

## Creating Custom JSP Tag Libraries: The Basics

Core Servlets & JSP book: [www.coreservlets.com](http://www.coreservlets.com)  
 More Servlets & JSP book: [www.moreservlets.com](http://www.moreservlets.com)  
 Servlet and JSP Training Courses: [courses.coreservlets.com](http://courses.coreservlets.com)

Slides © Marty Hall, <http://www.moreservlets.com>, book © Sun Microsystems Press

## Agenda

- Components of a tag library
- Basic tags
- Tags that use attributes
- Tags that use body content
- Tags that optionally use body content

www.moreservlets.com

## Uses of JSP Constructs

Simple Application

↓

Complex Application

- Scripting elements calling servlet code directly
- Scripting elements calling servlet code indirectly (by means of utility classes)
- Beans
- Custom tags
- Servlet/JSP combo (MVC), with beans and possibly custom tags

www.moreservlets.com

## Components That Make Up a Tag Library

- **The Tag Handler Class**
  - Java code that says how to actually translate tag into code
  - Must implement javax.servlet.jsp.tagext.Tag interface
  - Usually extends TagSupport or BodyTagSupport
  - Goes in same directories as servlet class files and beans
- **The Tag Library Descriptor File**
  - XML file describing tag name, attributes, and implementing tag handler class
  - Goes with JSP file or at arbitrary URL
- **The JSP File**
  - Imports a tag library (referencing URL of descriptor file)
  - Defines tag prefix
  - Uses tags

www.moreservlets.com

## Defining a Simple Tag Handler Class

- **Extend the TagSupport class**
- **Import needed packages**
  - import javax.servlet.jsp.\*;
  - import javax.servlet.jsp.tagext.\*;
  - import java.io.\*;
- **Override doStartTag**
  - Obtain the JspWriter by means of pageContext.getOut()
  - Use the JspWriter to generate JSP content
  - Return SKIP\_BODY
  - Translated into servlet code at page-translation time
  - Code gets called at *request* time

www.moreservlets.com

## Defining a Simple Tag Handler Class: Example

```
package coreservlets.tags;
import javax.servlet.jsp.*;
import javax.servlet.jsp.tagext.*;
import java.io.*;
import java.math.*;
import coreservlets.*;

public class SimplePrimeTag extends TagSupport {
    protected int len = 50;

    public int doStartTag() {
        try {
            JspWriter out = pageContext.getOut();
            BigInteger prime =
                Primes.nextPrime(Primes.random(len));
            out.print(prime); // Primes class defined in Sect. 7.3
        } catch (IOException ioe) {
            System.out.println("Error generating prime: " + ioe);
        }
        return(SKIP_BODY);
    }
}
```

www.moreservlets.com

## Defining a Simple Tag Library Descriptor

- Start with XML header and DOCTYPE
- Top-level element is `taglib`
- Each tag defined by `tag` element with:
  - **name**, whose body defines the base tag name.  
In this case, I use `<name>simplePrime</name>`
  - **tagclass**, which gives the fully qualified class name of the tag handler. In this case, I use `<tagclass>coreservlets.tags.SimplePrimeTag</tagclass>`
  - **bodycontent**, which gives hints to development environments. Optional.
  - **info**, which gives a short description. Here, I use `<info>Outputs a random 50-digit prime.</info>`

www.moreservlets.com

## TLD File for SimplePrimeTag

```

<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE taglib PUBLIC
"-//Sun Microsystems, Inc.//DTD JSP Tag Library 1.1//EN"
"http://java.sun.com/j2ee/dtds/web-jsptaglibrary_1_1.dtd">
<taglib>
... // A few standard items -- just copy these
<tag>
  <name>simplePrime</name>
  <tagclass>coreservlets.tags.SimplePrimeTag</tagclass>
  <info>Outputs a random 50-digit prime.</info>
</tag>
</taglib>
    
```

