

# J-CASE



JSP Custom Taglib for Writing Storyboards

## **J-CASE Tag Library**

### **Tag Library Reference**

**Version 0.2.1**

Revised: December 2009

**Contents**

---

1	Introduction .....	4
1.1	Declaration .....	4
1.2	EL (Expression Language) .....	4
1.3	Default Values .....	4
2	Common Tags .....	5
2.1	<c:head> .....	5
2.2	<c:init> .....	5
3	Use Case Tags .....	6
3.1	<c:useCase> .....	6
3.2	<c:description> <c:desc> .....	8
3.3	<c:insertFlows> .....	9
3.4	<c:newTable> .....	9
3.5	<c:chapter> .....	10
3.6	<c:section> .....	11
3.7	<c:pageBreak> .....	11
4	Action Tags .....	12
4.1	<c:actions> .....	12
4.2	<c:action> .....	12
5	Screen Tags .....	14
5.1	<c:screen> .....	14
5.2	<c:dialog> .....	15
5.3	<c:image> .....	16
6	Conditional Tags .....	17
6.1	<c:if> .....	17
6.2	<c:choose>, <c:when>, <c:otherwise> .....	17
7	Tool Tags .....	18
7.1	<c:generateUseCase> .....	18
8	Other Tags .....	20
8.1	<c:title> .....	20
8.2	<c:processResults> .....	21
8.3	<c:urlParameters> .....	21
8.4	<c:feedback> .....	22
8.5	<c:instruction> <c:inst> .....	23
8.6	<c:link> .....	23
8.7	<c:set> .....	24
8.8	<c:remove> .....	25
8.9	<c:catch> .....	25
8.10	<c:include> .....	26
8.11	<c:terminate> .....	26
8.12	<c:tag> .....	26
9	Functions .....	28
9.1	<fn:queryString> .....	28
9.2	<fn:includesQueryString> .....	28
9.3	<fn:excludesQueryString> .....	28
9.4	<fn:directoryName> .....	29
9.5	<fn:replaceAll> .....	29
9.6	<fn:substringAfterLast> .....	29
9.7	<fn:substringBeforeLast> .....	30
9.8	<fn:ifNull> .....	30
9.9	<fn:ifEmpty> .....	30
9.10	<fn:ifTest> .....	30

9.11	<fn:urlEncode>.....	31
9.12	<fn:urlDecode>.....	31
10	Aspect Weaving.....	32
10.1	<c:parent>.....	33
10.2	<c:replace>.....	33

## 1 Introduction

---

This document describes specifications of J-CASE tag library to describe Storyboards on JSP.

### 1.1 Declaration

In order to use J-CASE taglib on JSP, the following prefix and uri must be declared using taglib directive:

```
<%@ taglib prefix="c" uri="http://www5f.biglobe.ne.jp/~webtest/jcase/core" %>
```

In order to use J-CASE functions on JSP, the following prefix and uri must be declared using taglib directive:

```
<%@ taglib prefix="fn" uri="http://www5f.biglobe.ne.jp/~webtest/jcase/functions" %>
```

### 1.2 EL (Expression Language)

EL of JSP2.0 can be described in attributes of J-CASE tags.

**Example 1)** If result variable equals 'error', display the instruction text.

```
<c:instruction test="${result == 'error'}">  
    Failed to register. Please try again.  
</c:instruction>
```

**Example 2)** Add a value of type URL parameter into link.

```
<c:actions>  
    <c:action name="Next Page" link="next.jsp?type=${param.type}">  
        The user selects "Next" to process the next task.  
    </c:action>  
</c:actions>
```

### 1.3 Default Values

The default value of the most attributes in J-CASE tags can be changed using jcase.xml file. Once you describe your own default values in the XML, the specified values are effective in the web site. For further information, please see "jcase.xml" in User's Guide.

## 2 Common Tags

---

### 2.1 <c:head>

The <c:head> tag initialize variables and load resources such as JavaScript, CSS. This tag must be put in <HEAD> HTML tag of each JSP.

#### Tag attributes

The <c:head> tag's attributes are as follows:

Attribute	Description	Required	Default
id	Identifier of this tag	No	empty

#### Example)

```

<%@ taglib prefix="c" uri="http://www5f.biglobe.ne.jp/~webtest/jcase/core" %>
<html>
  <head>
    <c:head />
    <title>Page Title</title>
  </head>
<body>
...
...

```

Not available to access variables and call other J-CASE tags

### 2.2 <c:init>

The <c:init> tag initializes variables for J-CASE. This is internally called in <c:head> tag, so, normally you don't have to call this. However, the <c:head> must be located in <HEAD> HTML tag, so you can't access variables or call other J-CASE tags before the <HEAD>. In this case, <c:init> tag can be used to only initialize variables, not load resources.

#### Tag attributes

The <c:init> tag's attributes are as follows:

Attribute	Description	Required	Default
id	Identifier of this tag	No	empty

#### Example)

```

<%@ taglib prefix="c" uri="http://www5f.biglobe.ne.jp/~webtest/jcase/core" %>
<c:init />
<html>
  <head>
    <c:head />
    <title>Page Title</title>
  </head>
<body>
...
...

```

Available to access variables and call other J-CASE tags

<c:head> must be put to load resource files.

### 3 Use Case Tags

#### 3.1 <c:useCase>

The <c:useCase> tag generates a Use Case frame that contains section descriptions, such as Actor, Preconditions, except flows of events.

##### Tag attributes

The <c:useCase> tag's attributes are as follows:

Attribute	Description	Required	Default
test	Generates the Use Case frame if the test result is true.	No	true
label	Frame label	No	Use Case
id	Identifier of the Use Case	Yes	--
name	Use Case name	No	Note1
useCaseId	Target Use Case ID linked from this use case	Yes	--
useCaseName	Use Case name of the target use case	No	Note1
link	Link URL to the top page of flow pages.	Yes	--
show	Show table contents when the page is displayed. <b>true:</b> Show <b>false:</b> Do not show	No	true
width	Frame width (pixel or percent)	No	100%
pageBreak	Make a page break before the new use case table in use case document.	No	true

Note 1: Value of useCase tag in jcase.xml is used as default. If the value is not specified there, name/useCaseName attribute is required.

