

## JSTL Expression Language Definition

Implicit Object	Contains
pageScope	variables which have the scope page.
requestScope	variables which have the scope request.
sessionScope	variables which have the scope session.
applicationScope	variables which have the scope of the application.
pageContext	JSP PageContext object
cookie	Cookie values
param	Request parameters as strings.
header	HTTP request headers as strings.
headerValues	HTTP request headers as a collection of strings
initParam	Context initialization parameters

Autor: Sabine Winkler

Terminе unserer offenen Seminare in Mannheim  
 Kurs-ID: J.SERV / Schulung "Servlets und JavaServer™ Pages JSP"  
 30. - 31.01., 19. - 20.03., 11. - 12.06., 17. - 18.09., 12. - 13.11.2012

Orientation in Objects GmbH  
 Weinheimer Str.68 www.oio.de  
 68309 Mannheim info@oio.de

Kurs-ID: J.FACE / Schulung "JavaServer Faces"

Schulung, Beratung und  
 Entwicklung mit Java und XML

01. - 03.02., 21. - 23.03., 13. - 15.06., 19. - 21.09., 14. - 16.11.2012

## JSTL Operators

<b>Comparison operators:</b>	== / eq, != / ne, < / lt, > / gt, <= / le, >= / ge
<b>Logical operators:</b>	&& / and,    / or, ! / not
<b>Arithmetic operators:</b>	+, -, *, /, % / mod
<b>Property access:</b>	<b>Dot operator (.)</b> can be used to retrieve a named property. To access numbered or named properties use the <b>bracket operator ( [ ] )</b> . Example: <code>\$user.name</code> , <code>\$(row[0])</code> or <code>\$(order.items.price)</code> Use the <b>empty operator</b> to determine whether a collection or string is empty or null. Example: <code>&lt;c:if test="\$ {empty pageScope.users}"&gt;</code>

## Legend

All tags are case sensitive. A pair of single quotes is equivalent to a pair of double quotes. Spaces are not allowed between an equals sign and an attribute value.  
 plain text = required *italics* = user-defined underline = default value attribute = using a body, the content is associated with this attribute  
 [ ] = required choice | = or [ ] = optional ... = list of items ex. = exported sc. = scoped

