



Cognizant  
Technology  
Solutions



Cognizant  
Academy



# JSP - Custom Tags

Basic

C3: Protected

# About the Author

<b>Created By:</b>	<b>Mohamed, Hajura (Cognizant), 117960</b>
<b>Credential Information:</b>	<b>Executive- Cognizant Academy</b>
<b>Version and Date:</b>	<b>J2EE/PPT/0306/1.0</b>

## Cognizant Certified Official Curriculum



# Icons Used



Questions



Hands-on Exercise



A Welcome Break



Test Your Understanding



Coding Standards



Reference









Demo



Key Contacts







## Introduction:

This chapter covers the following:

-  Use of custom tags
-  Basic principals to develop custom tags
-  Interfaces, fields, and methods
-  Tag handlers
-  Body tag lifecycle
-  Iteration tag lifecycle

## **Objective:**

After completing this chapter, you will be able to:

-  Explain the role of custom tags
-  Describe the basic principals of custom tags
-  Explain tag interface
-  Define custom tags
-  Explain body tag lifecycle
-  Explain Iteration tag lifecycle

# Custom Tags: Features

- Custom tags simplify the coding efforts.
- Custom tags make the JSP page more modular.
- Custom tags promote reusability.
- The three interfaces that play a key role in custom tag development are:
  - Tag Interface
  - BodyTag Interface
  - IterationTag Interface
- The `TagSupport` classes implement the preceding interfaces.

# Role of Custom Tags

- The tags are interpreted by the J2EE Server.
- The tags promote modularity and code re-use.
- The coding effort of a JSP page developer is minimized.

- Your tag should have a Java class associated with it.
- This Java class provides the required functionality for the tag.
- This Java class is called the tag handler.
- The tag handler should extend any of the helper classes, which are `TagSupport` and `BodyTagSupport`, and override either `doStartTag()` or `doEndTag()` methods.
- These classes in turn implement the `Tag` interface, `IterationTag` interface, and `BodyTag` interface.



The implementation hierarchy is as follows:

- IterationTag Interface *extends* Tag Interface
- TagSupport Class *implements* IterationTag interface
- BodyTag Interface *extends* IterationTag Interface
- BodyTagSupport class *implements* BodyTag Interface

To use a JSP tag:

1. Build a tag library descriptor - `.tld` file. TLD is an XML file that describes your tags and associates them with their respective tag handler classes.
2. Make the container aware of the location of your TLD file by making entries in the `web.xml` file.
3. Use the tag in your JSP page.

# The Interfaces, Fields, and Methods

The three important interfaces for custom tag development are:

- Tag interface
- BodyTag interface
- IterationTag interface



Cognizant  
Technology  
Solutions

# The Tag Interface



Cognizant  
Academy

## Description:

- Simplest interface for JSP tag handlers
- Classes that implement this interface can perform custom actions at their end tags.
- These classes have no control over body content of the tag and cannot perform iterations.

# The Tag Interface (Contd.)

## Fields:

- **`public final static int EVAL_BODY_INCLUDE`**
  - This field instructs the server to evaluate the body content and place it in the out stream.
  - This is the value returned by `TagSupport.doStartTag()`, if it is not overridden.
- **`public final static int SKIP_BODY`**
  - This field instructs the server explicitly not to process any body content.
  - This value is returned by `doStartTag()`.
- **`public final static int EVAL_PAGE`**
  - This field instructs the server to continue evaluating the page.
  - This value is returned by `TagSupport.doStartTag()`, if it is not overridden.
- **`public final static int SKIP_PAGE`**
  - This field instructs the server explicitly not to process the rest of the page.
  - This value is returned by `doEndTag()`.

## Methods:

- **public abstract int doEndTag() throws javax.servlet.jsp.jspException**
  - The server invokes this method on all tag handlers when it encounters a closing tag for this handler instance.
  - A call to `doStartTag()` should already have been made and the body content may or may not have been processed.
  - If `SKIP_PAGE` is returned, no more evaluation is done on the page and the request is completed.
  - If `EVAL_PAGE` is returned, the page evaluation continues as normal.
- **public abstract int doStartTag() throws javax.servlet.jsp.jspException**
  - The server invokes this method on all tag handlers when it encounters an opening tag for this handler instance.
  - If `SKIP_BODY` is returned, no body content is evaluated and `doEndTag()` is called.
  - If `EVAL_BODY` is returned, the body content is evaluated as normal and placed in the out stream.

