



Web Application Framework Tag Library Reference

Sun Java™ Studio Enterprise 7 2004Q4

Sun Microsystems, Inc.
www.sun.com

Part No. 819-0730-10
December 2004, Revision A

Submit comments about this document at: <http://www.sun.com/hwdocs/feedback>

Copyright © 2004 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, U.S.A. All rights reserved.

U.S. Government Rights - Commercial software. Government users are subject to the Sun Microsystems, Inc. standard license agreement and applicable provisions of the FAR and its supplements.

This distribution may include materials developed by third parties. Sun, Sun Microsystems, the Sun logo and Java are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and other countries. Products bearing SPARC trademarks are based upon architecture developed by Sun Microsystems, Inc.

UNIX is a registered trademark in the U.S. and other countries, exclusively licensed through X/Open Company, Ltd.

Products covered by and information contained in this service manual are controlled by U.S. Export Control laws and may be subject to the export or import laws in other countries. Nuclear, missile, chemical biological weapons or nuclear maritime end uses or end users, whether direct or indirect, are strictly prohibited. Export or reexport to countries subject to U.S. embargo or to entities identified on U.S. export exclusion lists, including, but not limited to, the denied persons and specially designated nationals lists is strictly prohibited.

DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

Copyright © 2004 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, Etats-Unis. Tous droits réservés.

Cette distribution peut comprendre des composants développés par des tierces parties. Sun, Sun Microsystems, le logo Sun et Java sont des marques de fabrique ou des marques déposées de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays.

Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux Etats-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc.

UNIX est une marque déposée aux Etats-Unis et dans d'autres pays et licenciée exclusivement par X/Open Company, Ltd.

Les produits qui font l'objet de ce manuel d'entretien et les informations qu'il contient sont regis par la législation américaine en matière de contrôle des exportations et peuvent être soumis au droit d'autres pays dans le domaine des exportations et importations. Les utilisations finales, ou utilisateurs finaux, pour des armes nucléaires, des missiles, des armes biologiques et chimiques ou du nucléaire maritime, directement ou indirectement, sont strictement interdites. Les exportations ou réexportations vers des pays sous embargo des Etats-Unis, ou vers des entités figurant sur les listes d'exclusion d'exportation américaines, y compris, mais de manière non exclusive, la liste de personnes qui font l'objet d'un ordre de ne pas participer, d'une façon directe ou indirecte, aux exportations des produits ou des services qui sont régis par la législation américaine en matière de contrôle des exportations et la liste de ressortissants spécifiquement désignés, sont rigoureusement interdites. LA

DOCUMENTATION EST FOURNIE "EN L'ETAT" ET TOUTES AUTRES CONDITIONS, DECLARATIONS ET GARANTIES EXPRESSES OU TACITES SONT FORMELLEMENT EXCLUES, DANS LA MESURE AUTORISEE PAR LA LOI APPLICABLE, Y COMPRIS NOTAMMENT TOUTE GARANTIE IMPLICITE RELATIVE A LA QUALITE MARCHANDE, A L'APTITUDE A UNE UTILISATION PARTICULIERE OU A L'ABSENCE DE CONTREFAÇON.



Adobe PostScript

Contents

Before You Begin 9

1. Overview 15

Tag Library Overview 15

How the Tag Library Interacts With Web Application Framework 15

Using the Tag Library in Your Application 16

Using the Tag Library in non-HTML JSPs 16

Including JSP Content: Pagelets 17

Tag Overview 19

Context Tags 19

Value Tags 20

Visual (HTML) Tags 20

Tree Tags 22

Miscellaneous Tags 22

Tag Reference 22

2. containerView 23

<jato:containerView> 23

3. tiledView 25

<jato:tiledView> 25

4. **treeView** 27
 <jato:treeView> 27
5. **useViewBean** 29
 <jato:useViewBean> 29
6. **getDisplayFieldValue** 31
 <jato:getDisplayFieldValue> 31
7. **getModelFieldValue** 33
 <jato:getModelFieldValue> 33
8. **setDisplayFieldValue** 35
 <jato:setDisplayFieldValue> 35
9. **setModelFieldValue** 37
 <jato:setModelFieldValue> 37
10. **button** 39
 <jato:button> 39
11. **checkbox** 43
 <jato:checkbox> 43
12. **combobox** 47
 <jato:combobox> 47
13. **fileUpload** 51
 <jato:fileUpload> 51
14. **form** 55
 <jato:form> 55
15. **frameSrc** 59

- `<jato:frameSrc>` 59
- 16. hidden 61**
 - `<jato:hidden>` 61
- 17. href 63**
 - `<jato:href>` 63
- 18. image 67**
 - `<jato:image>` 67
- 19. listBox 71**
 - `<jato:listbox>` 71
- 20. password 75**
 - `<jato:password>` 75
- 21. radioButtons 79**
 - `<jato:radioButtons>` 79
- 22. text 83**
 - `<jato:text>` 83
- 23. textArea 87**
 - `<jato:textArea>` 87
- 24. textField 91**
 - `<jato:textField>` 91
- 25. validatingTextArea 95**
 - `<jato:validatingTextArea>` 95
- 26. validatingTextField 99**
 - `<jato:validatingTextField>` 99

- 27. **content** 103
 <jato:content> 103
- 28. **pagelet** 107
 <jato:pagelet> 107
- 29. **treeNode** 109
 <jato:treeNode> 109
- 30. **treeLevel** 111
 <jato:treeLevel> 111
- 31. **treeNodeHandle** 113
 <jato:treeNodeHandle> 113
- 32. **datasetLocator** 115
 <jatox:datasetLocator> 115
- 33. **datasetNavigator** 117
 <jatox:datasetNavigator> 117
- 34. **dateTimeView** 119
 <jatox:dateTimeView> 119
- 35. **dateView** 121
 <jatox:dateView> 121
- 36. **goToPage** 123
 <jatox:goToPage> 123
- 37. **maskedTextField** 125
 <jatox:maskedTextField> 125
- 38. **menu** 129

<jatox:menu> 129

39. staticBreadCrumb 131

<jatox:staticBreadCrumb> 131

40. timeView 133

<jatox:timeView> 133

Index 135

Before You Begin

The *Web Application Framework Tag Library Reference* is a brief introduction to the tag library, as well as a comprehensive reference to the tags available within the library of the Web Application Framework.

Before You Read This Book

Before starting, you should be familiar with concepts used in building web applications using existing J2EE web technologies, such as servlets and JavaServlet Pages™ (JSP™ pages).

The following resources can provide additional information :

- *Java 2 Platform, Enterprise Edition Specification*
<http://java.sun.com/j2ee/download.html#platformspec>
- *The J2EE Tutorial*
<http://java.sun.com/j2ee/tutorial>
- *Java Servlet Specification Version 2.3*
<http://java.sun.com/products/servlet/download.html#specs>
- *JavaServer Pages Specification Version 1.2*
<http://java.sun.com/products/jsp/download.html#specs>

Note – Sun is not responsible for the availability of third-party Web sites mentioned in this document. Sun does not endorse and is not responsible or liable for any content, advertising, products, or other materials that are available on or through such sites or resources. Sun will not be responsible or liable for any actual or alleged damage or loss caused by or in connection with the use of or reliance on any such content, goods, or services that are available on or through such sites or resources.

Typographic Conventions

Typeface	Meaning	Examples
AaBbCc123	The names of commands, files, and directories; on-screen computer output	Edit your <code>.cvspass</code> file. Use <code>DIR</code> to list all files. Search is complete.
AaBbCc123	What you type, when contrasted with on-screen computer output	> login Password:
<i>AaBbCc123</i>	Book titles, new words or terms, words to be emphasized	Read Chapter 6 in the <i>User's Guide</i> . These are called <i>class</i> options. You <i>must</i> save your changes.
<code>AaBbCc123</code>	Command-line variable; replace with a real name or value	To delete a file, type DEL <i>filename</i> .

Related Documentation

Java Studio Enterprise documentation includes books and tutorials delivered in Acrobat Reader (PDF) format, release notes, online help, and tutorials delivered in HTML format.

Documentation Available Online

The documents described in this section are available from the `docs.sun.com`SM web site and from the Documentation link from the Sun Java Studio Enterprise Developers Source portal (<http://developers.sun.com/jsenterprise>).

The `docs.sun.com` web site (<http://docs.sun.com>) enables you to read, print, and buy Sun Microsystems manuals through the Internet.

- *Sun Java Studio Enterprise 7 Release Notes* - part no. 819-0905-10
Describes last-minute release changes and technical notes.

- *Sun Java Studio Enterprise 7 Installation Guide* (PDF format) - part no. 817-7971-10
 Describes how to install the Sun Java Studio Enterprise 7 integrated development environment (IDE) on each supported platform and includes other pertinent information, such as system requirements, upgrade instructions, server information, command-line switches, installed subdirectories, database integration, and information on how to use the Update Center.
- *Building J2EE Applications* - part no. 819-0819-10
 Describes how to assemble EJB modules and web modules into a J2EE application and how to deploy and run a J2EE application.
- Web Application Framework documentation (PDF format)
 - *Web Application Framework Component Author's Guide* - part no. 819-0724-10
 Describes the Web Application Framework component architecture and the process to design, create, and distribute new components.
 - *Web Application Framework Component Reference Guide* - part no. 819-0725-10
 Describes the components available in the Web Application Framework Library.
 - *Web Application Framework Overview* - part no. 819-0726-10
 Introduces the Web Application Framework and what it is, how it works, and what sets it apart from other application frameworks.
 - *Web Application Framework Tutorial*- part no. 819-0727-10
 Introduces the mechanics and techniques to build a web application using the Web Application Framework tools.
 - *Web Application Framework Developer's Guide* - part no. 819-0728-10
 Provides the steps to create and use application components that can be assembled to develop an application using the Web Application Framework and explains how to deploy the application in most J2EE containers.
 - *Web Application Framework IDE Guide* - part no. 819-0729-10
 Describes the various parts of the Sun Java Studio Enterprise 7 2004Q4 IDE and emphasizes the use of the visual tools for developing a Web Application Framework application.
 - *Web Application Framework Tag Library Reference* - part no. 819-0730-10
 Gives a brief introduction to the Web Application Framework tag library, as well as a comprehensive reference to the tags available within the library.

Tutorials

Sun Java Studio Enterprise 7 tutorials help you understand the features of the IDE. Each tutorial provides techniques and code samples that you can use or modify in developing more substantial applications. All tutorials illustrate deployment with Sun Java System Application Server.

All tutorials are available from the Tutorials and Code Camps link off the Developers Source portal, which you can access from within the IDE by choosing Help > Examples and Tutorials.

- **QuickStart guides** provide an introduction to the Sun Java Studio IDE. Start with a QuickStart tutorial if you are either new to the Sun Java Studio IDE or want a quick introduction to a particular feature. These tutorials describe how to develop simple web and J2EE applications, generate web services, and how to get started with UML modeling and Refactoring. QuickStarts take minutes to complete.
- **Tutorials** focus on a single feature of the Sun Java Studio IDE. Try these if you are interested in the details of a particular feature. Some tutorials build an application from the ground up, while others build on provided source files, depending on the focus of the example. You can complete a tutorial in an hour or less.
- **Narrated Tutorials** use video to illustrate a feature or technique. Try a narrated tutorial for a visual overview of the IDE or an in-depth presentation of a particular feature. You can complete a narrated tutorial in a few minutes. You can also start and stop a narrated tutorial at any point you wish.

Online Help

Online help is available in the Sun Java Studio Enterprise 7 IDE. You can open help by pressing the help key (F1 in Microsoft Windows environments, Help key in the Solaris environment), or by choosing Help → Contents. Either action displays a list of help topics and a search facility.

Documentation in Accessible Formats

The documentation is provided in accessible formats that are readable by assistive technologies for users with disabilities. You can find accessible versions of documentation as described in the following table.

Type of Documentation	Format and Location of Accessible Version
Books and tutorials	HTML at http://docs.sun.com
Tutorials	HTML at the Examples and Code Camps link from the Developers Source portal at http://developers.sun.com/jsenterprise
Release notes	HTML at http://docs.sun.com

Sun Welcomes Your Comments

Sun is interested in improving its documentation and welcomes your comments and suggestions. Email your comments to Sun at this address:

`docfeedback@sun.com`

Please include the book's title (*Web Application Framework Tag Library Reference*) and its part number (819-0730-10) in the subject line of your email.

Overview

Tag Library Overview

The Web Application Framework tag library is a powerful tool for rendering dynamic JSP output when used in conjunction with the Web Application Framework. This document is intended to be a brief introduction to the tag library, as well as a comprehensive reference to the tags available within the library. See the supplemental Web Application Framework documentation for more complete information on writing a Web Application Framework application.

How the Tag Library Interacts With Web Application Framework

Web Application Framework is architected around the idea of arbitrarily nested View objects. Some of these views are `ContainerViews`, containing other `Views` (including other `ContainerViews`), and others are `DisplayFields`, with a notion of a value which can be accessed.

Each JSP has a single `ViewBean` associated with it. This object is also referred to as the *root view* of the JSP, as it is the top-level view object containing all other views used in that JSP. Each tag in the Web Application Framework tag library is a reference to one of these contained `Views`. The objects the tags refer to are called *peer views*, or simply *peers*. Each tag a developer uses in his or her JSP must refer to an existing peer object in that JSP's `ViewBean` or one of the `ViewBean`'s child views.

The bulk of the tags in the Web Application Framework tag library establish either a context within which other peers can be easily referenced, or they actually reference a peer view to render a visualHTML fragment such as a string value, a text field, or a list box. When a JSP using the tag library is rendered, it contains visual HTML form controls or other content derived from the interactions with each peer from within each tag. Thus, tags work in conjunction with a set of predefined peers to render a dynamic HTML page from a JSP.

Using the Tag Library in Your Application

To use the Web Application Framework tag library within your Web application, each JSP using the tag library must reference the Web Application Framework Tag Library Descriptor, or TLD. This is accomplished by including the following directive at the top of your JSP:

```
<%@taglib uri="/WEB-INF/jato.tld" prefix="jato"%>
```

(Note: If you are using the Web Application Framework IDE toolset, this is automatically managed for you.) The location of the TLD specified by the `uri` attribute, as well as the tag prefix name, are arbitrary. However, it is recommended that you use the above values and including the Web Application Framework TLD file in the root of your application WAR file's `WEB-INF` directory. Within this document, assume the `jato` tag prefix.

Using the Tag Library in non-HTML JSPs

The Web Application Framework tag library contains both visual and non-visual tags. The visual tags render HTML 4.01 compliant markup, but non-visual tags have no associated markup, or allow developers to specify the markup they wish to render. Non-visual tags fall into several categories, but generally they represent the structure of your Web Application Framework `View` hierarchy in an abstract way. Because this hierarchy remains the same across content types, you can develop non-HTML markup-based JSPs using Web Application Framework, with the only additional requirement being able to specify visual markup for the appropriate content type.

The existing `getDisplayFieldValue` tag provides a way to easily inline dynamic display field data without any associated markup, while still firing display events and working with the current `ContainerView` context. This allows developers to add dynamic values to static markup declared in a JSP. Use of this tag is an easy way to develop non-HTML (or even HTML) pages without creating new JSP tags. However, in general, creating new JSP tags is the easier approach to render non-HTML JSPs.

Beginning with Sun ONE Application Framework 2.0, component libraries gave developers the ability to easily package and deliver components that automatically manage their own tags for multiple content types, whether they are the standard Web Application Framework tag library tags or custom-built tags. Furthermore, beginning with Sun ONE Application Framework 2.1, Web Application Framework's `taglib` package has been reimplemented to provide much easier ways for developers to create new custom tags that render View components in arbitrary ways. Using these tools, it is easy for developers to either find an existing component library that handles visual markup of the content type they require, or create a component library that renders components using non-HTML content types. See the documentation for the `taglib` package for more information on creating your own visual tags.

Including JSP Content: Pagelets

JSPs allows inclusion of outside content in the currently rendering page (see the JSP 1.1 specification for full details). This allows developers to modularize JSP content and then combine this content into compound documents. This capability has interesting implications for Web Application Framework applications.

There are two ways to include content in a JSP: *translation-time includes*, and *request-time includes*. A *translation-time include* pulls an outside file's content into the enclosing JSP before it is translated into a servlet by the container. The benefit of this approach is that it performs well, and the included content acts just like it were part of the enclosing page. The downside is that it is not at all dynamic—the included content is statically enclosed and cannot be chosen or replaced at runtime.

A *request-time include* does a `RequestDispatcher.include()` operation on a target URL/JSP, dynamically inlining the target content into the enclosing page. This include is done every time the enclosing page is rendered. The benefit of this approach is that the included file can be chosen dynamically. The downside is that there is some performance overhead in dispatching the request to the included resource.

It is generally advocated *translation-time includes* of JSP fragments in Web Application Framework applications because they perform well, and can include just the views/fields appropriate for the scope of the inclusion. This approach lets a component writer compose a fragment of a JSP page or Web Application Framework content and conveniently reuse it in several pages. But, as noted above, the choice of which fragment to include in the enclosing page is determined at translation time and is then fixed, significantly limiting the dynamism of the rendered page.

Request-time includes in Web Application Framework applications would be a nice way to get around the static nature of *translation-time includes*, except that in prior versions of Web Application Framework, there was not any way to request-time include anything but a full `ViewBean` and its contents. This meant that developers

had to basically include a root view inside another root view, which was both confusing and not always guaranteed to work correctly—there should only be one root view per logical page. The other downside to that approach is that the included content lost the scope of the enclosing JSP's container view; the scope of objects accessed in the included page is limited by the immediately enclosing ViewBean, meaning that the page cannot necessarily be arbitrarily included in other JSPs like a translation-time include would allow.

Since JSun ONE Application Framework 1.2, a solution has been offered to these limitations in the form of *pagelets*, accompanied by a `<jato:pagelet>` tag. In Web Application Framework, a pagelet is distinguished from an arbitrary non-pagelet JSP fragment by the fact that it can be seamlessly request-time included in a Web Application Framework JSP page. This is accomplished by the use of the `<jato:pagelet>` tag, which *connects* the view tags in the included JSP to the enclosing JSP's container scope, so that an included pagelet uses the enclosing JSP's current container view as its container view scope.

