

# The Salmon Open Framework for Internet Applications (SOFIA)

---

Tag Library Specification

Salmon LLC  
220 Old Country Road  
Mineola, NY 11501-4271  
(516) 742-7888

Version 2.3

## Tag Index

HTML Tags .....	5
a .....	5
applet .....	5
box .....	6
brand .....	6
body .....	7
calendar .....	7
container .....	8
cookiecrumble .....	8
datatable .....	8
datasource .....	9
detailform .....	13
detailformdisplaybox .....	13
displaybox .....	14
div .....	15
expand .....	15
expandall .....	15
font .....	15
form .....	16
img .....	16
input .....	17
lookup .....	21
listform .....	22
listformdisplaybox .....	22
navbar .....	24
page .....	25
param .....	26
property .....	26
raw .....	26
rowselector .....	27
rowhighlighter .....	27
searchformdisplaybox .....	27
style .....	28
subreportcontainer .....	28
table .....	29
text .....	29
td .....	29
tr .....	30
tree .....	30
validator .....	31
WML Tags .....	35
a .....	35
card .....	35
datasource .....	35
do .....	38
go .....	39
img .....	39
input .....	40
page .....	40

param .....	41
property .....	41
raw.....	41
select.....	42
submit.....	42
text.....	43

## Copyright Notice

Licensed Material - Property of Salmon LLC  
(C) Copyright Salmon LLC. 1999 – 2004

The Salmon Open Framework for Internet Applications (SOFIA) has been released under dual licensing schemes. Unregistered versions of SOFIA fall under the GNU General Public License. This in effect means that non-internal projects that you create with SOFIA will also have to be released under the GNU GPL as open source. Once you register for SOFIA, you will be able to use it under the Salmon license as well. The Salmon license allows you to distribute SOFIA binaries with your SOFIA application without having to make your software open source. Registration is free. To register your copy of SOFIA, please visit <http://www.salmonllc.com/register>.

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License version 2 as published by the Free Software Foundation; Registered versions can be distributed under the terms of the Salmon Software License version 1 as well;

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

To get a copy of the GNU GPL, write to the Free Software Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA.

To view the full GNU GPL and Salmon Software License on-line please visit <http://www.salmonllc.com/license>.

This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>).

## HTML Tags

To use SOFIA HTML tags, include taglib.tld in a JSP taglib tag in your page.

### a

Provides an anchor tag. Used to identify an area on the page or to make an area on the page an underline link. This will create a JspLink component.

Attributes:

- name: (Required) The name of the component
- href: (Required) The url this link will zoom to if the user clicks anything in the component.
- classname: (Optional) The name of the style to use
- onclick: (Optional) Action on Click
- onmouseout: (Optional) action on mouse out
- onmouseover: (Optional) action on mouse over
- target: (Optional) Target for href
- datasource: (optional) The datastore expression to compute the href attribute. If used it will overwrite the attribute with the results of the expression.
- title (optional) A title that will be displayed when the user hovers over the link with the mouse pointer.

### applet

Provides an applet tag. Used to embed an applet in a page. The applet can use the Java plug-in VM or not depending on a parameter. In addition, the applet will be automatically be passed a parameter: `sessionId` which will identify the session id for the page that loaded the applet. This parameter can be used for proxy datastores and other objects that use the server session from the client.

Attributes:

- name: (Required) The name of the component
- code: (Required) The name of the Java class to run
- align: Sets the alignment for the applet. Valid values are LEFT, RIGHT, TOP, ABSMIDDLE, ABSBOTTOM, TEXTTOP, MIDDLE, BASELINE, BOTTOM
- alt: Text to display for browsers that don't support java
- appletdatasource: used to specify the datasource which is used for evaluating the values of the other attributes as expressions
- archive: a comma separated list of jars that contain the code for the applet to run.
- codebase: Gets the directory containing the applet class file and any resources the applet needs. The value is a URL for an absolute or a relative pathname. An absolute URL is used as is without modification and is not affected by the document's BASE tag. A relative CODEBASE attribute is relative to the document's base URL defined by the BASE tag. If the document does not define a BASE tag, it is relative to the directory containing the HTML file.
- height: The height of the applet in pixels
- hspace: The horizontal spacing
- interactwithform: Set the value to true if you want the applet to interact with the HTML form that it's in. Interaction consists of the applet returning a string value to the page when it is submitted (which can be queried via the `getValue()` method in the component representing this tag) and the page returning the state string to the applet which tells it how to initialize. The applet itself must have 2 public

methods for this to work: `public String getValue()` - This will return the value that the applet should return to the page. `public String getState()` - This method will return a state string to the page. The page will return this string back to the applet the next time it goes back to the browser as an applet parameter: "state". The applet can then restore it's state using the state string specified. This is so the applet can be made to retain it's state across page submits. In addition, two other parameters will be passed to each applet: "value" contains the value of the applet on the server and "servletSessionID" contains the session id of the session on the server.

- mayscript: This method sets whether the applet can be accessed via javaScript in the browser
- object: Sets the object attribute of the applet. It specifies the serialized Java Applet
- pluginclassid: Sets the Java Plug-in Class ID. Used by I.E. <Object Tag>
- plugincodebase: Sets the Java Plug-in Code Base. Used by I.E. <Object Tag>
- pluginspage: Sets the Java Plug-in Download Page. Used by Netscape <Embed Tag>
- state: Sets the original state string for the applet. See description of the interact with form attribute for more info
- useplugin: A true or false value indicating whether or not to use the plug-in
- visible: A true or false indicating whether or not the applet should display
- vspace: The vertical space around the applet
- width: The width of the applet in pixels

## box

This tag is used to draw a box border around the components within it.

