Session 7 -Custom Tags

Objectives

- **Tag Libraries**
- **The Custom Tag Development Process**
- **Hunting the Tag**
- **Custom Tag Example**
- **JSTL (read at home)**
- **Tag Interfaces and Classes in javax.servlet.jsp.tagext**
- **The “Classic” Custom Tag Event Model**
- **Tags and Implicit Variables**

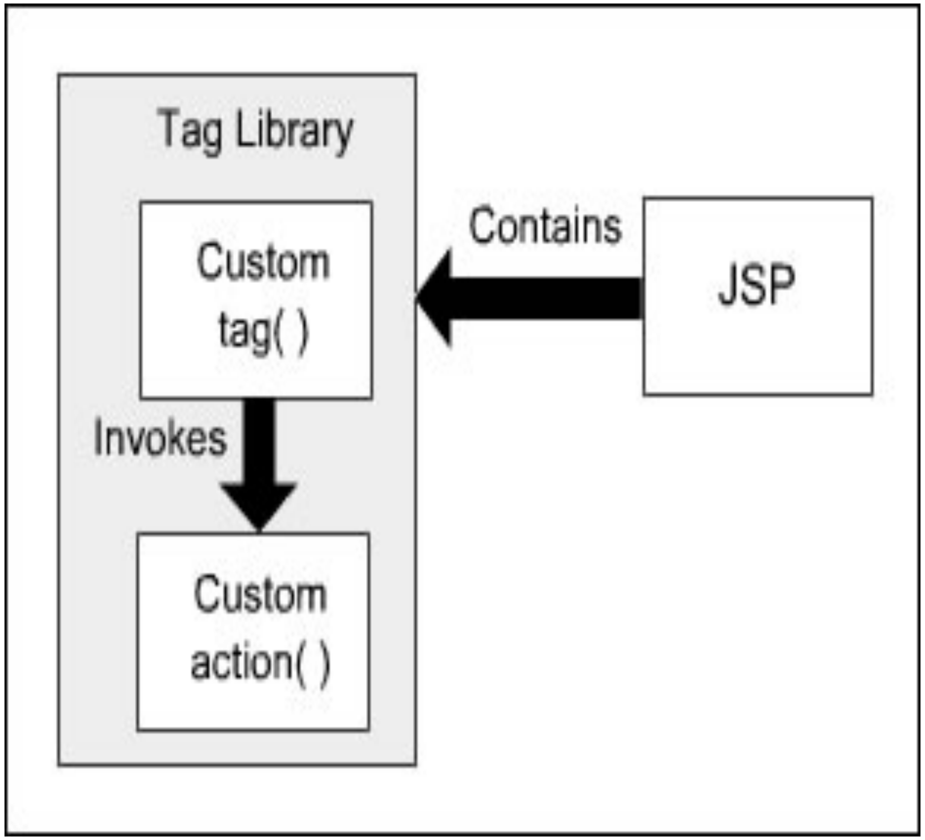
DREAM OF INNOVATION

Tag Extension Mechanism

- Enables creation of custom tags
- Java code can be avoided using custom tags
- The different terminologies related to tag extension mechanism are:
  - Tag library
  - TLD
  - Custom action
  - Custom

Tag extension mechanism..

- Tag library is a collection of custom actions and tags
- Custom tag encapsulates complex recurring code.
- Custom tag invokes a custom action



Custom Tags

- Separates the work profiles of Web designers and developers
- Custom tags can be reused
- Written using XML syntax
- The different types of custom tags are:
  - Empty tag
  - Tag with attributes
  - Body tags

