ORACLE®

Introduction to ATG Page Development

Presenter’s Name
Presenter’s Title

Safe Harbor Statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions.
The development, release, and timing of any features or functionality described for Oracle’s products remains at the sole discretion of Oracle.

Oracle Training Materials – Usage Agreement

Use of this Site (“Site”) or Materials constitutes agreement with the following terms and conditions:

1. Oracle Corporation (“Oracle”) is pleased to allow its business partner (“Partner”) to download and copy the information, documents, and the online training courses (collectively, “Materials") found on this Site. The use of the Materials is restricted to the non-commercial, internal training of the Partner’s employees only. The Materials may not be used for training, promotion, or sales to customers or other partners or third parties.

2. All the Materials are trademarks of Oracle and are proprietary information of Oracle. Partner or other third party at no time has any right to resell, redistribute or create derivative works from the Materials.

3. Oracle disclaims any warranties or representations as to the accuracy or completeness of any Materials. Materials are provided "as is" without warranty of any kind, either express or implied, including without limitation warranties of merchantability, fitness for a particular purpose, and non-infringement.

4. Under no circumstances shall Oracle or the Oracle Authorized Delivery Partner be liable for any loss, damage, liability or expense incurred or suffered which is claimed to have resulted from use of this Site of Materials. As a condition of use of the Materials, Partner agrees to indemnify Oracle from and against any and all actions, claims, losses, damages, liabilities and expenses (including reasonable attorneys' fees) arising out of Partner’s use of the Materials.

5. Reference materials including but not limited to those identified in the Boot Camp manifest can not be redistributed in any format without Oracle written consent.

Learning Objectives

At the end of this lesson, you should be able to:

- Construct JSP Pages using the most common JSP Elements and JSTL Tags
- Use Expression Language Constructs in JSP Pages
- Understand DSP Tag Library and how to use them
- Reuse JSP pages by including them in other pages
- Embed Nucleus components to expose business functionality

Agenda

- Java Server Pages
- Dynamo Server Pages
- DSP: Reusing Application Pages

Section 1:
Java Server Pages

JSP : An introduction

- JSP page is a text based document that contains two types of text:
  - Static template data (such as HTML, XML, etc.)
  - JSP Elements which constructs dynamic content
- JSP technology uses XML-like tags encapsulating logic that generates the content for the page typically in HTML.
- The application logic can reside in server-based resources (such as Droplets).
- By separating the page logic from its design and display and supporting a reusable component-based design, JSP technology makes it faster and easier than ever to build Web-based applications.

JSP: Advantages

- Together JSP technology and droplets provide the following advantages:
  - Platform independence
  - Enhanced performance
  - Separation of logic from display
  - Ease of administration
  - Extensibility into the enterprise
  - Ease of use
- JSP specification is an open standard that is the product of industry wide colloboration.

Types of JSP Elements (1)

- Directives
  - An option setting.

```
<%@page import="java.util.*" %>
```

- Expressions
  - Any java expression. The expression is evaluated and the text inserted into the html.

```
<%=new Date()%>
```

- Scriptlets
  - Sequence of java statements. Result is evaluated and text is inserted into the html.

```
<%
    out.println("Hello World");
%>
```

Types of JSP Elements (2)