Attributes:

- name: (Required) The name of the component
- width (optional) : sets the width of the table used to dray the box.
- linewidth (optional) : sets the width in pixels of the line that makes up the box
- bgcolor (optional) : sets the background color for the box contents.
- linecolor (optional) : sets color of the line that makes up the box.
- margin (optional): sets the width of the space in pixels between the box and its contents.

## brand

This tag is used to display a different image from a series of images on every request to the page. It cycles through the different images to display a new image wherever this tag is placed on the page, so as to give a more dynamic look and feel to the website.

Attributes:

- name: (Required) The name of the component
- tablename (optional) : sets the name of the table in the the database. It is assumed that a table named "brand" will be used to store the information about the Branding Images if this attribute is not gived/used.
- imagecolumn (optional) : sets the name of the column in the database table that will hold the names of the branding images. It is assumed that the default name of the column that holds the image names is "image"

- imagealtcolumn (optional) : sets the name of the column in the database table that will hold the alt text of the branding images. It is assumed that the default name of the column that holds the image alt values is "image\_alt"
- imagedir (optional) : sets the directory on the file system where the branding images physically reside. It is assumed that the default directory in which the branding images are stored is "images"

## body

This tag is used to replace the html body tag. It will generate the same HTML as a standard HTML body tag, but its attributes will be initialized from property files. Although this is a body tag in HTML is empty in the JSP so it can allow jsp includes to be nested within it. It should be closed with the <prefix:endbody/> tag.

Attributes (All optional):

- background: The URL of the background image for the page
- bgcolor: The color for the background
- text: The default text color
- link: The default color for links on the page
- alink: The default color for active links on the page
- vlink: The default color for visited links on the page
- onload: Javascript function to execute when the page loads
- classname: The style class to use for the page
- leftmargin: The pages left margin
- rightmargin: The pages left margin
- topmargin: The pages top margin
- marginwidth: The width of the margin
- marginheight: The height of the margin

## calendar

Creates a visual representation of a calendar in HTML.

Attributes

- name: (required) The name of the component
- headingbgcolor: (optional) The background color for the heading
- headingfgcolor: (optional) The foreground color for the heading
- weekbgcolor: (optional) The background color for the weeks
- weekfgcolor: (optional) The foreground color for the weeks
- daybgcolor: (optional) The background color for the days
- dayfgemphasiscolor: (optional) The foreground color for the days that are emphasized
- dayfgnormalcolor: (optional) The foreground color for the days that are not emphasized
- dayfgcurrentcolor: (optional) The foreground color for today's date
- fontface (optional) The font face to use for all text
- fontsize (optional) The font size to use for all text
- displaysize (optional) LARGE for a large calendar or SMALL for a small one
- currentmonth (optional) The month to display on the calendar (0=January, 11=December)
- currentyear (optional) The year to display on the calendar
- width (optional) The width of the table displaying the calendar

- border (optional) The width of the calendar's border in pixels

### container

This tag is used to allow HTML and framework tags to be treated as a group for programmatic access in the controller. All tags nested in the container can be made invisible or disabled by calling the setVisible or setEnabled method of the JspContainer class that this creates.

Attributes:

- name (required): The name of the component.
- visible (optional): Whether or not the item is initially visible
- enabled (optional): Whether or not the components in the container are enabled.
- containerclass (optional): A classname of a subclass of JspContainer that will be used in place of JspContainer for this component.

### cookiecrumble

Allows you to create a tag that you can specify browser pages as a series of links separated by colons. Ex. Home:product:cameras:digital

Attributes

- name (required): The name of the component
- linkfont (optional): The font to use for items in the cookie crumble that are links
- font: (optional): font to used for items in the crumble that are links

Nested Tag:

**crumble:** Adds crumbles to the cookiecrumble tag

Attributes:

- key: (Required) This value is used as a lookup. ( I recommend that these be as short as possible)
- display: (Required) The display value for the crumble
- href: (Optional) The href. For example, the last crumble will probably be the page that you're on so a href would be redundant.

### datatable

This tag will cause iterate through all the rows in a datasource and provide a grid presentation of the data in it. It supports headings, striping of repeating rows, automatic sorting of the table by clicking on one of the headings and paging functionality.

Attributes:

- align (optional): "LEFT|RIGHT"
- border (optional): width of the boder in pixels
- cellpadding (optional): the cell padding of each cell in the table
- cellspacing (optional):the spacing of each cell in the table
- height (optional): the height of the table
- width (optional):the width of the table in pixels or percent
- cols (optional):The number of columns in the table

- hspace (optional):the horizontal margin
- vspace (optional):the vertical margin
- name (required): The name of the component.
- theme (optional): The theme to of the datatable. Used to find look and feel defaults for the table.
- groupby (optional): A comma separated list of component names that will be used for grouping the rows in the table. The table should be sorted by these components. Each time the group changes, the table will generate a groupheader band and a groupdetail band.
- groupbycont (optional): A component name that will be set to visible if a group begins on a page prior to the current one displayed. It give you the opportunity to displayed a “continued” message.
- rowsperpage (optional): List the number of rows to appear on each page.
- pageselectortype (optional): Enter the type of page selector to use for the page. Values “SUBMIT” for submit buttons,”IMAGE” for image buttons, “TEXT” for text links and “SCROLLBAR” for a scrollbar (scrollbar requires a height attribute).
- rowsperpageselector (optional): A true false value indicating whether or not allow the user to select the number of rows on each page.
- clicksort (optional): A true/false flag indicating whether the heading tags should allow sorting of columns by clicking.
- visible (optional): Whether or not the item is initially visible
- enabled (optional): Whether or not the components in the container are enabled.
- maxpagingbuttons (optional): The maximum number of paging buttons for the page
- datasource (optional): The name of the datasource to use for this table. If not filled in, the datasource must be set in the controller.

