence only
 - Display order may change on page refresh

Components

- **Task List Data**
 - Task.java – Java Bean for a single task
 - TaskList.java – a list of tasks for a single user
- **iView Components**
 - JSPIView.java – dispatches iView requests to JSP pages
 - view.jsp – iView display
- **AJAX Components**
 - tasklist.js – AJAX routines
 - TasklistServlet.java – process AJAX requests
- **Other Components**
 - Utility.java – HTML escape routine
 - Debug.js – JavaScript utility routines

Initial Page Request

- **Browser initiates a request for a portal page**
- **Portal framework invokes JSP iView**
- **JSP iView includes view.jsp**



Task.java (1 of 2)

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

import java.io.Serializable;

public class Task implements Serializable, Comparable
{
    private static final long serialVersionUID = 1L;

    private int _id;
    private String _name;
    private boolean _completed;

    Task();
    Task(int id, String name, boolean completed);
```

Task.java (2 of 2)

```
public int getId();
public String getName();
public boolean getCompleted();
public void setName(String name);
public void setCompleted(boolean completed);
public int compareTo(Object obj)
{
    Task otherTask = (Task)obj;
    if (!_completed && otherTask._completed)
        return -1;
    else if ((_completed && !otherTask._completed))
        return 1;
    else
    {
        return _name.compareTo(otherTask._name);
    }
}
```


Tasklist.java (1 of 2)

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

```
import java.io.Serializable;
```

```
import java.util.*;
```

```
public class Tasklist implements Serializable
```

```
{
```

```
    private List _tasks;
```

```
    private int _nextTaskId;
```

```
    private Tasklist() {
```

```
        _tasks = new LinkedList();
```

```
        _nextTaskId = 1;
```

```
    }
```

Tasklist.java (2 of 2)

```
public void sort();
public Iterator iterator();
public boolean isEmpty();
public int getNextTaskId();
public Task addTask(String name);
public Task getTask(int id);
public Task removeTask(int id);
private static Map s_allLists = new HashMap();
public static synchronized Tasklist getTasklistForUser(Object
userID)
{
    Tasklist tasklist = (Tasklist)s_allLists.get(userID);
    if (null == tasklist) {
        tasklist = new Tasklist();
        s_allLists.put(userID, tasklist);
    }
    return tasklist;
}
}
```

JSPIView.java (1 of 2)

```
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" );
    }
}
```

JSPView.java (2 of 2)

```
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 (1 of 5)

```
<%@ page import=" com.sapportals.portal.prt.component.*,  
    com.wingspan.example.sap.*" %>  
<%  
    IPortalComponentRequest sapRequest =  
    (IPortalComponentRequest)request.getAttribute("sap.request");  
    IPortalComponentResponse sapResponse =  
    (IPortalComponentResponse)request.getAttribute("sap.response");  
  
    String userID = request.getUserPrincipal().getName();  
    Tasklist tasklist = Tasklist.getTasklistForUser(userID);  
  
    String tasklistStyle = "";  
    String noListStyle = "";  
    if (tasklist.isEmpty())  
        tasklistStyle = "style=\"display: none\"";  
    else  
        noListStyle = "style=\"display: none\"";  
%>
```

view.jsp (2 of 5)

```
<span id="hastasks" <%= tasklistStyle %>>
  <h3>Tasklist for <%= userID %></h3>
  <input type="button" name="add" value="Add Task"
onclick="addTask()" />
  <table id="tasklist">
    <thead>
      <tr>
        <th style="width: 1%">Completed</th>
        <th style="width: 98%">Task</th>
        <th style="width: 1%">Delete</th>
      </tr>
    </thead>
    <tbody>
```

view.jsp (3 of 5)

```
<%
    tasklist.sort();
    Iterator it = tasklist.iterator();
    while (it.hasNext())
    {
        Task task = (Task)it.next();

        String checkboxState = "";
        if (task.getCompleted())
            checkboxState = "checked=\"checked\"";
    }
%>
```

view.jsp (4 of 5)