For example, assume you have the following JSP and pagelet:

EnclosingPage.jsp

```
<%@page info="E0130" language="java"%>
<%@taglib uri="/WEB-INF/jato.tld" prefix="jato"%>

<jato:useViewBean ...>

<jato:containerView name="foo">
    ...
    <jsp:include page="MyPagelet.jsp"/>
    ...
</jato:containerView>

</jato:useViewBean>
```

MyPagelet.jsp

```
<%@page info="MyPagelet" language="java"%>
<%@taglib uri="/WEB-INF/jato.tld" prefix="jato"%>

<jato:pagelet>
    <jato:combobox name="month"/>...
</jato:pagelet>
```

The “month” field sees container view “foo” as its enclosing container, and thus can be declared a proper child of “foo”. The `<jato:pagelet>` tag acts as a proxy to the enclosing page’s container view tag, connecting the two JSPs as though they were part of the same rendering, when in fact they are two separate renderings.

Even more interesting, developers can use the display events in Web Application Framework to dynamically choose which pagelet to include at request time (by using the `PageContext.include()` method in the event). This technique is demonstrated in the Web Application Framework sample application.

The following matrix summarizes the types of inclusion techniques developers can use in Web Application Framework:

JSP Fragment Type	Translation-time Include	Runtime Include
Pagelet	Yes*	Yes
Fragment (no ViewBean)	Yes	No
Fragment (with ViewBean)	No	Yes

* Technically, this works; however, there is no reason other than consistency to use the `<jato:pagelet>` tag in this situation, since the included JSP will simply become part of the enclosing JSP. The pagelet tag is written to do nothing in this situation.

Tag Overview

The tags in the Web Application Framework tag library fall into three basic groups: context tags, value tags, and visual tags. See the following sections for more information.

Context Tags

These tags are oriented toward declaring a peer view’s scope, within which the referenced object will define the current context for other embedded tags. Specifically, this means that each of these tags declare usage of a `ContainerView` (or a specialization of `ContainerView`). Each such declaration establishes a name scope in which child Views can be referred to by their short, non-qualified names. `ContainerView` contexts may be nested, and do not directly result in any rendered HTML.

Tag Name	Alternate Names	Description
containerView		Declare usage of <code>ContainerView</code>
tiledView		Declare usage of a <code>TiledView</code>
treeView		Declare usage of a <code>TreeView</code>
useViewBean	<code>viewBean</code>	Declare usage of a <code>ViewBean</code>

Value Tags

These tags allow for direct manipulation of `DisplayField` and `Model` values. Developers may embed these tags in scriptlets, expressions, or HTML within the JSP. Unlike the visual tags described below, these tags do not render an HTML form control; rather, these tags manipulate values directly. The value tags can be used to get/set values on any of the `DisplayField` subtypes, or any `Model` instance available in the application. These tags must appear with the scope of a context tag.

Tag Name	Description
getDisplayFieldValue	Retrieve a value from a <code>DisplayField</code>
getModelFieldValue	Retrieve a value from a <code>Model</code> field
setDisplayFieldValue	Set the value of a <code>DisplayField</code>
setModelFieldValue	Set the value of a <code>Model</code> field

Visual (HTML) Tags

These tags use a a combination of tag attributes and values from `DisplayField` peers to render HTML form controls. These tags must also appear with the scope of a context tag.

Tag Name	Alternate Names	Description
button		Render a button control
checkbox	<code>checkBox</code>	Render a checkbox control
combobox	<code>comboBox</code>	Render a combobox control
fileUpload		Render a file upload element
form		Define an HTML form

Tag Name	Alternate Names	Description
frameSrc		Render a frame source element
hidden		Render a hidden element
href		Render a hyperlink element
image		Render an image
listbox	listBox	Render a list control
password		Render a password control
radioButtons		Render a group of radio button controls
text	staticText	Render arbitrary text
textArea		Render a multi-line text area control
textField		Render a single line text control
validatingTextArea		Render a multi-line text area control
validatingTextField		Render a single line text control
datasetLocator		Renders a static text display of the current location and number of records currently being displayed
datasetNavigator		Renders four dataset navigation actions: First, Previous, Next and Last.
dateTimeView		Render month, day, year range, hour, minute and am/pm options
dateView		Renders month, day and year range options
goToPage		Render a visible, clickable href element
maskedTextField		Render a text field element
menu		Render a menu element
staticBreadCrumb		Render bread crumb element
timeView		Renders 3 combo boxes (hour, minute and am/pm options)

Tree Tags

These tags can be combined with the **treeTag** to specify tree rendering logic.

Tag Name	Description
treeLevel	Denote a content section that will be rendered “level-times” for matching tree levels
treeNode	Denote a content section that will be rendered for matching tree nodes
treeNodeHandle	Renders a expand/collapse node control

Miscellaneous Tags

These tags provide additional features.

Tag Name	Description
content	Denote a content section and associate it with a display event
pagelet	Allows Web Application Framework tags in an included JSP fragment to inherit the enclosing page’s container view scope, and thus be included at request-time

Tag Reference

Note Legend:

*RTE**Expr* = Attribute value can be a runtime-evaluated expression

Req = Required attribute

containerView

<jato:containerView>

A containerView declares use of a `ContainerView` peer instance available from the view bean or one of its child `ContainerViews`. All `View` or `DisplayField` references enclosed by this tag are assumed to reference children of the associated `ContainerView` instance, and their name attributes are resolved within the namespace of the parent container. This tag is only valid when nested inside a `useViewBean`, `tiledView`, or another `containerView` tag.

The use of the `containerView` tag causes a new scripting variable to be defined within the scope of the enclosed tag body. The name of this scripting variable is the value of the `name` attribute, or the name of the `id` attribute if specified. When processing a `containerView` tag, the implicit scripting variable `currentContainerView` is set to the `ContainerView` instance referred to by this tag.

The `useViewBean` and `tiledView` tags are special cases of the `containerView` tag, adding additional behavior to the basic behavior defined by this tag.

Example:

```
<jato:containerView name="header">  
...  
</jato:containerView>
```

Attribute Name	Description	Notes
id	<p>Specifies a name used to identify the implicit scripting variable that will refer to the <code>ContainerView</code> instance. The name specified is case sensitive and shall conform to the current scripting language variable-naming conventions.</p> <p>If not specified, the scripting variable is named with the value specified in the "name" attribute.</p>	
name	<p>The name of the <code>ContainerView</code> peer instance as declared in its parent (a <code>ContainerView</code>, <code>TiledView</code>, or <code>ViewBean</code>).</p> <p>This name may be a qualified view path, using forward slashes ("/") as delimiters. All components in the path except the last must refer to a <code>ContainerView</code> or a derivative of <code>ContainerView</code> (such as <code>TiledView</code>). Both relative and absolute paths are possible. If a name path begins with a forward slash, the name is assumed to be relative to the root view (the <code>ViewBean</code>). If the path does not begin with a forward slash, the name is assumed to refer to a child relative to the current container. Two dots ("..") may be used to refer to the container that is the parent of the current container.</p> <p>Examples:</p> <ul style="list-style-type: none"> <code>/header/orderList/customerName</code> (absolute from root view) <code>orderList/customerName</code> (relative to current container) <code>../footer/orderList/customerName</code> (relative to parent) 	Req, RTExpr
type	<p>If specified, defines the type of the scripting variable defined within the scope of this tag. This allows the type of the scripting variable to be distinct from, but related to, that of the implementation class specified. The type is required to be either the specific <code>ContainerView</code> class itself, a superclass of the class, or an interface implemented by the class specified. The object referenced is required to be of this type, otherwise a <code>java.lang.ClassCastException</code> shall occur at request time when the assignment of the object referenced to the scripting variable is attempted.</p> <p>If unspecified, the scripting variable is declared to be of type <code><JATO package>.view.ContainerView</code></p>	

tiledView

<jato:tiledView>

A tiledView tag declares a block of body content which is evaluated and rendered repeatedly during the display of the JSP. The tiledView must be associated with a TiledView object available from the view bean or one of its child ContainerViews. Callbacks to the TiledView object control the display of the tiledView body content. A tiledView is particularly useful for displaying multiple rows an HTML table; however, a tiledView's usage is not restricted to dynamic HTML row generation.

The use of the tiledView tag causes a new scripting variable to be defined within the scope of the enclosed tag body. The name of this scripting variable is the value of the name attribute, or the name of the id attribute if specified. When processing a tiledView tag, the implicit scripting variables currentContainerView and currentTiledView are set to the TiledView instance referred to by this tag.

A tiledView tag can be considered a special case of containerView tag. This tag is only valid when nested inside a useViewBean, containerView, or another tiledView tag.

Example:

```
<jato:tiledView name="orderItems" maxTiles="10">  
...  
</jato:tiledView>
```

Attribute Name	Description	Notes
id	<p>Specifies a name used to identify the implicit scripting variable that will refer to the <code>TiledView</code> instance. The name specified is case sensitive and shall conform to the current scripting language variable-naming conventions.</p> <p>If not specified, the scripting variable is named with the value specified in the <code>name</code> attribute.</p>	
maxTiles	<p>Valid values: positive number</p> <p>Specifies the maximum number of times this <code>TiledView</code> will render its body content. This value will override the corresponding value in the <code>TiledView</code> object. If not specified, <code>maxTiles</code> is determined by the <code>TiledView</code> peer</p>	RTEExpr
name	<p>The name of the <code>TiledView</code> peer instance as declared in its parent (a <code>ContainerView</code>, <code>TiledView</code>, or <code>ViewBean</code>).</p> <p>This name may be a qualified view path, using forward slashes ("/") as delimiters. All components in the path except the last must refer to a <code>ContainerView</code> or a derivative of <code>ContainerView</code> (such as <code>TiledView</code>). Both relative and absolute paths are possible. If a name path begins with a forward slash, the name is assumed to be relative to the root view (the <code>ViewBean</code>). If the path does not begin with a forward slash, the name is assumed to refer to a child relative to the current container. Two dots ("..") may be used to refer to the container that is the parent of the current container.</p> <p>Examples:</p> <ul style="list-style-type: none"> <code>/header/orderList/customerName</code> (absolute from root view) <code>orderList/customerName</code> (relative to current container) <code>../footer/orderList/customerName</code> (relative to parent) 	Req, RTEExpr
type	<p>If specified, defines the type of the scripting variable defined within the scope of this tag. This allows the type of the scripting variable to be distinct from, but related to, that of the implementation class specified. The type is required to be either the specific <code>TiledView</code> class itself, a superclass of the class, or an interface implemented by the class specified. The object referenced is required to be of this type, otherwise a <code>java.lang.ClassCastException</code> shall occur at request time when the assignment of the object referenced to the scripting variable is attempted.</p> <p>If unspecified, the scripting variable is declared to be of type <code><JATO package>.view.TiledView</code></p>	
reset	<p>Valid values: "true" or "false"</p> <p>A value of "true" specifies that the tag will call the <code>TiledView</code>'s <code>resetTileIndex()</code> method immediately after invoking the <code>beginDisplay()</code> event. This will ensure that the view's primary model is properly reset for display. If not specified, a value of "true" is assumed. Note, this is a change from previous versions of Web Application Framework. Previous versions never performed a reset, leaving this behavior up to the developer. It is expected that this change will provide more ease of use, while providing backward compatibility for the rare application that relied upon the previous default behavior.</p>	

treeView

<jato:treeView>

A `treeView` tag declares a block of body content which is evaluated and rendered repeatedly during the display of the JSP. The `treeView` must be associated with a `TreeView` object available from the view bean or one of its child `ContainerViews`. Callbacks to the `TreeView` object control the display of the `treeView` body content. A `treeView` is particularly useful for displaying hierarchical data. Typically, though not by definition, a **treeView** tag will contain one or more `treeNode` tags.

The use of the `treeView` tag causes a new scripting variable to be defined within the scope of the enclosed tag body. The name of this scripting variable is the value of the `name` attribute, or the name of the `id` attribute if specified. When processing a `treeView` tag, the implicit scripting variables `currentContainerView` and `currentTreeView` are set to the `TreeView` instance referred to by this tag.

A `treeView` tag can be considered a special case of `containerView` tag. This tag is only valid when nested inside a `useViewBean`, `containerView`, `tilableView`, or another `treeView` tag.

See also `treeNode`, `treeLevel`, `treeNodeHandle`

Example:

```
<jato:treeView name="treeMenu">
...
</jato:treeView>
```

Attribute Name	Description	Notes
id	<p>Specifies a name used to identify the implicit scripting variable that will refer to the <code>ContainerView</code> instance. The name specified is case sensitive and shall conform to the current scripting language variable-naming conventions.</p> <p>If not specified, the scripting variable is named with the value specified in the "name" attribute.</p>	
name	<p>The name of the <code>TreeView</code> peer instance as declared in its parent container view. This name may be a qualified view path, using forward slashes ("/") as delimiters. All components in the path except the last must refer to a <code>ContainerView</code> or a derivative of <code>ContainerView</code> (such as <code>TiledView</code>). Both relative and absolute paths are possible. If a name path begins with a forward slash, the name is assumed to be relative to the root view (the <code>ViewBean</code>). If the path does not begin with a forward slash, the name is assumed to refer to a child relative to the current container. Two dots ("..") may be used to refer to the container that is the parent of the current container.</p> <p>Examples:</p> <ul style="list-style-type: none"> <code>/header/orderList/customerName</code> (absolute from root view) <code>orderList/customerName</code> (relative to current container) <code>../footer/orderList/customerName</code> (relative to parent) 	Req, RTExpr
type	<p>If specified, defines the type of the scripting variable defined within the scope of this tag. This allows the type of the scripting variable to be distinct from, but related to, that of the implementation class specified. The type is required to be either the specific <code>ContainerView</code> class itself, a superclass of the class, or an interface implemented by the class specified. The object referenced is required to be of this type, otherwise a <code>java.lang.ClassCastException</code> shall occur at request time when the assignment of the object referenced to the scripting variable is attempted.</p> <p>If unspecified, the scripting variable is declared to be of type <code><JATO package>.view.TreeView</code>.</p>	
reset	<p>Valid values: "true" or "false"</p> <p>A value of "true" specifies that the tag will call the <code>TreeView</code>'s <code>resetNodeLocation()</code> method immediately after invoking the <code>beginDisplay()</code> event. This will ensure that the view's primary model is properly reset for display. If not specified, a value of "true" is assumed. Note, this is a change from previous versions of Web Application Framework. Previous versions never performed a reset, leaving this behavior up to the developer. It is expected that this change will provide more ease of use, while providing backward compatibility for the rare application that relied upon the previous default behavior.</p>	

useViewBean

<jato:useViewBean>

A `useViewBean` tag establishes the root `ViewBean` peer used for rendering the current JSP. Only one `useViewBean` tag is permitted within a JSP, and it must enclose all other Web Application Framework tags on that page. The `useViewBean` tag establishes a root name space for a given JSP and acts as controller during rendering of the page.

The use of the `useViewBean` tag causes a new scripting variable called `viewBean` to be defined within the scope of the enclosed tag body. When processing a `useViewBean` tag, the implicit scripting variables `currentContainerView` and `currentTiledView` are established. The value of the `currentContainerView` variable is set to the `ViewBean` instance referred to by this tag.

A `useViewBean` tag can be considered a special case of `containerView` tag.

Example:

```
<jato:useViewBean className="com.mycomp.myapp.MyViewBean">  
...  
</jato:useViewBean>
```

Attribute Name	Description	Notes
className	The fully qualified name of the <code>ViewBean</code> peer class. The class name is case sensitive.	Req
type	If specified, defines the type of the <code>viewBean</code> scripting variable defined within the scope of this tag. This allows the type of the scripting variable to be distinct from, but related to, that of the implementation class specified. The type is required to be either the specific <code>ViewBean</code> class itself, a superclass of the class, or an interface implemented by the class specified. The object referenced is required to be of this type, otherwise a <code>java.lang.ClassCastException</code> shall occur at request time when the assignment of the object referenced to the scripting variable is attempted. If unspecified, the scripting variable is declared to be of the type specified by the <code>className</code> attribute.	

getDisplayFieldValue

<jato:getDisplayFieldValue>

This tag invokes the `DisplayField.getValue()` method on the named display field and inlines the resulting value in the HTML output stream. This tag may be embedded within any arbitrary HTML to manually construct HTML form controls or provide values to other tag attributes that allow runtime value expressions.

This tag is only valid when enclosed by a `useViewBean`, `containerView`, or `tilableView` tag. This tag may not contain any body content.

Example:

```
<jato:getDisplayFieldValue name="salutation" defaultValue="Mr." />
```

Attribute Name	Description	Notes
defaultValue	The value to display if the <code>DisplayField</code> peer value is null	RTEExpr
escape	<p>Valid values: "true" or "false"</p> <p>A value of "true" specifies that characters with special meaning in HTML (such as < and >) will be changed to the equivalent entity representation before being output.</p> <p>If not specified, a value of "true" is assumed.</p>	
name	<p>The name of the <code>DisplayField</code> peer. This peer must be a child of the current parent <code>ContainerView</code>, <code>TiledView</code>, or <code>ViewBean</code>. The <code>DisplayField</code> name is resolved relative to the current parent view.</p> <p>This name may be a qualified view path, using forward slashes ("/") as delimiters. All components in the path except the last must refer to a <code>ContainerView</code> or a derivative of <code>ContainerView</code> (such as <code>TiledView</code>). Both relative and absolute paths are possible. If a name path begins with a forward slash, the name is assumed to be relative to the root view (the <code>ViewBean</code>). If the path does not begin with a forward slash, the name is assumed to refer to a child relative to the current container. Two dots ("..") may be used to refer to the container that is the parent of the current container.</p> <p>Examples:</p> <ul style="list-style-type: none"> <code>/header/orderList/customerName</code> (absolute from root view) <code>orderList/customerName</code> (relative to current container) <code>../footer/orderList/customerName</code> (relative to parent) 	Req, RTEExpr

getModelFieldValue

<jato:getModelFieldValue>

This tag invokes the `Model.getValue()` method on the named model instance and inlines the resulting value in the HTML output stream. This tag may be embedded within any arbitrary HTML to manually construct HTML form controls or provide values to other tag attributes that allow runtime value expressions.

This tag is only valid when enclosed by a `useViewBean`, `containerView`, or `tilableView` tag. This tag may not contain any body content.