## Action

Description	Syntax	Attributes																																
<b>Core Tag Library - General Purpose Actions</b>																																		
<b>&lt;&lt;out&gt;</b> Evaluates an expression and outputs the result to the current JspWriter object.	<code>&lt;&lt;out value="value" [escapeXml="{true false}"] [default="{defaultValue}"]&gt;</code>	<table border="1"> <thead> <tr> <th>name</th> <th>dyn*</th> <th>type</th> <th>description</th> </tr> </thead> <tbody> <tr> <td>value</td> <td>true</td> <td>Object</td> <td>Expression to be evaluated.</td> </tr> <tr> <td>escapeXml</td> <td>true</td> <td>boolean</td> <td>Determines whether "&lt;&gt;,&amp;," in the result should be converted to their corresponding character entity codes.</td> </tr> <tr> <td>default</td> <td>true</td> <td>Object</td> <td>Default value if the resulting value is null.</td> </tr> </tbody> </table>	name	dyn*	type	description	value	true	Object	Expression to be evaluated.	escapeXml	true	boolean	Determines whether "<>,&," in the result should be converted to their corresponding character entity codes.	default	true	Object	Default value if the resulting value is null.																
name	dyn*	type	description																															
value	true	Object	Expression to be evaluated.																															
escapeXml	true	boolean	Determines whether "<>,&," in the result should be converted to their corresponding character entity codes.																															
default	true	Object	Default value if the resulting value is null.																															
<b>&lt;&lt;set&gt;</b> Sets the value of a scoped variable or a property of a target object.	<code>&lt;&lt;set value="value" var="varName" [scope="{page request session application}"]&gt;</code> or <code>&lt;&lt;set value="value" target="target" property="propertyName"&gt;</code>	<table border="1"> <thead> <tr> <th>name</th> <th>dyn*</th> <th>type</th> <th>description</th> </tr> </thead> <tbody> <tr> <td>value</td> <td>true</td> <td>Object</td> <td>Expression to be evaluated.</td> </tr> <tr> <td>var</td> <td>false</td> <td>String</td> <td>Scoped variable to hold the value specified in the action.</td> </tr> <tr> <td>scope</td> <td>false</td> <td>String</td> <td>Scope for var.</td> </tr> <tr> <td>target</td> <td>true</td> <td>Object</td> <td>JavaBeans object with setter property, or to a java.util.Map object.</td> </tr> <tr> <td>property</td> <td>true</td> <td>String</td> <td>Name of the property to be set in the target object.</td> </tr> </tbody> </table>	name	dyn*	type	description	value	true	Object	Expression to be evaluated.	var	false	String	Scoped variable to hold the value specified in the action.	scope	false	String	Scope for var.	target	true	Object	JavaBeans object with setter property, or to a java.util.Map object.	property	true	String	Name of the property to be set in the target object.								
name	dyn*	type	description																															
value	true	Object	Expression to be evaluated.																															
var	false	String	Scoped variable to hold the value specified in the action.																															
scope	false	String	Scope for var.																															
target	true	Object	JavaBeans object with setter property, or to a java.util.Map object.																															
property	true	String	Name of the property to be set in the target object.																															
<b>&lt;&lt;remove&gt;</b> Removes a scoped variable.	<code>&lt;&lt;remove var="varName" [scope="{page request session application}"]&gt;</code>	<table border="1"> <thead> <tr> <th>name</th> <th>dyn*</th> <th>type</th> <th>description</th> </tr> </thead> <tbody> <tr> <td>var</td> <td>false</td> <td>String</td> <td>Name of the scoped variable to be removed.</td> </tr> <tr> <td>scope</td> <td>false</td> <td>String</td> <td>Scope for var.</td> </tr> </tbody> </table>	name	dyn*	type	description	var	false	String	Name of the scoped variable to be removed.	scope	false	String	Scope for var.																				
name	dyn*	type	description																															
var	false	String	Name of the scoped variable to be removed.																															
scope	false	String	Scope for var.																															
<b>&lt;&lt;catch&gt;</b> Catches a java.lang.Throwable thrown by any of its nested actions.	<code>&lt;&lt;catch [var="varName"]&gt;</code> <i>nested actions</i> <code>&lt;/catch&gt;</code>	<table border="1"> <thead> <tr> <th>name</th> <th>dyn*</th> <th>type</th> <th>description</th> </tr> </thead> <tbody> <tr> <td>var</td> <td>false</td> <td>String</td> <td>Variable to expose information about error.</td> </tr> </tbody> </table>	name	dyn*	type	description	var	false	String	Variable to expose information about error.																								