The <c:useCase> tag also accepts the following attributes. The information specified in the attribute appears in Use Case Diagram:

Attribute	Description	Required	Default
actor	Actor name or ID (Note 2)	No	--
includes	Use Case IDs (comma separated) that this use case includes	No	--
extends	Use Case IDs (comma separated) that this use case extends	No	--
generalizes	Use Case IDs (comma separated) that this use case is specialized from	No	--

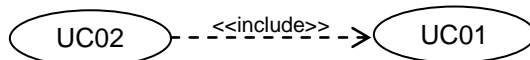
Note 2: Actor ID can be defined with actor tag in jcase.xml.

##### Tips:

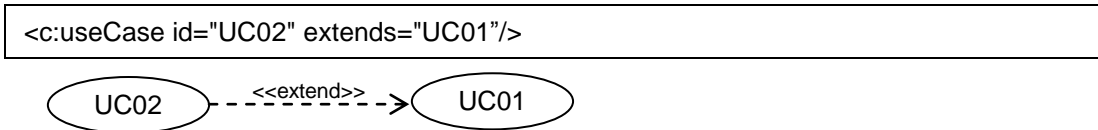
It is easy to confuse extends relationships between use cases with generalization. Follow the instructions for "include", "extend" and "generalization" use.

Use "includes" if a use case uses another use case. The <c:useCase> tag below represents that the UC02 uses (includes) the UC01 use case. (The UC01 is a part of the UC02)

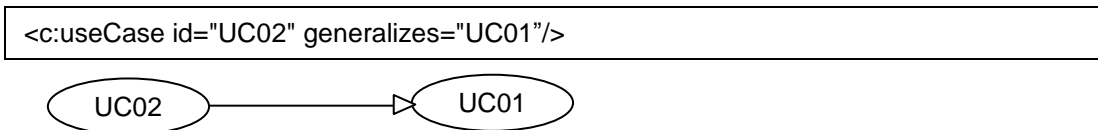
```
<c:useCase id="UC02" includes="UC01"/>
```



Use “extends” if there is an alternative use case. The <c:useCase> tag below represents that the UC02 is the alternative use case of the UC01 use case. (If UC01 may not work or is not available, the UC02 is executed instead.)



Use “generalizes” if the is specialized based on a generic (parent) use case. The <c:useCase> tag below represents that the UC02 is a specialized use case inherited from the UC01.



**Tag body**

The <c:useCase> tag’s body accepts the following tags:

- <c:description> (<c:desc>)
- <c:insertFlows>
- <c:newTable>

**Example 1) Make a Use Case**

Make a use case “UC01”.

```
<c:useCase id="UC01" link="uc01/reg.jsp">
  <c:desc name="Brief Description">
    The user registers on the site
  </c:desc>
  <c:desc name="Actor">Consumer user</c:desc>
</c:useCase>
```

When the JSP is executed, the following Use Case frame is generated:

<b>Use Case: <a href="#">[UC01] Register User</a></b>	
<b>Brief Description</b>	The user registers on the site
<b>Actor</b>	Consumer user

**Example 2) Link to another Use Case**

Make a link to UC01 from UC02 at the top level. The use case “UC02” doesn’t have any JSPs.

```
<c:useCase id="UC02" useCaseId="UC02" link="uc01/reg.jsp">
  <c:desc name="Brief Description">
    The user registers by calling to support center
  </c:desc>
  <c:desc name="Actor">Consumer user</c:desc>
</c:useCase>
```

When the JSP is executed, the following Use Case frame is generated:

<b>Use Case: [UC02] Register User by Phone (Use Case UC01)</b>	
<b>Brief Description</b>	The user registers by calling t support center
<b>Actor</b>	Consumer user

### 3.2 <c:description> <c:desc>

The <c:description> tag inserts a row in the Use Case frame. So, this tag must be described in <c:useCase> tag.

The <c:desc> can be also used for short.

#### Tag attributes

The <c:description> tag's attributes are as follows:

Attribute	Description	Required	Default
test	Generates this content if the test result is true.	No	true
id	Identifier of this tag	No	empty
name	Row name	Yes	--

#### Tag body

The <c:description> tag's body accepts description text shown in the row.

#### Example 1) Actor Row

Add actor row in the Use Case.

```
<c:useCase id="UC01" name="Register User" link="uc01/reg.jsp">
  <c:description name="Actor">
    Consumer user
  </c:description>
</c:useCase>
```

The above JSP can be also described with <c:desc> tag for short:

```
<c:useCase id="UC01" name="Register User" link="uc01/reg.jsp">
  <c:desc name="Actor">
    Consumer user
  </c:desc>
</c:useCase>
```

When the JSP is executed, the following row is added in the Use Case:

<b>Actor</b>	Consumer user
--------------	---------------

#### Example 2) Title Row

Add a title row.

```
<c:useCase id="UC01" name="Register User" link="uc01/reg.jsp">
  <c:desc name="ACTORS" />
  <c:desc name="Primary Actor">Consumer user</c:desc>
  <c:desc name="Secondary Actor">Support user</c:desc>
</c:useCase>
```

When the JSP is executed, the following row is added in the Use Case:

ACTORS	
<b>Primary Actor</b>	Consumer user
<b>Secondary Actor</b>	Support user

### 3.3 <c:insertFlows>

The <c:insertFlows> tag marks the line where flows of events for the Use Case is inserted when executing UseCaseGenerator command or <c:generateUseCase> tag. This tag must be described in <c:useCase> tag.

#### Tag attributes

The <c:insertFlows> tag's attributes are as follows:

Attribute	Description	Required	Default
test	Generates this content if the test result is true.	No	true
id	Identifier of this tag	No	empty

#### Example)

Mark to insert flows after Preconditions row:

```
<c:useCase id="UC01" name="Register User" link="uc01/reg.jsp">
  <c:desc name="Actor">Consumer user</c:desc>
  <c:desc name="Preconditions">The user is not registered.</c:desc>
  <c:insertFlows />
  <c:desc name="Postconditions">The user is registered.</c:desc>
</c:useCase>
```

When the JSP is executed, flows are not inserted yet:

<b>Actor</b>	Consumer user
<b>Preconditions</b>	The user is not registered.
<b>Postconditions</b>	The user is registered.

When you execute UseCaseGenerator command or <c:generateUseCase> tag, the flows are inserted between Preconditions and Postconditions in the Use Case document generated.