Example:

```
<jato:getModelFieldValue modelClass=  
"com.mycompany.myapp.MyModel" name="salutation" defaultValue=  
"Mr." />
```

Attribute Name	Description	Notes
defaultValue	The value to display if the <code>Model</code> field value is <code>null</code>	RTEExpr
escape	Valid values: "true" or "false" A value of "true" specifies that characters with special meaning in HTML (such as < and >) will be changed to the equivalent entity representation before being output. If not specified, a value of "true" is assumed.	
lookInSession	Valid values: "true" or "false" A value of "true" specifies that the tag will specify to the <code>ModelManager</code> that it should look in the session for the model named by the <code>modelName</code> attribute. The <code>modelName</code> attribute must be specified if this attribute is specified. If not specified, "false" is assumed.	RTEExpr
modelClass	The fully-qualified model class interface or implementation name (see the documentation for <code><JATO package>.ModelManager</code> for more information on which class to specify). The instance returned from the <code>ModelManager</code> in response to this class name must implement the <code><JATO package>.model.Model</code> interface.	Req, RTEExpr
modelName	If specified, attempts to obtain the model named by the value of this attribute. If this attribute is not specified, a default model instance will be used. This attribute is mandatory if the <code>lookInSession</code> attribute is set to "true". If the <code>lookInSession</code> attribute is "true", the <code>ModelManager</code> will attempt to obtain the model from the session using the name specified by this attribute.	RTEExpr
name	The name of the model field.	Req, RTEExpr

setDisplayFieldValue

<jato:setDisplayFieldValue>

This tag invokes the `DisplayField.setValue(Object)` method with the provided value. This tag does not render any HTML, and may be embedded within any arbitrary HTML on the page.

This tag is only valid when enclosed by a `useViewBean`, `containerView`, or `tilableView` tag. This tag may not contain any body content.

Example:

```
<jato:setDisplayFieldValue name="numItems" value="8"
valueType="int" />
```

Attribute Name	Description	Notes
name	<p>The name of the <code>DisplayField</code> peer. This peer must be a child of the current parent <code>ContainerView</code>, <code>TiledView</code>, or <code>ViewBean</code>. The <code>DisplayField</code> name is resolved relative to the current parent view.</p> <p>The view peer must be of type <code><JATO package>.view.DisplayField</code></p> <p>This name may be a qualified view path, using forward slashes ("/") as delimiters. All components in the path except the last must refer to a <code>ContainerView</code> or a derivative of <code>ContainerView</code> (such as <code>TiledView</code>). Both relative and absolute paths are possible. If a name path begins with a forward slash, the name is assumed to be relative to the root view (the <code>ViewBean</code>). If the path does not begin with a forward slash, the name is assumed to refer to a child relative to the current container. Two dots ("..") may be used to refer to the container that is the parent of the current container.</p> <p>Examples:</p> <pre> /header/orderList/customerName (absolute from root view) orderList/customerName (relative to current container) ../footer/orderList/customerName (relative to parent) </pre>	Req, RTEExpr
value	<p>The value that will be passed into the <code>DisplayField.setValue(Object)</code> method of the specified <code>DisplayField</code> peer.</p>	Req RTEExpr
valueType	<p>Valid values: "string", "int", "long", "float", "double", "short", "bigdecimal", "boolean", "byte", "char", "sqldate", "sqltime", "sqltimestamp"</p> <p>The type to which the string representation of the value attribute will be converted before being set on the target <code>DisplayField</code>. The type conversion from the string representation must be legal or an exception will be thrown.</p> <p>If this attribute is omitted, the set value will be of type <code>String</code>.</p>	

setModelFieldValue

<jato:setModelFieldValue>

This tag invokes the `Model.setValue(String, Object)` method with the provided value. This tag does not render any HTML, and may be embedded within any arbitrary HTML on the page.

This tag is only valid when enclosed by a `useViewBean`, `containerView`, or `tiledView` tag. This tag may not contain any body content.

Example:

```
<jato:setModelFieldValue modelClass=
  "com.mycompany.myapp.MyModel" name="numItems" value="8"
  valueType="int" />
```

Attribute Name	Description	Notes
lookInSession	Valid values: "true" or "false" A value of "true" specifies that the tag will specify to the <code>ModelManager</code> that it should look in the session for the model named by the <code>modelName</code> attribute. The <code>modelName</code> attribute must be specified if this attribute is specified. If not specified, "false" is assumed.	RTEExpr
name	The name of the <code>DisplayField</code> peer. This peer must be a child of the current parent <code>ContainerView</code> , <code>TiledView</code> , or <code>ViewBean</code> . The <code>DisplayField</code> name is resolved relative to the current parent view. The view peer must be of type <code><JATO package>.view.DisplayField</code>	Req, RTEExpr
value	The value that will be passed into the <code>DisplayField.setValue(Object)</code> method of the specified <code>DisplayField</code> peer.	Req RTEExpr

Attribute Name	Description	Notes
modelClass	The fully-qualified model class interface or implementation name (see the documentation for <code><JATO package>.ModelManager</code> for more information on which class to specify). The instance returned from the <code>ModelManager</code> in response to this class name must implement the <code><JATO package>.model.Model</code> interface.	Req, RTEExpr
modelName	If specified, attempts to obtain the model named by the value of this attribute. If this attribute is not specified, a default model instance will be used. This attribute is mandatory if the <code>lookInSession</code> attribute is set to "true". If the <code>lookInSession</code> attribute is "true", the <code>ModelManager</code> will attempt to obtain the model from the session using the name specified by this attribute.	RTEExpr
valueType	Valid values: "string", "int", "long", "float", "double", "short", "bigdecimal", "Boolean", "byte", "char", "sqldate", "sqltime", "sqltimestamp" The type to which the string representation of the value attribute will be converted before being set on the target <code>DisplayField</code> . The type conversion from the string representation must be legal or an exception will be thrown. If this attribute is omitted, the set value will be of type <code>String</code> .	

button

`<jato:button>`

Renders an HTML `<input type="submit">` element.

This tag is only valid when enclosed by both an HTML `<form>` element and a `useViewBean`, `containerView`, or `tiledView` tag. This tag may not contain any body content.

Example:

```
<jato:button name="processForm"/>
```

The above would be rendered into the following HTML:

```
<input type="submit" name="PageFoo.processForm" value="Process">
```

Attribute Name	Description	Notes
accessKey	Assigns an access key to the rendered HTML element. In general, the access key should be included in label text or wherever the access key is to apply.	RTEExpr
defaultValue	The value to use, if the <code>DisplayField</code> view peer's value is <code>null</code> . For a button tag, "value" refers to the value that will be rendered in the HTML element's "value" attribute	RTEExpr
disabled	Valid values: "true" or "false" Disables the HTML control for user input. <ul style="list-style-type: none"> • Disabled controls do not receive focus • Disabled controls are skipped in tabbing navigation If not specified, "false" is assumed.	RTEExpr
elementId	Assigns a name to an element. This name must be unique in an HTML document. <i>Please note, this corresponds to the HTML "id" attribute. It has no specific meaning to the Web Application Framework.</i> The id attribute has several roles in HTML: <ul style="list-style-type: none"> • As a style sheet selector • As a target anchor for hypertext links • As a means to reference a particular element from a script 	RTEExpr
extraHtml	Arbitrary HTML that will be rendered just prior to the closing ">" of this HTML element	RTEExpr
name	The name of the <code>DisplayField</code> peer. This peer must be a child of the current parent <code>ContainerView</code> , <code>TiledView</code> , or <code>ViewBean</code> . The <code>DisplayField</code> name is resolved relative to the current parent view. The view peer must be of type <code><JATO package>.view.CommandField</code> This name may be a qualified view path, using forward slashes ("/") as delimiters. All components in the path except the last must refer to a <code>ContainerView</code> or a derivative of <code>ContainerView</code> (such as <code>TiledView</code>). Both relative and absolute paths are possible. If a name path begins with a forward slash, the name is assumed to be relative to the root view (the <code>ViewBean</code>). If the path does not begin with a forward slash, the name is assumed to refer to a child relative to the current container. Two dots ("..") may be used to refer to the container that is the parent of the current container. Examples: <pre> /header/orderList/customerName (absolute from root view) orderList/customerName (relative to current container) ../footer/orderList/customerName (relative to parent) </pre>	Req
src	An image URL if this button is to be rendered as an image button.	RTEExpr
style	CSS styles to be applied to this HTML element.	RTEExpr

Attribute Name	Description	Notes
styleClass	CSS stylesheet class to be applied to this HTML element.	RTExpr
tabIndex	<p>Specifies the position of the current element in the tabbing order for the current document. This value must be a number between 0 and 32767.</p> <p>This tag also supports the following JavaScript events: onBlur, onChange, onClick, onDbClick, onFocus, onKeyDown, onKeyPress, onKeyUp, onMouseDown, onMouseMove, onMouseOut, onMouseOver, onMouseUp</p>	

checkbox

`<jato:checkbox>`

Renders an single HTML `<input type="checkbox">` element.

This tag is only valid when enclosed by both an HTML `<form>` element and a `useViewBean`, `containerView`, or `tilableView` tag. This tag may not contain any body content.

Example:

```
<jato:checkbox name="sendAdditionalInfo"/>
```

The above would be rendered into the following HTML:

```
<input type="checkbox" name="PageFoo.sendAdditionalInfo" value="true">
```

Attribute Name	Description	Notes
accessKey	Assigns an access key to the rendered HTML element. In general, the access key should be included in label text or wherever the access key is to apply.	RTEExpr
disabled	Valid values: "true" or "false" Disables the HTML control for user input. <ul style="list-style-type: none"> • Disabled controls do not receive focus • Disabled controls are skipped in tabbing navigation If not specified, "false" is assumed.	RTEExpr
elementId	Assigns a name to an element. This name must be unique in an HTML document. <i>Please note, this corresponds to the HTML "id" attribute. It has no specific meaning to the Web Application Framework.</i> The id attribute has several roles in HTML: <ul style="list-style-type: none"> • As a style sheet selector • As a target anchor for hypertext links • As a means to reference a particular element from a script 	RTEExpr
extraHtml	Arbitrary HTML that will be rendered just prior to the closing ">" of this HTML element	RTEExpr
label	The label for the checkbox which will be rendered to the right of the control	RTEExpr
name	The name of the <code>DisplayField</code> peer. This peer must be a child of the current parent <code>ContainerView</code> , <code>TiledView</code> , or <code>ViewBean</code> . The <code>DisplayField</code> name is resolved relative to the current parent view. The view peer must be of type <code><JATO package>.view.BooleanDisplayField</code> This name may be a qualified view path, using forward slashes ("/") as delimiters. All components in the path except the last must refer to a <code>ContainerView</code> or a derivative of <code>ContainerView</code> (such as <code>TiledView</code>). Both relative and absolute paths are possible. If a name path begins with a forward slash, the name is assumed to be relative to the root view (the <code>ViewBean</code>). If the path does not begin with a forward slash, the name is assumed to refer to a child relative to the current container. Two dots ("..") may be used to refer to the container that is the parent of the current container. Examples: <pre> /header/orderList/customerName (absolute from root view) orderList/customerName (relative to current container) ../footer/orderList/customerName (relative to parent) </pre>	Req
style	CSS styles to be applied to this HTML element.	RTEExpr

Attribute Name	Description	Notes
styleClass	CSS stylesheet class to be applied to this HTML element.	RTEExpr
tabIndex	Specifies the position of the current element in the tabbing order for the current document. This value must be a number between 0 and 32767. This tag also supports the following JavaScript events: onBlur, onChange, onClick, onDbClick, onFocus, onKeyDown, onKeyPress, onKeyUp, onMouseDown, onMouseMove, onMouseOut, onMouseOver, onMouseUp, onSelect	

combobox

<jato:combobox>

Renders an HTML combobox as comprised of an HTML `<input type="select">` element containing an arbitrary number of HTML `<option>` elements. This tag offers the convenience of treating the combobox as a single element and the flexibility of dynamic list content generation.

This tag is only valid when enclosed by both an HTML `<form>` element and a `useViewBean`, `containerView`, or `tilableView` tag. This tag may not contain any body content.

Example:

```
<jato:combobox name="itemCondition"/>
```

The above would be rendered into the following HTML:

```
<select name="PageFoo.itemCondition">
  <option value="" selected>None Selected</option>
  <option value="Excellent">Excellent</option>
  <option value="Good">Good</option>
  <option value="Average">Average</option>
  <option value="Poor">Poor</option>
</select>
```

Attribute Name	Description	Notes
accessKey	Assigns an access key to the rendered HTML element. In general, the access key should be included in label text or wherever the access key is to apply.	RTEExpr
disabled	Valid values: "true" or "false" Disables the HTML control for user input. <ul style="list-style-type: none"> • Disabled controls do not receive focus • Disabled controls are skipped in tabbing navigation If not specified, "false" is assumed.	RTEExpr
elementId	Assigns a name to an element. This name must be unique in an HTML document. <i>Please note, this corresponds to the HTML "id" attribute. It has no specific meaning to the Web Application Framework.</i> The id attribute has several roles in HTML: <ul style="list-style-type: none"> • As a style sheet selector • As a target anchor for hypertext links • As a means to reference a particular element from a script 	RTEExpr
extraHtml	Arbitrary HTML that will be rendered just prior to the closing ">" of this HTML element	RTEExpr
name	The name of the <code>DisplayField</code> peer. This peer must be a child of the current parent <code>ContainerView</code> , <code>TiledView</code> , or <code>ViewBean</code> . The <code>DisplayField</code> name is resolved relative to the current parent view. The view peer must be of type <code><JATO package>.view.html.SelectableGroup</code> This name may be a qualified view path, using forward slashes ("/") as delimiters. All components in the path except the last must refer to a <code>ContainerView</code> or a derivative of <code>ContainerView</code> (such as <code>TiledView</code>). Both relative and absolute paths are possible. If a name path begins with a forward slash, the name is assumed to be relative to the root view (the <code>ViewBean</code>). If the path does not begin with a forward slash, the name is assumed to refer to a child relative to the current container. Two dots ("..") may be used to refer to the container that is the parent of the current container. Examples: <pre> /header/orderList/customerName (absolute from root view) orderList/customerName (relative to current container) ../footer/orderList/customerName (relative to parent) </pre>	Req
style	CSS styles to be applied to this HTML element.	RTEExpr

Attribute Name	Description	Notes
styleClass	CSS stylesheet class to be applied to this HTML element.	RTEExpr
tabIndex	Specifies the position of the current element in the tabbing order for the current document. This value must be a number between 0 and 32767.	RTEExpr
<p>This tag also supports the following JavaScript events: onBlur, onChange, onClick, onDbClick, onFocus, onKeyDown, onKeyPress, onKeyUp, onMouseDown, onMouseMove, onMouseOut, onMouseOver, onMouseUp, onSelect</p>		

fileUpload

<jato:fileUpload>

Renders an HTML `<input type="file">` element. Used in conjunction with the `com.iplanet.jato.MultipartFormServletFilter` class, allows developers to easily provide file upload capabilities within their application.

The `fileUpload` tag must be used within a form declared with the `multipart/form-data` content type:

```
<jato:form name="form1" method="post" encType="multipart/form-data">
  <jato:fileUpload name="fileUpload1"/>
</jato:form>
```

This tag may not contain body content.

Example:

```
<jato:fileUpload name="fileUpload1" target="_top"/>
```

The above would be rendered into the following HTML:

```
<input type="file" name="Page1.fileUpload1">
```

Attribute Name	Description	Notes
accept	Specifies a comma-separated list of content types that a server processing this input tag will handle correctly. User agents may use this information to filter out non-conforming files when prompting a user to select files to be sent to the server.	
accessKey	Assigns an access key to the rendered HTML element. In general, the access key should be included in label text or wherever the access key is to apply.	RTEExpr
disabled	Valid values: "true" or "false" Disables the HTML control for user input. <ul style="list-style-type: none"> • Disabled controls do not receive focus • Disabled controls are skipped in tabbing navigation If not specified, "false" is assumed.	RTEExpr
elementId	Assigns a name to an element. This name must be unique in an HTML document. <i>Please note, this corresponds to the HTML "id" attribute. It has no specific meaning to the Web Application Framework.</i> The id attribute has several roles in HTML: <ul style="list-style-type: none"> • As a style sheet selector • As a target anchor for hypertext links • As a means to reference a particular element from a script 	RTEExpr
extraHtml	Arbitrary HTML that will be rendered just prior to the closing ">" of this HTML element	RTEExpr
maxLength	Valid values: positive number Specifies the maximum number of characters the user may enter in the field. If not specified, an unlimited number of characters may be entered into the field.	RTEExpr
name	The name of the FileUpload peer. This peer must be a child of the current parent ContainerView, TiledView, or ViewBean. The name is resolved relative to the current parent view. The view peer must be of type <JATO package>.view.html2.FileUpload This name may be a qualified view path, using forward slashes ("/") as delimiters. All components in the path except the last must refer to a ContainerView or a derivative of ContainerView (such as TiledView). Both relative and absolute paths are possible. If a name path begins with a forward slash, the name is assumed to be relative to the root view (the ViewBean). If the path does not begin with a forward slash, the name is assumed to refer to a child relative to the current container. Two dots ("..") may be used to refer to the container that is the parent of the current container. Examples: <pre> /header/orderList/customerName (absolute from root view) orderList/customerName (relative to current container) ../footer/orderList/customerName (relative to parent) </pre>	Req
size	The size of the file input text field	RTEExpr

Attribute Name	Description	Notes
style	CSS styles to be applied to this HTML element.	RTEExpr
styleClass	CSS stylesheet class to be applied to this HTML element.	RTEExpr
tabIndex	Specifies the position of the current element in the tabbing order for the current document. This value must be a number between 0 and 32767. This tag also supports the following JavaScript events: onBlur, onChange, onClick, onDbClick, onFocus, onKeyDown, onKeyPress, onKeyUp, onMouseDown, onMouseMove, onMouseOut, onMouseOver, onMouseUp, onSelect	

form

<jato:form>

Renders an HTML `<form>` element. The form contents are described by the body content of this tag. The HTML form attribute `action` is not exposed in this tag because the tag handler dynamically constructs this attribute to be appropriate for the Web Application Framework runtime environment.