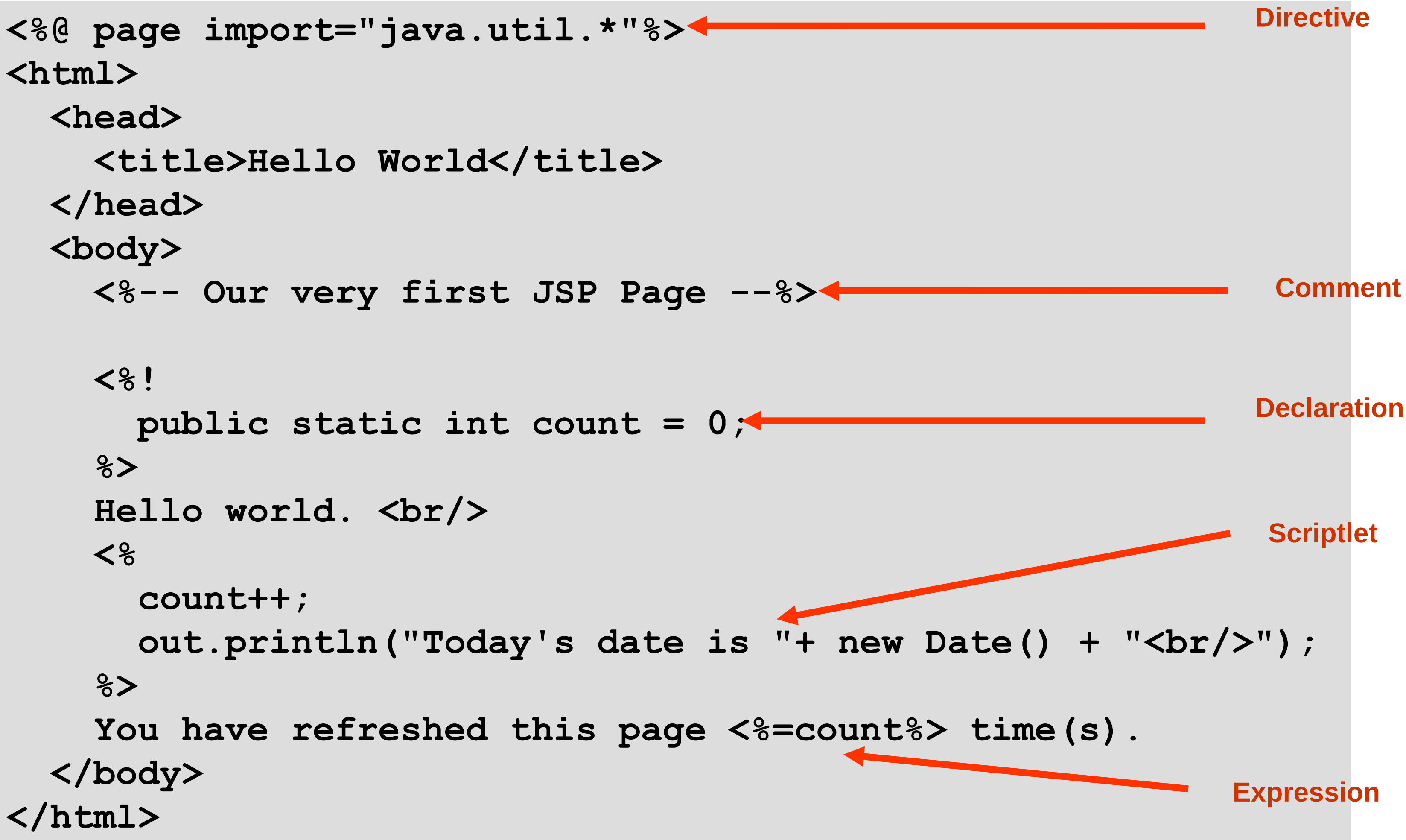
- Declarations
  - Declaration is used to create variables that can be used in expressions and scriptlets.

```
<%!
   private static int count = 1;
%>
```

- Comment
  - A comment is used to document a JSP Page.

```
<%--  This is a comment --%>
```

Sample JSP Page



```
<%@ page import="java.util.*"%>
<html>
  <head>
    <title>Hello World</title>
  </head>
  <body>
    <%-- Our very first JSP Page --%>

    <%!
      public static int count = 0;
    %>
    Hello world. <br/>
    <%
      count++;
      out.println("Today's date is "+ new Date() + "<br/>");
    %>
    You have refreshed this page <%=count%> time(s).
  </body>
</html>
```