### 3.4 <c:newTable>

The <c:newTable> tag generates a new table in the Use Case. This tag must be described in <c:useCase> tag.

#### Tag attributes

The <c:newTable> tag's attributes are as follows:

Attribute	Description	Required	Default
test	Generates this content if the test result is true.	No	true
id	Identifier of this tag	No	empty
pageBreak	Make a page break before the new table in use case document.	No	false

#### Example)

Display Pre and Postconditions in a new table:

```

<c:useCase id="UC01" name="Register User" link="uc01/reg.jsp">
  <c:desc name="Actor">Consumer user</c:desc>
  <c:newTable />
  <c:desc name="Preconditions">The user is not registered.</c:desc>
  <c:desc name="Postconditions">The user is registered.</c:desc>
</c:useCase>
```

When the JSP is executed, the following rows are displayed in the Use Case:

<b>Actor</b>	Consumer user
<b>Preconditions</b>	The user is not registered.
<b>Postconditions</b>	The user is registered.

### 3.5 <c:chapter>

The <c:chapter> tag generates a Chapter frame that is also inserted in use case document to be generated by <c:generateUseCase>.

#### Tag attributes

The <c:chapter> tag's attributes are as follows:

Attribute	Description	Required	Default
test	Generates the Chapter frame if the test result is true.	No	true
id	Identifier of this tag	No	empty
label	Chapter label to be displayed on the frame bar.	No	Chapter
name	Chapter title name to be displayed next to the label.	Yes	--
show	Show table contents when the page is displayed. <b>true:</b> Show <b>false:</b> Do not show	No	true
width	Frame width (pixel or percent)	No	empty
border	The thickness of the frame	No	1
align	Horizontal alignment of the table contents <b>left:</b> Align left <b>center:</b> Centering <b>right:</b> Align right	No	left
valign	Vertical alignment of the table contents <b>top:</b> Align top <b>center:</b> Centering <b>bottom:</b> Align bottom	No	top
bgcolor	Background color of the contents area	No	empty
cellpadding	Space between frame and content area. (pixel)	No	5
pageBreak	Make a page break before the new chapter in use case document.	No	true

#### Tag body

The <c:screen> tag's body accepts the following tags:

- <c:instruction> (<c:inst>)
- HTML text

#### Example) Instruction chapter

Add Instruction chapter:

```

<c:chapter name="Introduction">
  This document describes storyboards for Sample Web site.<br>
  <br>

  <c:section name="Assumptions" />
  <ul>
    <li>Only main use cases are described.</li>
  </ul>

  <c:section name="Use Case Diagram" />
  <br>
  
</c:chapter>

```

### 3.6 <c:section>

The <c:section> tag generates a section title in chapter. This tag must be described in <c:chapter> tag.

#### Tag attributes

The <c:section> tag's attributes are as follows:

Attribute	Description	Required	Default
test	Generates this content if the test result is true.	No	true
id	Identifier of this tag	No	empty
name	Section name	Yes	--
pageBreak	Make a page break before the new section in use case document.	No	false

### 3.7 <c:pageBreak>

The <c:pageBreak> tag inserts a page break in chapter. This tag must be described in <c:chapter> tag.

#### Tag attributes

The <c:pageBreak> tag's attributes are as follows:

Attribute	Description	Required	Default
test	Generates this content if the test result is true.	No	true
id	Identifier of this tag	No	empty

## 4 Action Tags

This section shows tags to generate actions frame including actions actor can take as the next action.

### 4.1 <c:actions>

The <c:actions> tag generates an Actions frame that contains actions actor will take.

#### Tag attributes

The <c:actions> tag's attributes are as follows:

Attribute	Description	Required	Default
test	Generates the Actions frame if the test result is true.	No	true
id	Identifier of this tag	No	empty
label	Frame label	No	Actions
name	Actions title to be displayed next to the frame label.	No	empty
show	Show table contents when the page is displayed. <b>true:</b> Show the contents <b>false:</b> Hide the contents	No	true
width	Frame width (pixel or percent)	No	100%
border	The thickness of the frame	No	1
showFrameBar	Show the frame bar. <b>true:</b> Show, <b>false:</b> Hide	No	true
showTableTitle	Show the table title. <b>true:</b> Show, <b>false:</b> Hide	No	true

#### Tag body

The <c:actions> tag's body accepts the following tags:

- <c:action>

### 4.2 <c:action>

The <c:action> tag generates a row in the Actions frame. Each action row represents an action actor can take. This tag must be described in <c:actions> tag.

#### Tag attributes

The <c:action> tag's attributes are as follows:

Attribute	Description	Required	Default
test	Generates the action row if the test result is true.	No	true
id	Identifier of this tag	No	empty
name	Name of action.	Yes	--
useCaseId	Use Case ID linked from this action	No	--
useCaseName	Use Case name	No	Note1
link	Link URL to the next page.	No	empty
bgcolor	Background color of the row	No	Note 2
useCaseBgcolor	Background color of the row when useCaseId is specified	No	lightyellow w

Note 1: Value of useCase tag in jcase.xml is used as default. If the value is not specified there, useCaseName attribute is required.

Note 2: tableBodyBgcolorDark (#F7F7F7) or tableBodyBgcolorLight (white)

### Tag body

The `<c:action>` tag's body accepts HTML text as action description.

### Example 1) Actions

Display an Action tables containing two actions:

```
<c:actions>
  <c:action name="Valid Credencial" link="loginProc.jsp?email=test@test&pwd=test">
    The user enters correct email address and password registered in database.
  </c:action>
  <c:action name="Invalid Format" link="loginProcess.jsp">
    The user enters invalid format of email or password.
  </c:action>
</c:actions>
```

### Example 2) Action link

Display an Action tables containing an action and an action link linked to the use case UC03:

```
<c:actions>
  <c:action name="Valid Credencial" link="loginProc.jsp?email=test@test&pwd=test">
    The user enters correct email address and password registered in database.
  </c:action>
  <c:action name="Forgot Password"
    useCaseId="UC03"
    useCaseName="Forgot Password"
    link="../uc03/forgot.jsp">
    The user forgot her/his password. The user goes to "Forgot Password" flow.
  </c:action>
</c:actions>
```

### Example 3) link with URL parameters

If you want to add some specific URL parameters in the link attribute, you can specify each parameter using param implicit object. (`${param.<parameter name>}`)

If you need to add all URL parameters, specify `${pageContext.request.queryString}` or use `<c:queryString>` function.

```
<c:actions>
  <c:action name="Next" link="next.jsp?id=${param.id}&session=${param.session}">
    The user selectes to go next step with specific parameters.
  </c:action>

  <c:action name="Next" link="next.jsp?${pageContext.request.queryString}">
    The user selects to go next step with all URL parameters.
  </c:action>
</c:actions>
```

## 5 Screen Tags

This section shows tags to represent a screen, dialog, email, etc. on event page.