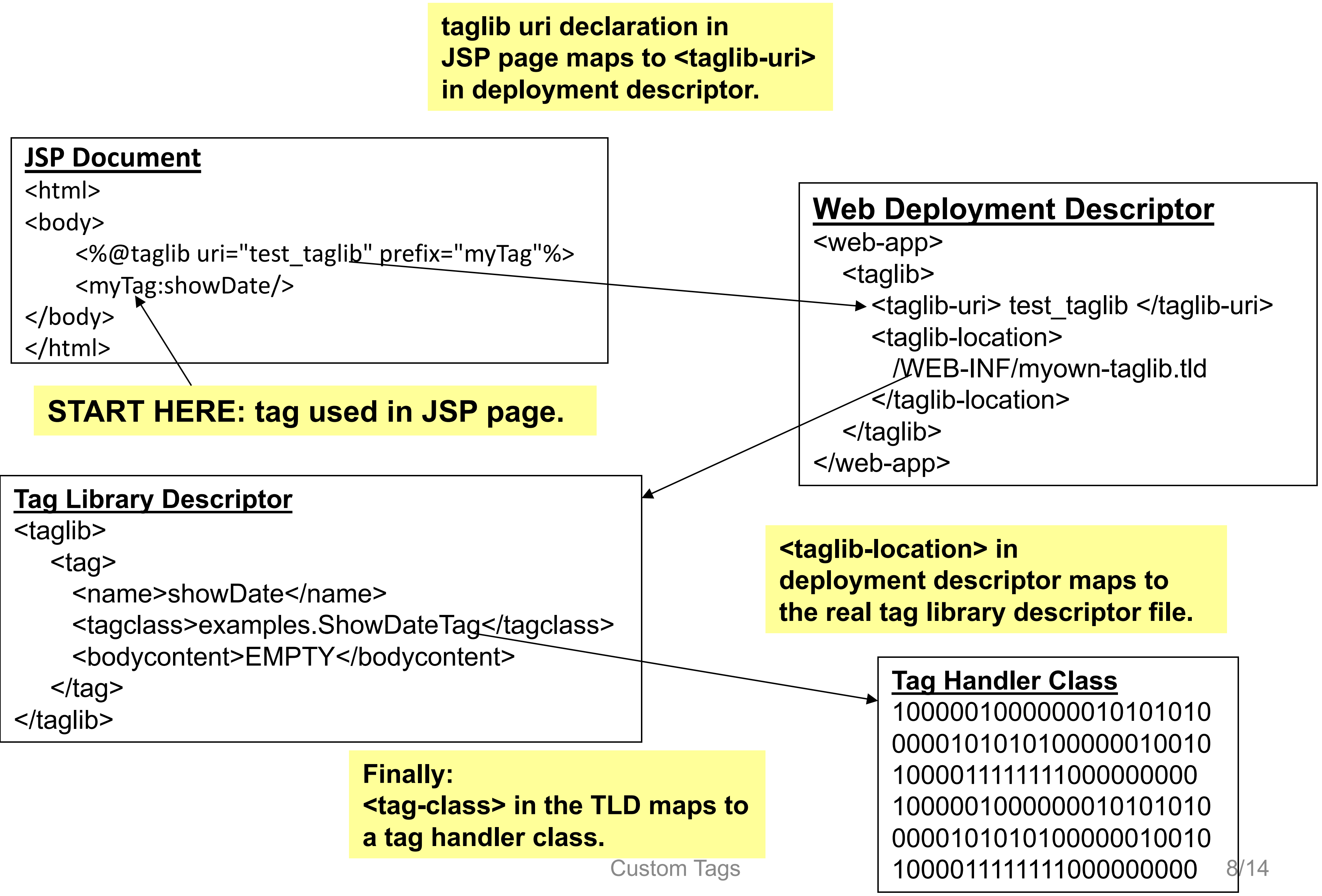
Custom Tags…

- The function of the tag can be customized using the attributes in the custom tags
- TLD file contains the details of the tag attributes
- Custom tag body can include static text, HTML, and JSP elements like scriptlets, between the start and the end tag.

The Custom Tag Development Process

- There are four essential steps to writing a custom tag for use in your JavaServer Pages:
  - Writing a Java class called a tag handler
  - Defining the tag within a tag library definition (TLD) file.
  - Providing details of where to find the TLD file in the deployment descriptor, web.xml
  - Referencing the TLD file in your JSP page source and using the tags from it.

Hunting the Tag



taglib uri declaration in JSP page maps to <taglib-uri> in deployment descriptor.

JSP Document

```
<html>
<body>
      <%@taglib uri="test_taglib" prefix="myTag"%>
      <myTag:showDate/>
</body>
</html>
```

START HERE: tag used in JSP page.

Web Deployment Descriptor

```
<web-app>
   <taglib>
      <taglib-uri> test_taglib </taglib-uri>
      <taglib-location>
        /WEB-INF/myown-taglib.tld
      </taglib-location>
   </taglib>
</web-app>
```

Tag Library Descriptor

```
<taglib>
    <tag>
      <name>showDate</name>
      <tagclass>examples.ShowDateTag</tagclass>
      <bodycontent>EMPTY</bodycontent>
    </tag>
</taglib>
```

<taglib-location> in deployment descriptor maps to the real tag library descriptor file.