#### Nested Tags:

- **datatableheader**: The first line in the data table.
- **datatablerows**:The rows of the data table that get repeated for each row in the datasource.
- **datatablefooter**: The band of the data table that appears at the end of the list.
- **datatablegroupheader**: The band of the data table that appears at the top when a group changes.
- **datatablegroupfooter**: The band of the data table that appears at the bottom when a group changes.

#### **datasource**

This tag allows access to relational database data from other components on the page, The tag set describes a select statement or model class in which the resulting columns can be bound to GUI components.

#### Attributes:

- name: (required)The name of the datasource. Used to reference the datasource in the controller and in other tags.
- type: (required) Valid Values: CONTENTCLASS,XML, SQL,MODEL, QBE. If type=XML the contents of the datasourcedef will be an XML representation of the datasouce structure.If type=CONTENTCLASS the FatWire Update Engine contentclass attribute must be filled in and the datasourcedef is only required if

the user wants to limit the number of columns the datasource retrieves. If type=SQL the datasourcedef must completely describe the datastore using the tags described below. If Type=MODEL, the class name of the model datastore or Java bean is required. In this case the class that gets the data for this datasource must be pre-built and placed in the class path for the web application. If type=QBE then the tag will create a QBEBUILDER datasource.

- contentclass: (optional) For type="CONTENTCLASS". The name of the Fatwire Update Engine Content class to retrieve data from.
- xmlfile: (optional) For type="XML". The name of the xml file to be loaded containing the datastore definition. The complete datastore xml definition may optionally be embedded inside the datasourcedef tag and this attribute left out.
- model: (optional) For type="MODEL". The class name of the datastore or Java Bean that implements this model.
- sessionid: (optional) For type="MODEL". If you want to share a model among multiple pages specify a sessionid in addition to a model attribute. The first page that uses a model with a sessionid will create the model and place it on the session. Subsequent pages with the same sessionid will retrieve the model from the session instead of creating a new one.
- defaulttable: (optional) The default table to use for all columns in the datastore where a table isn't described.
- autobind: (optional) Whether or not to automatically bind columns in the datasource to components on the page (default = yes)
- autoretrieve: (optional) When to retrieve the data from the database. NEVER: don't do an autoretrieve, "ONCHANGE" retrieve when the page is requested and some selection criteria changes, "ALWAYS" retrieve every time the page is requested. (not required, default="ONCHANGE")
- distinct: (optional) A true/false flag indicating whether a "distinct" should be added to the select statement (for SQL datasources)
- maxrows: (optional) The maximum number of rows the datastore should retrieve.
- updatemethod: (optional) Describes the SQL technique used to update the database from the datastore. Valid values are: UPDATE (use a SQL update statement) and DELETEINSERT (use SQL delete to first remove changed rows and then SQL insert the put them back with changed values).
- application: (optional) For SQL datastores, this will describe the application that the datastore is part of. The datastore will then get its database connection from the default database profile for that application. If the attribute is left out, it will be assumed from the application of the page.
- dbprofile: (optional) For SQL datastores, this will describe the name of the database profile to use. Database connection information will be retrieved from the property files using the application and dbprofile specified. If left out, it will use the default profile for the application.
- beandatastore: (optional) A subclass of com.salmonllc.sql.BeanDataStore that will be used in place of that class when the datastore is created. Used with the model attribute and beanextrainfo attribute.
- beanextrainfo: (optional) Used with the beandatastore attribute. A string passed to the bean datastore sub class that can be used to help in configuration.

#### Nested Tags

**datasourcedef** : Demarcates the section of the datasource where the structure of the datastore will be described. Alias, field, bucket, join and group tags are nested within.

**alias:** For datastores that require table aliases. It can be used when a table is required more than once (example a table joining to itself) or when a table is used in a join or as selection criteria, but no columns from that table are included in the column list.

Attributes

- tablename (required): The name of the table
- alias: (optional) The name of the alias. Leave it null to just include the table name as is in the from clause

**field:** If nested inside datasource def, each tag will specify one column to retrieve. If nested inside a group or order tag, it will specify a column to sort of group by respectively. The values required depend on the context of the tag. For example, sortdir has no meaning unless the tag is nested inside an order tag.