Note, the form tag does not have a view peer, nor can it be nested inside another JATO or HTML `<form>` element. However, it must be enclosed ultimately by a `useViewBean` tag. This tag may contain body content.

Example:

```
<jato:form name="form1" target="_top" />
```

The above would be rendered into the following HTML:

```
<form name="form1" method="post" action="...(JATO URL)...">
...
</form>
```

Property Name	Description	Notes
accept	Specifies a comma-separated list of content types that a server processing this form will handle correctly. User agents may use this information to filter out non-conforming files when prompting a user to select files to be sent to the server.	
acceptCharset	Specifies the list of character encodings for input data that is accepted by the server processing this form. The value is a space- and/or comma-delimited list of charset values. The client must interpret this list as an exclusive-or list, i.e., the server is able to accept any single character encoding per entity received. The default value for this attribute is the reserved string "UNKNOWN". User agents may interpret this value as the character encoding that was used to transmit the document containing this FORM element.	
defaultCommandChild	The name of a <code>CommandField</code> child which will be activated during a default request. A default request is a request that does not include specification of an activated command child. Such a situation will typically occur when the user presses the "Enter" key while in a form field, thereby resulting in submission of the enclosing form without choosing a particular button. The view peer must be of type <code><JATO package>.view.CommandField</code> . This name may be a qualified view path, using forward slashes ("/") as delimiters. All components in the path except the last must refer to a <code>ContainerView</code> or a derivative of <code>ContainerView</code> (such as <code>TiledView</code>). Both relative and absolute paths are possible. If a name path begins with a forward slash, the name is assumed to be relative to the root view (the <code>ViewBean</code>). If the path does not begin with a forward slash, the name is assumed to refer to a child relative to the current container. Two dots (".") may be used to refer to the container that is the parent of the current container. Examples: <pre>defaultCommandChild="/okButton" (absolute from root view) defaultCommandChild="okButton" (relative to current container) defaultCommandChild="../okButton" (relative to parent)</pre>	RTEExpr
elementId	Assigns a name to an element. This name must be unique in an HTML document. <i>Please note, this corresponds to the HTML "id" attribute. It has no specific meaning to the Web Application Framework.</i> The id attribute has several roles in HTML: <ul style="list-style-type: none"> • As a style sheet selector • As a target anchor for hypertext links • As a means to reference a particular element from a script 	RTEExpr
encType	Specifies the content type used to submit the form to the server (when the value of method is "post"). The default value for this attribute is "application/x-www-form-urlencoded". The value "multipart/form-data" should be used in combination with the <code>INPUT</code> element, type="file".	RTEExpr

Property Name	Description	Notes
method	Valid values: "post" or "get" The HTTP method that will be used to submit this request. If not specified, value will be "post".	
name	The name of the form element within the enclosing HTML document. This name is arbitrary and not related to any peer instance	
style	CSS styles to be applied to this HTML element.	RTEExpr
styleClass	CSS stylesheet class to be applied to this HTML element.	RTEExpr
target	Window target to which this form is submitted.	RTEExpr
	This tag also supports the following JavaScript events: onReset, onSubmit	

frameSrc

<jato:frameSrc>

Renders the `src` and `name` attributes for a HTML `<frame>` tag.

This tag is only valid when enclosed by both an HTML `<form>` element and a `useViewBean`, `containerView`, or `tiledView` tag. This tag may not contain any body content.

NOTE: This tag does not attempt to entirely replace the HTML `<frame>` tag because HTML editors will not allow you to edit your frameset in a WYSIWYG fashion without the presence of HTML `<frame>` tags. Therefore, this tag is designed to supply the HTML `<frame>` tag with a dynamic `src` value only.

Example:

```
<frameset rows="30%,70%">
  <frame <jato:frameSrc name="Frame1" location="internal" />
  >
  <frame <jato:frameSrc name="Frame2" location="external" />
  >
</frameset>
```

The above would be rendered into the following HTML:

```
<frameset rows="30%,70%">
  <frame src="...(JATO page)..." name="Frame1" >
  <frame src="/blank.html" name="Frame2" >
</frameset>
```

Attribute Name	Description	Notes
defaultValue	The value to use, if the DisplayField view peer's value is null. For a frameSrc tag, "value" refers to a JATO page name when the location attribute is "internal". If the location attribute is "external", then the "value" refers to an arbitrary non-JATO URL.	RTEExpr
name	The name of the DisplayField peer. This peer must be a child of the current parent ContainerView, TiledView, or ViewBean. The DisplayField name is resolved relative to the current parent view. The view peer must be of type <code><JATO package>.view.DisplayField</code> This name may be a qualified view path, using forward slashes ("/") as delimiters. All components in the path except the last must refer to a ContainerView or a derivative of ContainerView (such as TiledView). Both relative and absolute paths are possible. If a name path begins with a forward slash, the name is assumed to be relative to the root view (the ViewBean). If the path does not begin with a forward slash, the name is assumed to refer to a child relative to the current container. Two dots ("..") may be used to refer to the container that is the parent of the current container. Examples: <pre> /header/orderList/customerName (absolute from root view) orderList/customerName (relative to current container) ../footer/orderList/customerName (relative to parent) </pre>	Req
location	Valid values: "internal" or "external" A value of "internal" specifies that the source document for this frame will be a JATO page in the current application. A value of "external" specifies that the source document for this frame will be an arbitrary URL external to the current application. If not specified, "internal" is assumed. This tag does not support any JavaScript events. You can add JavaScript directly in the HTML <code><frame></code> element that contains this tag.	

hidden

`<jato:hidden>`

Renders an HTML `<input type="hidden">` element.

This tag is only valid when enclosed by both an HTML `<form>` element and a `useViewBean`, `containerView`, or `tiledView` tag. This tag may not contain any body content.

Example:

```
<jato:hidden name="nameFilter" defaultValue="*" />
```

The above would be rendered into the following HTML:

```
<input type="hidden" name="PageFoo.nameFilter" value="Jo*">
```

Attribute Name	Description	Notes
defaultValue	<p>The value to use, if the <code>DisplayField</code> view peer's value is null.</p> <p>For a hidden tag, "value" refers to the value that will be rendered in the HTML Hidden element's "value" attribute</p>	RTExpr
name	<p>The name of the <code>DisplayField</code> peer. This peer must be a child of the current parent <code>ContainerView</code>, <code>TiledView</code>, or <code>ViewBean</code>. The <code>DisplayField</code> name is resolved relative to the current parent view.</p> <p>The view peer may be any subtype of <code><JATO package>.view.DisplayField</code></p> <p>This name may be a qualified view path, using forward slashes ("/") as delimiters. All components in the path except the last must refer to a <code>ContainerView</code> or a derivative of <code>ContainerView</code> (such as <code>TiledView</code>). Both relative and absolute paths are possible. If a name path begins with a forward slash, the name is assumed to be relative to the root view (the <code>ViewBean</code>). If the path does not begin with a forward slash, the name is assumed to refer to a child relative to the current container. Two dots ("..") may be used to refer to the container that is the parent of the current container.</p> <p>Examples:</p> <ul style="list-style-type: none"> <code>/header/orderList/customerName</code> (absolute from root view) <code>orderList/customerName</code> (relative to current container) <code>../footer/orderList/customerName</code> (relative to parent) 	Req

This tag does not support any JavaScript events.

href

<jato:href>

Renders an HTML `...` element.

This tag is only valid when enclosed by a `useViewBean`, `containerView`, or `tiledView` tag. This tag may contain body content. This content will be included between the `<a>` and `` tags of the rendered element and thus appear as the visible, clickable href element.

Example:

```
<jato:href name="orderDrillDown">See orders</jato:href>
```

The above would be rendered into the following HTML:

```
<a href="...(JATO URL)...?PageFoo.orderDrillDown=10345">See
orders</a>
```

Attribute Name	Description	Notes
accessKey	Assigns an access key to the rendered HTML element. In general, the access key should be included in label text or wherever the access key is to apply.	RTEExpr
anchor	The anchor value to append to the generated HREF.	RTEExpr
disabled	Valid values: "true" or "false" Disables the HTML control for user input. <ul style="list-style-type: none"> • Disabled controls do not receive focus • Disabled controls are skipped in tabbing navigation If not specified, "false" is assumed.	RTEExpr

Attribute Name	Description	Notes
defaultValue	<p>The value to use, if the <code>DisplayField</code> view peer's value is null.</p> <p>For an href tag, "value" refers to a special name-value pair that will be automatically included as part of the HREF's query string. The format for the implicit name-value pair is the following:</p> <pre>...?<qualified display field name>=<value>&...</pre> <p>This mechanism allows an href to have a display string as well as a value important to the application.</p>	RTExpr
elementId	<p>Assigns a name to an element. This name must be unique in an HTML document. <i>Please note, this corresponds to the HTML "id" attribute. It has no specific meaning to the Web Application Framework.</i></p> <p>The id attribute has several roles in HTML:</p> <ul style="list-style-type: none"> • As a style sheet selector • As a target anchor for hypertext links • As a means to reference a particular element from a script 	RTExpr
extraHtml	Arbitrary HTML that will be rendered just prior to the closing ">" of this HTML element	RTExpr
name	<p>The name of the <code>DisplayField</code> peer. This peer must be a child of the current parent <code>ContainerView</code>, <code>TiledView</code>, or <code>ViewBean</code>. The <code>DisplayField</code> name is resolved relative to the current parent view.</p> <p>The view peer must be of type <code><JATO package>.view.CommandField</code></p> <p>This name may be a qualified view path, using forward slashes ("/") as delimiters. All components in the path except the last must refer to a <code>ContainerView</code> or a derivative of <code>ContainerView</code> (such as <code>TiledView</code>). Both relative and absolute paths are possible. If a name path begins with a forward slash, the name is assumed to be relative to the root view (the <code>ViewBean</code>). If the path does not begin with a forward slash, the name is assumed to refer to a child relative to the current container. Two dots ("..") may be used to refer to the container that is the parent of the current container.</p> <p>Examples:</p> <pre>/header/orderList/customerName (absolute from root view) orderList/customerName (relative to current container) ../footer/orderList/customerName (relative to parent)</pre>	Req
queryParams	<p>Specifies an arbitrary number of name value pairs which will be appended to the HREF's query string.</p> <p>The name value pairs should follow the format "name=value", where value is URL encoded by the JSP author as needed. Additional name value pairs must be delimited by the '&' character.</p> <p>Example:</p> <pre>queryParams="fname=Mike&lname=Jones"</pre>	RTExpr
style	CSS styles to be applied to this HTML element.	RTExpr
styleClass	CSS stylesheet class to be applied to this HTML element.	RTExpr

Attribute Name	Description	Notes
tabIndex	Specifies the position of the current element in the tabbing order for the current document. This value must be a number between 0 and 32767.	
target	A frame target	RTEExpr
title	Offers advisory information about this element.	RTEExpr
trim	<p>Valid values: "true" or "false"</p> <p>A value of "true" specifies that the tag content should be trimmed of all leading and trailing whitespace. A value of "false" preserves the whitespace.</p> <p>If not specified, "true" is assumed.</p> <p>This tag also supports the following JavaScript events: onBlur, onChange, onClick, onDoubleClick, onFocus, onKeyDown, onKeyPress, onKeyUp, onMouseDown, onMouseMove, onMouseOut, onMouseOver, onMouseUp, onSelect</p>	

image

<jato:image>

Renders an HTML element.

This tag is only valid when enclosed by both an HTML <form> element and a useViewBean, containerView, or tiledView tag. This tag may not contain any body content.

Example:

```
<jato:image name="employeePhoto" defaultValue=
"/images/nophoto.jpg"/>
```

The above would be rendered into the following HTML:

```

```

Attribute Name	Description	Notes
align	<p>Specifies the position of an IMG with respect to its context.</p> <p>The following values for align concern the object's position with respect to surrounding text:</p> <ul style="list-style-type: none"> • bottom: means that the bottom of the object should be vertically aligned with the current baseline. This is the default value. • middle: means that the center of the object should be vertically aligned with the current baseline. • top: means that the top of the object should be vertically aligned with the top of the current text line. • left causes the image to float to the current left margin. • right causes the image to float to the current right margin. 	RTEExpr
alt	Specifies alternate text for user agents that cannot display the image	RTEExpr
border	Specifies the width of the IMG border, in pixels. The default value for this attribute depends on the user agent.	RTEExpr
defaultValue	<p>The value to use, if the <code>DisplayField</code> view peer's value is <code>null</code>.</p> <p>For an image tag, "value" refers to the value that will be rendered in the HTML img element's <code>src</code> attribute</p>	RTEExpr
elementId	<p>Assigns a name to an element. This name must be unique in an HTML document. <i>Please note, this corresponds to the HTML "id" attribute. It has no specific meaning to the Web Application Framework.</i></p> <p>The id attribute has several roles in HTML:</p> <ul style="list-style-type: none"> • As a style sheet selector • As a target anchor for hypertext links • As a means to reference a particular element from a script 	RTEExpr
extraHtml	Arbitrary HTML that will be rendered just prior to the closing ">" of this HTML element	RTEExpr
height	Image height override. When specified, the width and height attributes tell user agents to override the natural image or object size in favor of these values.	RTEExpr
hspace	Specifies the amount of white space to be inserted to the left and right of the IMG. The default value is not specified, but is generally a small, non-zero length.	RTEExpr
isMap	<p>Valid values: "true" or "false"</p> <p>Specifies that the IMG is a server side image map.</p> <p>When the user activates the link by clicking on the image, the screen coordinates are sent directly to the server where the document resides. Screen coordinates are expressed as screen pixel values relative to the image.</p> <p>If not specified, "false" is assumed.</p>	RTEExpr
longDesc	Specifies a link to a long description of the image. This description should supplement the short description provided using the alt attribute.	RTEExpr

Attribute Name	Description	Notes
name	<p>The name of the <code>DisplayField</code> peer. This peer must be a child of the current parent <code>ContainerView</code>, <code>TiledView</code>, or <code>ViewBean</code>. The <code>DisplayField</code> name is resolved relative to the current parent view.</p> <p>The view peer may be any subtype of <code><JATO package>.view.DisplayField</code>. This name may be a qualified view path, using forward slashes ("/") as delimiters. All components in the path except the last must refer to a <code>ContainerView</code> or a derivative of <code>ContainerView</code> (such as <code>TiledView</code>). Both relative and absolute paths are possible. If a name path begins with a forward slash, the name is assumed to be relative to the root view (the <code>ViewBean</code>). If the path does not begin with a forward slash, the name is assumed to refer to a child relative to the current container. Two dots ("..") may be used to refer to the container that is the parent of the current container.</p> <p>Examples:</p> <ul style="list-style-type: none"> <code>/header/orderList/customerName</code> (absolute from root view) <code>orderList/customerName</code> (relative to current container) <code>../footer/orderList/customerName</code> (relative to parent) 	Req
normalize	<p>Valid values: "true" or "false"</p> <p>Specifies that the src URL should be automatically normalized. This means that an absolute URL value will have the current servlet context prepended. A relative URL value will not have current servlet context prepended. Both relative and absolute src URL values will also be URL encoded.</p> <p>If not specified, "false" is assumed.</p>	RTEExpr
style	CSS styles to be applied to this HTML element.	RTEExpr
styleClass	CSS stylesheet class to be applied to this HTML element.	RTEExpr
title	Offers advisory information about this element.	RTEExpr
useMap	Associates an image map with an element. The image map is defined by a <code>MAP</code> element. The value of <code>usemap</code> must match the value of the <code>name</code> attribute of the associated <code>MAP</code> element.	RTEExpr
vspace	Specifies the amount of white space to be inserted above and below the <code>IMG</code> . The default value is not specified, but is generally a small, non-zero length.	RTEExpr
width	Image width override. When specified, the width and height attributes tell user agents to override the natural image or object size in favor of these values.	RTEExpr
	<p>This tag also supports the following JavaScript events:</p> <p><code>onBlur</code>, <code>onChange</code>, <code>onClick</code>, <code>onDbClick</code>, <code>onFocus</code>, <code>onKeyDown</code>, <code>onKeyPress</code>, <code>onKeyUp</code>, <code>onMouseDown</code>, <code>onMouseMove</code>, <code>onMouseOut</code>, <code>onMouseOver</code>, <code>onMouseUp</code>, <code>onSelect</code></p>	

listbox

<jato:listbox>

Renders an HTML list as comprised of an HTML `<input type="select">` element containing an arbitrary number of HTML `<option>` elements. This tag offers the convenience of treating the list as a single element and the flexibility of dynamic list content generation.

This tag is only valid when enclosed by both an HTML `<form>` element and a `useViewBean`, `containerView`, or `tilableView` tag. This tag may not contain any body content.

Example:

```
<jato:listbox name="itemSize" size="4" multiple="false"/>
```

The above would be rendered into the following HTML:

```
<select name="PageFoo.itemSize" size="4">
  <option value="S">Small</option>
  <option value="M">Medium</option>
  <option value="L">Large</option>
  <option value="XL">Extra Large</option>
</select>
```

Attribute Name	Description	Notes
accessKey	Assigns an access key to the rendered HTML element. In general, the access key should be included in label text or wherever the access key is to apply.	RTEExpr
disabled	Valid values: "true" or "false" Disables the HTML control for user input. <ul style="list-style-type: none"> • Disabled controls do not receive focus • Disabled controls are skipped in tabbing navigation If not specified, "false" is assumed.	RTEExpr
elementId	Assigns a name to an element. This name must be unique in an HTML document. <i>Please note, this corresponds to the HTML "id" attribute. It has no specific meaning to the Web Application Framework.</i> The id attribute has several roles in HTML: <ul style="list-style-type: none"> • As a style sheet selector • As a target anchor for hypertext links • As a means to reference a particular element from a script 	RTEExpr
extraHtml	Arbitrary HTML that will be rendered just prior to the closing ">" of this HTML element	RTEExpr
multiple	Valid values: "true" or "false" A value of "true" specifies that multiple items can be selected by the user. A value of "false" specifies that only a single item can be selected by the user. If not specified, "false" is assumed.	
name	The name of the <code>DisplayField</code> peer. This peer must be a child of the current parent <code>ContainerView</code> , <code>TiledView</code> , or <code>ViewBean</code> . The <code>DisplayField</code> name is resolved relative to the current parent view. The view peer must be of type <code><JATO package>.view.html.SelectableGroup</code> This name may be a qualified view path, using forward slashes ("/") as delimiters. All components in the path except the last must refer to a <code>ContainerView</code> or a derivative of <code>ContainerView</code> (such as <code>TiledView</code>). Both relative and absolute paths are possible. If a name path begins with a forward slash, the name is assumed to be relative to the root view (the <code>ViewBean</code>). If the path does not begin with a forward slash, the name is assumed to refer to a child relative to the current container. Two dots ("..") may be used to refer to the container that is the parent of the current container. Examples: <pre> /header/orderList/customerName (absolute from root view) orderList/customerName (relative to current container) ../footer/orderList/customerName (relative to parent) </pre>	Req

Attribute Name	Description	Notes
size	Valid values: positive number Specifies the number of rows in the list that should be visible at the same time. If not specified, the size attribute is omitted from the rendered HTML	RTEExpr
style	CSS styles to be applied to this HTML element.	RTEExpr
styleClass	CSS stylesheet class to be applied to this HTML element.	RTEExpr
tabIndex	Specifies the position of the current element in the tabbing order for the current document. This value must be a number between 0 and 32767.	
	<p>This tag also supports the following JavaScript events: onBlur, onChange, onClick, onDbClick, onFocus, onKeyDown, onKeyPress, onKeyUp, onMouseDown, onMouseMove, onMouseOut, onMouseOver, onMouseUp, onSelect</p>	

password

<jato:password>

Renders an HTML `<input type="password">` element.