- Don't memorize XML header and DOCTYPE; modify version from `moreservlets.com`

www.moreservlets.com

## Accessing Custom Tags From JSP Files

- Import the tag library
  - Specify location of TLD file  
`<%@ taglib uri="csajsp-taglib.tld" prefix="csajsp" %>`
    - Eventually, you should move the TLD file to the WEB-INF directory and use `uri="/WEB-INF/whatever.tld"`.
  - Define a tag prefix (namespace)  
`<%@ taglib uri="csajsp-taglib.tld" prefix="csajsp" %>`
- Use the tags
  - `<prefix:tagName />`
    - Tag name comes from TLD file
    - Prefix comes from taglib directive
  - E.g., `<csajsp:simplePrime />`

www.moreservlets.com

## Using simplePrime Tag

```

...
<BODY>
<H1>Some 50-Digit Primes</H1>

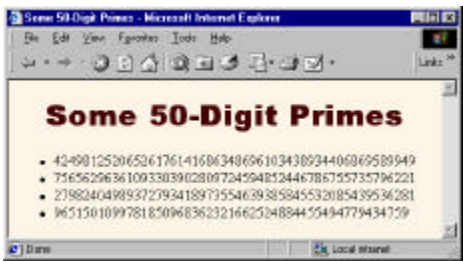
<%@ taglib uri="csajsp-taglib.tld"
  prefix="csajsp" %>

<UL>
  <LI><csajsp:simplePrime />
  <LI><csajsp:simplePrime />
  <LI><csajsp:simplePrime />
  <LI><csajsp:simplePrime />
</UL>

</BODY>
</HTML>
    
```

www.moreservlets.com

## Using simplePrime Tag: Result



www.moreservlets.com

## Assigning Attributes to Tags

- Allowing tags like
  - `<prefix:name attribute1="value1" attribute2="value2" ... attributeN="valueN" />`
- Tags are still standalone
  - No body between start and end tags

www.moreservlets.com

## Attributes: The Tag Handler Class

- Use of an attribute called `attribute1` simply results in a call to a method called `setAttribute1`
  - Attribute value is supplied to method as a String
- Example
  - To support

```
<prefix:tagName attribute1="Test" />
```

add the following to tag handler class:

```
public void setAttribute1(String value1) {
    doSomethingWith(value1);
}
```

12

Basic Custom Tags

www.moreservlets.com

## Attributes: PrimeTag.java

```
package coreservlets.tags;

import javax.servlet.jsp.*;
import javax.servlet.jsp.tagext.*;
import java.io.*;
import java.math.*;
import coreservlets.*;

public class PrimeTag extends SimplePrimeTag {
    public void setLength(String length) {
        try {
            // len used by parent class
            len = Integer.parseInt(length);
        } catch (NumberFormatException nfe) {
            len = 50;
        }
    }
}
```

14

Basic Custom Tags

www.moreservlets.com

## PrimeTag (Continued)

```
public void release() {
    len = 50;
}
}
```

- Why `release`?
  - Servers are permitted to reuse tag instances
    - No synchronization issues, since reuse occurs only after tag processing is totally finished.
  - Few current servers reuse tags, but you should plan ahead
  - Not needed if attribute is required. (Why not?)

15

Basic Custom Tags

www.moreservlets.com

## Attributes: The Tag Library Descriptor File

- The tag element must contain a nested attribute element
- The attribute element has three further-nested elements
  - `name`, a required element that defines the case-sensitive attribute name. In this case, I use `<name>length</name>`
  - `required`, a required element that stipulates whether the attribute must always be supplied (true) or is optional (false). Here, to indicate that length is optional, I use `<required>false</required>`
  - `rtexprvalue`, an optional attribute that indicates whether the attribute value can be a JSP expression like `<%= expression %>` (true) or whether it must be a fixed string (false). The default value is false.

16

Basic Custom Tags

www.moreservlets.com

## TLD File for PrimeTag

```
...
<taglib>
  <tag>
    <name>prime</name>
    <tagclass>coreservlets.tags.PrimeTag</tagclass>
    <info>Outputs a random N-digit prime.</info>
    <attribute>
      <name>length</name>
      <required>false</required>
    </attribute>
  </tag>
</taglib>
```

17

Basic Custom Tags

www.moreservlets.com

## Using prime Tag

```
...
<BODY>
  <H1>Some N-Digit Primes</H1>

  <%@ taglib uri="csajsp-taglib.tld"
    prefix="csajsp" %>

  <UL>
    <LI>20-digit: <csajsp:prime length="20" />
    <LI>40-digit: <csajsp:prime length="40" />
    <LI>80-digit: <csajsp:prime length="80" />
    <LI>Default (50-digit): <csajsp:prime />
  </UL>

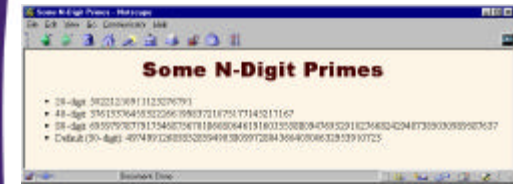
</BODY>
</HTML>
```