name	dyn*	type	description																															
var	false	String	Variable to expose information about error.																															
<b>Core Tag Library - Conditional Actions</b>																																		
<b>&lt;&lt;if&gt;</b> Evaluates its body content if the expression is true.	<code>&lt;&lt;if test="testCondition" [var="varName"] [scope="{page request session application}"]&gt;</code> <i>body content</i> <code>&lt;/if&gt;</code>	<table border="1"> <thead> <tr> <th>name</th> <th>dyn*</th> <th>type</th> <th>description</th> </tr> </thead> <tbody> <tr> <td>test</td> <td>true</td> <td>boolean</td> <td>Test condition that determines whether or not the body content should be processed.</td> </tr> <tr> <td>var</td> <td>false</td> <td>String</td> <td>Name of the exported scoped variable for the resulting test condition value.</td> </tr> <tr> <td>scope</td> <td>false</td> <td>String</td> <td>Scope for var.</td> </tr> </tbody> </table> <p><i>Constraints: If scope is specified, var must also be specified.</i></p>	name	dyn*	type	description	test	true	boolean	Test condition that determines whether or not the body content should be processed.	var	false	String	Name of the exported scoped variable for the resulting test condition value.	scope	false	String	Scope for var.																
name	dyn*	type	description																															
test	true	boolean	Test condition that determines whether or not the body content should be processed.																															
var	false	String	Name of the exported scoped variable for the resulting test condition value.																															
scope	false	String	Scope for var.																															
<b>&lt;&lt;choose&gt;</b> <b>&lt;&lt;when&gt;</b> <b>&lt;&lt;otherwise&gt;</b>	<code>&lt;&lt;choose&gt;</code> <code>&lt;&lt;when test="testCondition1"&gt;body content&lt;/when&gt;</code> <code>&lt;&lt;when test="testCondition2"&gt;body content&lt;/when&gt;</code> ... <code>&lt;&lt;otherwise&gt;conditional block&lt;/otherwise&gt;</code> <code>&lt;/choose&gt;</code>	<table border="1"> <thead> <tr> <th>name</th> <th>dyn*</th> <th>type</th> <th>description</th> </tr> </thead> <tbody> <tr> <td>test</td> <td>true</td> <td>boolean</td> <td>Test condition - determines whether or not the body content should be processed.</td> </tr> </tbody> </table> <p><i>Constraints: The body of &lt;c:choose&gt; can only contain 1 or more &lt;c:when&gt; actions (must all appear before &lt;c:otherwise&gt;) and 0 or 1 &lt;c:otherwise&gt; action (must be the last action)</i></p>	name	dyn*	type	description	test	true	boolean	Test condition - determines whether or not the body content should be processed.																								
name	dyn*	type	description																															
test	true	boolean	Test condition - determines whether or not the body content should be processed.																															
<b>Core Tag Library - Iterator Actions</b>																																		
<b>&lt;&lt;forEach&gt;</b> Repeats its nested body content over a collection of objects, or repeats it a fixed number of times.	<code>&lt;&lt;forEach [var="varName"] items="collection" [varStatus="varStatusName"] [begin="begin" ] [end="end" ] [step="step" ]&gt;</code> <i>body content</i> <code>&lt;/forEach&gt;</code>	<table border="1"> <thead> <tr> <th>name</th> <th>dyn*</th> <th>type</th> <th>description</th> </tr> </thead> <tbody> <tr> <td>var</td> <td>false</td> <td>String</td> <td>Name of the exported scoped variable for the current item. Its type depends on the object of the underlying collection.</td> </tr> <tr> <td>items</td> <td>true</td> <td>(see descr.)</td> <td>Collection of items to iterate over. Types can contain all supported types.</td> </tr> <tr> <td>varStatus</td> <td>false</td> <td>String</td> <td>Name of the exported scoped variable for the status of the iteration.</td> </tr> <tr> <td>begin / end</td> <td>true</td> <td>int</td> <td>Iteration begins / ends at the specified index (items) or at the specified value.</td> </tr> <tr> <td>step</td> <td>true</td> <td>int</td> <td>Iteration will only process every step items.</td> </tr> </tbody> </table> <p><i>Constraints: If items is not specified, begin and end must be specified.</i></p>	name	dyn*	type	description	var	false	String	Name of the exported scoped variable for the current item. Its type depends on the object of the underlying collection.	items	true	(see descr.)	Collection of items to iterate over. Types can contain all supported types.	varStatus	false	String	Name of the exported scoped variable for the status of the iteration.	begin / end	true	int	Iteration begins / ends at the specified index (items) or at the specified value.	step	true	int	Iteration will only process every step items.								