Attributes

- tablename (optional) The name of the table or alias. If left out, use the default table for the datasource
- fieldname (required) The name of the column.
- datatype(optional) The datatype of the column. If left out, the DataStore will figure out the datatype by examining database system tables. Valid values are STRING, INT, DATETIME, DOUBLE, BYTEARRAY, SHORT, LONG, FLOAT, DATE, TIME
- primarykey (optional) A true/false value indicating whether the column is a primary key. If left out, the DataStore will figure whether or not the field is part of the primary key by examining database system tables
- updateable (optional) A true false value indicating whether or not the column will be updated when the datastore update method is called.
- name (optional) The internal name of the column in the datasource (default is table.column)
- format (optional) A java.text string indicating how the column should be formatted for display
- sortdir (optional) When nested in an order tag, this attribute will indicate the direction of the sort (ASC or DESC);

**bucket:** For columns in the datastore that don't represent SQL columns

Attributes

- name (required) The name for the bucket
- datatype (required) The datatype for the bucket. Valid values are STRING, INT, DATETIME, DOUBLE, BYTEARRAY, SHORT, LONG, FLOAT, DATE, TIME
- format (optional) A java.text string indicating how the column should be formatted for display

**join:** Describes a join used in the datastore. It is used in conjunction with one or more joinfield tag which must be nested within it.

Attributes

- outer (required) : A true/false indicating whether the join is an outer join. The fields in fieldname1 in the join field would be the outer part of the join if this is true.

**joinfield:** Describes a column pair used in a join. Each column pair would require one of these tags.

Attributes

- fieldname1 (required) The name of the column in the first table

- fieldname2 (required) The name of the column in the second table

**order:** Demarcates the section of the datasource tag where the order of retrieved items is specified. It is used in conjunction with field tags (field tag indicates which field to sort by) which are nested within.

**group:** Demarcates the section of the datasource tag where the grouping of items is specified. It is used in conjunction with field tags which are nested within.

**selection** Demarcates the section of the datasource tag where selection criteria is specified (generate the where clause for the datasource). It is used in conjunction with the selectioncriteria tags which are nested within.

**groupselection:** Demarcates the section of the datasource tag where selection criteria for grouped columns are specified (generate the having clause for the datasource). It is used in conjunction with selectioncriteria tags which are nested within.

**selectioncriteria:** Describes a one element of selection criteria used in a query. Used in conjunction with the selection and groupselection tags

Attributes

- prefix (optional) If multiple criteria need to be grouped, the prefix may hold one or more open parenthesis.
- fieldname (required) The name of the field to use for selection criteria
- operator(required) The comparison operator used in the query (=, <>, >, <, like ) because some of these operators could cause problems with JSP engines (< and >), text can be substituted (EQUAL, NOT\_EQUAL, GREATER\_THAN, GREATER\_THAN\_OR\_EQUALS, LESS\_THAN, LESS\_THAN\_OR\_EQUALS)
- connector (optional) For multiple criteria, this describes how a query is logically connected to the one after it (and, or)
- suffix (optional) If multiple criteria need to be grouped, the suffix may hold one or more close parenthesis.

**qbecriteria:** Used only for QBEBuilders. Describes a one element of selection criteria used by the builder.

Attributes

- name (required) bucket name for the selection criteria column. Any unique name is OK, but no spaces in
- columns (required) Comma separated list of table columns to apply to this criteria. Valid values are: \* = any table is ok, table.\* = any column in table is OK, table.column = only the specific column is OK
- type (required) The type of selection criteria. Valid types are: EQUALS\_IGNORE\_CASE, STARTS\_WITH, STARTS\_WITH\_IGNORE\_CASE, CONTAINS, CONTAINS\_IGNORE\_CASE, LTE, LT, GTE, GT, EQUALS, NOT\_EQUALS, COMPLEX

### detailform

This tag is used to create Detail Form. **(deprecated, replaced by detailformdisplaybox)**

Attributes:

- name (required): The name of the component.
- datadef (required): The name of the XML file which describes data definition for this detail form.
- caption (optional): Caption for the heading of the datatable portion of ListForm.
- saveButton (optional): The name of the button component clicked to search (For User Defined Detail Form) or true | false (Standard Detail Form).
- deleteButton (optional): The name of the button component clicked to delete record(For User Defined Detail Form) or true | false (Standard Detail Form).

### detailformdisplaybox

Provides a display box component with embedded controller logic for editing or displaying a single database row.

Attributes:

- name (required): Identifier for the component
- caption (required): Caption on the title bar of the display box
- bgcolor (optional): Background Color
- headerbgcolor (optional): Header Background Color
- headerfont (optional): font for the header caption
- border (optional): The table border
- cellpadding (optional): The table cell padding
- cellspacing (optional): The table cell spacing
- width (optional): A string with the width of the table expressed in percent or pixels
- theme (optional): A string indicating the theme to use
- visible (optional): true/false whether the component is visible
- enabled (optional): true/false whether the component is enabled
- classname (optional): Style sheet class name
- buttondisplaylocation (optional): Location of the buttons. Valid values are IN\_HEADER, BELOW\_TABLE, IN\_HEADER\_AND\_BELOW\_TABLE
- buttonbgcolor (optional): Background color for the buttons
- buttonfontstyle (optional): A style sheet style for the buttons
- addbuttoncaption (optional): String representing the caption for the add button
- addbuttonvisible (optional): A true/false value indicating whether or not the button is visible
- addbuttonaccesskey (optional): A single character from a to z indicating the access key for the add button
- savebuttoncaption (optional): String representing the caption for the save button
- savebuttonvisible (optional): A true/false value indicating whether or not the button is visible
- savebuttonaccesskey (optional): A single character from a to z indicating the access key for the save button
- deletebuttoncaption (optional): String representing the caption for the delete button