Directive

Comment

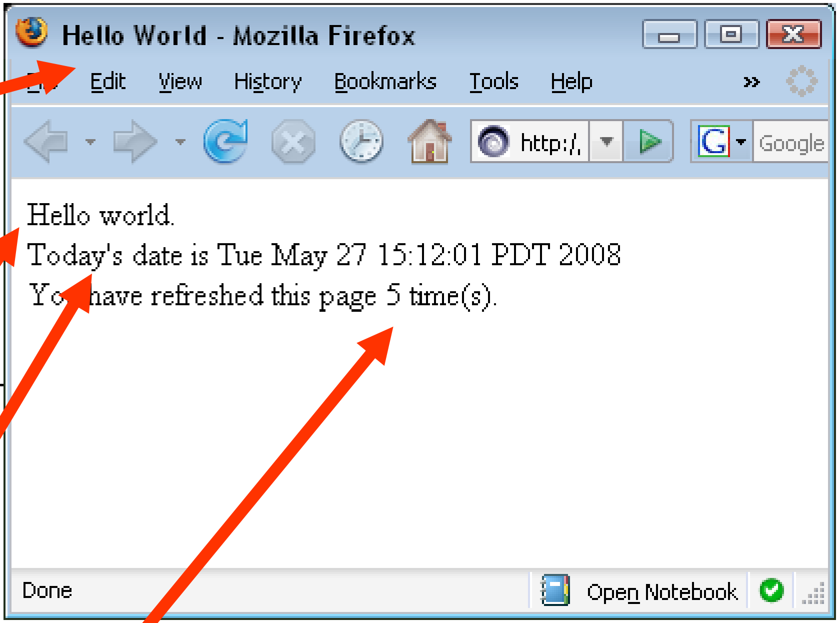
Declaration

Scriptlet

Expression

Result of execution

```
<%@ page import="java.util.*"%>
<html>
  <head>
    <title>Hello World</title>
  </head>
  <body>
    <%-- Our very first JSP Page -
    <%!
  public static int count = 0;
    %>
     Hello world. <br/>
    <%
      count++;
      out.println("Today's date is "+ new Date() + "<br/>");
    %>
     You have refreshed this page <%=count%> time(s).
  </body>
</html>
```