name	dyn*	type	description																															
var	false	String	Name of the exported scoped variable for the current item. Its type depends on the object of the underlying collection.																															
items	true	(see descr.)	Collection of items to iterate over. Types can contain all supported types.																															
varStatus	false	String	Name of the exported scoped variable for the status of the iteration.																															
begin / end	true	int	Iteration begins / ends at the specified index (items) or at the specified value.																															
step	true	int	Iteration will only process every step items.																															
<b>&lt;&lt;forEachTokens&gt;</b> Iterates over tokens, separated by the supplied delimiters.	<code>&lt;&lt;forEachTokens items="stringOfTokens" delims="delimiters" [var="varName" ] [varStatus="varStatusName" ] [begin="begin" ] [end="end" ] [step="step" ]&gt;</code> <i>body content</i> <code>&lt;/forEachTokens&gt;</code>	<table border="1"> <thead> <tr> <th>name</th> <th>dyn*</th> <th>type</th> <th>description</th> </tr> </thead> <tbody> <tr> <td>var</td> <td>false</td> <td>String</td> <td>Name of the exported scoped variable for the current item of the iteration.</td> </tr> <tr> <td>items</td> <td>true</td> <td>String</td> <td>String of tokens to iterate over.</td> </tr> <tr> <td>delims</td> <td>true</td> <td>String</td> <td>The set of delimiters (the characters that separate the tokens in the string).</td> </tr> <tr> <td>varStatus</td> <td>false</td> <td>String</td> <td>Name of the exported scoped variable for the status of the iteration.</td> </tr> <tr> <td>begin / end</td> <td>true</td> <td>int</td> <td>Iteration begins / ends at the token located at the specified index.</td> </tr> <tr> <td>step</td> <td>true</td> <td>int</td> <td>Iteration will only process every step tokens.</td> </tr> </tbody> </table>	name	dyn*	type	description	var	false	String	Name of the exported scoped variable for the current item of the iteration.	items	true	String	String of tokens to iterate over.	delims	true	String	The set of delimiters (the characters that separate the tokens in the string).	varStatus	false	String	Name of the exported scoped variable for the status of the iteration.	begin / end	true	int	Iteration begins / ends at the token located at the specified index.	step	true	int	Iteration will only process every step tokens.				
name	dyn*	type	description																															
var	false	String	Name of the exported scoped variable for the current item of the iteration.																															
items	true	String	String of tokens to iterate over.																															
delims	true	String	The set of delimiters (the characters that separate the tokens in the string).																															
varStatus	false	String	Name of the exported scoped variable for the status of the iteration.																															
begin / end	true	int	Iteration begins / ends at the token located at the specified index.																															
step	true	int	Iteration will only process every step tokens.																															
<b>Core Tag Library - URL Related Actions</b>																																		
<b>&lt;&lt;redirect&gt;</b> Sends an HTTP redirect to the client.	<code>&lt;&lt;redirect url="value" [context="context" ]&gt;</code> <i>optional &lt;c:param&gt; subtags within the body</i>	<table border="1"> <thead> <tr> <th>name</th> <th>dyn*</th> <th>type</th> <th>description</th> </tr> </thead> <tbody> <tr> <td>url</td> <td>true</td> <td>String</td> <td>The URL of the resource to redirect to.</td> </tr> <tr> <td>context</td> <td>true</td> <td>String</td> <td>Context name when specifying a relative URL that belongs to a foreign context.</td> </tr> </tbody> </table>	name	dyn*	type	description	url	true	String	The URL of the resource to redirect to.	context	true	String	Context name when specifying a relative URL that belongs to a foreign context.																				
name	dyn*	type	description																															
url	true	String	The URL of the resource to redirect to.																															