# The Tag Interface (Contd.)

- **`public abstract javax.servlet.jsp.tagext.Tag getParent()`**
  - This method gets the closest tag handler that encloses the tag handler, or null if one cannot be found.
- **`public abstract void release()`**
  - This method is called by the server to allow the tag handler to release any resources it has acquired throughout its life.
  - The JSP standard states that the server must call the `release()` method for each tag handler object it creates, but it also states this tag handling objects may be re-used if the same tag is used in several places on the same page.

# The Tag Interface (Contd.)

- **public abstract void**  
**setPageContext (javax.servlet.jsp.PageContext pc)**
  - This method sets the current PageContext for the tag handler, and is called by the server before any calls to `doStartTag()`.
- **public abstract void setParent (javax.servlet.jsp.tagext.Tag param1)**
  - This method sets the parent tag handler that will be returned by `getParent()`.
  - It is called by the server before any call to `doStartTag()`, so that `doStartTag()` has access to its parent, if required.

- **Why to use custom tags?**
  - Improved separation of presentation and implementation by:
    - Reducing / eliminating scripting tags
    - Encapsulating common or application specific idioms
    - Providing an HTML-friendly layer of abstraction
  - Web designers understand tags better than java code
  - Portable and reusable: Write once and use it in many applications
  - Improves maintainability

## **Example:**

- Without Custom Tag

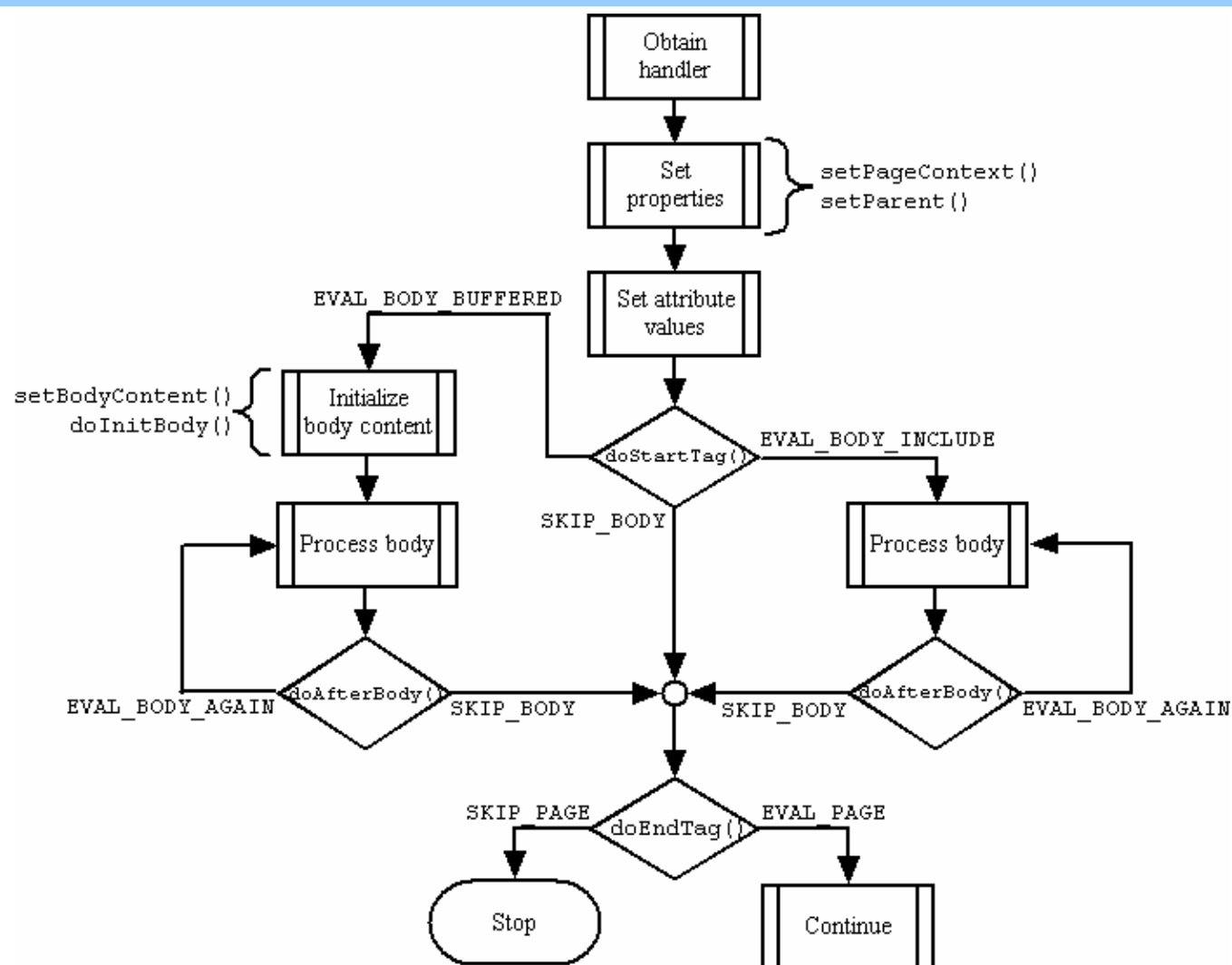
```
<%= (value!=null) ? ("The value is " + value) : "Value not  
found"%>
```