### 5.1 <c:screen>

The <c:screen> tag generates a Screen frame that represents abstract UI including messages and form elements.

#### Tag attributes

The <c:screen> tag's attributes are as follows:

Attribute	Description	Required	Default
test	Generates the Screen frame if the test result is true.	No	true
id	Identifier of this tag	No	empty
label	Screen label to be displayed on the frame bar.	No	Screen
name	Screen title name to be displayed next to the label.	No	empty
show	Show table contents when the page is displayed. <b>true:</b> Show <b>false:</b> Do not show	No	true
width	Frame width (pixel or percent)	No	empty
border	The thickness of the frame	No	3
align	Horizontal alignment of the table contents <b>left:</b> Align left <b>center:</b> Centering <b>right:</b> Align right	No	left
valign	Vertical alignment of the table contents <b>top:</b> Align top <b>center:</b> Centering <b>bottom:</b> Align bottom	No	center
bgcolor	Background color of the contents area	No	#F7F7F7
cellpadding	Space between frame and content area. (pixel)	No	5

#### Tag body

The <c:screen> tag's body accepts the following tags:

- <c:instruction> (<c:inst>)
- <c:link>
- HTML text

#### Example 1) Login Form

Display a page containing a login form:

```
<c:screen name="Login Form" align="center">
  <table>
    <tr><td>User ID</td><td><input type="text" size="30"></td></tr>
    <tr><td>Password</td><td><input type="text" size="30"></td></tr>
  </table>
  <input type="button" value="Submit">
</c:screen>
```

#### Example 2) Email Message

Display an email message:

```
<c:screen label="E-mail" name="Thank You e-mail">
```

```
Thank you for your registration!<br>
<br>
<u>Click here</u> to log in.
</c:screen>
```

## 5.2 <c:dialog>

The <c:dialog> tag generates a Dialog table that represents a dialog screen.

### Tag attributes

The <c:dialog> tag's attributes are as follows:

Attribute	Description	Required	Default
test	Generates the Dialog frame if the test result is true.	No	true
id	Identifier of this tag	No	empty
label	Screen label to be displayed on the frame bar.	No	Dialog
name	Screen title name to be displayed next to the label.	No	empty
show	Show table contents when the page is displayed. <b>true:</b> Show <b>false:</b> Do not show	No	false
width	Frame width (pixel or percent)	No	empty
border	The thickness of the frame	No	3
align	Horizontal alignment of the table contents <b>left:</b> Align left <b>center:</b> Centering <b>right:</b> Align right	No	center
valign	Vertical alignment of the table contents <b>top:</b> Align top <b>center:</b> Centering <b>bottom:</b> Align bottom	No	center
bgcolor	Background color of the contents area	No	#E0E0E0
cellpadding	Space between frame and content area. (pixel)	No	5
button	Name of button to be displayed at the bottom line of the dialog. In case of multiple buttons, separate using comma, e.g. "OK,Cancel"	No	OK
disabled	Disable buttons specified in button attribute. <b>true:</b> Disabled <b>false:</b> Not disabled	No	false

### Tag body

The <c:dialog> tag's body accepts the following tags:

- <c:instruction> (<c:inst>)
- <c:link>
- HTML text

### Example)

Display a confirmation dialog:

```
<c:dialog name="Delete files" button="OK,Cancel">
  Are you sure to delete all files?
</c:dialog>
```

### 5.3 <c:image>

The <c:image> tag generates an Image frame that basically displays images.

#### Tag attributes

The <c:image> tag's attributes are as follows:

Attribute	Description	Required	Default
test	Generates the Image frame if the test result is true.	No	true
id	Identifier of this tag	No	empty
label	Image label to be displayed on the frame bar.	No	Image
name	Image title name to be displayed next to the label.	No	empty
show	Show table contents when the page is displayed. <b>true:</b> Show <b>false:</b> Do not show	No	true
src	URL of image	No	empty
height	Image height (pixel)	No	empty
width	Image width (pixel)	No	empty
border	The thickness of the frame	No	3
title	Screen title to be displayed next to the name.	No	empty
align	Horizontal alignment of the table contents <b>left:</b> Align left <b>center:</b> Centering <b>right:</b> Align right	No	center
valign	Vertical alignment of the table contents <b>top:</b> Align top <b>center:</b> Centering <b>bottom:</b> Align bottom	No	center
bgcolor	Background color of the contents area	No	white
cellpadding	Space between frame and content area. (pixel)	No	5

#### Tag body

The <c:screen> tag's body accepts the following tags:

- <c:instruction> (<c:inst>)
- <c:link>
- HTML text

#### Example 1) Server Image

Display an image of web server:

```
<c:image name="Web Server" src="${user.images}/server.gif"/>
```

#### Example 2) Image with text

Display an image with text:

```
<c:image name="Web Server" src="${user.images}/server.gif" height="100" width="200">
  <br>
  Note that this is a tentative sample image.
</c:image>
```

## 6 Conditional Tags

---

### 6.1 <c:if>

The <c:if> tag is a simple conditional tag. If the condition specified in test attribute is true, the body content is processed.

#### Tag attributes

The <c:if> tag's attributes are as follows:

Attribute	Description	Required	Default
test	Condition that determines if the body content should be processed.	Yes	--

#### Tag body

The <c:if> tag accepts HTML text and other J-CASE tags.

#### Example) Simple condition

If result parameter equals 'success', display a success message.

```
<c:if test="${param.result == 'success'}">
  Your request has been processed successfully!
</c:if>
```

### 6.2 <c:choose>, <c:when>, <c:otherwise>

The <c:choose>, <c:when> and <c:otherwise> tags are conditional tags for mutually exclusive conditional operations. If the condition in <c:when> tag is true, the body content is processed. When all conditions are false, the body content in <c:otherwise> is processed.