context	true	String	Context name when specifying a relative URL that belongs to a foreign context.																															
<b>&lt;&lt;c:param&gt;</b> Adds request parameters to a URL.	<code>&lt;&lt;c:param name="name" value="value"&gt;</code>	<table border="1"> <thead> <tr> <th>name</th> <th>dyn*</th> <th>type</th> <th>description</th> </tr> </thead> <tbody> <tr> <td>name</td> <td>true</td> <td>String</td> <td>Name of the query string parameter.</td> </tr> <tr> <td>value</td> <td>true</td> <td>String</td> <td>Value of the parameter.</td> </tr> </tbody> </table>	name	dyn*	type	description	name	true	String	Name of the query string parameter.	value	true	String	Value of the parameter.																				
name	dyn*	type	description																															
name	true	String	Name of the query string parameter.																															
value	true	String	Value of the parameter.																															
<b>&lt;&lt;import&gt;</b> Imports the content of a URL-based resource.	<code>&lt;&lt;import url="url" [context="context" ] [var="varName" ] [scope="{page request session application}"] [varReader="varReaderName" ] [charEncoding="charEncoding" ]&gt;</code> <i>optional &lt;c:param&gt; subtags within the body</i>	<table border="1"> <thead> <tr> <th>name</th> <th>dyn*</th> <th>type</th> <th>description</th> </tr> </thead> <tbody> <tr> <td>url</td> <td>true</td> <td>String</td> <td>The URL of the resource to import.</td> </tr> <tr> <td>context</td> <td>true</td> <td>String</td> <td>Context name when specifying a relative URL that belongs to a foreign context.</td> </tr> <tr> <td>var</td> <td>false</td> <td>String</td> <td>Name of the exported scoped variable for the resource's content.</td> </tr> <tr> <td>scope</td> <td>false</td> <td>String</td> <td>Scope for var.</td> </tr> <tr> <td>charEncoding</td> <td>true</td> <td>String</td> <td>Character encoding of the content at the input resource.</td> </tr> <tr> <td>varReader</td> <td>false</td> <td>String</td> <td>Name of the exported scoped variable for the resource's content. (Type: Reader)</td> </tr> </tbody> </table> <p><i>Constraints: If varReader is specified, var and scope must not be specified.</i></p>	name	dyn*	type	description	url	true	String	The URL of the resource to import.	context	true	String	Context name when specifying a relative URL that belongs to a foreign context.	var	false	String	Name of the exported scoped variable for the resource's content.	scope	false	String	Scope for var.	charEncoding	true	String	Character encoding of the content at the input resource.	varReader	false	String	Name of the exported scoped variable for the resource's content. (Type: Reader)				
name	dyn*	type	description																															
url	true	String	The URL of the resource to import.																															
context	true	String	Context name when specifying a relative URL that belongs to a foreign context.																															
var	false	String	Name of the exported scoped variable for the resource's content.																															
scope	false	String	Scope for var.																															
charEncoding	true	String	Character encoding of the content at the input resource.																															
varReader	false	String	Name of the exported scoped variable for the resource's content. (Type: Reader)																															
<b>&lt;&lt;url&gt;</b> Builds a URL with the proper rewriting rules applied.	<code>&lt;&lt;url value="value" [context="context" ] [var="varName" ] [scope="{page request session application}"]&gt;</code> <i>optional &lt;c:param&gt; subtags within the body</i>	<table border="1"> <thead> <tr> <th>name</th> <th>dyn*</th> <th>type</th> <th>description</th> </tr> </thead> <tbody> <tr> <td>value</td> <td>true</td> <td>String</td> <td>URL to be processed.</td> </tr> <tr> <td>context</td> <td>true</td> <td>String</td> <td>Name of the ex. sc. variable for the resource's content.</td> </tr> <tr> <td>var</td> <td>false</td> <td>String</td> <td>Name of the ex. sc. variable for the processed url.</td> </tr> <tr> <td>scope</td> <td>false</td> <td>String</td> <td>Scope for var.</td> </tr> </tbody> </table>	name	dyn*	type	description	value	true	String	URL to be processed.	context	true	String	Name of the ex. sc. variable for the resource's content.	var	false	String	Name of the ex. sc. variable for the processed url.	scope	false	String	Scope for var.												
name	dyn*	type	description																															
value	true	String	URL to be processed.																															