18

Basic Custom Tags

www.moreservlets.com

## Using prime Tag: Result



16

Basic Custom Tags

www.moreservlets.com

## Including the Tag Body

- **Simplest tags**
  - `<prefix:tagName />`
- **Tags with attributes**
  - `<prefix:tagName att1="val1" ... />`
- **Now**
  - `<prefix:tagName>`  
**JSP Content**  
`</prefix:tagName>`
  - `<prefix:tagName att1="val1" ... >`  
**JSP Content**  
`</prefix:tagName>`

20

Basic Custom Tags

www.moreservlets.com

## Using Tag Body: The Tag Handler Class

- **doStartTag**
  - Usually returns `EVAL_BODY_INCLUDE` instead of `SKIP_BODY`
- **doEndTag**
  - Define this method if you want to take action after handling the body
  - Return `EVAL_PAGE`

21

Basic Custom Tags

www.moreservlets.com

## Using Tag Body: HeadingTag.java

```
package coreservlets.tags;
import javax.servlet.jsp.*;
import javax.servlet.jsp.tagext.*;
import java.io.*;

public class HeadingTag extends TagSupport {
    private String bgColor; // The one required attribute
    private String color = null;
    ...

    public void setBgColor(String bgColor) {
        this.bgColor = bgColor;
    }

    public void setColor(String color) {
        this.color = color;
    }
    ...
}
```

22

Basic Custom Tags

www.moreservlets.com

## Using Tag Body: HeadingTag.java (Continued)

```
public int doStartTag() {
    try {
        JspWriter out = pageContext.getOut();
        out.print("<TABLE BORDER=" + border +
            " BGCOLOR=\"" + bgColor + "\"" +
            " ALIGN=\"" + align + "\"");
        if (width != null) {
            out.print(" WIDTH=\"" + width + "\"");
        }
        ...
    } catch (IOException ioe) {
        System.out.println("Error in HeadingTag: " + ioe);
    }
    return(EVAL_BODY_INCLUDE); // Include tag body
}
```

23

Basic Custom Tags

www.moreservlets.com

## Using Tag Body: HeadingTag.java (Continued)

```
public int doEndTag() {
    try {
        JspWriter out = pageContext.getOut();
        out.print("</SPAN></TABLE>");
    } catch (IOException ioe) {
        System.out.println("Error in HeadingTag: " + ioe);
    }
    return(EVAL_PAGE); // Continue with rest of JSP page
}
```

24

Basic Custom Tags

www.moreservlets.com

### Using Tag Body: The Tag Library Descriptor File

- **Only difference is bodycontent element**
  - Should be JSP instead of empty:
 

```
<tag>
  <name>...</name>
  <tagclass>...</tagclass>
  <bodycontent>JSP</bodycontent>
  <info>...</info>
</tag>
```
- **Purpose is primarily for development environments**
  - Advanced IDEs may have drag-and-drop support for custom tags
  - Defined as optional in JSP specification
- **I will omit in my examples**

www.moreservlets.com

### TLD File for HeadingTag

```
...
<taglib>
  <tag>
    <name>heading</name>
    <tagclass>coreservlets.tags.HeadingTag</tagclass>
    <info>Outputs a 1-cell table used as a heading.</info>
    <attribute>
      <name>bgColor</name>
      <required>true</required> <!-- bgColor required -->
    </attribute>
    <attribute>
      <name>color</name>
      <required>false</required>
    </attribute>
    ...
  </tag>
</taglib>
```

www.moreservlets.com

### Using heading Tag

```
<%@ taglib uri="csajsp-taglib.tld" prefix="csajsp" %>
<csajsp:heading bgColor="#C0C0C0">
Default Heading
</csajsp:heading>
<P>
<csajsp:heading bgColor="BLACK" color="WHITE">
White on Black Heading
</csajsp:heading>
<P>
<csajsp:heading bgColor="#EF8429" fontSize="60" border="5">
Large Bordered Heading
</csajsp:heading>
<P>
<csajsp:heading bgColor="CYAN" width="100%">
Heading with Full-Width Background
</csajsp:heading>
...