- With Custom Tag

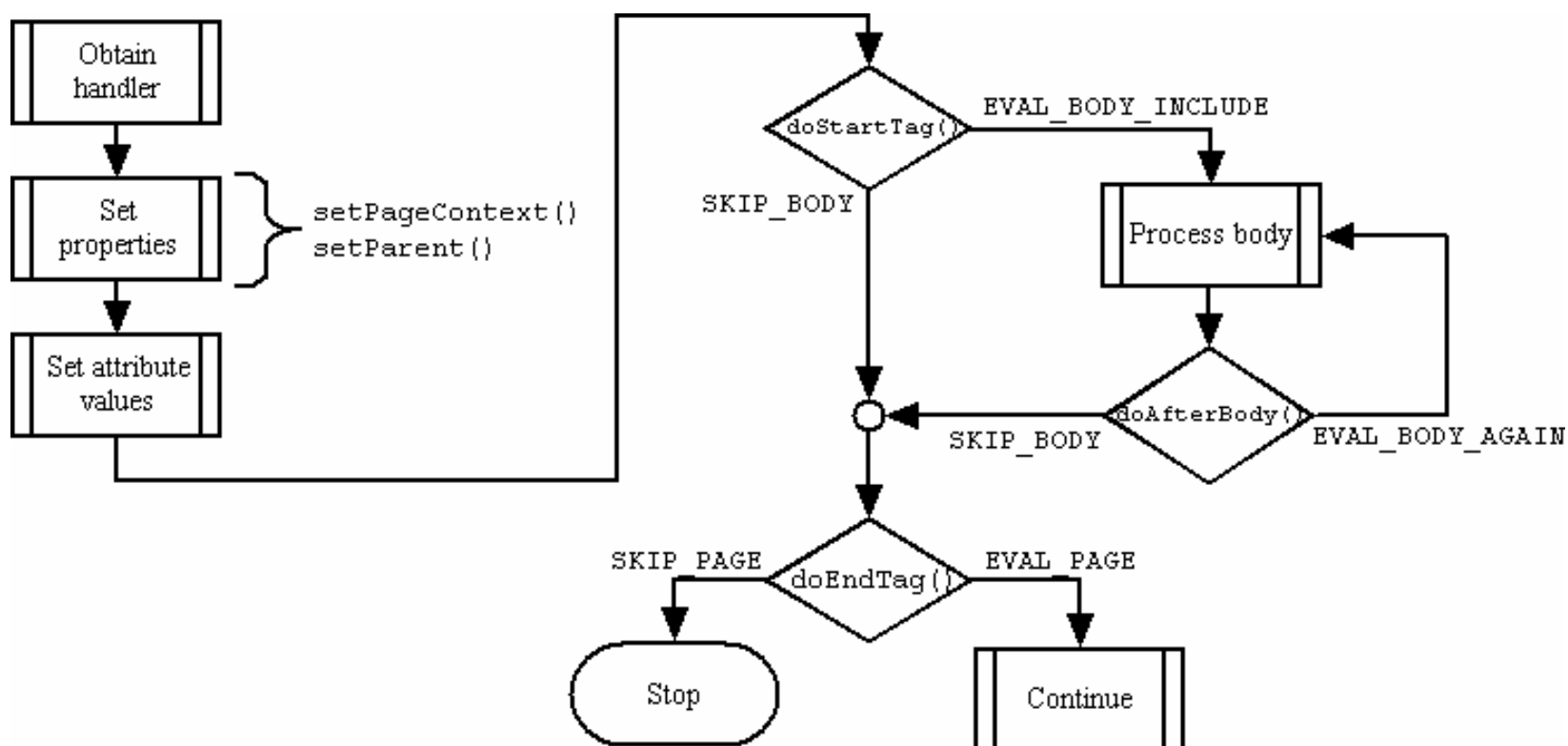
```
<lib:display name="value"/>
```



# Body Tag Life Cycle



# Iteration Tag Life cycle



# Demo 1 - Simple Custom Tag

Simple Custom Tag: Demonstrates a simple JSP tag handler to print a simple message.

```
<!-- JSPCustomTags\customhello\WEB-INF\classes\helloworld\
HelloWorld.java - - >
package helloworld;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.jsp.*;
import javax.servlet.jsp.tagext.*;
public class HelloWorld extends TagSupport{
    public int doStartTag() throws JspException {
        try{
            pageContext.getOut().print("Hello Jsp World!");
        }
        catch(Exception ioException)
        {
            System.err.println("IOException thrown in
HelloWorld.doStartTag():");
            System.err.println(ioException.toString());
            throw new JspException(ioException);
        }
    }
}
```

# Demo 1 - Simple Custom Tag (Contd.)



```
        return SKIP_BODY;
    }
    public int doEndTag() {
        return EVAL_PAGE;
    }
}
```

# Demo 1 - Simple Custom Tag (Contd.)



```
<!-- ?JSPCustomTags\customhello\WEB-INF\helloworldtags.tld - -->
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE taglibPUBLIC "-//Sun Microsystems, Inc.//DTD JSP Tag Library