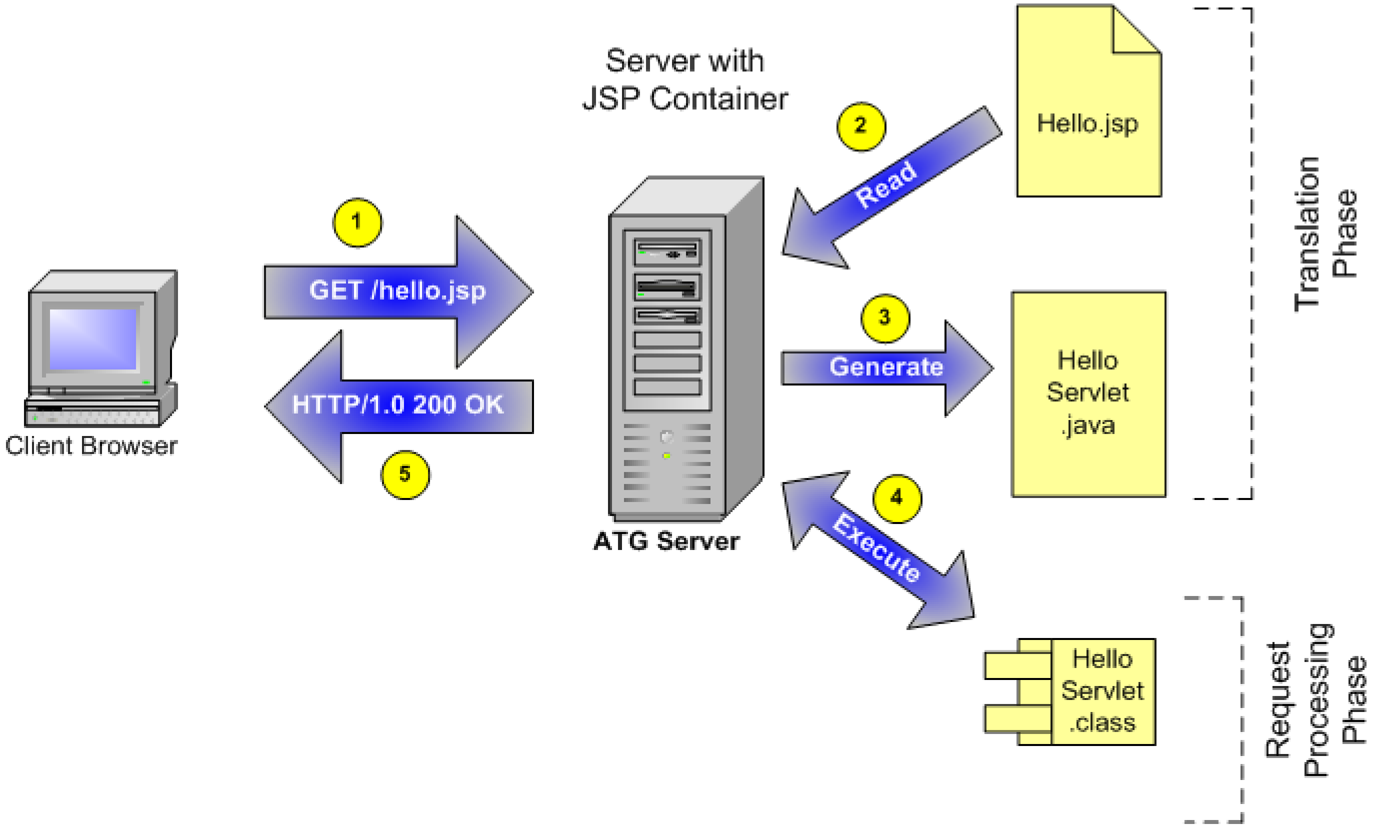


Hello world.
Today's date is Tue May 27 15:12:01 PDT 2008
You have refreshed this page 5 time(s).

JSP Container

- JSP Page is represented at execution time by a JSP page implementation object, which is a servlet (javax.servlet.Servlet) and is executed by JSP container.
- JSP container delivers requests from a client to a JSP servlet and responses from servlet to a client.
- Container locates the appropriate instance of the JSP page implementation class and delivers requests to it using servlet protocol.
- Most JSP pages use HTTP as the communication protocol, and their implementation classes must implement the HttpJspPage interface, which extends JspPage.

JSP Container Illustrated



- Steps 2 and 3 are executed only the first time the jsp is accessed after it has been created or modified..

Introduction to Java Servlets

- Java Servlets are the technology of choice for extending and enhancing Web servers:
  - They run on a server.
  - They respond to client requests.
  - They typically generate dynamic content (HTML).
  - Servlets implement the javax.servlet.Servlet interface.

Java Servlet Example

- The javax.servlet.Servlet interface contains a service() method that:
  - Performs the work of the servlet.
  - Services requests and generates HTML.

```
public void service
        (HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException, IOException
```

Commonly Used JSP Directives

- Page directive
  - Lets you set page directives such as buffer length, content type, import classes, etc.

```
<%@ page content-Type=“text/html” %>
```

- Include directive
  - Lets you include JSP fragments into the JSP Page.

```
<%@ include file="file" %>
```

- Taglib directive
  - Used to define custom tag libs.

```
<%@ taglib uri="uriToTagLibrary" prefix="prefix" %>
```

JSP Taglib Overview

- Tags encapsulate functionality and are used in JSP.
- Component-based programming and code reuse.
- Usage:
  - Reference a TLD file via taglib directive:

```
<%@ taglib uri="http://java.sun.com/jstl/core"
  prefix="c” %>
```

  - Use in jsp file:

```
<c:if test=”${empty param.name}”>
     Please specify your name.
</c:if>
```

The JSP Standard Tag Library (JSTL)

- JSTL is the Java Server Pages Standard Tag Library.
- Provides a set of standard tags for common operations such as:
  - Iteration and conditionals
  - XML document manipulation
  - Internationalization
  - Database access
- EL(Expression language)
  - Makes it easily to access application data stored in JavaBeans

JSTL Commonly Used Core Tags

Area	Function	Explanation
Variable	set	To set a variable
Flow Control	choose	Used for multiple if (like a switch in java)
	when	Used by choose to specify conditions
	otherwise	Used by choose when no condition applies
	foreach	Iterations
	if	Conditional sentence
Page	redirect	Redirect to another page
	param	Param to be passed
	url	Used to rewrite urls
Others	out	Print out variable on a page

Expression Language Concept

- JSTL includes an expression language to minimize the use of JSP scriptlets and expressions.
  - In the past, we used Java-like codes to retrieve a bean property.

    Example:
    ```
    <%= person.getAddress().getState() %>
    ```

  - With JSTL and expression language, Syntax is used instead.

    Example:
    ```
    <c:out value="${person.address.state}"/>
    ```

- The expression language usually handles data type conversion and null values.

Expression Language Basics

- EL expression always starts with ‘${’ and ends with ‘ }’.
- EL expressions can include:
  - literals (“1”, “100” etc)
  - variables
  - implicit variables
- EL expression can be used inside the other JSTL.
- EL implicit objects:

  pageContext, session, request, etc.
- Examples:

```
${100}  ${100*3+2}  ${a+2}
 ${requestScope.loginUser}
 ${myMessage.message}
```

Expression Language Operators

- Arithmetic:

+, - (binary), *, / and div, % and mod, - (unary)

- Logical:

and, &&, or, ||, not, !