- deletebuttonvisible (optional): A true/false value indicating whether or not the button is visible
- deletebuttonaccesskey (optional): A single character from a to z indicating the access key for the delete button
- cancelbuttoncaption (optional): String representing the caption for the caption button
- cancelbuttonvisible (optional): A true/false value indicating whether or not the button is visible
- cancelbuttonaccesskey (optional): A single character from a to z indicating the access key for the cancel button
- datasource (optional): The Datasource used for this component
- validator (optional): The validator for this component
- supressheading (optional): A true/false indicating whether or not the heading should display
- listformdisplaybox (optional): The list form display box that will be linked to this component.
- detailformclass (optional): The name of the Java class to use instead of the generic JspDetailFormDisplayBox. The class must extend com.salmonllc.jsp.JspDetailFormDisplayBox.
- confirmdelete (optional) A true/false value indicating whether or not a confirm message should pop up before a delete.
- reloadrowaftersave (optional) A true/false value indicating whether a reload row should be issued on each saved row. If true, the row will be reloaded from the database.

### displaybox

This tag creates a special display type that looks somewhat like a window in a Windows system. The display box consists of a header with a caption and buttons and a box area whatever other JSP/HTML the box should contain nested within.

#### Attributes:

- name (required) The name of the component
- border (optional):border width
- cellpadding (optional): HTML table cell padding
- cellspacing (optional): HTML table cell padding
- width (optional): The width of the box in pixels or percent
- caption (optional): The caption to appear in the heading
- headerbgcolor (optional): The background color for the heading
- headerfont (optional): The font name to use for the heading text. The actual font tag will be loaded from the properties file.
- bgcolor (optional): The background color for the box.
- visible (optional): Whether or not the item is initially visible
- enabled (optional): Whether or not the components in the container are enabled.

#### Nested Tags:

- **displayBoxHeader**: The components that go in the header of the box (usually buttons are nested within this tag).
- **displayBoxContents**: The HTML /JSP to display in the box are nested within this tag.

## div

Adds a div to your jsp. The div is a region of the page that can be displayed, hidden or moved via javascript. Note: This is an empty tag `<salmon:div />` which is ended with a `<salmon:enddiv/>` rather than a standard closing tag.

Attributes:

- name: (Required) The identifier for the div.
- hidden: true/false whether or not the div starts out hidden

## expand

Provides a button to expand or contract a div on the page using javascript. Used in conjunction with the JspDiv tag.

Attributes:

- name (required): Identifier for the component
- imgexp (required): The source for the expand button image
- imgcont (required): The source for the contract button image
- altexp: Message to display when the mouse hovers over the expand image
- altcont : Message to display when the mouse hovers over the contract image
- keepstate: A true/false flag indicating whether the expand component should keep information on the client as to which div rows in a datatable are expanded or not. It will cause the expanded state to be remembered when the datatable pages.
- div (required): The div to expand or contract

## expandall

Provides a button to expand or contract all the divs on the page using javascript. Used in conjunction with the Jsp div and expand tag.

Attributes:

- name (required): Identifier for the component
- imgexp (required): The source for the expand button image
- imgcont (required): The source for the contract button image
- altexp: Message to display when the mouse hovers over the expand image
- altcont : Message to display when the mouse hovers over the contract image
- expand (required): The name of the expand tag to expand (or contract) all for

## font

Provides a font tag. This differs from the HTML font tag in that the font is described with only a type attribute. This type is read from the appropriate property file at run time, which sets the actual font.

Attributes:

- type: (Required) The type of font as specified in the property file.

## form

Provides the equivalent of an HTML form tag. For pages with multiple forms, the salmon form tag is desirable because it can inform other framework components of the presence of multiple forms.

### Attributes:

- name: (Required) The identifier for the form.
- enabled: Indicates whether or not the components in the form can accept user input
- visible: Indicates whether the components in the for can be seen
- action: The url to post the results of the form to. If left out, the framework will decide the appropriate value for this attribute.
- enctype: The encoding type for submitting the form. "application/x-www-form-urlencoded" is the default for regular forms. For forms with a file upload inside, use "multipart/form-data" instead.
- method: The HTTP method for submitting the form (GET or POST). POST is default.
- onreset: Javascript to execute when the form is reset.
- onsubmit: Javascript to execute when the form is submitted.
- target: The frame where the results of the submit should be sent.
- autosegment: A true/false flag indicating whether or not the form should scroll back to it's previous screen position once a page is submitted and returned back to the browser.

## img

This tag creates an HTMLImage component which in turn puts an image in the page.

### Attributes:

- src (required): The URL of the location of the image file
- name (required): The name of the component
- alt (optional): Bubble help text when the user mouses over the image
- align (optional): Position of the image LEFT, RIGHT, TOP, ABSMIDDLE, ABSBOTTOM, TEXTTOP, MIDDLE, BASELINE, BOTTOM
- border (optional): border width
- height (optional): height of the image
- vspace (optional): vertical margin
- usemap (optional): The image map to use for the image
- datasource (optional): An expression that computes the src attribute from data in a datasource.
- srclocalekey (optional): The key used to replace the value in the src attribute in this component when used with internationalization.
- altlocalekey (optional): The key used to replace the value in the alt attribute in this component when used with internationalization.

## input

Allows for form components (from the com.salmonllc.html package) that can be controlled programmatically by your controller and bound to datasource columns. To function properly, you should embed these inside an HTML form tag.