1.2//EN" "http://java.sun.com/dtd/web-jsptaglibrary_1_2.dtd">
<taglib>
  <tlib-version>1.0</tlib-version>
  <jsp-version>1.2</jsp-version>
  <short-name>custhello</short-name>
  <!-- Path to this tld file. -->
  <uri>/WEB-INF/helloworldtags.tld</uri>
  <tag>
    <name>hw</name>
    <tag-class>hellopack.HelloWorld</tag-class>
    <body-content>JSP</body-content>
  </tag>
</taglib>
```

# Demo 1 - Simple Custom Tag (Contd.)



```
<!-- ? web.xml file - ->
JSPCustomTags\customhello\WEB-INF\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/j2ee/dtds/web-
  app_2_3.dtd">
<web-app>
  <display-name>Custom examples</display-name>
  <description>
    This is the collection of simple tag libraries
  </description>
  <taglib>
    <taglib-uri>/helloworldtags</taglib-uri>
    <taglib-location>/WEB-INF/helloworldtags.tld
    </taglib-location>
  </taglib>
</web-app>
```

# Demo 1 - Simple Custom Tag (Contd.)



```
<!-- JSPCustomTags\customhello\hello.jsp - - >
<%@ taglib prefix="custhello" uri="/WEB-INF/helloworldtags.tld" %>
<%@ page errorPage="error.jsp"%>

<HTML>

<HEAD>

<TITLE>Hello World Example</TITLE>

</HEAD>

<BODY>

<custhello:hw />

</BODY>

</HTML>
```