This tag is only valid when enclosed by both an HTML `<form>` element and a `useViewBean`, `containerView`, or `tilableView` tag. This tag may not contain any body content.

Example:

```
<jato:password name="newPassword" size="8" maxLength="16"/>
```

The above would be rendered into the following HTML:

```
<input type="password" name="PageFoo.newPassword" value=""  
maxLength="16" size="8">
```

Attribute Name	Description	Notes
accessKey	Assigns an access key to the rendered HTML element. In general, the access key should be included in label text or wherever the access key is to apply.	RTEExpr
defaultValue	The value to use, if the <code>DisplayField</code> view peer's value is <code>null</code> . For a password tag, "value" refers to the value that will be rendered in the HTML password element's "value" attribute	RTEExpr
disabled	Valid values: "true" or "false" Disables the HTML control for user input. <ul style="list-style-type: none"> • Disabled controls do not receive focus • Disabled controls are skipped in tabbing navigation If not specified, "false" is assumed.	RTEExpr
elementId	Assigns a name to an element. This name must be unique in an HTML document. <i>Please note, this corresponds to the HTML "id" attribute. It has no specific meaning to the Web Application Framework.</i> The id attribute has several roles in HTML: <ul style="list-style-type: none"> • As a style sheet selector • As a target anchor for hypertext links • As a means to reference a particular element from a script 	RTEExpr
extraHtml	Arbitrary HTML that will be rendered just prior to the closing ">" of this HTML element	RTEExpr
maxLength	Valid values: positive number Specifies the maximum number of characters the user may enter in the field. If not specified, an unlimited number of characters may be entered into the field.	RTEExpr
name	The name of the <code>DisplayField</code> peer. This peer must be a child of the current parent <code>ContainerView</code> , <code>TiledView</code> , or <code>ViewBean</code> . The <code>DisplayField</code> name is resolved relative to the current parent view. The view peer must be of type <code><JATO package>.view.DisplayField</code> This name may be a qualified view path, using forward slashes ("/") as delimiters. All components in the path except the last must refer to a <code>ContainerView</code> or a derivative of <code>ContainerView</code> (such as <code>TiledView</code>). Both relative and absolute paths are possible. If a name path begins with a forward slash, the name is assumed to be relative to the root view (the <code>ViewBean</code>). If the path does not begin with a forward slash, the name is assumed to refer to a child relative to the current container. Two dots ("..") may be used to refer to the container that is the parent of the current container. Examples: <pre> /header/orderList/customerName (absolute from root view) orderList/customerName (relative to current container) ../footer/orderList/customerName (relative to parent) </pre>	Req

Attribute Name	Description	Notes
readOnly	Valid values: "true" or "false" A value of "true" specifies that the text contained within this element cannot be changed by the user. If not specified, "false" is assumed.	RTEExpr
size	Valid values: positive number Specifies the initial width of the control in number of characters. If not specified, the size attribute is omitted from the rendered HTML	RTEExpr
style	CSS styles to be applied to this HTML element.	RTEExpr
styleClass	CSS stylesheet class to be applied to this HTML element.	RTEExpr
tabIndex	Specifies the position of the current element in the tabbing order for the current document. This value must be a number between 0 and 32767.	
	<p>This tag also supports the following JavaScript events: onBlur, onChange, onClick, onDbClick, onFocus, onKeyDown, onKeyPress, onKeyUp, onMouseDown, onMouseMove, onMouseOut, onMouseOver, onMouseUp, onSelect</p>	

radioButtons

<jato:radioButtons>

Renders a group of HTML radio buttons as comprised of an arbitrary number of HTML `<input type="radio">` elements which share the same name attribute value. This tag offers the convenience of treating the set of radio buttons as a single element and the flexibility of dynamic choice generation.

This tag is only valid when enclosed by both an HTML `<form>` element and a `useViewBean`, `containerView`, or `tiledView` tag. This tag may not contain any body content.

Example:

```
<jato:radioButtons name="incomeRange" layout="vertical"/>
```

The above would be rendered into the following HTML:

```
<input type="radio" name="PageFoo.incomeRange" value=""
checked>Nondisclosed
<br>
<input type="radio" name="PageFoo.incomeRange" value="19">$0
to $19,999
<Br>
<input type="radio" name="PageFoo.incomeRange" value=
"49">$20,000-$49,999
<Br>
<input type="radio" name="PageFoo.incomeRange" value=
"50">$50,000 or more
<Br>
```

Attribute Name	Description	Notes
accessKey	Assigns an access key to the rendered HTML element. In general, the access key should be included in label text or wherever the access key is to apply.	RTEExpr
disabled	Valid values: "true" or "false" Disables the HTML control for user input. <ul style="list-style-type: none"> • Disabled controls do not receive focus • Disabled controls are skipped in tabbing navigation If not specified, "false" is assumed.	RTEExpr
elementId	Assigns a name to an element. This name must be unique in an HTML document. <i>Please note, this corresponds to the HTML "id" attribute. It has no specific meaning to the Web Application Framework.</i> The id attribute has several roles in HTML: <ul style="list-style-type: none"> • As a style sheet selector • As a target anchor for hypertext links • As a means to reference a particular element from a script 	RTEExpr
extraHtml	Arbitrary HTML that will be rendered just prior to the closing ">" of this HTML element	RTEExpr
layout	Valid values: "h", "horizontal" or anything beginning with "h" A value beginning with "h" specifies a horizontal arrangement of radio buttons within this group. No value, or a value beginning with a letter other than "h" specifies a vertical alignment. The vertical arrangement is achieved by appending a element after each of the rendered HTML radio button controls.	RTEExpr
name	The name of the <code>DisplayField</code> peer. This peer must be a child of the current parent <code>ContainerView</code> , <code>TiledView</code> , or <code>ViewBean</code> . The <code>DisplayField</code> name is resolved relative to the current parent view. The view peer must be of type <code><JATO package>.view.html.SelectableGroup</code> This name may be a qualified view path, using forward slashes ("/") as delimiters. All components in the path except the last must refer to a <code>ContainerView</code> or a derivative of <code>ContainerView</code> (such as <code>TiledView</code>). Both relative and absolute paths are possible. If a name path begins with a forward slash, the name is assumed to be relative to the root view (the <code>ViewBean</code>). If the path does not begin with a forward slash, the name is assumed to refer to a child relative to the current container. Two dots ("..") may be used to refer to the container that is the parent of the current container. Examples: <pre> /header/orderList/customerName (absolute from root view) orderList/customerName (relative to current container) ../footer/orderList/customerName (relative to parent) </pre>	Req
style	CSS styles to be applied to this HTML element.	RTEExpr

Attribute Name	Description	Notes
styleClass	CSS stylesheet class to be applied to this HTML element.	RTEExpr
tabIndex	<p>Specifies the position of the current element in the tabbing order for the current document. This value must be a number between 0 and 32767.</p> <p>This tag also supports the following JavaScript events: onBlur, onChange, onClick, onDbClick, onFocus, onKeyDown, onKeyPress, onKeyUp, onMouseDown, onMouseMove, onMouseOut, onMouseOver, onMouseUp, onSelect</p>	

text

<jato:text>

Renders a display field value as plain HTML body text. This tag is only valid when nested inside of a `useViewBean` or `containerView` or `tiledView` tag body.

This tag does not require an enclosing `<form>` element but must be enclosed by a `useViewBean`, `containerView`, or `tiledView` tag. This tag may not contain any body content.

Example:

```
<jato:text name="ssn" formatType="string" formatMask="???-??-????" />
```

The above would be rendered into the following HTML:

```
123-45-6789
```

Attribute Name	Description	Notes
defaultValue	The value to render if the <code>DisplayField</code> view peer's value is <code>null</code> .	RTExpr
escape	Valid values: "true" or "false" A value of "true" specifies that the value be HTML escaped. If not specified, a value of "true" is assumed.	
formatMask	The valid values for the <code>formatMask</code> attribute depend on the value of <code>formatType</code> attribute. For <code>formatType</code> values of "string" or "alpha": The format mask consists of any arbitrary combination of static text and the reserved metacharacter "?". Each "?" metacharacter serves as a placeholder for a single character in the value being formatted, proceeding from left to right. A literal "?" can be specified by escaping the "?" character with "\\". For <code>formatType</code> values of "dec" or "curr": The format mask consists of two patterns separated by a semicolon: a primary pattern (specifies precision and thousandths separator) and a negative number pattern (see <code>java.text.DecimalFormat</code> for pattern details). (*See table below.)	

Attribute Name	Description	Notes
formatType	<p>Valid values: "string" (also "alpha"), "dec", or "curr", or "date"</p> <p>Each type has specific masking rules (see the <code>formatMask</code> attribute for details).</p> <p>A value of "string" supports the formatting of arbitrary text via the application of a simple text mask. This allows dynamic content to be arbitrarily combined with static text. A value of "alpha" is identical to "string".</p> <p>A value of "dec" supports the formatting of numeric values via the application of a numeric mask.</p> <p>A value of "curr" supports the formatting of currency values via the application of a currency mask. The currency format should simply take a decimal specification and apply the locale specific currency format. Currently, currency formatting is not working due to an unresolved bug.</p> <p>A value of "date", although legal, at present has no supporting date formatting implementation, due to the fact that the <code>java.text.DateFormat</code> class does not lend itself easily to the generic formatting support that a <code>taglib</code> implementation would require. We are investigating ways around this limitation.</p> <p>Therefore, it is suggested that developers wishing to apply date formatting implement the <code>begin<fieldName>Display(ChildDisplayEvent event)</code> and/or <code>end<fieldName>Display(ChildContentDisplayEvent event)</code> events and therein take explicit advantage of the date formatting support available in the <code>java.text.DateFormat</code> class.</p> <p>The Web Application Framework implementation of text formatting relies heavily on the <code>java.text</code> package and the format related classes therein. Please see that package for more information. See also the <code><JATO package>.util.HtmlUtil</code> class.</p>	
name	<p>The name of the <code>DisplayField</code> peer. This peer must be a child of the current parent <code>ContainerView</code>, <code>TiledView</code>, or <code>ViewBean</code>. The <code>DisplayField</code> name is resolved relative to the current parent view.</p> <p>The view peer must be of type <code><JATO package>.view.DisplayField</code></p> <p>This name may be a qualified view path, using forward slashes ("/") as delimiters. All components in the path except the last must refer to a <code>ContainerView</code> or a derivative of <code>ContainerView</code> (such as <code>TiledView</code>). Both relative and absolute paths are possible. If a name path begins with a forward slash, the name is assumed to be relative to the root view (the <code>ViewBean</code>). If the path does not begin with a forward slash, the name is assumed to refer to a child relative to the current container. Two dots ("..") may be used to refer to the container that is the parent of the current container.</p> <p>Examples:</p> <pre> /header/orderList/customerName (absolute from root view) orderList/customerName (relative to current container) ../footer/orderList/customerName (relative to parent) </pre>	Req

This tag does not support any JavaScript events.

(*See `formatMask` above.)

Format Type	Format Mask	Sample Value	Rendered Value
string	??-??-????	123456789	123-45-6789
string	??HELLO??	abcd	abHELLOcd
string	??HELLO\ \ ???	abcd	abHELLO?cd
dec	###0; (-#)	1000000.99	1000001
dec	#,##0.00; (-#)	1000000.99	1,000,000.99
dec	###0.00; (-#)	1000000.99	1000000.99
dec	###0; (-#)	-1000000.99	(-1000001)
dec	###0; (#)	-1000000.99	(1000001)
dec	#,##0.00; -#	-1000000.99	-1,000,000.99

textArea

<jato:textArea>

Renders an HTML `<textarea>` element.

This tag is only valid when enclosed by both an HTML `<form>` element and a `useViewBean`, `containerView`, or `tiledView` tag. This tag may not contain any body content.

Example:

```
<jato:textArea name="comments" cols="80" rows="5" wrap="virtual"/>
```

The above would be rendered into the following HTML:

```
<textarea name="PageFoo.comments" cols="80" rows="5" wrap="virtual">
...
</textarea>
```

Attribute Name	Description	Notes
accessKey	Assigns an access key to the rendered HTML element. In general, the access key should be included in label text or wherever the access key is to apply.	RTEExpr
cols	Valid values: positive number Specifies the visible width of the field in average character widths.	Req, RTEExpr
defaultValue	The value to use, if the <code>DisplayField</code> view peer's value is <code>null</code> . For a <code>textArea</code> tag, "value" refers to the text that will be rendered between the opening and closing tag of the <code>textArea</code> element	RTEExpr
disabled	Valid values: "true" or "false" Disables the HTML control for user input. <ul style="list-style-type: none"> • Disabled controls do not receive focus • Disabled controls are skipped in tabbing navigation If not specified, "false" is assumed.	RTEExpr
elementId	Assigns a name to an element. This name must be unique in an HTML document. <i>Please note, this corresponds to the HTML "id" attribute. It has no specific meaning to the Web Application Framework.</i> The <code>id</code> attribute has several roles in HTML: <ul style="list-style-type: none"> • As a style sheet selector • As a target anchor for hypertext links • As a means to reference a particular element from a script 	RTEExpr
extraHtml	Arbitrary HTML that will be rendered just prior to the closing ">" of the opening <code><textarea></code> element	RTEExpr
formatMask	The valid values for the <code>formatMask</code> attribute depend on the value of <code>formatType</code> attribute. For <code>formatType</code> values of "string" or "alpha": The format mask consists of any arbitrary combination of static text and the reserved metacharacter "?". Each "?" metacharacter serves as a placeholder for a single character in the value being formatted, proceeding from left to right. A literal "?" can be specified by escaping the "?" character with "\\?". For <code>formatType</code> values of "dec" or "curr": The format mask consists of two patterns separated by a semicolon: a primary pattern (specifies precision and thousandths separator) and a negative number pattern (see <code>java.text.DecimalFormat</code> for pattern details). (*See table below.)	

Attribute Name	Description	Notes
formatType	<p>Valid values: "string" (also "alpha"), "dec", or "curr", or "date"</p> <p>Each type has specific masking rules (see the <code>formatMask</code> attribute for details).</p> <p>A value of "string" supports the formatting of arbitrary text via the application of a simple text mask. This allows dynamic content to be arbitrarily combined with static text. A value of "alpha" is identical to "string".</p> <p>A value of "dec" supports the formatting of numeric values via the application of a numeric mask.</p> <p>A value of "curr" supports the formatting of currency values via the application of a currency mask. The currency format should simply take a decimal specification and apply the locale specific currency format. Currently, currency formatting is not working due to an unresolved bug.</p> <p>A value of "date", although legal, at present has no supporting date formatting implementation, due to the fact that the <code>java.text.DateFormat</code> class does not lend itself easily to the generic formatting support that a <code>taglib</code> implementation would require. We are investigating ways around this limitation.</p> <p>Therefore, it is suggested that developers wishing to apply date formatting implement the <code>begin<fieldName>Display(ChildDisplayEvent event)</code> and/or <code>end<fieldName>Display(ChildContentDisplayEvent event)</code> events and therein take explicit advantage of the date formatting support available in the <code>java.text.DateFormat</code> class.</p> <p>The Web Application Framework implementation of text formatting relies heavily on the <code>java.text</code> package and the format related classes therein. Please see that package for more information. See also the <code><JATO package>.util.HtmlUtil</code> class.</p>	
name	<p>The name of the <code>DisplayField</code> peer. This peer must be a child of the current parent <code>ContainerView</code>, <code>TiledView</code>, or <code>ViewBean</code>. The <code>DisplayField</code> name is resolved relative to the current parent view.</p> <p>The view peer must be of type <code><JATO package>.view.DisplayField</code></p> <p>This name may be a qualified view path, using forward slashes ("/") as delimiters. All components in the path except the last must refer to a <code>ContainerView</code> or a derivative of <code>ContainerView</code> (such as <code>TiledView</code>). Both relative and absolute paths are possible. If a name path begins with a forward slash, the name is assumed to be relative to the root view (the <code>ViewBean</code>). If the path does not begin with a forward slash, the name is assumed to refer to a child relative to the current container. Two dots ("..") may be used to refer to the container that is the parent of the current container.</p> <p>Examples:</p> <pre> /header/orderList/customerName (absolute from root view) orderList/customerName (relative to current container) ../footer/orderList/customerName (relative to parent) </pre>	Req
readOnly	<p>Valid values: "true" or "false"</p> <p>A value of "true" specifies that the text contained within this element cannot be changed by the user.</p> <p>If not specified, "false" is assumed.</p>	RTExpr

Attribute Name	Description	Notes
rows	Valid values: positive number Specifies the number of visible text lines in the rendered control	Req, RTEExpr
style	CSS styles to be applied to this HTML element.	RTEExpr
styleClass	CSS stylesheet class to be applied to this HTML element.	RTEExpr
tabIndex	Specifies the position of the current element in the tabbing order for the current document. This value must be a number between 0 and 32767.	
wrap	Valid values: valid wrap type supported by the browser, such as "virtual" or "physical" Specifies the value that will be rendered in the HTML textarea element's "wrap" attribute. If not specified, the wrap attribute is omitted from the rendered HTML	RTEExpr
<p>This tag also supports the following JavaScript events: onBlur, onChange, onClick, onDbClick, onFocus, onKeyDown, onKeyPress, onKeyUp, onMouseDown, onMouseMove, onMouseOut, onMouseOver, onMouseUp, onSelect</p>		

(*See formatMask above.)