```

www.moreservlets.com

### Using heading Tag: Result

www.moreservlets.com

### Optional Tag Bodies

- **First examples had no tag bodies**
  - doStartTag returned SKIP\_BODY
- **Most recent examples always included tag bodies**
  - doStartTag returned EVAL\_BODY\_INCLUDE
- **Now: decide whether or not to include tag body at request time**
  - Have doStartTag return either SKIP\_BODY or EVAL\_BODY\_INCLUDE depending on values of request time information

www.moreservlets.com

### Optional Tag Bodies: DebugTag.java

```
package coreservlets.tags;
import javax.servlet.jsp.*;
import javax.servlet.jsp.tagext.*;
import java.io.*;
import javax.servlet.*;

public class DebugTag extends TagSupport {
  public int doStartTag() {
    ServletRequest request =
      pageContext.getRequest();
    String debugFlag = request.getParameter("debug");
    if ((debugFlag != null) &&
        (!debugFlag.equalsIgnoreCase("false"))) {
      return(EVAL_BODY_INCLUDE);
    } else {
      return(SKIP_BODY);
    }
  }
}
```

www.moreservlets.com

### TLD File for DebugTag

```

...
<taglib>
  <tag>
    <name>debug</name>
    <tagclass>coreservlets.tags.DebugTag</tagclass>
    <info>
      Includes body only if debug param is set.
    </info>
  </tag>
</taglib>
    
```

31 Basic Custom Tags www.moreservlets.com

### Using debug Tag

```

<%@ taglib uri="csajsp-taglib.tld" prefix="csajsp" %>
Top of regular page. Blah, blah, blah.
Yadda, yadda, yadda.
<P>
<csajsp:debug>
<B>Debug:</B>
<UL>
  <LI>Current time: <%= new java.util.Date() %>
  <LI>Requesting hostname: <%= request.getRemoteHost() %>
  <LI>Session ID: <%= session.getId() %>
</UL>
</csajsp:debug>
<P>
Bottom of regular page. Blah, blah, blah.
Yadda, yadda, yadda.
    
```

32 Basic Custom Tags www.moreservlets.com

### Using debug Tag: Results

33 Basic Custom Tags www.moreservlets.com

### Open Source Tag Libraries

<http://jakarta.apache.org/taglibs/>

- JSP Standard Tag Library (JSTL)
  - Final specification released mid 2002.
    - Covered in later lecture
- Internationalization (I18N)
- Database access
- Sending email
- JNDI
- Date/time
- Populating/validating form fields
- Perl regular expressions
- Extracting data from other Web pages
- XSL transformations
- Etc.

34 Basic Custom Tags www.moreservlets.com

### Summary

- **For each custom tag, you need**
  - A tag handler class (usually extending TagSupport or BodyTagSupport)
  - An entry in a Tag Library Descriptor file
  - A JSP file that imports it, specifies prefix, and uses it
- **Simple tags**
  - Generate output in doStartTag, return SKIP\_BODY
- **Attributes**
  - Define setAttributeName method. Update TLD file.
  - Use release if attributes are optional.
- **Body content**
  - doStartTag returns EVAL\_BODY\_INCLUDE
  - Add doEndTag method

35 Basic Custom Tags www.moreservlets.com


### Questions?

Core Servlets & JSP book: [www.coreservlets.com](http://www.coreservlets.com)  
 More Servlets & JSP book: [www.moreservlets.com](http://www.moreservlets.com)  
 Servlet and JSP Training Courses: [courses.coreservlets.com](http://courses.coreservlets.com)

Slides © Marty Hall, <http://www.moreservlets.com>, book © Sun Microsystems Press

## More Information

- **Source code for examples**
  - <http://www.moreservlets.com>
- **More Servlets & JSP**
  - <http://www.moreservlets.com>
  - Site includes info on servlet and JSP training courses
- **Core Servlets & JSP**
  - Prequel to *More Servlets & JSP*
  - <http://www.coreservlets.com>
- **Servlet home page**
  - <http://java.sun.com/products/servlet/>
- **JavaServer Pages home page**
  - <http://java.sun.com/products/jsp/>



www.moreservlets.com