# Demo 1 - Simple Custom Tag (Contd.)



```
<!-- JSPCustomTags\customhello\error.jsp - ->
<%@ page import="java.io.*" isErrorPage="true" %>
<html>
    <head>
        <title>Error!</title>
    </head>
    <body>
        <h1>Error: <%=exception%><h1>
        <%
            StringWriter errorWriter = new StringWriter();
            PrintWriter errorStream = new PrintWriter(errorWriter);
            exception.printStackTrace (errorStream);
        %>
        <p>Stack trace:
        <blockquote>
            <pre><%=errorWriter%></pre>
        </blockquote>
    </body>
</html>
```



## Demo 2 - Custom Tag with Attributes

Demonstrates a JSP tag handler that accepts different usernames as attributes when accessed through their respective JSPs.

```
<!-- JSPCustomTags\helloattribute\WEB-  
INF\classes\attrpack\HelloUser.java - ->  
package attrpack;  
import javax.servlet.jsp.*;  
import javax.servlet.jsp.tagext.*;  
public class HelloUser extends TagSupport{  
    private String username="";  
    public String getUsername() {  
        return this.username;  
    }  
}
```



Cognizant  
Technology  
Solutions

# Demo 2 - Custom Tag with Attributes (Contd.)



Cognizant  
Academy

```
public void setUsername(String newUsername){
    this.username=newUsername;
}
public int doStartTag() throws JspException {
    try
    {
        pageContext.getOut().print("Hello " + this.getUsername() +
"!");
    }
    catch(Exception ioException) {
        System.err.println("IO Exception thrown in
HelloUser.doStartTag():");
        System.err.println(ioException.toString());
        throw new JspException(ioException);
    }
    return SKIP_BODY;
}
}
```



Cognizant  
Technology  
Solutions

# Demo 2 - Custom Tag with Attributes (Contd.)



Cognizant  
Academy

```
<!--JSPCustomTags\helloattribute\WEB-INF\hellouser1tags.tld - ->
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE taglib PUBLIC "-//Sun Microsystems, Inc.//DTD JSP Tag
  Library 1.2//EN" "http://java.sun.com/dtd/web-
  jsptaglibrary_1_2.dtd">
<taglib>
  <tlib-version>1.0</tlib-version>
  <jsp-version>1.2</jsp-version>
  <short-name>Hello User</short-name>
  <uri>/WEB-INF/hellouser1tags.tld</uri>
  <tag>
    <name>hellouser</name>
    <tag-class>attrpack.HelloUser</tag-class>
    <body-content>empty</body-content>
    <attribute>
      <name>username</name>
    </attribute>
  </tag>
</taglib>
```



Cognizant  
Technology  
Solutions

# Demo 2 - Custom Tag with Attributes (Contd.)



Cognizant  
Academy

```
<!--JSPCustomTags\helloattribute\WEB-INF\hellouser2tags.tld - ->
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE taglib
PUBLIC "-//Sun Microsystems, Inc.//DTD JSP Tag Library 1.2//EN"
"http://java.sun.com/dtd/web-jsptaglibrary_1_2.dtd">
<taglib>
  <tlib-version>1.0</tlib-version>
  <jsp-version>1.2</jsp-version>
  <short-name>Hello User</short-name>
  <uri>/WEB-INF/hellouser2tags.tld</uri>
  <tag>
    <name>hellouser</name>
    <tag-class>attrpack.HelloUser</tag-class>
    <body-content>empty</body-content>
    <attribute>
      <name>username</name>
      <rtexprvalue>true</rtexprvalue>
    </attribute>
  </tag>
</taglib>
```



Cognizant  
Technology  
Solutions

# Demo 2 - Custom Tag with Attributes (Contd.)



Cognizant  
Academy

```
JSPCustomTags\helloattribute\WEB-INF\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/j2ee/dtds/web-
  app_2_3.dtd">
<web-app>
  <display-name>Custom examples</display-name>
  <description>
    This is the collection of simple tag libraries
  </description>
  <taglib>
    <taglib-uri>/hellouser1tags.tld</taglib-uri>
    <taglib-location>/WEB-INF/hellouser1tags.tld</taglib-location>
  </taglib>
  <taglib>
    <taglib-uri>/hellouser2tags.tld</taglib-uri>
    <taglib-location>/WEB-INF/hellouser2tags.tld</taglib-location>
  </taglib>
</web-app>
```