#### Tag attributes

The <c:choose> and <c:otherwise> don't accept attributes.

The <c:when> tag's attributes are as follows:

Attribute	Description	Required	Default
test	Condition that determines if the body content should be processed.	Yes	--

#### Tag body

The <c:choose> tag accepts <c:when> and <c:otherwise> tags in the body.

The <c:when> and <c:otherwise> tags accept HTML text and other J-CASE tags.

#### Example) Choose a message

Display a message according to state variable.

```
<c:choose>
  <c:when test="${state == 'init'}">
    Welcome to our site!
  </c:when>
  <c:when test="${state == 'done'}">
    You have already processed.
  </c:when>
  <c:otherwise>
    Unknown state!
```

```
</c:otherwise>
</c:choose>
```

## 7 Tool Tags

---

### 7.1 <c:generateUseCase>

The <c:generateUseCase> tag generates a Use Case document by accessing pages linked from the specified link.

**Note that <c:generateUseCase> tag doesn't work properly on some servlet container, also it's not secure to use this tag on Web Server connected to Internet. Please use this tag only for internal use. We strongly recommend using the UseCaseGenerator Java application for other cases.**

#### Tag attributes

The <c:generateUseCase> tag's attributes are as follows:

Attribute	Description	Required	Default
test	Execute this tag if the test result is true.	No	true
id	Identifier of this tag	No	empty
input	Link URL to web page containing Use Cases. In case of multiple links, it can be separated by commas, e.g. input="uc1/index.jsp,uc2/index.jsp"	Yes	--
output	File path of Use Case document to be created.	Yes	--
cover	Path of cover HTML/JSP file. In case of multiple covers, it can be separated by commas, e.g. cover="cover1.html,cover2.html"	No	empty
appendix	Path to appendix HTML/JSP file. In case of multiple appendixes, it can be separated by commas, e.g. appendix="appendix1.html,appendix2.html"	No	empty
showContents	Insert contents page. <b>{true false}</b>	No	true
pageBreak	Insert a page break before starting each Use Case. <b>{true false}</b>	No	true
pageBreakContents	Insert a page break before Contents page. <b>{true false}</b>	No	true
pageBreakAppendix	Insert a page break before Appendix page. <b>{true false}</b>	No	true
debug	Display debugging messages. <b>{true false}</b>	No	false
encoding	Encoding code, e.g. UTF-8	No	empty
width	Table width (pixel or percent)	No	100%
title	Document title to be inserted in <TITLE> tag.	No	empty
align	Horizontal alignment of tables <b>left:</b> Align left <b>center:</b> Centering	No	left
user	User ID to authorize for Basic authentication	No	--
password	Password of the user	No	empty

showName	Display name as title on each event.	No	true
showScreen	Insert screen tables. {true false}	No	true
showProcessResults	Insert process results tables. {true false}	No	true
useCaseDiagram	File path of Use Case Diagram be created.	No	--
navigationMap	File path of Navigation Map Diagrams to be created.	No	--
urlHost	URL that only contains protocol, host name and port number. e.g. http://localhost:8080	No	--

**Tag body**

The <c:generateuserCase> tag has no body text.

**Example 1) Storyboard**

Generate a storyboard document that contains screens and process results.

```
<c:generateUseCase
  input="http://localhost:8080/test"
  output="storyboard.html"
  cover="cover.html,history.html"
  title="Use-Case Storyboard"
  pageBreakContents="false"
  width="90%"
  align="center"
  debug="true" />
```

**Example 2) High-level Use Case**

Generate a high-level use case document that doesn't contain screens and process results.

```
<c:generateUseCase
  input="http://localhost:8080/test"
  output="useCase.html"
  cover="cover.html,history.html"
  title="High-level Use Case"
  pageBreakContents="false"
  width="90%"
  align="center"
  debug="true"
  showName="false"
  showScreen="false"
  showProcessResults="false" />
```

**Example 3) UML Diagrams**

Generate a use case diagram and navigation maps.

```
<c:generateUseCase
  input="http://localhost:8080/test"
  output="useCase.html"
  useCaseDiagram="useCase.uud"
  navigationMap="navMap.uad" />
```

## 8 Other Tags

---

### 8.1 <c:title>

The <c:title> tag generates a title box that shows the current Use Case information and the page title specified in the name attribute and body.

#### Tag attributes

The <c:title> tag's attributes are as follows:

Attribute	Description	Required	Default
test	Generates the title box if the test result is true.	No	true
id	Identifier of this tag	No	empty
name	Title name.	No	empty
width	Box width (pixel or percent)	No	100%
border	The thickness of the box frame	No	1
align	Horizontal alignment of the table contents <b>left:</b> Align left <b>center:</b> Centering <b>right:</b> Align right	No	center
color	Text color	No	black
bgcolor	Background color	No	lightblue

#### Tag body

The <c:title> tag accepts HTML text to be displayed in the title box.

#### Example 1) Title name

Display the current Use Case information with a title name:

```
<c:title name="Homepage" />
```

When the JSP is executed in use case [UC01] Log in, the following box is generated in the HTML:

**[UC01] Log in: Homepage**

#### Example 2) Title name and Description

Display the current Use Case information with a title name and description:

```
<c:title name="Homepage">
  This is a sample title for homepage.
</c:title>
```

When the JSP is executed in use case [UC01] Log in, the following box is generated in the HTML:

**[UC01] Log in: Homepage**  
This is a sample title for homepage

## 8.2 <c:processResults>

The <c:processResults> tag generates a Process Results frame that shows results of process execution.

### Tag attributes

The <c:processResults> tag's attributes are as follows:

Attribute	Description	Required	Default
test	Generates the frame content if the test result is true.	No	true
id	Identifier of this tag	No	empty
label	Frame label.	No	Process Results
title	Title name.	No	empty
show	Show table contents when the page is displayed. <b>true:</b> Show <b>false:</b> Do not show	No	true
width	Table width (pixel or percent)	No	empty
border	The thickness of the frame	No	1
errorBgcolor	Background color to be used in case of error	No	pink
skipBgcolor	Background color to be used in case of skip	No	#E0E0E0
cellpadding	Space between frame and content area. (pixel)	No	5

### Tag body

The <c:processResults> tag has no body text.