- Relational:

==, eq, !=, ne, <, lt, >, gt, <=, ge, >=, le

- Empty:

The empty operator is a prefix operation that can be used to determine whether a value is null or empty

- Conditional:

A ? B : C

Advanced Expression Language

- Evaluating map:
  - The value of the entry named deptName in the department’s map.

```
    ${departments[deptName]}
```

- Evaluating array:

```
public static String[] colors = new String[] {“red”,
  “green”, “blue”, “orange”, “black”};
pageContext.setAttribute(“colors”, colors);
…..
<font color=“${colors[1]}”> User </font>
```

- Using function:

```
    <c:out value="${fn:toUpperCase(sentence)}"/>
```

Section 1 ✔
Check Your Understanding

What output do JSP pages typically produce?

a. XML that JBOSS interprets

b. HTML content that browsers interpret

c. JSTL, DSP and EL syntax

d. Text content

e. A rendered image of the web page

Answer: b

Section 1 ✔
Check Your Understanding

Name two of the advantages of JSP and Droplet Technology?

a. Enhanced Performance

b. Pre compiled shopping cart solutions

c. Separation of logic from Display

d. Analytics on page usage

e. Customer User Experience monitoring

Answer: a, c

Section 1 ✔
Check Your Understanding

A Server Side JSP Comment is defined using the tags

a. <!-- Comment -->

b. <%-- Comment -->

c. <%-- Comment --%>

d. <!-- Comment --!>

e. <% Comment %>

f. <%! Comment %>

Answer: c

Section 1 ✓
Check Your Understanding

When a request is sent to the server, the server locates the .class file corresponding to the JSP, executes it and returns output to the client. When is the .class file generated

a. On application deployment

b. On server startup

c. On first access of the page

d. On every request

e. On Server startup. However, if page does not exist on first access, it is created.

Answer: c

For Oracle employees and authorized partners only. Do not distribute to third parties.

Section 1 ✔
Check Your Understanding

Name two commonly used page directives

a. <%@ page ..

b. <%@ dsptaglib

c. <%@ jstl

d. <%@ include ..

e. <%@ forEach

Answer: a,d

Section 1
Check Your Understanding

Which of the following is not allowed in a EL expression

a. + (Plus)

b. ge (Comparator)

c. <= (Comparator)

d. A?B:C (Tertiary Conditional Operator)

e. - (Unary negation)

f. if (Conditional)

Answer: f

Summary

- Java Server Page or JSP is a technology used to produce dynamic web pages.
- JSP is server based and has an XML like tag syntax.
- Typical JSP Page has Directives, comments, declarations, scriptlets, and expressions along with HTML markup.
- A JSP Page is represented as an execution time as a servlet.
- Tag libraries can be added to the application to get added functionality.
- JSTL and DSP are examples of such tag libraries.

Section 2:
Dynamo Server Pages

Introduction

- ATG ships with JSTL and DSP Tag Libraries out of the box.
- Dynamo Server Pages or DSPs are JSP Pages that use DSP Tag Libraries to display dynamic values on the page.
- DSP Tag Libraries are used to provide:
  - Dynamic values display.
  - Accessing and setting nucleus component properties.
  - Use page parameters to get dynamic capability.
  - Use tag converters for formatting.
  - Working with EL variables in conjunction with nucleus components.

What is DSP Tag Library

- In the previous section, we learned that tag Libraries provide additional capability to the JSP Page and application.
- ATG platform ships with Dynamo Server Pages (DSP) and JSTL.
- DSP tag libraries let you access all data types in ATG’s Nucleus framework.
- Other functions provided by these tags manage transaction and control data rendering in JSP.

For Oracle employees and authorized partners only. Do not distribute to third parties.