Cognizant  
Technology  
Solutions

# Demo 2 - Custom Tag with Attributes (Contd.)



Cognizant  
Academy

```
JSPCustomTags\helloattribute\HelloUser1.jsp
```

```
<%@ page errorPage="error.jsp" %>
```

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>Hello User Example</TITLE>
```

```
</HEAD>
```

```
<BODY>
```

```
<%@ taglib uri="/hellouser1tags.tld" prefix="HelloAttrib" %>
```

```
<HelloAttrib:hellouser username="Homer"/>
```

```
</BODY>
```

```
</HTML>
```



Cognizant  
Technology  
Solutions

# Demo 2 - Custom Tag with Attributes (Contd.)



Cognizant  
Academy

```
JSPCustomTags\helloattribute\HelloUser2.jsp
```

```
<%@ page errorPage="error.jsp" %>
```

```
<HTML>
```

```
<HEAD>
```

```
    <TITLE>Hello User    Example</TITLE>
```

```
</HEAD>
```

```
<BODY>
```

```
    <%@ taglib uri="/hellouser2tags.tld" prefix="HelloAttrib" %>
```

```
    <% String user = "Marge"; %>
```

```
    <HelloAttrib:hellouser username="<%=user%>" />
```

```
</BODY>
```

```
</HTML>
```

## Demo 3 - Custom Tag - Iteration Tag

Demonstrates a JSP tag handler that makes an array values iterate using page attributes and an iterating tag.

```
JSPCustomTags\IterateArray\WEB-INF\classes\IterateArrayPack\CreatearrayClass.java
package IterateArrayPack;
import javax.servlet.jsp.*;
import javax.servlet.jsp.tagext.*;
import javax.servlet.jsp.*;
import javax.servlet.jsp.tagext.*;
public class CreatearrayClass extends TagSupport {
    private String arrayName = "";
    public String getName() {
        return this.arrayName;
    }
}
```





Cognizant  
Technology  
Solutions

## Demo 3 - Custom Tag - Iteration Tag (Contd.)



Cognizant  
Academy

```
public void setName(String newArrayName) {  
    this.arrayName = newArrayName;  
}  
  
public int doStartTag() throws JspException {  
    String[] myArray = new String[5];  
    myArray[0] = "Homer";  
    myArray[1] = "Marge";  
    myArray[2] = "Bart";  
    myArray[3] = "Lisa";  
    myArray[4] = "Maggie";  
    pageContext.setAttribute(this.getName(), myArray);  
    return SKIP_BODY;  
}  
}
```



Cognizant  
Technology  
Solutions

# Demo 3 - Custom Tag - Iteration Tag (Contd.)



Cognizant  
Academy

```
JSPCustomTags\IterateArray\WEB-INF\classes\IterateArrayPack\IterateArrayClass.java
package IterateArrayPack;
import java.lang.*;
import javax.servlet.jsp.*;
import javax.servlet.jsp.tagext.*;
public class IterateArrayClass extends TagSupport {
    private String[] array = null;
    private int arrayCounter = 0;
    private String arrayName;
    public String getName() {
        return this.arrayName;
    }
    public void setName(String newArrayName) {
        this.arrayName = newArrayName;
    }
    public int doStartTag() throws JspException {
        this.array = (String[])
        pageContext.getAttribute(this.getName());
        return EVAL_BODY_INCLUDE;
    }
}
```



Cognizant  
Technology  
Solutions

# Demo 3 - Custom Tag - Iteration Tag (Contd.)



Cognizant  
Academy