### Example)

Define processes and Display the results:

```

<%@ page import="jcase.*" %>
...
<%!
    @JCaseProcess("Validate user")
    public void validate(PageContext pageContext) throws Exception {
        String email = JCaseUtils.getParameter(pageContext, "email");
        String password = JCaseUtils.getParameter(pageContext, "password");

        if (!"test@test".equals(email) || !"testpass".equals(password)) {
            pageContext.setAttribute("loginResult", "errorOnServer");
            throw new Exception("Failed to login. Not found the email or password.");
        }

        pageContext.setAttribute("loginResult", "success");
    }
%>

<c:processResults />
```

## 8.3 <c:urlParameters>

The <c:urlParameters> tag generates a table that shows all URL parameter names and the values.

### Tag attributes

The <c:urlParameters> tag's attributes are as follows:

Attribute	Description	Required	Default
test	Generates the frame content if the test result is true.	No	true
id	Identifier of this tag	No	empty
label	Frame label.	No	URL Parameters
title	Title name.	No	empty
show	Show table contents when the page is displayed. <b>true:</b> Show <b>false:</b> Do not show	No	true
width	Frame width (pixel or percent)	No	100%
border	The thickness of the frame	No	1

**Tag body**

The <c:urlParameters> tag has no body text.

**Example)**

Display an URL Parameters table:

```
<c:urlParameters />
```

**8.4 <c:feedback>**

The <c:feedback> tag generates a feedback form and/or comment list table.

**Tag attributes**

The <c:feedback> tag's attributes are as follows:

Attribute	Description	Required	Default
test	Generates the frame content if the test result is true.	No	true
id	Identifier of this tag	No	empty
label	Frame label	No	Feedback
name	Title name.	No	empty
code	Feedback code. If you don't specify or specify 'all', all comments are displayed	No	all
show	Show table contents when the page is displayed. <b>true:</b> Show <b>false:</b> Do not show	No	true
width	Frame width (pixel or percent)	No	100%
border	The thickness of the frame	No	1
cols	The number of columns of feedback text area	No	70
rows	The number of rows of feedback text area	No	6
maxLimit	Maximum number of comments to be displayed in a Feedback table.	No	1000
showForm	Show feedback form to submit a new comment. <b>true:</b> Show <b>false:</b> Do not show	No	true
showComments	Show comments submitted. <b>true:</b> Show <b>false:</b> Do not show	No	true
showDelete	Show DELETE button to delete a comment.	No	false

	<b>true:</b> Show <b>false:</b> Do not show		
escapeXml	Escape <, >, &, \, ', " in comment. <b>true:</b> Escape <b>false:</b> No escape	No	true

Note: Either showForm or showComments must be 'true'.

**Tag body**

The <c:feedback> tag has no body text.

**Example 1) Feedback for Login**

Display a Feedback form and comments for 'Login' code.

```
<c:feedback code="Login" />
```

**Example 2) All Comments**

Display all comments without feedback form.

```
<c:feedback code="all" showForm="false" />
```

**8.5 <c:instruction> <c:inst>**

The <c:instruction> tag displays notes or messages to guide user.

The <c:inst> can be used for short.

**Tag attributes**

The <c:instruction> tag's attributes are as follows:

Attribute	Description	Required	Default
test	Generates the content if the test result is true.	No	true
id	Identifier of this tag	No	empty
color	Text color.	No	green
size	Font size 1 (small) to 7 (big)	No	empty

**Tag body**

The <c:instruction> tag accepts HTML text in its body.

**Example)**

Display a message.

```
<c:instruction>Please choose an action in actions table.</c:instruction>
```

The above code can be replaced with <c:inst> for short.

```
<c:inst>Please choose an action in actions table.</c:inst>
```

**8.6 <c:link>**

The <c:link> tag generates a link. If you need to make a link that appears in use case document as well as UseCase/Action JSP, <c:link> can be used.

**Tag attributes**

The <c:link> tag's attributes are as follows:

Attribute	Description	Required	Default
-----------	-------------	----------	---------

test	Generates the content if the test result is true.	No	true
id	Identifier of this tag	No	empty
name	Link name to be used in use case document	Yes	--
href	Link URL to be used in Use Case/Action JSPs	Yes	--
target	target window name	No	empty

**Tag body**

The <c:link> tag's body accepts HTML text to be shown as its link text.

**Example)**

Display a message.

```
For detail screen, see
<c:link name="loginScreen" href="../screens.jsp" target="_blank">Login Screen</c:link>
```

The above code generates the following <a> tag in UseCase/Action JSPs:

```
For detail screen, see
<a href="../screens.jsp#loginScreen" target="_blank">Login Screen</a>
```

And below is generated in use case document:

```
For detail screen, see
<a href="#loginScreen">Login Screen</a>
```

**8.7 <c:set>**

The <c:set> tag sets an evaluated value in the specified scope.

**Tag attributes**

The <c:set> tag's attributes are as follows:

Attribute	Description	Required	Default
id	Identifier of this tag	No	empty
var	Variable name	Yes	---
value	Value or expression to be evaluated	No	body text
scope	Scope for var <b>page:</b> Page Scope <b>request:</b> Request Scope <b>session:</b> Session Scope <b>application:</b> Application Scope	No	empty

**Tag body**

The <c:set> tag accepts body text as its value.

**Example 1) Use value attribute**

Set a message specified in value attribute.

```
<c:set var="msg" value="test message" />
```

**Example 2) Specify scope**

Set a message in session scope.

```
<c:set var="msg" value="test message" scope="session" />
```

**Example 3) Body text**

Set a message in its body in the variable.

```
<c:set var="msg">
  test message
</c:set>
```

**8.8 <c:remove>**

The <c:remove> tag removes a variable in the specified scope.

**Tag attributes**

The <c:remove> tag's attributes are as follows:

Attribute	Description	Required	Default
id	Identifier of this tag	No	empty
var	Variable name	Yes	---
scope	Scope for var <b>page:</b> Page Scope <b>request:</b> Request Scope <b>session:</b> Session Scope <b>application:</b> Application Scope	No	empty

**Tag body**

The <c:remove> tag doesn't accept body text.

**Example 1) No scope**

Remove a variable.

```
<c:remove var="msg" />
```

**Example 2) Specify scope**

Remove a variable from application scope.

```
<c:remove var="msg" scope="application" />
```

**8.9 <c:catch>**

The <c:remove> tag catches any Throwable exceptions.