Frequently Used DSP Tags

- Tags for working with pages:
  - dsp:page
  - dsp:valueof
- Tags for including and linking to other pages:
  - dsp:include
  - dsp:a
- Tags for working with Nucleus components:
  - dsp:importbean
  - dsp:setvalue
  - dsp:getvalueof
  - dsp:droplet
  - dsp:form

DSP Tags for working with a page

- dsp:page encloses a JSP.
- dsp:valueof allows for display of page parameters, component properties, and constant values.
- DSP tag names and attributes that have equivalent names in HTML typically replicate the functionality of the corresponding tag and provide extra functionality:
  - dsp:a is similar to a html tag but allows passing of component properties
  - dsp:frame embeds a page
  - dsp:iframe embeds a dynamic page
  - dsp:img is equivalent to an image tag
  - dsp:link is similar to link html tag

dsp:page

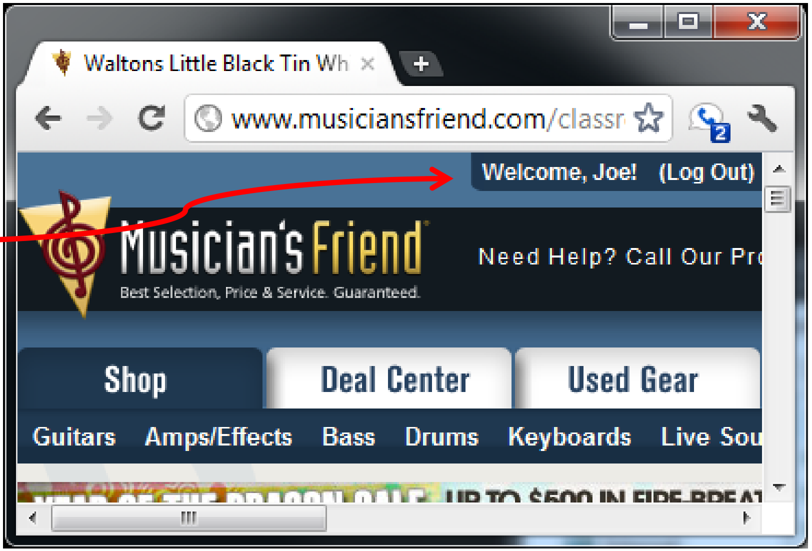
- dsp:page encloses a JSP.
- The first tag follows the page directive and the last tag ends the page.
- dsp:page invokes the JSP handler, which calls the servlet pipeline and generates HTTPServletRequest.
- The servlet pipeline manages generic page-related tasks, such as retrieving the servlet session components, instructing the page to render its contents.

```
<%@ taglib uri=http://www.atg.com/taglibs/daf/dspjspTaglib1_0
     prefix="dsp" %>
<dsp:page>
        Hello world!
</dsp:page>
```

Displaying values with dsp:value

- dsp:value renders a page parameter, bean property, or a static value.
- The value tag can include a tag converter such as date, number, currency, etc.
- It can also accept EL parameters as source.

```
   Welcome,
   <dsp:valueof param=“name">
      Guest
   </dsp:valueof>
```



Using dsp equivalent of html tags

- dsp:a defines a link to a target destination.

```
<dsp:a href="homepage.jsp">Go to your home page
<dsp:param name="city"
    beanvalue="Profile.homeAddress.city"/>
</dsp:a>
```

- dsp:link references a file such as stylesheet.

```
<dsp:link rel="stylesheet‟ type="text/css"
href="Style.css‟ title=‟My StyleSheet"/>
```

- dsp:img inserts an image into a page.

```
<dsp:img page="en/images/IdPhoto.gif">
```

DSP tags for working with other pages

- dsp:include allows page fragments to be embedded in a page.
- dsp:param is used to pass page parameters.
- dsp:a allows linkage between pages.

Page Parameters Overview

- What are page parameters?
  - Page Parameter is a named value that one page can pass to another.
  - It can be set to a component property, another parameter, or a constant value.
- Why do we need page parameters?
  - To pass data around and make the pages perform the correct logic.
- They can be passed using:
  - dsp:a
  - dsp:frame
  - dsp:iframe
  - dsp:img
  - dsp:include
  - dsp:link

For Oracle employees and authorized partners only. Do not distribute to third parties.

Passing Page Parameters

- You can pass parameters to a target page by embedding one or more **`<dsp:param>`** tags.

```
<dsp:include page=“pageFragment.jsp">
  <dsp:param name=“firstName" value=“Joe"/>
</dsp:include>
```

- Parameters can be passed to other pages via a hyperlink using a DSP anchor tag (dsp:a).

```
<dsp:a page=“pageLink.jsp">
  <dsp:param name=“firstName" value=“Joe"/>
</dsp:a>
```

- Parameters passed to embedded pages may be of any type (primitives or Java objects).