Tag Handler Class

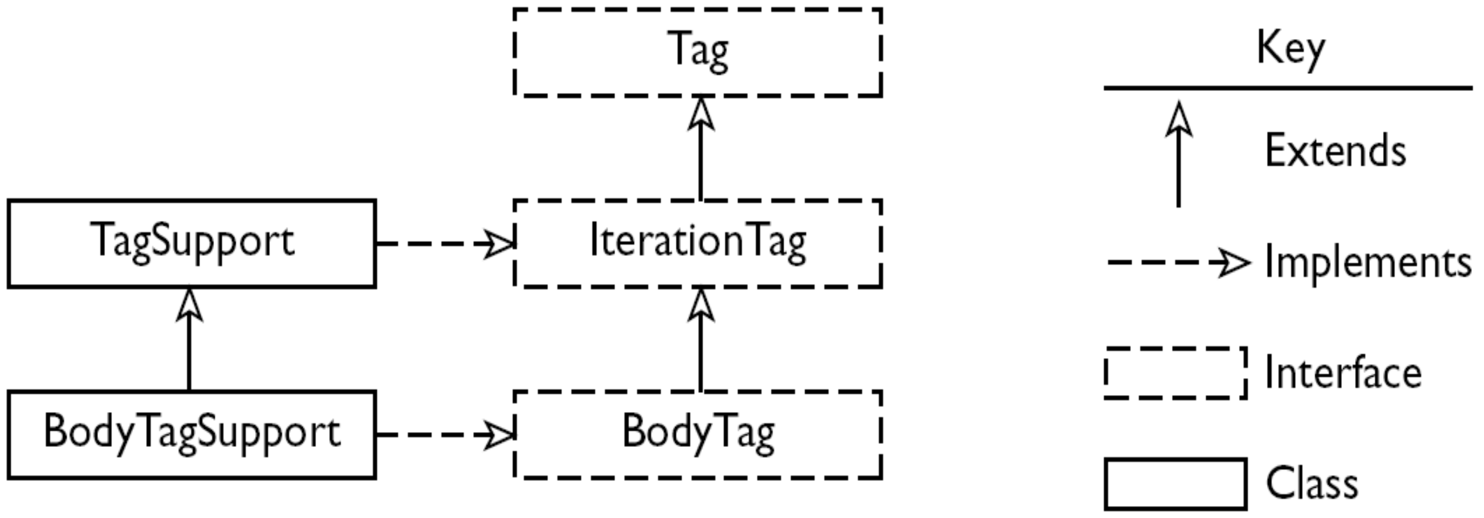
```
10000010000000010101010
00001010101000000010010
10000111111110000000000
10000010000000010101010
00001010101000000010010
10000111111110000000000
```

Finally:
<tag-class> in the TLD maps to a tag handler class.