Format Type	Format Mask	Sample Value	Rendered Value
string	??-??-????	123456789	123-45-6789
string	??HELLO??	abcd	abHELLOcd
string	??HELLO\???	abcd	abHELLO?cd
dec	###0; (-#)	1000000.99	1000001
dec	#,###0.00; (-#)	1000000.99	1,000,000.99
dec	###0.00; (-#)	1000000.99	1000000.99
dec	###0; (-#)	-1000000.99	(-1000001)
dec	###0; (#)	-1000000.99	(1000001)
dec	#,###0.00; -#	-1000000.99	-1,000,000.99

textField

<jato:textField>

Renders an HTML `<input type="text">` element.

This tag is only valid when enclosed by both an HTML `<form>` element and a `useViewBean`, `containerView`, or `tilableView` tag. This tag may not contain any body content.

Example:

```
<jato:textField name="firstName" size="12" maxLength="32"/>
```

The above would be rendered into the following HTML:

```
<input type="text" name="PageFoo.firstName" value="John"
maxLength="32" size="12">
```

Attribute Name	Description	Notes
accessKey	Assigns an access key to the rendered HTML element. In general, the access key should be included in label text or wherever the access key is to apply.	RTEExpr
defaultValue	The value to use, if the <code>DisplayField</code> view peer's value is <code>null</code> . For a <code>textField</code> tag, "value" refers to the value that will be rendered in the HTML text element's "value" attribute	RTEExpr
disabled	Valid values: "true" or "false" Disables the HTML control for user input. <ul style="list-style-type: none"> • Disabled controls do not receive focus • Disabled controls are skipped in tabbing navigation If not specified, "false" is assumed.	RTEExpr
elementId	Assigns a name to an element. This name must be unique in an HTML document. <i>Please note, this corresponds to the HTML "id" attribute. It has no specific meaning to the Web Application Framework.</i> The id attribute has several roles in HTML: <ul style="list-style-type: none"> • As a style sheet selector • As a target anchor for hypertext links • As a means to reference a particular element from a script 	RTEExpr
extraHtml	Arbitrary HTML that will be rendered just prior to the closing ">" of this HTML element	RTEExpr
formatMask	The valid values for the <code>formatMask</code> attribute depend on the value of <code>formatType</code> attribute. For <code>formatType</code> values of "string" or "alpha": The format mask consists of any arbitrary combination of static text and the reserved metacharacter "?". Each "?" metacharacter serves as a placeholder for a single character in the value being formatted, proceeding from left to right. A literal "?" can be specified by escaping the "?" character with "\\?". For <code>formatType</code> values of "dec" or "curr": The format mask consists of two patterns separated by a semicolon: a primary pattern (specifies precision and thousandths separator) and a negative number pattern (see <code>java.text.DecimalFormat</code> for pattern details). (*See table below.)	

Attribute Name	Description	Notes
formatType	<p>Valid values: "string" (also "alpha"), "dec", or "curr", or "date"</p> <p>Each type has specific masking rules (see the <code>formatMask</code> attribute for details).</p> <p>A value of "string" supports the formatting of arbitrary text via the application of a simple text mask. This allows dynamic content to be arbitrarily combined with static text. A value of "alpha" is identical to "string".</p> <p>A value of "dec" supports the formatting of numeric values via the application of a numeric mask.</p> <p>A value of "curr" supports the formatting of currency values via the application of a currency mask. The currency format should simply take a decimal specification and apply the locale specific currency format. Currently, currency formatting is not working due to an unresolved bug.</p> <p>A value of "date", although legal, at present has no supporting date formatting implementation, due to the fact that the <code>java.text.DateFormat</code> class does not lend itself easily to the generic formatting support that a <code>taglib</code> implementation would require. We are investigating ways around this limitation.</p> <p>Therefore, it is suggested that developers wishing to apply date formatting implement the <code>begin<fieldName>Display(ChildDisplayEvent event)</code> and/or <code>end<fieldName>Display(ChildContentDisplayEvent event)</code> events and therein take explicit advantage of the date formatting support available in the <code>java.text.DateFormat</code> class.</p> <p>The Web Application Framework implementation of text formatting relies heavily on the <code>java.text</code> package and the format related classes therein. Please see that package for more information. See also the <code><JATO package>.util.HtmlUtil</code> class.</p>	
maxLength	<p>Valid values: positive number</p> <p>Specifies the maximum number of characters the user may enter in the field.</p> <p>If not specified, an unlimited number of characters may be entered into the field.</p>	RTExpr
name	<p>The name of the <code>DisplayField</code> peer. This peer must be a child of the current parent <code>ContainerView</code>, <code>TiledView</code>, or <code>ViewBean</code>. The <code>DisplayField</code> name is resolved relative to the current parent view.</p> <p>The view peer must be of type <code><JATO package>.view.DisplayField</code></p> <p>This name may be a qualified view path, using forward slashes ("/") as delimiters. All components in the path except the last must refer to a <code>ContainerView</code> or a derivative of <code>ContainerView</code> (such as <code>TiledView</code>). Both relative and absolute paths are possible. If a name path begins with a forward slash, the name is assumed to be relative to the root view (the <code>ViewBean</code>). If the path does not begin with a forward slash, the name is assumed to refer to a child relative to the current container. Two dots ("..") may be used to refer to the container that is the parent of the current container.</p> <p>Examples:</p> <pre> /header/orderList/customerName (absolute from root view) orderList/customerName (relative to current container) ../footer/orderList/customerName (relative to parent) </pre>	Req

Attribute Name	Description	Notes
readOnly	Valid values: "true" or "false" A value of "true" specifies that the text contained within this element cannot be changed by the user. If not specified, "false" is assumed.	RTEExpr
size	Valid values: positive number Specifies the initial width of the control in number of characters. If not specified, the size attribute is omitted from the rendered HTML	RTEExpr
style	CSS styles to be applied to this HTML element.	RTEExpr
styleClass	CSS stylesheet class to be applied to this HTML element.	RTEExpr
tabIndex	Specifies the position of the current element in the tabbing order for the current document. This value must be a number between 0 and 32767.	
<p>This tag also supports the following JavaScript events: onBlur, onChange, onClick, onDbClick, onFocus, onKeyDown, onKeyPress, onKeyUp, onMouseDown, onMouseMove, onMouseOut, onMouseOver, onMouseUp, onSelect</p>		

(*See formatMask above.)

Format Type	Format Mask	Sample Value	Rendered Value
string	??-??-????	123456789	123-45-6789
string	??HELLO??	abcd	abHELLOcd
string	??HELLO\ \ ???	abcd	abHELLO?cd
dec	###0; (-#)	1000000.99	1000001
dec	#,##0.00; (-#)	1000000.99	1,000,000.99
dec	###0.00; (-#)	1000000.99	1000000.99
dec	###0; (-#)	-1000000.99	(-1000001)
dec	###0; (#)	-1000000.99	(1000001)
dec	#,##0.00; -#	-1000000.99	-1,000,000.99

validatingTextArea

<jato:validatingTextArea>

Renders an HTML `<textarea>` element.

This tag is only valid when enclosed by both an HTML `<form>` element and a `useViewBean` or `containerView` tag. This tag may not contain any body content and may not appear within a `tiledView` tag.

The type of this tag's peer component must be `ValidatingDisplayField`.

Example:

```
<jato:validatingTextArea name="comments" cols="80" rows="5"
wrap="virtual" />
```

The above would be rendered into the following HTML:

```
<textarea name="PageFoo.comments" cols="80" rows="5" wrap=
"virtual">
...
</textarea>
```

Attribute Name	Description	Notes
accessKey	Assigns an access key to the rendered HTML element. In general, the access key should be included in label text or wherever the access key is to apply.	RTEExpr
cols	Valid values: positive number Specifies the visible width of the field in average character widths.	Req, RTEExpr
defaultValue	The value to use, if the <code>DisplayField</code> view peer's value is <code>null</code> . For a <code>validatingTextArea</code> tag, "value" refers to the text that will be rendered between the opening and closing tag of the <code>textarea</code> element	RTEExpr
disabled	Valid values: "true" or "false" Disables the HTML control for user input. <ul style="list-style-type: none"> • Disabled controls do not receive focus • Disabled controls are skipped in tabbing navigation If not specified, "false" is assumed.	RTEExpr
elementId	Assigns a name to an element. This name must be unique in an HTML document. <i>Please note, this corresponds to the HTML "id" attribute. It has no specific meaning to the JWeb Application Framework.</i> The <code>id</code> attribute has several roles in HTML: <ul style="list-style-type: none"> • As a style sheet selector • As a target anchor for hypertext links • As a means to reference a particular element from a script 	RTEExpr
extraHtml	Arbitrary HTML that will be rendered just prior to the closing ">" of the opening <code><textarea></code> element	RTEExpr
formatMask	The valid values for the <code>formatMask</code> attribute depend on the value of <code>formatType</code> attribute. For <code>formatType</code> values of "string" or "alpha": The format mask consists of any arbitrary combination of static text and the reserved metacharacter "?". Each "?" metacharacter serves as a placeholder for a single character in the value being formatted, proceeding from left to right. A literal "?" can be specified by escaping the "?" character with "\\?". For <code>formatType</code> values of "dec" or "curr": The format mask consists of two patterns separated by a semicolon: a primary pattern (specifies precision and thousandths separator) and a negative number pattern (see <code>java.text.DecimalFormat</code> for pattern details). (*See table below.)	

Attribute Name	Description	Notes
formatType	<p>Valid values: "string" (also "alpha"), "dec", or "curr", or "date"</p> <p>Each type has specific masking rules (see the <code>formatMask</code> attribute for details).</p> <p>A value of "string" supports the formatting of arbitrary text via the application of a simple text mask. This allows dynamic content to be arbitrarily combined with static text. A value of "alpha" is identical to "string".</p> <p>A value of "dec" supports the formatting of numeric values via the application of a numeric mask.</p> <p>A value of "curr" supports the formatting of currency values via the application of a currency mask. The currency format should simply take a decimal specification and apply the locale specific currency format. Currently, currency formatting is not working due to an unresolved bug.</p> <p>A value of "date", although legal, at present has no supporting date formatting implementation, due to the fact that the <code>java.text.DateFormat</code> class does not lend itself easily to the generic formatting support that a <code>taglib</code> implementation would require. We are investigating ways around this limitation.</p> <p>Therefore, it is suggested that developers wishing to apply date formatting implement the <code>begin<fieldName>Display(ChildDisplayEvent event)</code> and/or <code>end<fieldName>Display(ChildContentDisplayEvent event)</code> events and therein take explicit advantage of the date formatting support available in the <code>java.text.DateFormat</code> class.</p> <p>The Web Application Framework implementation of text formatting relies heavily on the <code>java.text</code> package and the format related classes therein. Please see that package for more information. See also the <code><JATO package>.util.HtmlUtil</code> class.</p>	
name	<p>The name of the <code>ValidatingDisplayField</code> peer. This peer must be a child of the current parent <code>ContainerView</code> or <code>ViewBean</code>. The name is resolved relative to the current parent view.</p> <p>The view peer must be of type <code><JATO package>.view.DisplayField</code></p> <p>This name may be a qualified view path, using forward slashes ("/") as delimiters. All components in the path except the last must refer to a <code>ContainerView</code> or a derivative of <code>ContainerView</code>. Both relative and absolute paths are possible. If a name path begins with a forward slash, the name is assumed to be relative to the root view (the <code>ViewBean</code>). If the path does not begin with a forward slash, the name is assumed to refer to a child relative to the current container. Two dots ("..") may be used to refer to the container that is the parent of the current container.</p> <p>Examples:</p> <pre> /header/orderList/customerName (absolute from root view) orderList/customerName (relative to current container) ../footer/orderList/customerName (relative to parent) </pre>	Req
readOnly	<p>Valid values: "true" or "false"</p> <p>A value of "true" specifies that the text contained within this element cannot be changed by the user.</p> <p>If not specified, "false" is assumed.</p>	RTExpr

Attribute Name	Description	Notes
rows	Valid values: positive number Specifies the number of visible text lines in the rendered control	Req, RTExpr
style	CSS styles to be applied to this HTML element.	RTExpr
styleClass	CSS styles to be applied to this HTML element.	RTExpr
tabIndex	Specifies the position of the current element in the tabbing order for the current document. This value must be a number between 0 and 32767.	
wrap	Valid values: valid wrap type supported by the browser, such as "virtual" or "physical" Specifies the value that will be rendered in the HTML textarea element's "wrap" attribute. If not specified, the wrap attribute is omitted from the rendered HTML	RTExpr
<p>This tag also supports the following JavaScript events: onBlur, onChange, onClick, onDblClick, onFocus, onKeyDown, onKeyPress, onKeyUp, onMouseDown, onMouseMove, onMouseOut, onMouseOver, onMouseUp, onSelect</p>		

(*See formatMask above.)

Format Type	Format Mask	Sample Value	Rendered Value
string	??-??-????	123456789	123-45-6789
string	??HELLO??	abcd	abHELLOcd
string	??HELLO\ \ ???	abcd	abHELLO?cd
dec	###0; (-#)	1000000.99	1000001
dec	#,###0.00; (-#)	1000000.99	1,000,000.99
dec	###0.00; (-#)	1000000.99	1000000.99
dec	###0; (-#)	-1000000.99	(-1000001)
dec	###0; (#)	-1000000.99	(1000001)
dec	#,###0.00; -#	-1000000.99	-1,000,000.99

validatingTextField

<jato:validatingTextField>

Renders an HTML `<input type="text">` element.

This tag is only valid when enclosed by both an HTML `<form>` element and a `useViewBean` or `containerView` tag. This tag may not contain any body content, and may not be used within a `tiledView` tag.

The type of this tag's peer component must be `ValidatingDisplayField`.

Example:

```
<jato:validatingTextField name="firstName" size="12"
  maxLength="32" />
```

The above would be rendered into the following HTML:

```
<input type="text" name="PageFoo.firstName" value="John"
  maxLength="32" size="12">
```

Attribute Name	Description	Notes
accessKey	Assigns an access key to the rendered HTML element. In general, the access key should be included in label text or wherever the access key is to apply.	RTEExpr
defaultValue	The value to use, if the <code>DisplayField</code> view peer's value is <code>null</code> . For a <code>validatingTextField</code> tag, "value" refers to the value that will be rendered in the HTML text element's "value" attribute	RTEExpr
disabled	Valid values: "true" or "false" Disables the HTML control for user input. <ul style="list-style-type: none"> • Disabled controls do not receive focus • Disabled controls are skipped in tabbing navigation If not specified, "false" is assumed.	RTEExpr
elementId	Assigns a name to an element. This name must be unique in an HTML document. <i>Please note, this corresponds to the HTML "id" attribute. It has no specific meaning to the Web Application Framework.</i> The id attribute has several roles in HTML: <ul style="list-style-type: none"> • As a style sheet selector • As a target anchor for hypertext links • As a means to reference a particular element from a script 	RTEExpr
extraHtml	Arbitrary HTML that will be rendered just prior to the closing ">" of this HTML element	RTEExpr
formatMask	The valid values for the <code>formatMask</code> attribute depend on the value of <code>formatType</code> attribute. For <code>formatType</code> values of "string" or "alpha": The format mask consists of any arbitrary combination of static text and the reserved metacharacter "?". Each "?" metacharacter serves as a placeholder for a single character in the value being formatted, proceeding from left to right. A literal "?" can be specified by escaping the "?" character with "\\?". For <code>formatType</code> values of "DEC" or "curr": The format mask consists of two patterns separated by a semicolon: a primary pattern (specifies precision and thousandths separator) and a negative number pattern (see <code>java.text.DecimalFormat</code> for pattern details). (*See table below.)	

Attribute Name	Description	Notes
formatType	<p>Valid values: "string" (also "alpha"), "dec", or "curr", or "date"</p> <p>Each type has specific masking rules (see the <code>formatMask</code> attribute for details).</p> <p>A value of "string" supports the formatting of arbitrary text via the application of a simple text mask. This allows dynamic content to be arbitrarily combined with static text. A value of "alpha" is identical to "string".</p> <p>A value of "dec" supports the formatting of numeric values via the application of a numeric mask.</p> <p>A value of "curr" supports the formatting of currency values via the application of a currency mask. The currency format should simply take a decimal specification and apply the locale specific currency format. Currently, currency formatting is not working due to an unresolved bug.</p> <p>A value of "date", although legal, at present has no supporting date formatting implementation, due to the fact that the <code>java.text.DateFormat</code> class does not lend itself easily to the generic formatting support that a <code>taglib</code> implementation would require. We are investigating ways around this limitation.</p> <p>Therefore, it is suggested that developers wishing to apply date formatting implement the <code>begin<fieldName>Display(ChildDisplayEvent event)</code> and/or <code>end<fieldName>Display(ChildContentDisplayEvent event)</code> events and therein take explicit advantage of the date formatting support available in the <code>java.text.DateFormat</code> class.</p> <p>The Web Application Framework implementation of text formatting relies heavily on the <code>java.text</code> package and the format related classes therein. Please see that package for more information. See also the <code><JATO package>.util.HtmlUtil</code> class.</p>	
maxLength	<p>Valid values: positive number</p> <p>Specifies the maximum number of characters the user may enter in the field.</p> <p>If not specified, an unlimited number of characters may be entered into the field.</p>	RTExpr
name	<p>The name of the <code>ValidatingDisplayField</code> peer. This peer must be a child of the current parent <code>ContainerView</code> or <code>ViewBean</code>. The name is resolved relative to the current parent view.</p> <p>The view peer must be of type <code><JATO package>.view.DisplayField</code></p> <p>This name may be a qualified view path, using forward slashes ("/") as delimiters. All components in the path except the last must refer to a <code>ContainerView</code> or a derivative of <code>ContainerView</code>. Both relative and absolute paths are possible. If a name path begins with a forward slash, the name is assumed to be relative to the root view (the <code>ViewBean</code>). If the path does not begin with a forward slash, the name is assumed to refer to a child relative to the current container. Two dots ("..") may be used to refer to the container that is the parent of the current container.</p> <p>Examples:</p> <pre> /header/orderList/customerName (absolute from root view) orderList/customerName (relative to current container) ../footer/orderList/customerName (relative to parent) </pre>	Req

Attribute Name	Description	Notes
readOnly	Valid values: "true" or "false" A value of "true" specifies that the text contained within this element cannot be changed by the user. If not specified, "false" is assumed.	RTEExpr
size	Valid values: positive number Specifies the initial width of the control in number of characters. If not specified, the size attribute is omitted from the rendered HTML	RTEExpr
style	CSS styles to be applied to this HTML element.	RTEExpr
styleClass	CSS stylesheet class to be applied to this HTML element.	RTEExpr
tabIndex	Specifies the position of the current element in the tabbing order for the current document. This value must be a number between 0 and 32767.	
<p>This tag also supports the following JavaScript events: onBlur, onChange, onClick, onDbClick, onFocus, onKeyDown, onKeyPress, onKeyUp, onMouseDown, onMouseMove, onMouseOut, onMouseOver, onMouseUp, onSelect</p>		

(*See formatMask above.)