Accessing Page parameters

- The target page, whether included or linked to, can access the page parameter either with dsp:value of or as a EL variable.
- From the previous page’s example:

```
Welcome <dsp:valueof param=“firstName">
            Guest
 </dsp:valueof>
```

- Or:

```
Welcome ${param.firstName}
```

- Page parameter s get passed around in the query string.

Page Parameter Scope

- Page parameters are accessible only to the current page and its embedded child pages.
- A page that is embedded in another page automatically inherits the parameters of its parent page.
- Page parameters can be overridden with the dsp:setvalue tag.

```
<dsp:setvalue param=“firstName" beanvalue="member.name"/>
```

- When the page parameter is overridden, it takes into effect from that point forward on the page.

Section 2 ✔
Check Your Understanding

What are the two tag libaries that ATG ships with?

a. DSP Tag Libraries.

b. DYN Tag Libraries

c. Apache Tag Libraries

d. JSTL Tag Libaries.

e. JSP Tag Libraries

Answer: a, d

Section 2
Check Your Understanding

What two purposes is the dsp:valueof used for?

a. Copy nucleus component properties to EL variables

b. Display nucleus components properties

c. Display page parameters

d. Copy page parameters to EL variables

e. Log activity on web site for analytics processing

Answer: b, c

Section 2 ✔
Check Your Understanding

What tag should wrap all jsp pages that use DSP tag libraries?

a. <@ page directive

b. <dsp:page> tag

c. <dsp:include> tag

d. <dsp:process> tag

e. <dsp:valueof> tag

Answer: b

Section 2
Check Your Understanding

What is the visibility of the page parameters?

a. Page parameters are visible to the page only and not embedded pages

b. Page parameters are visible to the page and embedded pages.

c. Page parameters are visible to all pages on the site

d. Page parameters are not visible unless configured in the page.properties file. Visibility and scope depends on settings there

Answer: b

Summary

- Dynamic pages can be built with DSP Tag libraries.
- DSP Tag Libraries let you access ATG nucleus framework in JSP Page.
- ATG provides dsp:valueof to access page parameters and nucleus component properties.
- Page parameters are named values that are passed between pages.
- Page parameters help reuse page code by allowing the embedded page to react dynamically.

Section 3:

Dynamo Server Pages:

Reusing and Accessing Components

Reusing Application Components with dsp:include

- ATG Platform helps you to build an application from reusable components.
- A JSP can embed components such as other JSPs or images.
- Reusing the nucleus component (droplets) provides:
  - Single place to edit changes avoiding duplication.
  - Embedded files can respond dynamically via parameters.
- The dsp:include can be used as follows:

```
  <dsp:include page="RegistrationForm.jsp"/>
```

Including pages

- Simple static files can be included with the JSP include directive.

```
<%@include file=“header.jspf” %>
```

- A JSP can embed another JSP by calling dsp:include or jsp:include.

```
<jsp:include page="RegistrationForm.jsp"/>
or
<dsp:include page="RegistrationForm.jsp"/>
```

- dsp:include offers these benefits:
  - It can pass page parameters from parent to child page.
  - Its page attribute lets you set an absolute URL.
  - URLs resolve relative to current page.

Relative and Absolute URLs

- Relative URLs
  - Relative paths typically begin in the parent’s directory.
  - The src and page attributes of DSP tags such as dsp:include and dsp:a let you specify relative URLs.
  - Can be reused by pages in separate directories or moved to another portion of the tree.
- Absolute URLs
  - Absolute paths typically begin in the web application’s root directory.
  - Tags that invoke other files such as dsp:include can set their page attribute to an absolute path.

Invoking Pages in Other Web Applications

- Tag dsp:include lets you designate a target page in any J2EE Web application. You can do this in three ways:
  - Use the WebAppRegistry.
  - Use the src attribute.
  - Use attributes src and otherContext.

Passing Parameters to Embedded Pages

- Application typically contains embedded pages to promote code reuse.
- Page parameters can be sent to embedded pages using dsp:param tag.

```
<dsp:include page=“header.jsp”>
     <dsp:param name="storename" value="Spinning Wheel"/>
</dsp:include>
```

- For another page, the store name can be different.
- The embedded page can access the storename as:

```
<h1>Welcome to <dsp:valueof param="storename“ /></h1>
```

Tags for Working with Nucleus Components

- The following tags are used for working with nucleus components:
  - dsp:importbean
  - dsp:setvalue
  - dsp:getvalueof
  - dsp:droplet
  - dsp:form
- dsp:importbean is used to import a servlet or JavaBean component into a JSP.
- dsp:setvalue and getvalueof are used to set and get properties.