context	true	String	Name of the ex. sc. variable for the resource's content.																															
var	false	String	Name of the ex. sc. variable for the processed url.																															
scope	false	String	Scope for var.																															
<b>SQL Actions</b>																																		
<b>&lt;sql:query&gt;</b> Queries a database.	<code>&lt;sql:query sql="sqlQuery" var="varName" [scope="{page request session application}"] [dataSource="dataSource" ] [maxRows="maxRows" ] [startRow="startRow" ]&gt;</code> <i>optional &lt;c:param&gt; actions within the body</i>	<table border="1"> <thead> <tr> <th>name</th> <th>dyn*</th> <th>type</th> <th>description</th> </tr> </thead> <tbody> <tr> <td>sql</td> <td>true</td> <td>String</td> <td>SQL query statement.</td> </tr> <tr> <td>dataSource</td> <td>true</td> <td>String, javax.sql.DataSource</td> <td>JNDI Data source</td> </tr> <tr> <td>maxRows</td> <td>true</td> <td>int</td> <td>The maximum number of rows to be included in the query result.</td> </tr> <tr> <td>startRow</td> <td>true</td> <td>int</td> <td>The returned Result object includes the rows starting at the specified index.</td> </tr> <tr> <td>var</td> <td>false</td> <td>String</td> <td>Name of the exported scoped variable for the query result.</td> </tr> <tr> <td>scope</td> <td>false</td> <td>String</td> <td>Scope for var.</td> </tr> </tbody> </table>	name	dyn*	type	description	sql	true	String	SQL query statement.	dataSource	true	String, javax.sql.DataSource	JNDI Data source	maxRows	true	int	The maximum number of rows to be included in the query result.	startRow	true	int	The returned Result object includes the rows starting at the specified index.	var	false	String	Name of the exported scoped variable for the query result.	scope	false	String	Scope for var.				
name	dyn*	type	description																															
sql	true	String	SQL query statement.																															
dataSource	true	String, javax.sql.DataSource	JNDI Data source																															
maxRows	true	int	The maximum number of rows to be included in the query result.																															
startRow	true	int	The returned Result object includes the rows starting at the specified index.																															
var	false	String	Name of the exported scoped variable for the query result.																															
scope	false	String	Scope for var.																															
<b>&lt;sql:update&gt;</b> Executes an SQL INSERT, UPDATE, or DELETE statement. (and SQL statements that return nothing)	<code>&lt;sql:update sql="sqlUpdate" [dataSource="dataSource" ] [var="varName" ] [scope="{page request session application}"]&gt;</code> <i>optional &lt;c:param&gt; actions within the body</i>	<table border="1"> <thead> <tr> <th>name</th> <th>dyn*</th> <th>type</th> <th>description</th> </tr> </thead> <tbody> <tr> <td>sql</td> <td>true</td> <td>String</td> <td>SQL update statement.</td> </tr> <tr> <td>dataSource</td> <td>true</td> <td>String, javax.sql.DataSource</td> <td>JNDI Data source</td> </tr> <tr> <td>var</td> <td>false</td> <td>String</td> <td>Name of the exported scoped variable for the result of the database update. (Integer)</td> </tr> <tr> <td>scope</td> <td>false</td> <td>String</td> <td>Scope for var.</td> </tr> </tbody> </table>	name	dyn*	type	description	sql	true	String	SQL update statement.	dataSource	true	String, javax.sql.DataSource	JNDI Data source	var	false	String	Name of the exported scoped variable for the result of the database update. (Integer)	scope	false	String	Scope for var.												
name	dyn*	type	description																															
sql	true	String	SQL update statement.																															
dataSource	true	String, javax.sql.DataSource	JNDI Data source																															
var	false	String	Name of the exported scoped variable for the result of the database update. (Integer)																															
scope	false	String	Scope for var.																															
<b>&lt;sql:transaction&gt;</b> Establishes a transaction context for <sql:query> and <sql:update> subtags.	<code>&lt;sql:transaction [dataSource="dataSource" ] [isolation="isolationLevel" ]&gt;</code> <i>&lt;sql:query&gt; and &lt;sql:update&gt; statements</i> <code>&lt;/sql:transaction&gt;</code>	<table border="1"> <thead> <tr> <th>name</th> <th>dyn*</th> <th>type</th> <th>description</th> </tr> </thead> <tbody> <tr> <td>dataSource</td> <td>true</td> <td>String, javax.sql.DataSource</td> <td>JNDI Data source</td> </tr> <tr> <td>isolation</td> <td>true</td> <td>String</td> <td>Max contain read_committed   read_uncommitted   repeatable_read   serializable.</td> </tr> </tbody> </table> <p><i>Constraints: Any nested &lt;sql:query&gt; and &lt;sql:update&gt; actions must not specify a dataSource attribute.</i></p>	name	dyn*	type	description	dataSource	true	String, javax.sql.DataSource	JNDI Data source	isolation	true	String	Max contain read_committed   read_uncommitted   repeatable_read   serializable.																				