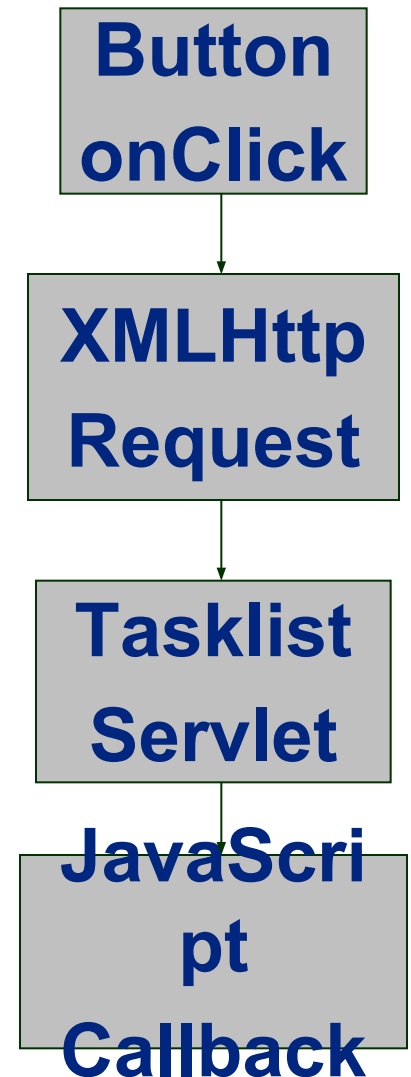
```
<tr id="row-<%= task.getId() %>">
  <td style="text-align: center">
    <input type="checkbox" value="true"
      onclick="AJAXCompleteTask(<%= task.getId()
%>, <%= !task.getCompleted() %>)" <%= checkboxState %> />
    </td>
    <td><%= Utility.escapeHTML(task.getName()) %></td>
    <td style="text-align: center"><input type="button"
value="X" onclick="AJAXDeleteTask(<%= task.getId() %>)"
/></td>
  </tr>
<%
  }
%>
</tbody>
</span>
```


view.jsp (5 of 5)

```
<span id="notasks" <%= noListStyle %>>
  <h3>No tasks for <%= userID %></h3>
  <input type="button" name="add" value="Add Task"
    onclick="addTask()" />
</span>
<script src="<%=
  sapRequest.getWebResourcePath("WingspanAJAXPar/js/debug.
  js") %>"></script>
<script src="<%=
  sapRequest.getWebResourcePath("WingspanAJAXPar/js/tasklist
  .js") %>"></script>
<script>
  initNextTaskId(<%= tasklist.getNextTaskId() %>);
  initAJAXURL('<%= TasklistServlet.getServletURL(sapRequest)
  %>');
</script>
```

Adding a New Task

- **User clicks “Add Task” button**
 - **onClick handler creates XMLHttpRequest object**
 - **XMLHttpRequest sends command to TaskListServlet**
 - **XML result is retrieved and parsed**
 - **Callback manipulates iView HTML to add the new entry**



Partial tasklist.js (1 of 2)

```
function AJAXGetRequestObject()
{
    var oRequest = null;
    if (window.XMLHttpRequest)
        oRequest = new XMLHttpRequest();
    else if (window.ActiveXObject)
        oRequest = new ActiveXObject("Microsoft.XMLHTTP");
    else
        throw new Error('Could not create XMLHttpRequest object');
    return oRequest;
}
```

Partial tasklist.js (2 of 2)

```
function AJAXExecuteCommand(sQueryString)
{
    var oRequest = AJAXGetRequestObject();

    oRequest.open("GET", gsAJAXURL + '?' + sQueryString, true);
    oRequest.onreadystatechange = function() {
AJAXCallback(oRequest); };
    oRequest.send(null);
}
```

```
function AJAXNewTask(sName)
{
    AJAXExecuteCommand('cmd=addtask&taskname=' + sName);
}
```

TasklistServlet.java (1 of 7)

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

import java.io.IOException;
import javax.servlet.*;
import javax.servlet.http.*;
import com.sapportals.portal.prt.component.*;

public class TasklistServlet extends HttpServlet
{
    public void service(HttpServletRequest request,
        HttpServletResponse response) throws IOException,
        ServletException
    {
        StringBuffer out = new StringBuffer();
    }
}
```

TasklistServlet.java (2 of 7)

```
try {  
    String userID = request.getUserPrincipal().getName();  
    if (null == userID) throw new Exception("Cannot determine  
calling user");  
    Tasklist tasklist = Tasklist.getTasklistForUser(userID);  
    if (null == tasklist) throw new Exception("Cannot load task  
list for user: " + userID);  
    String cmd = request.getParameter("cmd");  
    if (null == cmd) throw new Exception("No command  
specified");  
    ... (Process the command)
```

TasklistServlet.java (3 of 7)

```
} catch (Exception e) {  
    out.append("<?xml version=\"1.0\" ?>\n");  
    out.append("<error>");  
    out.append(Utility.escapeHTML(e.getMessage()));  
    out.append("</error>");  
}
```

... (Send the XML formatted response)