### General Attributes

- name (required): The name of the component
- type (required): The Type of component see below
- visible (optional): Whether or not the item is initially visible
- theme (optional): The theme to use to display the component.
- datasource (optional): The column to bind the input field to
- classname (optional): The name of the style class to use for this object
- style (optional): An inline CSS style for the input component

### Conditional Attributes (Depending on the type, the attributes will be used or not.)

- onclick: Javascript to execute when the component is clicked.
- checked: For check boxes, a true false value indicating whether the item is checked.
- value: The text to display on a button.
- align: "LEFT"|"RIGHT"|"TOP"|"ABSMIDDLE"|"ABSBOTTOM"|"TEXTTOP"|"MIDDLE"|"BASELINE"|"BOTTOM" for images
- src: The URL for an image
- font: The font name to use for the heading text. The actual font tag will be loaded from the properties file.
- fontface: The font face for generated images
- fontsize: The font size for generated images
- bgcolor: The background color for generated images
- textcolor: The text color for generated images
- topleftbordercolor: The top left border color for generated images
- bottomrighthbordercolor: The bottom right border color for generated images
- transparentcolor: For round images, the color that represents transparent (one not used in the button)
- imagetype: RECTANGLE, CIRCLE or OVAL for generated images.
- maxlength: For text entry, the maximum number of characters to use.
- minlength: For EMAIL – the minimum number of chars an email address can be default = 7(ex. 7 would allow 'a@a.com').
- separator: For SSNMULTI,PHONE – the separator to use between the edit fields the default is '-' a dash.
- codes: For STATE – A true false value indicating whether to use just code or to show full state names.
- areacodestart – For PHONE the starting character to use to enclose the area code
- areacodeend - For PHONE the ending character to use to enclose the area code.
- extendedzip- For ZIP A true false value indicating whether to use a 10 digit zip.
- fracsize –For FRACTION (int) number of palces for integer portion.
- decprecision –For DECIMAL (int) Total number of digits.
- scale–For FRACTION,DECIMAL (int) highest fraction proportion, e.g 64 for 1/64 .. 63/64. Or Number of decimal places (digits to right of decimal point)
- nopadding: For FRACTION,DECIMAL used to Pad left side with

- interval -For DATE (default=48) For the hour component, break up the day into this many intervals. For example, 48 = half-hour intervals. 0 means do not display the hour component.
- fromtime – For DATE The hour component, skip the hours before this time. (For example fromtime = 7 and totime = 19 => display times only between 7AM to 7PM)
- totime – For DATE For the hour component, skip the hours after this time. (For example fromtime = 7 and totime = 19 => display times only between 7AM to 7PM)
- dateflag- For DATE This is to signify how you want to see the month valid values are FULL , PARTIAL ,DIGIT
- onblur: Javascript to execute when the text field loses keyboard focus.
- onchange: Javascript to execute when the text field loses keyboard focus and the value has been modified.
- onfocus: Javascript to execute when the text field gains focus
- size: For text fields, the number of characters to display, for select lists, the number of items in the list
- multiple: A true false value indicating whether or not multiple entries are permitted
- group: For radio buttons, must be the same for each radio button in a group
- display:For Select Lists (LIST for a list box or RADIO for a radio button display)
- orientation: For select lists with a RADIO display type, orientation can be VERTICAL or HORIZONTAL
- captionpos: For select lists with a RADIO display type can be "TOP", "BOTTOM", "LEFT", "RIGHT"
- cols: For a TEXTAREA tag, the number of columns to display
- rows: For a TEXTAREA tag, the number of rows to display
- wrap: For a TEXTAREA tag, the way the text should wrap (HARD, SOFT, OFF)
- height: (optional) Height of Image
- width: (optional) Width of Image
- textlocalekey (optional): The key used to replace the value in the text attribute of this component when used with internationalization.
- urllocalekey (optional): The key used to replace the value in the src attribute of this component when used with internationalization
- displaynamelocalekey (optional): The key used to replace the value in the value in this component when used with internationalization
- displayformatlocalekey (optional): The key used to replace the display format used by this component when used with internationalization.
- displayformat (optional): The display format used for the this component. It is superseded by displayformatlocalekey if present.
- accesskey (optional): A keyboard character that when pressed in conjunction with the alt key will active the component (For example click a button)
- readonly (optional): For some components this will display the component like an editor, but not allow the user to edit.
- tabindex (optional): A number from 0 to 32000 that indicates what order the component will get focus when the user presses the tab or shift tab key.
- onmouseover (optional): For type="IMAGE", javascript that fires when the user mouses over the image.
- onmouseout (optional): For type="IMAGE", javascript that fires when the user mouses out the image.
- listtype (optional): For type="STATE", the type of states/provinces/countries that will populate the drop down. Valid Values: "states" for US states, "provinces" for

Canadian provinces, "provincesstates" for Canadian provinces follow by US states, "statesprovinces" for US states followed by Canadian provinces, "canusa" for states and provinces sorted together, "countries" for countries of the world.

- style (optional) an inline stylesheet style for text edit compents
- text (optional) for images and submit images, like an alt attribute but works in non-Internet Explorer browsers.

Nested Tag:

**option:** Adds options to an <input type="SELECT"> tag

Attributes:

- key: The key value (actual value) for the option
- display: The Display value for the option
- table: The table to use in the database for the option
- keycolumn: The column to use for key values for the option
- displaycolumn: The column to use for display values for the option
- nulloption: Add a blank at the top (normally used for a search field)
- toupper: Used to make sure all keys and display values are upper case

Types:

**BUTTON:** Creates a button that executes java script

Required Attributes: value and onclick.