name	dyn*	type	description																															
dataSource	true	String, javax.sql.DataSource	JNDI Data source																															
isolation	true	String	Max contain read_committed   read_uncommitted   repeatable_read   serializable.																															
<b>&lt;sql:setDataSource&gt;</b> Exports a data source either as a scoped variable or as the data source configuration variable (javax.servlet.jsp.jstl.sql.dataSource).	<code>&lt;sql:setDataSource [dataSource="dataSource" ] [url="jdbcUrl" ] [driver="driverClassName" ] [user="userName" ] [password="password" ] [var="varName" ] [scope="{page request session application}"]&gt;</code>	<table border="1"> <thead> <tr> <th>name</th> <th>dyn*</th> <th>type</th> <th>description</th> </tr> </thead> <tbody> <tr> <td>dataSource</td> <td>true</td> <td>String, javax.sql.DataSource</td> <td>JNDI Data source</td> </tr> <tr> <td>driver</td> <td>true</td> <td>String</td> <td>Class name of the JDBC driver.</td> </tr> <tr> <td>url</td> <td>true</td> <td>String</td> <td>JDBC database URL.</td> </tr> <tr> <td>user</td> <td>true</td> <td>String</td> <td>JDBC database user</td> </tr> <tr> <td>password</td> <td>true</td> <td>String</td> <td>JDBC user password</td> </tr> <tr> <td>var</td> <td>false</td> <td>String</td> <td>Name of the ex. sc. variable for the data source specified. Type: String / DataSource.</td> </tr> <tr> <td>scope</td> <td>false</td> <td>String</td> <td>Scope for var.</td> </tr> </tbody> </table>	name	dyn*	type	description	dataSource	true	String, javax.sql.DataSource	JNDI Data source	driver	true	String	Class name of the JDBC driver.	url	true	String	JDBC database URL.	user	true	String	JDBC database user	password	true	String	JDBC user password	var	false	String	Name of the ex. sc. variable for the data source specified. Type: String / DataSource.	scope	false	String	Scope for var.
name	dyn*	type	description																															
dataSource	true	String, javax.sql.DataSource	JNDI Data source																															
driver	true	String	Class name of the JDBC driver.																															
url	true	String	JDBC database URL.																															
user	true	String	JDBC database user																															
password	true	String	JDBC user password																															
var	false	String	Name of the ex. sc. variable for the data source specified. Type: String / DataSource.																															
scope	false	String	Scope for var.																															
<b>&lt;sql:param&gt;</b> Sets the values of parameter markers ("?" ) in a SQL statement.	<code>&lt;sql:param value="value"&gt;</code>	<table border="1"> <thead> <tr> <th>name</th> <th>dyn*</th> <th>type</th> <th>description</th> </tr> </thead> <tbody> <tr> <td>value</td> <td>true</td> <td>Object</td> <td>Parameter value.</td> </tr> </tbody> </table>	name	dyn*	type	description	value	true	Object	Parameter value.																								
name	dyn*	type	description																															
value	true	Object	Parameter value.																															
<b>&lt;sql:dateParam&gt;</b> Like <sql:param> for values of type java.util.Date. Subtag of <sql:query> and <sql:update>.	<code>&lt;sql:dateParam value="value" type="{timestamp time date}"&gt;</code>	<table border="1"> <thead> <tr> <th>name</th> <th>dyn*</th> <th>type</th> <th>description</th> </tr> </thead> <tbody> <tr> <td>value</td> <td>true</td> <td>java.util.Date</td> <td>Parameter value for DATE, TIME, or TIMESTAMP column.</td> </tr> <tr> <td>type</td> <td>true</td> <td>String</td> <td>One of "date", "time" or "timestamp".</td> </tr> </tbody> </table>	name	dyn*	type	description	value	true	java.util.Date	Parameter value for DATE, TIME, or TIMESTAMP column.	type	true	String	One of "date", "time" or "timestamp".																				
name	dyn*	type	description																															
value	true	java.util.Date	Parameter value for DATE, TIME, or TIMESTAMP column.																															
type	true	String	One of "date", "time" or "timestamp".																															