```
}  
}
```

TasklistServlet.java (4 of 7)

```
public class TasklistServlet extends HttpServlet
{
    public void service(HttpServletRequest request,
        HttpServletResponse response) throws IOException,
        ServletException
    {
        ...
        Task cmdTask = null;
        if ("addtask".equalsIgnoreCase(cmd)) {
            cmdTask =
tasklist.addTask(getRequiredParameter(request, "taskname"));
        } else if ("remtask".equalsIgnoreCase(cmd)) {
            cmdTask =
tasklist.removeTask(Integer.parseInt(getRequiredParameter(request,
"taskid")));
        } else if ("completetask".equalsIgnoreCase(cmd)) {
            cmdTask =
tasklist.getTask(Integer.parseInt(getRequiredParameter(request,
"taskid")));
        }
    }
}
```


TasklistServlet.java (5 of 7)

```
cmdTask.setCompleted(Boolean.valueOf(getRequiredParameter(request, "completed")).booleanValue());
    } else
        throw new Exception("Unrecognized command: " + cmd);

if (null == cmdTask)
    throw new Exception("Specified task was not found");
```

TasklistServlet.java (6 of 7)

```
        out.append("<?xml version=\"1.0\" ?>\n");
        out.append("<success>\n");
        out.append("\t<task>\n");
        out.append("\t\t<id>" + cmdTask.getId() + "</id>\n");
        out.append("\t\t<name>" +
Utility.escapeHTML(cmdTask.getName()) + "</name>\n");
        out.append("\t\t<completed>" + cmdTask.getCompleted() +
"</completed>\n");
        out.append("\t</task>\n");
        out.append("\t<command>");
        out.append(cmd);
        out.append("</command>\n");
        out.append("</success>\n");
        ...
    }
}
```

TasklistServlet (7 of 7)

```
public class TasklistServlet extends HttpServlet
{
    public void service(HttpServletRequest request,
        HttpServletResponse response) throws IOException,
        ServletException
    {
        ...
        try
        {
            response.setStatus (200);
            response.setContentType ("text/xml");
            response.setHeader("Cache-Control", "no-cache");
            response.setContentLength (out.length());
            response.getOutputStream ().print(out.toString());
        }
        catch (Exception e)
        {
            throw new ServletException(e);
        }
    }
}
```

Partial tasklist.js (1 of 7)

```
function AJAXCallback(oRequest)
{
    if (oRequest.readyState == 4)
    {
        if (oRequest.status == 200)
        {
            var aErrors = oRequest.responseXML.getElementsByTagName('error');
            if (null != aErrors && aErrors.length > 0)
            {
                var sErrorMsg = aErrors[0].childNodes[0].nodeValue;
                alert(sErrorMsg);
                return;
            }
        }
    }
}
```

Partial tasklist.js (2 of 7)

```
var oTask = null;
var aTasks = oRequest.responseXML.getElementsByTagName('task');
    if (null != aTasks && aTasks.length > 0)
    {
        var nID =
aTasks[0].getElementsByTagName('id')[0].childNodes[0].nodeValue;
        var sName =
aTasks[0].getElementsByTagName('name')[0].childNodes[0].nodeValue;
        var bCompleted = ('true' ==
aTasks[0].getElementsByTagName('completed')[0].childNodes[0].nodeValue);

        oTask = new Object();
        oTask.nID = nID;
        oTask.sName = sName;
        oTask.bCompleted = bCompleted;
    }
```

Partial tasklist.js (3 of 7)

```
var sCmd = null;
    var aCommands =
oRequest.responseXML.getElementsByTagName('command');
    if (null != aCommands && aCommands.length > 0)
    {
        sCmd = aCommands[0].childNodes[0].nodeValue;
    }
    ... (Process the command)
}
}
```

Partial tasklist.js (4 of 7)

```
function AJAXCallback(oRequest)
{
  if (oRequest.readyState == 4)
  {
    if (oRequest.status == 200)
    {
      ... (Parse the response)

      switch (sCmd)
      {
        case 'addtask':
          gnNextTaskId++;
          addRow(oTask);
          break;

        case 'remtask':
          removeRow(oTask);
          break;
```

Partial tasklist.js (5 of 7)

```
    case 'completetask':
        removeRow(oTask);
        addRow(oTask);
        break;

    default:
        log(kERROR, 'Unexpected command:' + sCmd);
        break;
}
}
}
```


Partial tasklist.js (6 of 7)