Format Type	Format Mask	Sample Value	Rendered Value
string	??-??-????	123456789	123-45-6789
string	??HELLO??	abcd	abHELLOcd
string	??HELLO\ \ ???	abcd	abHELLO?cd
dec	###0; (-#)	1000000.99	1000001
dec	#,##0.00; (-#)	1000000.99	1,000,000.99
dec	###0.00; (-#)	1000000.99	1000000.99
dec	###0; (-#)	-1000000.99	(-1000001)
dec	###0; (#)	-1000000.99	(1000001)
dec	#,##0.00; -#	-1000000.99	-1,000,000.99

content

<jato:content>

Denotes a section of JSP content that has `begin<name>Display()` and `end<name>Display()` events associated with it. The association of these events works in the same way as for visual tags, except that there need be no peer by the specified name in the parent container view.

The entire body content of this tag may be conditionally suppressed by returning a value of `false` from the corresponding `begin<name>Display()` event. If the body content is not suppressed, the processed body content will be provided to the `end<name>Display()` event. The body content will be fully processed, including any embedded JSP or Web Application Framework/JATO tags, before being passed to this event.

Unlike the firing of display events for the visual tags, which may be suppressed by a combination of attributes on the tag itself or its parent tag, the firing of display events for this tag always occurs. Therefore, for efficiency reasons, developers should always implement at least one of the display events for a content tag—if no events are implemented, there is no reason to use the tag.

This tag is only valid when enclosed by a `useViewBean`, `containerView`, or `tiledView` tag.

Example:

```
...
Username: <jato:textField name="username"/>
<jato:content name="superuser">
<br>
```

```
Password: <jato:textField name="password"/>
</jato:content>
<hr>
...
```

The above would be rendered into the following HTML if and only if the `beginSuperuserDisplay()` event in the current parent view returned `true`:

```
...
Username: <input type="text" name="PageFoo.username" value="scott">
<br>
Password: <input type="text" name="PageFoo.password" value="tiger">
<hr>
...
```

The `endSuperuserDisplay()` event would receive the following HTML as a method parameter:

```
<br>
Password: <input type="text" name="PageFoo.password" value="tiger">
```

If the `beginSuperuserDisplay()` event instead returned `false`, the following would be rendered:

```
...
Username: <input type="text" name="PageFoo.username" value="scott">
<hr>
...
```

The `endSuperuserDisplay()` event would not be invoked.

Attribute Name	Description	Notes
name	The name of the content section, used to fire display events in the current parent view.	Req
roles	A comma delimited list of role names. Role names must be limited to those declared in the deployment descriptor file (<code>web.xml</code> in the WEB-INF directory of the Web application). When this attribute is present, the display event is not fired. If any of the roles in the list is found in the roles of the authenticated principal (user) of the current request, then the body content of this tag will be displayed. In other words, unless the logged in user has at least one of the roles listed, the content in the JSP wrapped by this tag will be skipped.	
negate	<p>Valid values: "true" or "false"</p> <p>A value of "true" negates or reverses the result of the test done by the tag based on display event or role authorization. For example, if the display event returns false or if the roles listed under the roles attribute are not satisfied, then the body of the content tag will still be displayed.</p> <p>If not specified, "false" is assumed.</p>	

pagelet

<jato:pagelet>

Allows Web Application Framework/JATO tags in an included JSP fragment to inherit the enclosing page's container view scope, and thus be included at request-time.

This tag is only valid when used inside an included JSP fragment (also known as a *pagelet*). The **pagelet** tag should enclose all other Web Application Framework/JATO tags in the pagelet.

Example:

EnclosingPage.jsp

```
<%@page info="E0130" language="java"%>
<%@taglib uri="/WEB-INF/jato.tld" prefix="jato"%>

<jato:useViewBean ...>
<jato:containerView name="foo">
    ...
    <jsp:include page="MyPagelet.jsp"/>
    ...
</jato:containerView>
</jato:useViewBean>
```

MyPagelet.jsp

```
<%@page info="MyPagelet" language="java"%>
<%@taglib uri="/WEB-INF/jato.tld" prefix="jato"%>

<jato:pagelet>
  <jato:combobox name="month"/>...
</jato:pagelet>
```

treeNode

<jato:treeNode>

Denotes an arbitrary JSP content section that will be rendered for any matching tree nodes. There may be more than one `treeNode` tag contained in a `treeView` tag. With each cycle of the `TreeView` traversal, any contained `treeNode` tags are evaluated for a match against the current node. If the `treeNode` tag matches the current node, then the content of the `treeNode` tag is rendered.

A `treeNode` tag is said to "match" the current `TreeView` node if all of the following is true (each comparison is considered true if the tag attribute is unspecified).

- Current node name matches tag attribute "nodeName"
- Current node type matches tag attribute "nodeType"
- Current node children matches tag attribute "hasChildren"
- Current node expanded state matches tag attribute "isExpanded"

Note that the `nodeName` and `nodeType` attributes support comma delimited lists of potential match criteria. The expanded state of the current node is determined by end user interaction with the **treeNodeHandle** tags associated with the current node.

This tag is only valid when ultimately enclosed by a **treeView** tag; it does not have to be a direct child tag of the enclosing `treeView` tag, and can in fact be arbitrarily nested below the **treeView** parent tag, even within other **treeNode** tags.

Example:

```
...  
<jato:treeView name="treeMenu">
```

```

<jato:treeNode nodeName="menuCategory" hasChildren="true" >
  <jato:href name="category"/>
</jato:treeNode>

<jato:treeNode nodeName="menuCategory" hasChildren="false"
>
  <jato:text name="category"/>
</jato:treeNode>

<jato:treeNode nodeName="menuChoice,menuItem">
  <jato:href name="item"/>
</jato:treeNode>

</jato:treeView>

...

```

Attribute Name	Description	Notes
nodeName	Valid values: Arbitrary string of comma delimited node names. Specifies a match on current node name. The attribute may specify a comma delimited list of node names with the intent of matching the current node against any one of the specified names. If not specified, this does not factor in node match evaluation.	RTEExpr
nodeType	Valid values: Arbitrary string of comma delimited node types. Specifies a match on current node type. The attribute may specify a comma delimited list of node types with the intent of matching the current node against any one of the specified types. If not specified, this does not factor in node match evaluation.	RTEExpr
hasChildren	Valid values: "true" or "false" A value of "true" specifies that the current tag should only match the current node if the current node has children. If not specified, this does not factor in node match evaluation.	RTEExpr
isExpanded	Valid values: "true" or "false" A value of "true" specifies that the current tag should only match the current node if the current node is already "expanded". If not specified, this does not factor in node match evaluation.	RTEExpr

treeLevel

<jato:treeLevel>

Denotes an arbitrary JSP content section that will be rendered "level times" for matching node levels. There may be more than one `treeLevel` tag contained in a `treeView` tag. With each cycle of the `TreeView` traversal, any contained `treeLevel` tags are evaluated for a match with the current node depth level. If the `treeLevel` tag matches the current node depth level then the content of the tag is rendered a number of times equal to the current level.

A `treeLevel` tag is said to "match" the current level if all of the following is true (each comparison is considered true if the tag attribute is unspecified):

- Current level \geq tag attribute "minLevel"
- Current level \leq tag attribute "maxLevel"

If the `treeLevel` tag matches the current tree depth level, then the body content will be processed a number of times equal to the current level, including any embedded JSP or Web Application Framework/JATO tags. For example, if the current node level is 4, then the body content of any matching `treeLevel` tags would be rendered 4 times for the current node. Typically, one uses `treeLevel` tags to provide level-specific indentation or other structural markup.

This tag is only valid when ultimately enclosed by a `treeView` tag; it does not have to be a direct child tag of the enclosing `treeView` tag, and can in fact be arbitrarily nested below the `treeView` parent tag.

Example:

```
<jato:treeView name="treeMenu">
```

```

<!-- Add a space for each level of the current node -->
<jato:treeLevel>
  &nbsp;
</jato:treeLevel>

...

</jato:treeView>

```

Attribute Name	Description	Notes
minLevel	Valid values: positive number Specifies the minimum node depth level at which to match.	RTEExpr
maxLevel	Valid values: positive number Specifies the maximum node depth level at which to match.	RTEExpr
offset	Valid values: positive or negative integer Specifies a positive or negative node level depth modifier to apply when rendering this tag. An offset of $-n$ will cause the tag to render (level- n) times, whereas an offset of n will cause the tag to render (level+ n) times.	RTEExpr

treeNodeHandle

<jato:treeNodeHandle>

Denotes a visual control that end users can select to toggle the expanded/collapsed state of a particular tree node. This tag will be rendered as an HTML `...` element, with an implicit URL generated by the tag and handled by the parent tree view automatically when activated.

This tag may contain body content. This content will be included between the `<a>` and `` tags of the rendered element and thus appear as the visible, clickable HREF element.

This tag is only valid when ultimately enclosed by a `treeView` tag. This tag may be directly parented by a `treeNode` tag.

Example:

...

```
<jato:treeNodeHandle>
  
</jato:treeNodeHandle>
```

...

Attribute Name	Description	Notes
accessKey	Assigns an access key to the rendered HTML element. In general, the access key should be included in label text or wherever the access key is to apply.	RTEExpr
disabled	Valid values: "true" or "false" Disables the HTML control for user input. <ul style="list-style-type: none"> • Disabled controls do not receive focus • Disabled controls are skipped in tabbing navigation If not specified, "false" is assumed.	RTEExpr
elementId	Assigns a name to an element. This name must be unique in an HTML document. <i>Please note, this corresponds to the HTML "id" attribute. It has no specific meaning to the Web Application Framework.</i> The id attribute has several roles in HTML: <ul style="list-style-type: none"> • As a style sheet selector • As a target anchor for hypertext links • As a means to reference a particular element from a script 	RTEExpr
extraHtml	Arbitrary HTML that will be rendered just prior to the closing ">" of this HTML element	RTEExpr
queryParams	Specifies an arbitrary number of name value pairs which will be appended to the HREF's query string. The name value pairs should follow the format "name=value", where value is URL encoded by the JSP author as needed. Additional name value pairs must be delimited by the '&' character. Example: queryParams="fname=Mike&lname=Jones"	RTEExpr
style	CSS styles to be applied to this HTML element.	RTEExpr
styleClass	CSS stylesheet class to be applied to this HTML element.	RTEExpr
tabIndex	Specifies the position of the current element in the tabbing order for the current document. This value must be a number between 0 and 32767.	
title	Offers advisory information about this element.	RTEExpr
trim	Valid values: "true" or "false" A value of "true" specifies that the tag content should be trimmed of all leading and trailing whitespace. A value of "false" preserves the whitespace. If not specified, "true" is assumed. This tag also supports the following JavaScript events: onBlur, onChange, onClick, onDbClick, onFocus, onKeyDown, onKeyPress, onKeyUp, onMouseDown, onMouseMove, onMouseOut, onMouseOver, onMouseUp, onSelect	

datasetLocator

<jatox:datasetLocator>

Renders a static text display of the current location and number of records currently being displayed by its targeted container view (which is bound to some DatasetModel). This tag is only valid when nested inside of a useViewBean or containerView or tiledView tag body, although, it typically will not be used inside a tiledView tag.

This tag is only valid when enclosed by both an HTML <form> element and a useViewBean, containerView, or tiledView tag. This tag may not contain any body content.

Example:

```
<jatox:datasetLocator name="datasetLocator1">  
<jato:containerView name="datasetLocator1">  
<%@ include file="/com/sun/jatox/view/DatasetLocator.jsp" %>  
</jato:containerView>  
</jatox:datasetLocator>
```

The above would be rendered into the following various forms of HTML:

1. Records X to Y of Z (example: Records 11 to 20 of 53)
2. Records X to Y (example: Records 6 to 10)
3. Record X of Z (example: Record 23 of 41)
4. Record X (example: Record 14)

The included file is a precomposed pagelet that can be customized with some limits. Tweaking the wrong tag could produce undesired or unpredictable behavior. It is encouraged that only static content be modified or additional HTML markup be added when customizing this file.

Attribute Name	Description	Notes
name	<p>The name of the <code>DisplayField</code> peer. This peer must be a child of the current parent <code>ContainerView</code>, <code>TiledView</code>, or <code>ViewBean</code>. The <code>DisplayField</code> name is resolved relative to the current parent view.</p> <p>The view peer must be of type <code><JATOX package>.view.DatasetLocator</code></p> <p>This name may be a qualified view path, using forward slashes ("/") as delimiters. All components in the path except the last must refer to a <code>ContainerView</code> or a derivative of <code>ContainerView</code> (such as <code>TiledView</code>). Both relative and absolute paths are possible. If a name path begins with a forward slash, the name is assumed to be relative to the root view (the <code>ViewBean</code>). If the path does not begin with a forward slash, the name is assumed to refer to a child relative to the current container. Two dots ("..") may be used to refer to the container that is the parent of the current container.</p> <p>Examples:</p> <ul style="list-style-type: none"><code>/header/orderList/customerName</code> (absolute from root view)<code>orderList/customerName</code> (relative to current container)<code>../footer/orderList/customerName</code> (relative to parent) <p>This tag does not support any JavaScript events.</p>	Req

datasetNavigator

<jatox:datasetNavigator>

Renders four graphical buttons that represent the four dataset navigation actions: First, Previous, Next and Last. The actions work against the targeted container view that is configured in the component. This tag is only valid when nested inside of a `useViewBean` or `containerView` or `tiledView` tag body, although, it typically will not be used inside a `tiledView` tag.

The First and Previous buttons can be displayed as disabled or hidden (not displayed at all) when the first record displayed is determined to be the first record in the dataset. Same goes for the Next and Last buttons when it is determined that the last record (no more records) is being displayed. If for some reason no data was returned, the text (no data found) will be displayed instead of the buttons.

This tag is only valid when enclosed by both an HTML `<form>` element and a `useViewBean`, `containerView`, or `tiledView` tag. This tag may not contain any body content.

Example:

```
<jatox:datasetNavigator name="datasetNavigator1">  
<jato:containerView name="datasetNavigator1">  
<%@ include file="/com/sun/jatox/view/DatasetNavigator.jsp" %>  
</jato:containerView>  
</jatox:datasetNavigator>
```

The above would be rendered into the following various forms of HTML (bold indicates enabled, non-bold indicates disabled):

1. [`<<`] [`<`] [`>`] [`>>`]
2. [`<<`] [`<`] [`>`] [`>>`]
3. [`<<`] [`<`] [`>`] [`>>`]
4. [`<<`] [`<`]
5. [`>`] [`>>`]

The included file is a precomposed pagelet that can be customized with some limits. Tweaking the wrong tag could produce undesired or unpredictable behavior. It is encouraged that only static content be modified or additional HTML markup be added when customizing this file.

Attribute Name	Description	Notes
name	<p>The name of the <code>DisplayField</code> peer. This peer must be a child of the current parent <code>ContainerView</code>, <code>TiledView</code>, or <code>ViewBean</code>. The <code>DisplayField</code> name is resolved relative to the current parent view.</p> <p>The view peer must be of type <code><JATOX package>.view.DatasetLocator</code></p> <p>This name may be a qualified view path, using forward slashes ("<code>/</code>") as delimiters. All components in the path except the last must refer to a <code>ContainerView</code> or a derivative of <code>ContainerView</code> (such as <code>TiledView</code>). Both relative and absolute paths are possible. If a name path begins with a forward slash, the name is assumed to be relative to the root view (the <code>ViewBean</code>). If the path does not begin with a forward slash, the name is assumed to refer to a child relative to the current container. Two dots ("<code>..</code>") may be used to refer to the container that is the parent of the current container.</p> <p>Examples:</p> <p><code>/header/orderList/customerName</code> (absolute from root view)</p> <p><code>orderList/customerName</code> (relative to current container)</p> <p><code>../footer/orderList/customerName</code> (relative to parent)</p> <p>This tag does not support any JavaScript events.</p>	Req

dateTimeView

<jatox:dateTimeView>

Renders six combo boxes (month, day, year range, hour, minute and am/pm options), and optionally, a graphic button that pops up a small calendar from which a date can be selected and the month/day/year combo boxes will reflect this selection. This tag is only valid when nested inside of a `useViewBean` or `containerView` or `tiledView` tag body.

If the popup calendar property is enabled for this component, then the tag will require JavaScript. This JavaScript is automatically included in the JSP's header section as follows:

```
<script language="JavaScript" type="text/javascript" src='<%=
request.getContextPath()+
"/com/sun/jatox/view/popupminicalendar.js"%>'></script>
```

This tag is only valid when enclosed by both an HTML `<form>` element and a `useViewBean`, `containerView`, or `tiledView` tag. This tag may not contain any body content.