**CHECKBOX:** Creates a checkbox component

Optional Attributes: checked and onclick

**FILE:** Creates a file upload component

Optional Attributes: NONE

**HIDDEN:** Creates an Html Hidden component

Required Attributes: value

**IMAGE:** Creates an Html Image Submit Button

Required Attributes: src for a static image or value for a dynamically generated image.

Optional Attributes: align, fontface, fontsize, bgcolor, textcolor, topleftbordercolor, bottomrightbordercolor, transparentcolor, height, width, textlocalekey, urllocalekey.

**PASSWORD:** Creates a password entry field

Required Attributes: maxlength, size

Optional Attributes: value

**RADIO:** Creates a radio button

Required Attributes: checked, value, group

Optional Attributes: onclick

**SUBMIT:** Creates a submit button

Required Attributes: value

Optional Attributes: displaynamelocalekey

**TEXT:** Creates a text entry field

Required Attributes: maxlength, size  
Optional Attributes: onchange, onfocus, onselect, value, displayformatlocalekey, displayformat

*SELECT*: Creates list of items to choose from (either a list box or a drop down list box or a radio button group.

Required Attributes: maxlength, size, multiple  
Optional Attributes: display, orientation, captionpos, onblur, onclick, onfocus  
Note: The options appear in the OPTIONS tag

*TEXTAREA*: Create a multiline text edit box

Required Attributes: cols, rows, wrap  
Optional Attributes: onblur, onchange, onfocus, displayformatlocalekey  
Note: The text within the tag will initialize the value of the text box

*EMAIL* Create a text edit that knows how to do simple validation of an email address. Checks that is well formed (has @ and a .)

Required Attributes:none  
Optional Attributes: maxlength(default = 50) , size (default=20), minlength(default=7)

*SSNMULTI* Create a set of three edits that know how to do simple validation of a social security number. Checks that only numbers are entered and that it is the right length.

Required Attributes:none  
Optional Attributes: separator (default = '-')

*SSN* Create an edit field validation of a social security number. Checks that only numbers are entered and that it is the right length.

Required Attributes:none  
Optional Attributes: separator (default = '-')

*STATE* Create a dropdown that is prefilled with the states.

Required Attributes:none  
Optional Attributes: codes (default = 'true')

*PHONE* Create a set of three edits that know how to do simple validation of a phone number.

Required Attributes:none  
Optional Attributes: separator (default = '-'), areacodestart(default = '('), areacodeend(default = ')')

*ZIP* Creates an edit that knows how to validate a zip code. Makes sure only numbers are used.

Required Attributes:none  
Optional Attributes:extendedzip(default='false')

*FRACTION* Implements a two-part entry field for a decimal number

Required Attributes : fracsize ,scale  
Optional Attributes:nopadding

*DECIMAL* . Implements a two-part entry field for a decimal number.

Required Attributes: deprecision,scale  
Optional Attributes: nopadding

*DATE* . Use this class to represent an SQL DATE or TIMESTAMP value as a series of drop-down lists to support selection of year, month, day, and hour.

Required Attributes: none

Optional Attributes: interval (default=48),fromtime,totime,dateflag

## list

Repeats all nested tags once for every row in the datasource. Allows newspaper style display via the columns attribute or can render with no extra formatting (providing the HTML designer with complete control over the presentation) via the usetable attribute.

Attributes:

- name (required):The name of the component
- columns (optional):The number of columns to display in the list (default = 1)
- width (optional):The width of the table used to display the list in pixels or percent
- datasource (optional):The datasource used to enumerate the rows to display in the list
- columnwidth (optional):The width of each column in the list in pixels or percent
- type (optional):NEWSPAPER or STRAIGHT. Indicates how the columns should sort for multi-column lists. NEWSPAPER will sort down and then across, STRAIGHT will sort across and then down.
- visible (optional):True/False indicating whether or not the component is initially visible
- border (optional):The border width for the list in pixels
- cellpadding(optional):The cell padding for each cell in the table
- cellpadding(optional):The cell spacing for each cell in the table
- usetable(optional):A true false flag (default=true) to indicate whether the list will be rendered using an HTML table. If false, table attributes (width, columnwidth, border, cellpadding and cellspacing) will be ignored)

## lookup

This tag creates a Lookup Component with a Text Edit and a browse Image button.

Attributes:

- name (required): The name of the component
- browseimage (required): The name of the Image file which is shown as a button to be clicked to look up a specified URL.
- lookupurl (required): The url to be browsed when browse image is clicked.
- datasource (optional): A datasource expression used to bind the text edit to a column.
- size (optional): The size of the text edit field
- maxlength (optional): The maximum number of characters to enter on a field
- displayformat (optional): the format for displaying dates or numbers
- displayformatlocalekey (optional): The internationalization key for choosing the display format
- usepopup (optional): True/False indicating whether or not to use popup windows
- popupwidth (optional): The width of the popup window
- popupheight (optional): The height of the popup window
- descriptionfont (optional): The font to use for the description field