```
function addRow(oTask)
{
    var oTable = document.getElementById('tasklist');

    sCheckboxState = "";
    if (oTask.bCompleted)
        sCheckboxState = ' checked="checked"';

    var oTR = null;
    if (oTask.bCompleted)
    {
        oTR = document.createElement('tr');
        oTable.tBodies[0].appendChild(oTR);
    }
}
```

Partial tasklist.js (7 of 7)






```
else
    oTR = oTable.tBodies[0].insertRow(oTR, 0);
oTR.id = 'row-' + oTask.nID;
oTD = document.createElement('td');
oTD.style.cssText = "text-align: center;";
oTD.innerHTML = '<input type="checkbox" value="true"
onclick="AJAXCompleteTask(' + oTask.nID + ', ' +
!oTask.bCompleted + ')" + sCheckboxState + ' />';
oTR.appendChild(oTD);

oTD = document.createElement('td');
oTD.innerHTML = oTask.sName;
oTR.appendChild(oTD);

oTD = document.createElement('td');
oTD.style.cssText = "text-align: center;";
oTD.innerHTML = '<input type="button" value="X"
onclick="AJAXDeleteTask(' + oTask.nID + ')" />';
oTR.appendChild(oTD);
}
```

Deployment Archive

WingspanAJAXPar.par

Name	Type	Modified	Size	Ratio	Packed	Path ▲
 debug.js	JScript Script File	12/29/2005 3:51 PM	634	46%	343	js\
 tasklist.js	JScript Script File	12/29/2005 3:51 PM	5,767	67%	1,896	js\
 portalapp.xml	XML Document	12/29/2005 3:51 PM	900	67%	298	portal-inf\
 view.jsp	JSP File	12/29/2005 3:51 PM	3,061	59%	1,249	PORTAL-INF\jsp\tasklist\
 core.jar	JAR File	12/29/2005 3:51 PM	7,300	13%	6,354	PORTAL-INF\private\lib\

portalapp.xml (1 of 2)

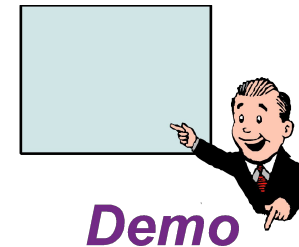
```
<?xml version="1.0" encoding="ISO-8859-1"?>
<application>
  <application-config>
    <property name="Vendor" value="wingspan.com" />
    <property name="SecurityArea" value="Wingspan" />
  </application-config>
  <components>
    <component name="Tasklist">
      <component-config>
        <property name="ClassName"
value="com.wingspan.example.sap.JSPIView" />
        <property name="SafetyLevel" value="no_safety" />
        <property name="portlet.jsp.path" value="/tasklist/" />
      </component-config>
      <component-profile/>
    </component>
  </components>
</application>
```

portalapp.xml (2 of 2)

```
<component name="TasklistServlet">  
  <component-config>  
    <property name="ClassName"  
value="com.wingspan.example.sap.TasklistServlet"/>  
    <property name="ComponentType" value="servlet"/>  
  </component-config>  
  <component-profile/>  
</component>  
</components>  
<services/>  
</application>
```

iView Demonstration

- **Build PAR**
- **Deploy PAR to SAP Enterprise Portal**
- **Create iView Instance**
- **Create Page**
- **Add iView to Page**



What We'll Cover ...

- Introduction
- **AJAX (Asynchronous JavaScript and XML)**
- Example iView
- **Debugging Techniques**
- **Advanced Topics**
- **Wrap-up**

Welcome to the World of AJAX

- **You are completely on your own**
- **JavaScript alerts can cause timing problems and are generally irritating**
- **The best approach is to build your own DHTML logging facility into your iViews**

Simple JavaScript Log Framework (tasklist.js) (1 of 2)

```
// Log Levels
var kNONE = 0;
var kERROR = 1;
var kWARN = 2;
var kINFO = 3;
var kDEBUG = 4;

var gnLogLevel = kNONE
var gfLogCallback = null;

function initLogging(nLevel, fLogCallback)
{
    gnLogLevel = nLevel;
    gfLogCallback = fLogCallback;
}
```

Simple JavaScript Log Framework (tasklist.js) (2 of 2)

```
function log(nLevel, sMsg)
{
    if (null != gfLogCallback && nLevel <= gnLogLevel)
        gfLogCallback(nLevel, sMsg);
}
```

Initializing the Log Framework (view.jsp) (1 of 2)