Example:

```
<jatox:dateTimeView name="dateTimeView1"/>
```

The above would be rendered into the following HTML (too much HTML to list here):

```
[month combo box] [day combo box] [year range combo box] [hour
combo box]: [minute combo box] [am/pm combo box] [popup
calendar graphic button]
```

Attribute Name	Description	Notes
name	<p>The name of the <code>DisplayField</code> peer. This peer must be a child of the current parent <code>ContainerView</code>, <code>TiledView</code>, or <code>ViewBean</code>. The <code>DisplayField</code> name is resolved relative to the current parent view.</p> <p>The view peer must be of type <code><JATOX package>.view.DateView</code></p> <p>This name may be a qualified view path, using forward slashes ("<code>/</code>") as delimiters. All components in the path except the last must refer to a <code>ContainerView</code> or a derivative of <code>ContainerView</code> (such as <code>TiledView</code>). Both relative and absolute paths are possible. If a name path begins with a forward slash, the name is assumed to be relative to the root view (the <code>ViewBean</code>). If the path does not begin with a forward slash, the name is assumed to refer to a child relative to the current container. Two dots ("<code>..</code>") may be used to refer to the container that is the parent of the current container.</p> <p>Examples:</p> <p><code>/header/orderList/customerName</code> (absolute from root view)</p> <p><code>orderList/customerName</code> (relative to current container)</p> <p><code>../footer/orderList/customerName</code> (relative to parent)</p> <p>This tag does not support any JavaScript events.</p>	Req

dateView

<jatox:dateView>

Renders 3 combo boxes (month, day and year range options), and optionally, a graphic button that pops up a small calendar from which a date can be selected and the month/day/year combo boxes will reflect this selection. This tag is only valid when nested inside of a `useViewBean` or `containerView` or `tiledView` tag body.

If the popup calendar property is enabled for this component, then the tag will require JavaScript. This JavaScript is automatically included in the JSP's header section as follows:

```
<script language="JavaScript" type="text/javascript" src='<%=
request.getContextPath()+
"/com/sun/jatox/view/popupminicalendar.js"%>'></script>
```

This tag is only valid when enclosed by both an HTML `<form>` element and a `useViewBean`, `containerView`, or `tiledView` tag. This tag may not contain any body content.

Example:

```
<jatox:dateView name="dateView1"/>
```

The above would be rendered into the following HTML (too much HTML to list here):

```
[month combo box] [day combo box] [year range combo box]
[popup calendar graphic button]
```

Attribute Name	Description	Notes
name	<p>The name of the <code>DisplayField</code> peer. This peer must be a child of the current parent <code>ContainerView</code>, <code>TiledView</code>, or <code>ViewBean</code>. The <code>DisplayField</code> name is resolved relative to the current parent view.</p> <p>The view peer must be of type <code><JATOX package>.view.DateView</code></p> <p>This name may be a qualified view path, using forward slashes ("/") as delimiters. All components in the path except the last must refer to a <code>ContainerView</code> or a derivative of <code>ContainerView</code> (such as <code>TiledView</code>). Both relative and absolute paths are possible. If a name path begins with a forward slash, the name is assumed to be relative to the root view (the <code>ViewBean</code>). If the path does not begin with a forward slash, the name is assumed to refer to a child relative to the current container. Two dots ("..") may be used to refer to the container that is the parent of the current container.</p> <p>Examples:</p> <p><code>/header/orderList/customerName</code> (absolute from root view)</p> <p><code>orderList/customerName</code> (relative to current container)</p> <p><code>../footer/orderList/customerName</code> (relative to parent)</p> <p>This tag does not support any JavaScript events.</p>	Req

goToPage

<jatox:goToPage>

Renders HTML as:

```
<a href=" ../<ParentContainerView>;jsessionid=
82BEsadfaeae?<FullQualifiedName>=&jato.pageSession=">Click Here</a>
```

This tag is only valid when enclosed by a `useViewBean`, `containerView`, or `tiledView` tag. This tag may contain body content. This content will be included between the `<a>` and `` tags of the rendered element and thus appear as the visible, clickable href element.

Example:

```
<jatox:goToPage name="goToHref1">Click Here</jato:href>
```

The attributes of this tag are identical to that of the `jato:href` tag with no additional attributes.

Attribute Name	Description	Notes
accessKey	Assigns an access key to the rendered HTML element. In general, the access key should be included in label text or wherever the access key is to apply.	RTExpr
anchor	The anchor value to append to the generated HREF.	RTExpr
disabled	Valid values: "true" or "false" Disables the HTML control for user input. <ul style="list-style-type: none"> • Disabled controls do not receive focus • Disabled controls are skipped in tabbing navigation If not specified, "false" is assumed.	RTExpr

Attribute Name	Description	Notes
defaultValue	<p>The value to use, if the <code>DisplayField</code> view peer's value is null.</p> <p>For an href tag, "value" refers to a special name-value pair that will be automatically included as part of the HREF's query string. The format for the implicit name-value pair is the following:</p> <pre>...?<qualified display field name>=<value>&...</pre> <p>This mechanism allows an href to have a display string as well as a value important to the application.</p>	RTEExpr
elementId	<p>Assigns a name to an element. This name must be unique in an HTML document. <i>Please note, this corresponds to the HTML "id" attribute. It has no specific meaning to the Web Application Framework.</i></p> <p>The id attribute has several roles in HTML:</p> <ul style="list-style-type: none"> • As a style sheet selector • As a target anchor for hypertext links • As a means to reference a particular element from a script 	RTEExpr
extraHtml	Arbitrary HTML that will be rendered just prior to the closing ">" of this HTML element	RTEExpr
name	<p>The name of the <code>DisplayField</code> peer. This peer must be a child of the current parent <code>ContainerView</code>, <code>TiledView</code>, or <code>ViewBean</code>. The <code>DisplayField</code> name is resolved relative to the current parent view.</p> <p>The view peer must be of type <code><JATO package>.view.CommandField</code></p> <p>This name may be a qualified view path, using forward slashes ("/") as delimiters. All components in the path except the last must refer to a <code>ContainerView</code> or a derivative of <code>ContainerView</code> (such as <code>TiledView</code>). Both relative and absolute paths are possible. If a name path begins with a forward slash, the name is assumed to be relative to the root view (the <code>ViewBean</code>). If the path does not begin with a forward slash, the name is assumed to refer to a child relative to the current container. Two dots ("..") may be used to refer to the container that is the parent of the current container.</p> <p>Examples:</p> <pre>/header/orderList/customerName (absolute from root view) orderList/customerName (relative to current container) ../footer/orderList/customerName (relative to parent)</pre>	Req

maskedTextField

<jatox:maskedTextField>

Renders HTML as:

```
<input type="text" name="<fieldName>" onkeypress = ' return  
anyMask(event, "null"); '>
```

The required JavaScript is automatically included in the in the header portion of the same JSP file.

```
<script language="javascript" src=  
'/<ServletContext>/com/sun/jatox/scripts/anyMask.js'></script>
```

This tag is only valid when enclosed by both an HTML <form> element and a useViewBean, containerView, or tiledView tag. This tag may not contain any body content.

Example:

```
<jatox:maskedTextField name="maskedTextField1"/>
```

The attributes of this tag are identical to that of the jato:textField tag with no additional attributes.

Attribute Name	Description	Notes
accessKey	Assigns an access key to the rendered HTML element. In general, the access key should be included in label text or wherever the access key is to apply.	RTEExpr
defaultValue	The value to use, if the <code>DisplayField</code> view peer's value is null. For a <code>textField</code> tag, "value" refers to the value that will be rendered in the HTML text element's "value" attribute	RTEExpr
disabled	Valid values: "true" or "false" Disables the HTML control for user input. <ul style="list-style-type: none"> • Disabled controls do not receive focus • Disabled controls are skipped in tabbing navigation If not specified, "false" is assumed.	RTEExpr
elementId	Assigns a name to an element. This name must be unique in an HTML document. <i>Please note, this corresponds to the HTML "id" attribute. It has no specific meaning to the Web Application Framework.</i> The id attribute has several roles in HTML: <ul style="list-style-type: none"> • As a style sheet selector • As a target anchor for hypertext links • As a means to reference a particular element from a script 	RTEExpr
extraHtml	Arbitrary HTML that will be rendered just prior to the closing ">" of this HTML element	RTEExpr
formatMask	The valid values for the <code>formatMask</code> attribute depend on the value of <code>formatType</code> attribute. For <code>formatType</code> values of "string" or "alpha": The format mask consists of any arbitrary combination of static text and the reserved metacharacter "?". Each "?" metacharacter serves as a placeholder for a single character in the value being formatted, proceeding from left to right. A literal "?" can be specified by escaping the "?" character with "\\?". For <code>formatType</code> values of "dec" or "curr": The format mask consists of two patterns separated by a semicolon: a primary pattern (specifies precision and thousandths separator) and a negative number pattern (see <code>java.text.DecimalFormat</code> for pattern details). (*See table below.)	

Attribute Name	Description	Notes
formatType	<p>Valid values: “string” (also “alpha”), “dec”, or “curr”, or “date”</p> <p>Each type has specific masking rules (see the <code>formatMask</code> attribute for details).</p> <p>A value of “string” supports the formatting of arbitrary text via the application of a simple text mask. This allows dynamic content to be arbitrarily combined with static text. A value of “alpha” is identical to “string”.</p> <p>A value of “dec” supports the formatting of numeric values via the application of a numeric mask.</p> <p>A value of “curr” supports the formatting of currency values via the application of a currency mask. The currency format should simply take a decimal specification and apply the locale specific currency format. Currently, currency formatting is not working due to an unresolved bug.</p> <p>A value of “date”, although legal, at present has no supporting date formatting implementation, due to the fact that the <code>java.text.DateFormat</code> class does not lend itself easily to the generic formatting support that a taglib implementation would require. We are investigating ways around this limitation.</p> <p>Therefore, it is suggested that developers wishing to apply date formatting implement the <code>begin<fieldName>Display(ChildDisplayEvent event)</code> and/or <code>end<fieldName>Display(ChildContentDisplayEvent event)</code> events and therein take explicit advantage of the date formatting support available in the <code>java.text.DateFormat</code> class.</p> <p>The Web Application Framework implementation of text formatting relies heavily on the <code>java.text</code> package and the format related classes therein. Please see that package for more information. See also the <code><JATO package>.util.HtmlUtil</code> class.</p>	
maxLength	<p>Valid values: positive number</p> <p>Specifies the maximum number of characters the user may enter in the field.</p> <p>If not specified, an unlimited number of characters may be entered into the field.</p>	RTExpr
name	<p>The name of the <code>DisplayField</code> peer. This peer must be a child of the current parent <code>ContainerView</code>, <code>TiledView</code>, or <code>ViewBean</code>. The <code>DisplayField</code> name is resolved relative to the current parent view.</p> <p>The view peer must be of type <code><JATO package>.view.DisplayField</code></p> <p>This name may be a qualified view path, using forward slashes (“/”) as delimiters. All components in the path except the last must refer to a <code>ContainerView</code> or a derivative of <code>ContainerView</code> (such as <code>TiledView</code>). Both relative and absolute paths are possible. If a name path begins with a forward slash, the name is assumed to be relative to the root view (the <code>ViewBean</code>). If the path does not begin with a forward slash, the name is assumed to refer to a child relative to the current container. Two dots (“..”) may be used to refer to the container that is the parent of the current container.</p> <p>Examples:</p> <p><code>/header/orderList/customerName</code> (absolute from root view)</p> <p><code>orderList/customerName</code> (relative to current container)</p> <p><code>../footer/orderList/customerName</code> (relative to parent)</p>	Req

Attribute Name	Description	Notes
readOnly	Valid values: "true" or "false" A value of "true" specifies that the text contained within this element cannot be changed by the user. If not specified, "false" is assumed.	RTEExpr
size	Valid values: positive number Specifies the initial width of the control in number of characters. If not specified, the size attribute is omitted from the rendered HTML	RTEExpr
style	CSS styles to be applied to this HTML element.	RTEExpr
styleClass	CSS stylesheet class to be applied to this HTML element.	RTEExpr
tabIndex	Specifies the position of the current element in the tabbing order for the current document. This value must be a number between 0 and 32767. This tag also supports the following JavaScript events: onBlur, onChange, onClick, onDbClick, onFocus, onKeyDown, onKeyPress, onKeyUp, onMouseDown, onMouseMove, onMouseOut, onMouseOver, onMouseUp, onSelect	

(*See formatMask above.)

Format Type	Format Mask	Sample Value	Rendered Value
string	??-??-????	123456789	123-45-6789
string	??HELLO??	abcd	abHELLOcd
string	??HELLO\\???	abcd	abHELLO?cd
dec	###0; (-#)	1000000.99	1000001
dec	#,##0.00; (-#)	1000000.99	1,000,000.99
dec	###0.00; (-#)	1000000.99	1000000.99
dec	###0; (-#)	-1000000.99	(-1000001)
dec	###0; (#)	-1000000.99	(1000001)
dec	#,##0.00; -#	-1000000.99	-1,000,000.99

menu

<jatox:menu>

This tag may not contain any body content.

Example:

```
<jatox:menu name="menu1">
```

Attribute Name	Description	Notes
name	<p>The name of the DisplayField peer. This peer must be a child of the current parent ContainerView, TiledView, or ViewBean. The DisplayField name is resolved relative to the current parent view.</p> <p>The view peer must be of type <JATO package>.view.Menu</p> <p>This name may be a qualified view path, using forward slashes ("/") as delimiters. All components in the path except the last must refer to a ContainerView or a derivative of ContainerView (such as TiledView). Both relative and absolute paths are possible. If a name path begins with a forward slash, the name is assumed to be relative to the root view (the ViewBean). If the path does not begin with a forward slash, the name is assumed to refer to a child relative to the current container. Two dots ("..") may be used to refer to the container that is the parent of the current container.</p> <p>Examples:</p> <p>/header/orderList/customerName (absolute from root view)</p> <p>orderList/customerName (relative to current container)</p> <p>../footer/orderList/customerName (relative to parent)</p> <p>This tag does not support any JavaScript events.</p>	Req

staticBreadCrumb

<jatox:staticBreadCrumb>

This tag may not contain any body content.

Example:

```
<jatox:staticBreadCrumb name="staticBreadCrumb1">
```

Attribute Name	Description	Notes
name	<p>The name of the DisplayField peer. This peer must be a child of the current parent ContainerView, TiledView, or ViewBean. The DisplayField name is resolved relative to the current parent view.</p> <p>The view peer must be of type <JATO package>.view.Menu</p> <p>This name may be a qualified view path, using forward slashes ("/") as delimiters. All components in the path except the last must refer to a ContainerView or a derivative of ContainerView (such as TiledView). Both relative and absolute paths are possible. If a name path begins with a forward slash, the name is assumed to be relative to the root view (the ViewBean). If the path does not begin with a forward slash, the name is assumed to refer to a child relative to the current container. Two dots ("..") may be used to refer to the container that is the parent of the current container.</p> <p>Examples:</p> <p>/header/orderList/customerName (absolute from root view)</p> <p>orderList/customerName (relative to current container)</p> <p>../footer/orderList/customerName (relative to parent)</p> <p>This tag does not support any JavaScript events.</p>	Req

timeView

<jatox:timeView>

Renders 3 combo boxes (hour, minute and am/pm options). This tag is only valid when nested inside of a `useViewBean` or `containerView` or `tiledView` tag body.

This tag is only valid when enclosed by both an HTML `<form>` element and a `useViewBean`, `containerView`, or `tiledView` tag. This tag may not contain any body content.

Example:

```
<jatox:dataView name="timeView1"/>
```

The above would be rendered into the following HTML (too much HTML to list here):

```
[hour combo box]: [minute combo box] [am/pm combo box]
```

Attribute Name	Description	Notes
name	<p>The name of the <code>DisplayField</code> peer. This peer must be a child of the current parent <code>ContainerView</code>, <code>TiledView</code>, or <code>ViewBean</code>. The <code>DisplayField</code> name is resolved relative to the current parent view.</p> <p>The view peer must be of type <code><JATO package>.view.TimeView</code></p> <p>This name may be a qualified view path, using forward slashes ("<code>/</code>") as delimiters. All components in the path except the last must refer to a <code>ContainerView</code> or a derivative of <code>ContainerView</code> (such as <code>TiledView</code>). Both relative and absolute paths are possible. If a name path begins with a forward slash, the name is assumed to be relative to the root view (the <code>ViewBean</code>). If the path does not begin with a forward slash, the name is assumed to refer to a child relative to the current container. Two dots ("<code>..</code>") may be used to refer to the container that is the parent of the current container.</p> <p>Examples:</p> <p><code>/header/orderList/customerName</code> (absolute from root view)</p> <p><code>orderList/customerName</code> (relative to current container)</p> <p><code>../footer/orderList/customerName</code> (relative to parent)</p> <p>This tag does not support any JavaScript events.</p>	Req

Index

A

application, use tag library in, 16

B

Before You Begin, 9 to 13

button, 39

C

checkbox, 43

combobox, 47

containerView, 23

content, 103

Context Tags, 19

D

datasetLocator, 115

datasetNavigator, 117

dateTimeView, 119

dateView, 121

F

fileUpload, 51

form, 55

frameSrc, 59

G

getDisplayFieldValue, 31

getModelFieldValue, 33

gotoPage, 123

H

hidden, 61

href, 63

I

image, 67

J

JSPs, non-HTML, use tag library in, 16

L

legend, 22

listbox, 71

M

maskedTextField, 125

menu, 129

Miscellaneous Tags, 22

N

non-HTML JSPs, use tag library in, 16

P

pagelet, 107

pagelets, including JSP content, 17

password, 75

R

radioButtons, 79

Req, 22

RTEExpr, 22

S

setDisplayFieldValue, 35
setModelFieldValue, 37
staticBreadCrumb, 131

T

Tag Library Overview, 15
tag library, use in your application, 16
tag overview, 19
Tag Reference, 22
text, 83
textArea, 87
textField, 91
tiledView, 25
timeView, 133
translation-time includes, 17
Tree Tags, 22
treeLevel, 111
treeNode, 109
treeNodeHandle, 113
treeView, 27

U

useViewBean, 29

V

validatingTextArea, 95
validatingTextField, 99
Value Tags, 20
Visual (HTML) Tags, 20

W

Web Application Framework, how the tag library
interacts with, 15