```
public int doAfterBody() throws JspException {
    try {
        JspWriter out = pageContext.getOut();
        out.print(" [" + this.arrayCounter + "]: ");
        out.print(this.array[this.arrayCounter] + "<BR>");
    } catch (Exception ioException) {
        System.err.println("Exception thrown in doAfterBody():");
        System.err.println(ioException.toString());
        throw new JspException(ioException);
    }
    int repeatOrSkip = SKIP_BODY;
    if (this.arrayCounter < (this.array.length - 1)) {
        repeatOrSkip = EVAL_BODY_AGAIN;
        this.arrayCounter++;
    }
    return repeatOrSkip;
}
```



# Demo 3 - Custom Tag - Iteration Tag (Contd.)



## **JSPCustomTags\IterateArray\WEB-INF\iteratearraytags.tld**

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE taglib
PUBLIC "-//Sun Microsystems, Inc.//DTD JSP Tag Library 1.2//EN"
"http://java.sun.com/dtd/web-jsptaglibrary_1_2.dtd">
<taglib>
  <tlib-version>1.0</tlib-version>
  <jsp-version>1.2</jsp-version>
  <short-name>IterateArray</short-name>
  <uri>/iteratearraytags.tld</uri>
  <tag>
    <name>createarray</name>
    <tag-class>IterateArrayPack.CreatearrayClass</tag-class>
  <attribute>
    <name>name</name>
  </attribute>
</tag>
```



Cognizant  
Technology  
Solutions

# Demo 3 - Custom Tag - Iteration Tag (Contd.)



Cognizant  
Academy

```
<tag>
  <name>iteratearray</name>
</tag-class>IterateArrayPack.IterateArrayClass</tag-class>
<attribute>
  <name>name</name>
</attribute>
</tag>
</taglib>
```



Cognizant  
Technology  
Solutions

# Demo 3 - Custom Tag - Iteration Tag (Contd.)



Cognizant  
Academy

## JSP CustomTags\IterateArray\WEB-INF\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/j2ee/dtds/web-app_2_3.dtd">
<web-app>
  <display-name>Custom examples</display-name>
  <description>
    This is the collection of simple tag libraries
  </description>
  <taglib>
    <taglib-uri>/iteratearraytags.tld</taglib-uri>
    <taglib-location>/WEB-INF/iteratearraytags.tld
    </taglib-location>
  </taglib>
</web-app>
```

## JSPCustomTags\IterateArray\IterateArray.jsp

```
<%@ page errorPage="error.jsp" %>
<HTML>
<HEAD>
  <TITLE>Iterator Example Using Page Attributes And An Iterating
  Tag</TITLE>
</HEAD>
<BODY>
  <H1>Array values iterated in a tag</H1>
  <%@ taglib uri="/iteratearraytags.tld" prefix="chap09" %>
  <chap09:createarray name="myArray"/>
  <chap09:iteratearray name="myArray">
    Array value
  </chap09:iteratearray>
</BODY>
</HTML>
```

- **Allow time for questions from participants**





# JSP Custom Tag: Summary

- Custom tags simplify the coding efforts
- The tags are interpreted by the J2EE Server
- Your tag should have a java class associated with it.
- This java class is called tag handler.
- The three interfaces that play a key role in custom tag development are:
  - Tag Interface
  - BodyTag Interface
  - IterationTag Interface
- Classes that implement tag interface can perform custom actions at their end tags. But these classes have no control over body content and cannot perform iterations
- The custom tags are used for Improved separation of presentation and implementation



Cognizant  
Technology  
Solutions

# JSP Custom Tag: Source



Cognizant  
Academy

- [http://java.sun.com/j2ee/tutorial/1\\_3-fcs/index.html](http://java.sun.com/j2ee/tutorial/1_3-fcs/index.html)
- Professional Java Server Programming J2EE Edition By Wrox Author Team

**Disclaimer:** Parts of the content of this course is based on the materials available from the Web sites and books listed above. The materials that can be accessed from linked sites are not maintained by Cognizant Academy and we are not responsible for the contents thereof. All trademarks, service marks, and trade names in this course are the marks of the respective owner(s).



Cognizant  
Technology  
Solutions



Cognizant  
Academy



You have  
successfully  
completed  
JSP Custom Tag.