```
<div id="logpanel"></div>
<script>
    function logCallback(nLevel, sMsg)
    {
        var sColor = 'black';
        switch (nLevel)
        {
            case kERROR: sColor = 'red'; break;
            case kWARN: sColor = 'yellow'; break;
            case kINFO: sColor = 'yellow'; break;
            case kDEBUG: sColor = 'gray'; break;
        }
    }
}
```

Initializing the Log Framework (view.jsp) (2 of 2)

```
var oEntry = document.createElement('div');  
oEntry.style.color = sColor;  
oEntry.innerHTML = sMsg;
```

```
var oPanel = document.getElementById('logpanel');  
oPanel.appendChild(oEntry);
```

```
}
```

```
initLogging(kERROR, logCallback);
```

```
</script>
```

JavaScript API Inspection

- **debug.js**

```
function objectToString(oObject)
{
    var sString = "Object Contents:"
    for(sProperty in oObject)
    {
        var sValue = oObject[sProperty];
        sString += '\n ' + sProperty + ' = ' + sValue;
    }
    return sString;
}
```

- **Binary objects cannot be inspected using this facility**
 - This includes the XMLHttpRequest

XMLHttpRequest Properties (1 of 2)

- **No obvious “authoritative” API resource**
- **Property: ReadyState**
 - 0 – Uninitialized
 - 1 – Loading
 - 2 – Loaded
 - 3 – Interactive
 - 4 – Completed
- **Property: Status**
 - HTTP Response Code (200, 404, 500, etc.)

XMLHttpRequest Properties (2 of 2)

- **Property: ResponseXML**

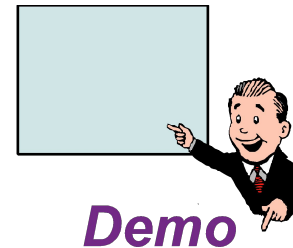
- Will be empty if response was not XML or if there was an error parsing the XML (no errors will be reported in this case)
- API of Internet Explorer's XML Data Island

- **Property: ResponseText**

- Always contains the unmodified response text

Logging Demonstration

- **Activate logging**
- **Rebuild PAR**
- **Redeploy PAR**
- **View existing page**



What We'll Cover ...

- **Introduction**
- **AJAX (Asynchronous JavaScript and XML)**
- **Example iView**
- **Debugging Techniques**
- **Advanced Topics**
- **Wrap-up**

Multiple iView Instances

- **As implemented, you cannot put two Tasklist iViews on the same page**
 - Only one will dynamically update
- **To address this, each HTML element ID in an AJAX iView must be per-page unique**
 - “Mangle” existing IDs by appending a consistent, unique value
 - ▶ `IPortalComponentRequest.getComponentContext().getContextName()`
- **JavaScript functions must accept this “mangler” as an argument for proper processing**
 - Alternatively, JavaScript functions can be name-mangled as well

Inter-iView Communication

- **These mechanisms can be used to allow iViews to communicate with each other**
- **When processing an AJAX response, multiple iViews can be updated**

AJAX Framework

- **Framework**
 - **Generic AJAX JavaScript library**
 - **Single AJAX Servlet**
 - **Interface for handling AJAX requests**
- **AJAX call**
 - **JavaScript invoke method**
 - **Specify server class name**
 - **JavaScript callback**

Third-Party Communications with AJAX

- **AJAX is restricted from accessing servers other than the one that delivered the initial page**
 - **Other servers can still be accessed, but a proxy must be installed**
- **Do not be afraid – this is not inconsistent with the general nature of portal servers**

What We'll Cover ...

- **Introduction**
- **AJAX (Asynchronous JavaScript and XML)**
- **Example iView**
- **Debugging Techniques**
- **Advanced Topics**
- **Wrap-up**

Resources

- **Wikipedia, the free encyclopedia**
 - <http://en.wikipedia.org/wiki/AJAX>
- **Adaptive Path**
 - <http://www.adaptivepath.com/publications/essays/archives/000385.php>
- **Sun Developer Network**
 - <http://java.sun.com/developer/technicalArticles/J2EE/AJAX/index.html?cid=59754>
- **Downloadable Examples**
 - <http://www.wingspan.com/downloads/SAPConference2006.zip>

7 Key Points to Take Home

- **AJAX improves the performance of portal-based applications**
- **AJAX interfaces are more dynamic and intuitive**
- **AJAX is a cross-browser technique**
- **AJAX is an advanced technique and requires a different thought process**
- **Tools are primitive or non-existent – develop your own log and debug facilities**
- **AJAX is a loose specification, use whatever similar model suits your needs and development style**
- **Download this presentation and example project**

Your Turn!



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