Custom Tag Example

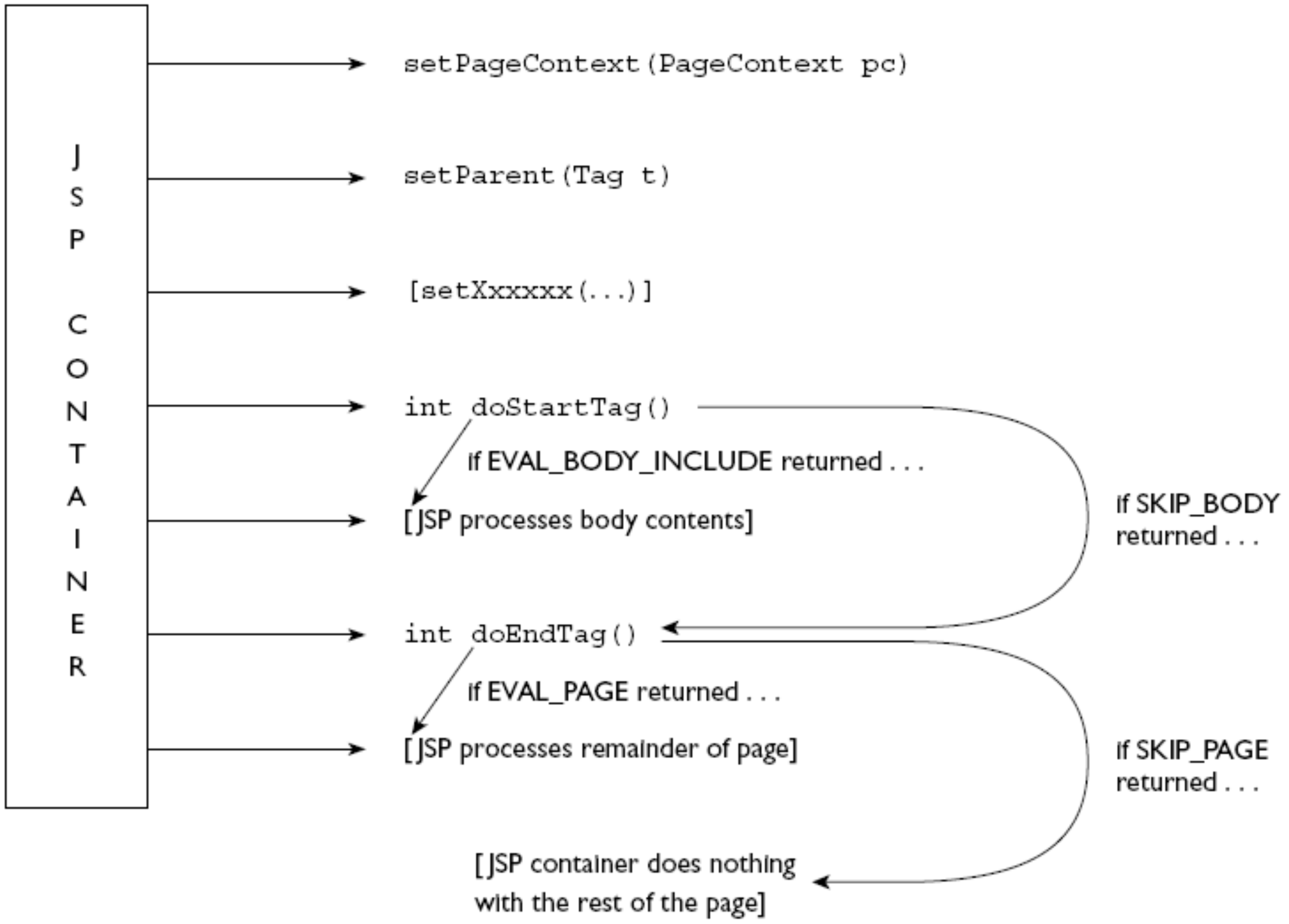
Demo\CusTags

Tag Interfaces and Classes in javax.servlet.jsp.tagext



Tag Interfaces and Classes in javax.servlet.jsp.tagext

The “Classic” Custom Tag Event Model


JSP CONTAINER
setPageContext(PageContext pc)
setParent(Tag t)
[setXxxxxx(...)]
int doStartTag()
If EVAL_BODY_INCLUDE returned . . .
[JSP processes body contents]
If SKIP_BODY returned . . .
int doEndTag()
If EVAL_PAGE returned . . .
[JSP processes remainder of page]
If SKIP_PAGE returned . . .
[JSP container does nothing with the rest of the page]

JSP container processing one occurrence of a tag implementing the Tag Interface in one JSP page.

Tags and Implicit Variables

- *Using the PageContext API we can write tag handler code to access the JSP implicit variables and access web application attributes.*

Tags and Implicit Variables…

JSP Implicit Variable	PageContext Method Used to Obtain Equivalent Object
Request	`getRequest()`
Response	`getResponse()`
Out	`getOut()` (inherited from PageContext's parent class JspWriter)
Session	`getSession()`
Config	`getServletConfig()`
Application	`getServletContext()`
Page	`getPage()`
PageContext	(This *is* the PageContext object passed to your tag handler)
Exception	`getException()`

Summary

- **Tag Libraries**
- **The Custom Tag Development Process**
- **Hunting the Tag**
- **Custom Tag Example**
- **Tag Interfaces and Classes in javax.servlet.jsp.tagext**
- **The “Classic” Custom Tag Event Model**
- **Tags and Implicit Variables**

Q&A