\* The "dyn" (dynamic) column title presents the rtxprvalue element defined in a TLD. It captures the dynamic behaviour of an attribute for both library versions.

Action	Description	Syntax	Attributes			
<b>XML Tag Library - XML Core Actions</b>						
<x:out>	Evaluates an XPath expression and outputs the result to the current JspWriter object.	<x:out select="XPathExpression" [escapeXml="true false"]/>	name	dyn*	type	description
<x:parse>	Parses an XML document.	<x:parse xml="XMLDocument" [var="var" [scope="page request session application]] [varDom="var" [scopeDom="scope"]] [systemId="systemId"] [filter="filter"]/>	xml	true	String, Reader	Source XML document to be parsed. The system identifier (URI) for parsing the XML document.
<x:set>	Evaluates an XPath expression and stores the result into a scoped variable.	<x:set select="XPathExpression" var="varName" [scope="page request session application"]/>	select	false	String	XPath expression to be evaluated.
<b>XML Tag Library - XML Flow Control Actions</b>						
<x:if>	Evaluates the XPath expression specified in the select attribute and renders its body content if the expression evaluates to true.	<x:if select="XPathExpression" [var="varName"] [scope="page request session application"]> body content </x:if>	select	false	String	The test condition that tells whether or not the body content should be processed.
<x:when>	Provides the context for mutually exclusive conditional execution.	<x:choose> <x:when select="XPathExpression">body content </x:when> <x:otherwise>conditional block </x:otherwise> </x:choose>	select	false	String	Test condition that determines whether or not the body content should be processed.
<x:forEach>	Evaluates the given XPath expression and repeats its nested body content over the result.	<x:forEach [var="varName"] select="XPathExpression"> body content </x:forEach>	var	false	String	Name of the exported scoped variable for the current item of the iteration.
<b>XML Tag Library - XML Transform Actions</b>						
<x:transform>	Applies an XSLT stylesheet transformation to an XML document.	<x:transform xml="XMLDocument" xslt="XSLTStylesheet" [xmlSystemId="XMLSystemId"] [xsltSystemId="XSLTSystemId"] [var="varName" [scope="page request session application"]] [result="resultObject"]/>	xml	true	String, Reader, javax.xml.transform.Source	Source XML document to be transformed. Type also: org.w3c.dom.Document, or object exported by <x:parse>, <x:set>.
<x:param>	Set transformation parameters. Nested	<x:param name="name" value="value"/>	name	true	String	Name of the transformation parameter.
<b>i18n-capable formatting Tag Library - Internationalization (i18n) Actions</b>						
<fmt:setLocale>	Stores the specified locale in the javax.servlet.jsp.jstl.fmt.locale configuration variable.	<fmt:setLocale value="locale" [variant="variant"] [scope="page request session application"]/>	value	true	String	A String value is interpreted as the printable representation of a locale. Language and country codes must be separated by hyphen ('-') or underscore ('_').
<fmt:bundle>	Creates an i18n localization context to be used by its body content.	<fmt:bundle basename="basename" [prefix="prefix"]> body content </fmt:bundle>	basename	true	String	Resource bundle base name. (fully-qualified)
<fmt:setBundle>	Creates an i18n localization context and stores it in the scoped variable or the localizationContext.	<fmt:setBundle basename="basename" [var="varName" [scope="page request session application"]/>	basename	true	String	Resource bundle base name. (fully-qualified)
<fmt:message>	Looks up a localized message in a resource bundle.	<fmt:message key="messageKey" [bundle="resourceBundle" [var="varName" [scope="page request session application"]]]/>	key	true	String	Message key to be looked up.
<b>i18n-capable formatting Tag Library - Internationalization (i18n) Actions</b>						
<fmt:param>	Supplies a single parameter within a <fmt:message> action.	<fmt:param value="messageParameter"/>	value	true	Object	Argument used for parametric replacement.
<fmt:requestEncoding>	Sets the request's char. encoding.	<fmt:requestEncoding [value="charsetName"]/>	value	true	String	Name of character encoding to be applied when decoding request parameters.
<b>i18n-capable formatting Tag Library - Formatting Actions</b>						
<fmt:timeZone>	Specifies the time zone in which time information is to be formatted or parsed in its body content.	<fmt:timeZone value="timeZone"> body content </fmt:timeZone>	value	true	String, java.util.TimeZone	The time zone. A String value is interpreted as a time zone ID.
<fmt:setTimeZone>	Stores the specified time zone in a scoped variable or the time zone configuration variable.	<fmt:setTimeZone value="timeZone" [var="varName" [scope="page request session application"]/>	value	true	String, java.util.TimeZone	The time zone. A String value is interpreted as a time zone ID.
<fmt:formatNumber>	Formats a numeric value in a locale-sensitive or customized manner as a number, currency, or percentage.	<fmt:formatNumber value="numericValue" [type="number currency percent"] [pattern="customPattern"] [currencyCode="currencyCode"] [currencySymbol="currencySymbol"] [groupingUsed="true false"] [maxIntegerDigits="maxIntegerDigits"] [minIntegerDigits="minIntegerDigits"] [maxFractionDigits="maxFractionDigits"] [minFractionDigits="minFractionDigits"] [var="varName" [scope="page request session application"]/>	value	true	String, Number	Numeric value to be formatted.
<fmt:parseNumber>	Parses the string representation of numbers, currencies, and percentages that were formatted in a locale-sensitive or customized manner.	<fmt:parseNumber value="numericValue" [type="number currency percent"] [pattern="customPattern"] [parseLocale="parseLocale"] [integerOnly="true false"] [var="varName" [scope="page request session application"]/>	value	true	String	String to be parsed.
<fmt:formatDate>	Allows the formatting of dates and times in a locale-sensitive or customized manner.	<fmt:formatDate value="date" [type="time date both"] [dateStyle="default short medium long full"] [timeStyle="default short medium long full"] [pattern="customPattern"] [timeZone="timeZone" [var="varName" [scope="page request session application"]/>	value	true	java.util.Date	Date and/or time to be formatted.
<fmt:parseDate>	Parses the string representation of dates and times that were formatted in a locale-sensitive or customized manner.	<fmt:parseDate value="dateString" [type="time date both"] [dateStyle="default short medium long full"] [timeStyle="default short medium long full"] [pattern="customPattern"] [timeZone="timeZone" [parseLocale="parseLocale"] [var="varName" [scope="page request session application"]/>	value	true	String	Date string to be parsed.

\* The "dyn" (dynamic) column title presents the rtxprvalue element defined in a TLD. It captures the dynamic behaviour of an attribute for both library versions.