

JSTL CORE <FMT:BUNDLE> TAG

http://www.tutorialspoint.com/jsp/jstl_format_bundle_tag.htm

Copyright © tutorialspoint.com

The <fmt:bundle> tag will make the specified bundle available to all <fmt:message> tags that occur between the bounding <fmt:bundle> and </fmt:bundle> tags. This saves you the extra step of having to specify the resource bundle for each of your <fmt:message> tags.

For example, the following two <fmt:bundle> blocks would produce the same output:

```
<fmt:bundle basename="com.tutorialspoint.Example">
  <fmt:message key="count.one"/>
</fmt:bundle>

<fmt:bundle basename="com.tutorialspoint.Example" prefix="count.">
  <fmt:message key="title"/>
</fmt:bundle>
```

Attribute:

The <fmt:bundle> tag has following attributes:

Attribute	Description	Required	Default
basename	Specifies the base name of the resource bundle that is to be loaded.	Yes	None
prefix	Value to prepend to each key name in <fmt:message> subtags	No	None

Example:

Resource bundles contain locale-specific objects. Resource bundles contain key/value pairs. When your program needs a locale-specific resource, you keep all the keys common to all the locale but you can have translated values specific to locale. Resource bundles helps in providing content specific to locale.

A Java resource bundle file contains a series of key-to-string mappings. The method that we focus on involves creating compiled Java classes that extend the java.util.ListResourceBundle class. You must compile these class files and make them available to the classpath of your Web application.

Let us define a default resource bundle as follows:

```
package com.tutorialspoint;

import java.util.ListResourceBundle;

public class Example_En extends ListResourceBundle {
    public Object[][] getContents() {
        return contents;
    }
    static final Object[][] contents = {
        {"count.one", "One"},
        {"count.two", "Two"},
        {"count.three", "Three"},
    };
}
```

Let us compile above class Example.class and make it available in the CLASSPATH of your Web application. Now you can use the following JSTL tags to display the three numbers as follows:

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<%@ taglib uri="http://java.sun.com/jsp/jstl/fmt" prefix="fmt" %>
<html>
<head>
<title>JSTL fmt:bundle Tag</title>
</head>
<body>

<fmt:bundle basename="com.tutorialspoint.Example" prefix="count.">
    <fmt:message key="one"/><br/>
    <fmt:message key="two"/><br/>
    <fmt:message key="three"/><br/>
</fmt:bundle>

</body>
</html>
```

This would produce following result:

```
One
Two
Three
```

Try above example without prefix as follows:

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<%@ taglib uri="http://java.sun.com/jsp/jstl/fmt" prefix="fmt" %>
<html>
<head>
<title>JSTL fmt:bundle Tag</title>
</head>
<body>

<fmt:bundle basename="com.tutorialspoint.Example">
    <fmt:message key="count.one"/><br/>
    <fmt:message key="count.two"/><br/>
    <fmt:message key="count.three"/><br/>
</fmt:bundle>

</body>
</html>
```

This would also produce following result:

```
One
Two
Three
```

Check [<fmt:setLocale>](#) and [<setBundle>](#) tags to understand complete concept.