**Tag attributes**

The <c:catch> tag's attributes are as follows:

Attribute	Description	Required	Default
var	Variable name	Yes	---

**Tag body**

The <c:catch> tag accepts a text in its body.

**Example 1) Catch an error**

Catch an error thrown from init tag.

```
<c:catch var="err">
  <c:init />
```

</c:catch>

### 8.10 <c:include>

The <c:include> tag includes another JSP.

#### Tag attributes

The <c:include> tag's attributes are as follows:

Attribute	Description	Required	Default
test	Includes the page if the test result is true.	No	true
id	Identifier of this tag	No	empty
page	Relative path to be included.  NOTE 1. If you specify an URL with a parameter that may contain special characters, the parameter has to be encode using urlencode function. 2. Even if the page is empty, this tag doesn't throw a JspException.	Yes	---
flush	The writer object is to be flushed before the 'include'. <b>true:</b> flush <b>false:</b> not flush	No	true

#### Example 1) Include a JSP

Include a JSP file without parameter.

<c:include page="include/test.jsp" />

#### Example 2) Include a JSP with parameters

Include a JSP file with parameters.

<c:include page="include/test.jsp?zip=12345&name=\${fn:urlEncode(userName)}" />

### 8.11 <c:terminate>

The <c:terminate> tag terminates the evaluation in JSP.

#### Tag attributes

The <c:terminate> tag has no attributes.

#### Example) Terminate the process

If the condition is true, terminate the process.

```
<c:if test="${condition}">
  <c:terminate />
</c:if>
```

### 8.12 <c:tag>

The <c:tag> tag itself does nothing. This tag is used for replacing with another tag in extended JSP or deleting the tag in parent JSP (servlet class).

**Tag attributes**

The <c:tag> tag's attributes are as follows:

Attribute	Description	Required	Default
id	Identifier of this tag	No	empty

**Example 1) Mark as an extended point**

Set extended point in storyboard library so that another library can override it.

```
<c:tag id="extendedPoint" />
```

**Example 2) Mark a string as an extended point**

Specify the ID for the label so that it can be replaced in inherited JSP or aspect JSP.

```
<c:tag id="label_userid">User ID</c:tag>
```

## 9 Functions

---

### 9.1 <fn:queryString>

The <fn:queryString> function returns a query string.

#### Parameters

The <fn:queryString> function's parameters are as follows:

Parameter	Description	Required
pageContext	PageContext object	Yes

#### Example) All Parameters

Returns all URL parameters.

```

<%@ taglib prefix="fn" uri="http://www5f.biglobe.ne.jp/~webtest/jcase/functions" %>
...
<c:action name="Next" link="next.jsp?${fn:queryString(pageContext)}" />
```

### 9.2 <fn:includesQueryString>

The <fn:includesQueryString> function returns a query string only including parameters specified in includes.

#### Parameters

The <fn:includesQueryString> function's parameters are as follows:

Parameter	Description	Required
pageContext	PageContext object	Yes
includes	URL parameter name(s) to be included.	Yes

#### Example) Parameters except Specific Parameters

Sets all parameters except targetUrl and session parameters:

```

<c:action name="Next"
link="Next.jsp?${fn:includesQueryString(pageContext, 'userId, session')}" />
```

### 9.3 <fn:excludesQueryString>

The <fn:excludesQueryString> function returns a query string except parameters specified in excludes.

#### Parameters

The <fn:excludesQueryString> function's parameters are as follows:

Parameter	Description	Required
pageContext	PageContext object	Yes
excludes	URL parameter name(s) to be excluded.	Yes

#### Example) Parameters except Specific Parameters

Sets all parameters except targetUrl and session parameters:

```

<c:action name="Next"
link="${targetUrl}?session=valid&${fn:excludesQueryString(pageContext, 'targetUrl,
```

```
session'}}" />
```

#### 9.4 <fn:directoryName>

The <fn:directoryName> function returns a directoryName in specified URL.

##### Parameters

The <fn:directoryName> function's parameters are as follows:

Parameter	Description	Required
url	URL	Yes
index	Index number of directory (starts with 0)	Yes

##### Example) The second directory name in URL

Gets the second directory (index = 1) in targetUrl parameter:

```
<c:action name="Next" useCaseId="{fn:directoryName(param.targetUrl, 1)}" />
```

#### 9.5 <fn:replaceAll>

The <fn:replaceAll> function replaces each substring of this string that matches the given regular expression with the given replacement.

##### Parameters

The <fn:replaceAll> function's parameters are as follows:

Parameter	Description	Required
str	Input string to be searched	Yes
regex	The regular expression to be matched	Yes
replacement	The string to be substituted for each match	Yes

##### Example) Replace all strings

Replace all "old" in the parent string with "new".

```
New string: {fn:replaceAll(parent, "old", "new")}
```

#### 9.6 <fn:substringAfterLast>

The <fn:substringAfterLast> function returns a subset of a string following a specific substring, searching backward.

##### Parameters

The <fn:substringAfterLast> function's parameters are as follows:

Parameter	Description	Required
str	Input string to be searched	Yes
substr	Substring to search for	Yes

##### Example) The use case short ID

Displays the short ID in the 'use case ID' string

```
Use Case Short ID: {fn:substringAfterLast(useCaseId, "-")}
```

### 9.7 <fn:substringBeforeLast>

The <fn:substringBeforeLast> function returns a subset of a string before a specific substring, searching backward.

#### Parameters

The <fn:substringBeforeLast> function's parameters are as follows:

Parameter	Description	Required
str	Input string to be searched	Yes
substr	Substring to search for	Yes

#### Example) The use case group ID

Displays the group ID in the 'use case ID' string

Use Case Group ID: <code>{fn:substringBeforeLast(useCaseId, "-")}</code>
--

### 9.8 <fn:ifNull>

The <fn:ifNull> function returns the specified default object if the first given object is null.

#### Parameters

The <fn:ifNull> function's parameters are as follows:

Parameter	Description	Required
obj	The object to be tested	Yes
defaultObj	The default object to be returned if the obj is null.	Yes

#### Example) The attribute in default variable

Displays the attribute in defaultVar variable if myVar is null

Attribute: <code>{fn:ifNull(myVar, defaultVar).attribute}</code>
--

### 9.9 <fn:ifEmpty>

The <fn:ifEmpty> function returns the specified default object if the first given object is empty.