Import Components

- dsp:importbean imports servletbeans and JavaBean component s into a JSP.

```
<dsp:importbean bean="/MyModule/MyComponent‟
                                   var=‟MyComponent”/>
```

- After a component is imported, references can omit the full nucleus path.

```
<dsp:valueof bean=‟MyComponent.name"/>
```

- EL –enabled resources on the same page can access the variable.

```
<dsp:setvalue bean="${Mycomponent.refresh}" value=" "/>
```

Setting Component Properties

- Two dsp tags can be used to set component properties.
- You can set a property in a page with the setvalue tag.

```
   <dsp:setvalue bean="Student_01.name" value="Brian"/>
```

- The anchor property can also be used to set properties. The above can also be achieved as:

```
<dsp:a href="page2.jsp"
       bean="/samples/Student_01.name" value="Ron">
    If your name is Ron, click here!
</dsp:a>
```

Using JSTL with DSP Tags

- dsp:importbean and dsp:droplet can produce scoped attributes that can be used by JSTL tags.

- ATG provides two versions of the DSP tag library. Each has the same tags but supports different expression languages for use within the tag:
  - EL – tags supporting the JSTL expression language.
  - Default (RT) – tags supporting JSP runtime expressions.
  - You cannot use both EL and RT statements in the same tag.

Exposing Nucleus Components to EL

- The dsp:importbean tag has been extended to accept two additional attributes:
  - var (optional) - the name of the attribute which will be set with the value of the nucleus component.
  - scope (optional) - the scope in which the attribute should be set ("page", "request", "session", "application") - it defaults to "page."

- The variable can be used in EL format to access the bean.

Code Sample

```
<%@ taglib uri="/dspELTaglib" prefix="dspel" %>
<dspel:importbean var="Profile"
       bean="/atg/userprofiling/Profile"/>
<c:choose>
    <c:when test="${Profile.transient == true}" >
       Welcome to Dynamusic.
        <a href="newProfile.jsp">Click here</a>
        to register. <p>
    </c:when>
    <c:otherwise>
       <p>Welcome,
          <dsp:valueof bean="Profile.firstName"/><p>
    </c:otherwise>
</c:choose>
```

Section 3 ✔
Check Your Understanding

What two mechanism does the ATG platform provide to promote code reuse?

a. dsp:link to link to other nucleus components
b. dsp:href to point to other JSP pages
c. Droplets
d. dsp:include to include other JSP Pages
e. A large number of utility classes that can be used as is

Answer: c, d

Section 3 ✓
Check Your Understanding

What are the advantages for dsp:include over jsp:include?

a. jsp:include is a static pre compile inclusion of jsp page where as dsp:include is a dynamic page include

b. dsp:include allows object parameters to be passed, jsp:include does not

c. dsp:include allows for jsps in other folders to be included, jsp:include only allows jsps in same folder.

d. dsp:include allows image embedding, jsp:include does not.

Answer: b

Section 3
Check Your Understanding

What are the two ways of setting nucleus component attributes in JSP pages?

a. dsp:setvalue tag

b. dsp:a tag

c. dsp:valueof tag

d. dsp:setattribute tag

e. dsp:set tag

Answer: a, b

Section 3 ✔
Check Your Understanding

How do you access entire nucleus components in the EL format?

a. Use the dsp:expose tag with var parameter

b. Use the dsp:importbean tag with var parameter

c. Use the dsp:setvalue tag

d. Use the dsp:set tag

Answer: b

Summary

- ATG supports and highly recommends application code reuse using the dsp:include and page parameters.
- Pages from other J2EE Applications can be embedded as well.
- Page parameters allow for dynamic behavior in included and forwarded pages.
- EL expressions can be used to access nucleus components using the importbean tag.

Key Points

- Nucleus is a key ATG feature for building component based web applications.
- Simple JavaBeans can be combined with configuration property files to make components.
- Components can have global, session, request, or window scope.
- Configuration property files have to be specified in the CONFIGPATH.
- ATG will smartly layer and merge property files for components.
- GenericService implements useful Nucleus interfaces to add additional functionality.

ORACLE IS THE INFORMATION COMPANY

ORACLE®