#### Parameters

The <fn:ifEmpty> function's parameters are as follows:

Parameter	Description	Required
obj	The object to be tested (String, Collection, Map or Array can be set)	Yes
defaultObj	The default object to be returned if the obj is empty.	Yes

#### Example) The attribute in default variable

Displays the attribute in defaultVar variable if myVar is empty

Attribute: <code>{fn:ifEmpty(myVar, defaultVar).attribute}</code>
---

### 9.10 <fn:ifTest>

The <fn:ifTest> function returns the first specified object if the test result is true. Otherwise, return the second object.

#### Parameters

The <fn:ifTest> function's parameters are as follows:

Parameter	Description	Required
result	Test result (true or false)	Yes
thenObj	The object to be returned if the test result is true.	Yes
elseObj	The object to be returned if the test result is false.	Yes

**Example) Return a different value according to the test result**

Displays 'Equal' if the myVar1 equals myVar2, otherwise display 'Not equal'.

```
${fn:ifTest(myVar1 == myVar2, 'Equal', 'Not equal')}
```

### 9.11 <fn:urlEncode>

The <fn:urlEncode> function translates a string into application/x-www-form-urlencoded format using UTF-8 encoding scheme.

**Parameters**

The <fn:urlEncode> function's parameters are as follows:

Parameter	Description	Required
str	Input string to be encoded	Yes

**Example) An encoded parameter**

Includes a JSP with an encoded parameter.

```
<c:include page="include/test.jsp?name=${fn:urlEncode(userName)}" />
```

### 9.12 <fn:urlDecode>

The <fn:urlDecode> function decodes a application/x-www-form-urlencoded string using UTF-8 encoding scheme.

**Parameters**

The <fn:urlDecode> function's parameters are as follows:

Parameter	Description	Required
str	Input string to be decoded	Yes

**Example) Decode a string**

Decodes an encoded string.

```
Decoded value: ${fn:urlDecode(encodedString)}
```

## 10 Aspect Weaving

With ServletGenerator tool, you can insert and replace with another J-CASE or HTML tag by extending the JSP or specifying parent class. To extend a JSP, a parent JSP or class path is specified in “extends” attribute of the page directive. To weave a J-CASE tag, a parent JSP or class path is specified in parentClass attribute of the tag. In both cases, parentId has to be specified to identify the tag to be updated and advice can be provided to determine where to be inserted or replaced.

### Test.jsp

This is a parent JSP which can execute direct on servlet container. You can specify id attribute to enable to be extended or weaved.

```
<c:tag id="message">User ID</c:tag>
```

### Inheritance.jsp

The tag in the parent JSP can be extended by this JSP.

```
<%@ page extends="jcase.jsp.test_jsp" %>

<c:tag
  id="message_new"
  parentId="message"
  advice="around">Company ID (<c:parent />)</c:tag>
```

The JSP will generate the code below:

```
Company ID (User ID)
```

### Aspect.jsp

Also, it can be weaved by this JSP, and you will get the same result.

```
<c:tag
  id="message_new"
  parentClass="jcase.jsp.test_jsp"
  parentId="message"
  advice="around">Company ID (<c:parent />)</c:tag>
```

In the inherited JSPs and aspect JSPs, you can specify the following attributes. They aren't explained in each tag section, but they are available in almost all J-CASE tags in extended and aspect JSPs.

#### Tag attributes available in extended and aspect JSP

The following attributes are accepted with all J-CASE tags **except** catch, if, when, otherwise, replace and parent tags.

Attribute	Description	Required	Default
parentClass	Regular expression that specifies JSP or class path to be weaved. e.g. .* (All JSPs and classes) ^test\.jsp\$ (test.jsp) .*test.* (JSPs and classes including test)	No (Note 1)	--
parentId	Regular expression that specifies the tag ID to be extended/weaved defined in the parent JSP or	Yes (Note 1)	--

	Servlet class.		
advice	<p>Advice type where to be inserted or replaced in the target tag.</p> <p><b>before:</b> Insert the tag before the target.</p> <p><b>after:</b> Insert the tag after the target.</p> <p><b>around:</b> Replace with the tag. The (original) target tag can be kept by specifying &lt;c:parent&gt; tag.</p> <p><b>top:</b> Insert the tag into inside top of the target tag.</p> <p><b>bottom:</b> Insert the tag into inside bottom of the target tag.</p>	Yes	--

Note 1: If parentId is specified, parent JSP path or Servlet class path has to be specified in parentClass attribute of the tag or extends attribute in page directive.

**Tags available in extended and aspect JSP**

The following tags are available to use only in extended and aspect JSP.

**10.1 <c:parent>**

If you specify around in advice attribute, you can specify the place where the (original) target tag should be inserted in the new tag.

The <c:parent> tag can be only described in inherited and aspect JSPs.

**Tag attributes**

The <c:parent> tag has no attributes.

**Example) Parent tag**

Specify the place where the parent tag is inserted.

```

<c:tag
  id="new_message"
  parentClass="jcase.jsp.test_jsp"
  parentId="message"
  advice="around">Company ID (<c:parent />)</c:tag>
```

**10.2 <c:replace>**

If you need to replace or delete messages in parent classes, which is not marked by J-CASE tag, <c:replace> tag can be used. The <c:replace> tag replaces the string in the generated HTML with the specified new string.

The <c:replace> tag can be only described in inherited and aspect JSPs.

**Tag attributes**

The <c:replace> tag's attributes are as follows:

Attribute	Description	Required	Default
old	The regular expression to be matched	Yes	--
new	The string to be substituted for each match	No	empty

**Example) Replace a word**

Replace a word "Keyword" in HTML with "New Keyword". But the word in link must not be replaced.

```
<c:replace old="(\\W)Keyword(\\W)" new="$1New Keyword$2" />
```

**NOTE:**

With the expression, “<u>Keyword</u>”, “Enter a Keyword” are replaced with “<u>New Keyword</u>”, “Enter a New Keyword”, respectively. However, “link=/SearchKeyword.jsp” is not replaced. The link URL is used to control the page flow, so it shouldn’t be replaced usually. In this case, we can specify “(\\W)Keyword(\\W)” instead of “Keyword” in the old attribute not to match with the word (Keyword) in the word combined with another word (SearchKeyword.jsp).