- showdescription (optional): True/False indicating whether or not the description should be displayed next to the text box.
- descriptiondatasource (optional): The datasource for the datastore column/bucket holding the description.
- usemodal (optional): A true/false flag indicating whether a popup window should be modal.
- usediv (optional): If using a popup, indicates whether the popup should be in a operating system window or a browser div tag.
- divscrolling(optional): A true/false flag indicating whether a popup in a div will have a scrollbar.
- divborderstyle(optional): A style sheet border style for the appearance of the border around a div popup window.
- popupposition(optional): Value indicating the position of a popup window or popup div. Valid values are “center” to center the popup on the screen, “relative” to make the popup position relative to the component and “custom” where you specify the position with the `popupleft` and `popuptop` attributes.
- popupleft(optional): Value indicating the left corner of a popup window or popup div in pixels (if `popupposition=“custom”`).
- popuptop(optional): Value indicating the top corner of a popup window or popup div in pixels (if `popupposition=“custom”`).
- popupattributes(optional): For popup windows, the Javascript attributes that the window will have. See attributes in the javascript function `window.open()` for a list of valid attributes.
- editdescription (optional): For lookups where the key field should not be not editable (for example an integer that is generated and has no meaning to the end user). In this case on returning from a lookup, the description returned will be displayed in the edit box and the key returned will go into a hidden field. Note: if the user should enter a value in the description field without doing a lookup, you must populate the hidden key field in your code, usually by using a validator lookup rule.
- theme(optional): A theme for choosing which SOFIA properties are used for the look and behavior of the component.

## listform

This tag is used to create a List Form component. (**deprecated, replaced by `listformdisplaybox`**)

Attributes:

- name (required): The name of the component.
- datadef (required): The name of the XML file which describes data definition for this listform.
- detailpage (optional): The name of the JSP page which describes the detail view of one row in List Form.
- caption (optional): Caption for the heading of the datatable portion of ListForm.
- buttons (optional): Buttons that are to be added on ListForm. This can be any combination of ADD, SEARCH, ADVANCE\_SEARCH delimited with ‘|’;

## listformdisplaybox

Provides a display box component with embedded controller logic for listing and/or editing the results of a database search.

## Attributes:

- name (required): Identifier for the component
- caption (required): Caption on the title bar of the display box
- bgcolor (optional): Background Color
- headerbgcolor (optional): Header Background Color
- headerfont (optional): font for the header caption
- border (optional): The table border
- cellpadding (optional): The table cell padding
- cellspacing (optional): The table cell spacing
- width (optional): A string with the width of the table expressed in percent or pixels
- theme (optional): A string indicating the theme to use
- visible (optional): true/false whether the component is visible
- enabled (optional): true/false whether the component is enabled
- classname (optional): Style sheet class name
- buttondisplaylocation (optional): Location of the buttons. Valid values are IN\_HEADER, BELOW\_TABLE, IN\_HEADER\_AND\_BELOW\_TABLE
- buttonbgcolor (optional): Background color for the buttons
- buttonfontstyle (optional): A style sheet style for the buttons
- addbuttoncaption (optional): String representing the caption for the add button
- addbuttonvisible (optional): A true/false value indicating whether or not the button is visible
- addbuttonaccesskey (optional): A single character from a to z indicating the access key for the add button
- savebuttoncaption (optional): String representing the caption for the save button
- savebuttonvisible (optional): A true/false value indicating whether or not the button is visible
- savebuttonaccesskey (optional): A single character from a to z indicating the access key for the save button
- datasource (optional): The Datasource used for this component
- validator (optional): The validator for this component
- supressheading (optional): A true/false indicating whether or not the heading should display
- listformclass (optional): The name of the Java class to use instead of the generic JspListFormDisplayBox. The class must extend com.salmonllc.jsp.JspListFormDisplayBox.
- autocreatelink (optional) A true/false value indicating whether or not a zoom link should be created.
- detailpageurl (optional): The url for the detail page this form will zoom to
- rowhighlightcolor (optional): A color indicating the background for a selected row
- maxrows (optional): An integer indicating the maximum number of rows to retrieve
- lookupreturnexp (optional): A datasource expression indicating what value to return in the event this list is used for a lookup
- lookupdescretreturnexp (optional): A datasource expression indicating what value to return for a description column the event this list is used for a lookup
- searchformdisplaybox (optional): The search form display box that will be linked to this component.
- detailformdisplaybox (optional): The detail form display box that will be linked to this component.

## navbar

This tag creates a navigation menu that can be controlled programmatically.

### Attributes:

- align (optional): "LEFT|RIGHT"
- bgcolor (optional): "color"
- border (optional): "value"
- cellpadding (optional): "value"
- cellspacing (optional): "value"
- height (optional): "height"
- width (optional): "width"
- cols (optional): "numOfCols"
- hspace (optional): "horizMargin"
- vspace (optional): "vertMargin"
- name (required): The name of the component.
- theme(optional): The theme to use for items in the container if they don't specify a theme.
- visible (optional): Whether or not the item is initially visible
- enabled (optional): Whether or not the components in the container are enabled.
- hrefstyle (required): "Style of the Href Text when not Hovering"
- hoverstyle (required): "Style of the Href Text while Hovering"
- textstyle (optional): "Style of text if not an href"
- selectedstyle (required): "Style of href when selected"
- selectedhoverstyle (required): "Style of selected href while hovering"
- showpopupstyle (optional): "Style of href when a submenu is displayed inside the Navbar"
- showpopuphoverstyle (optional): "Style of href when a submenu is displayed inside the Navbar while hovering"
- showpopupselectedstyle (optional): "Style of selected href when a submenu is displayed inside the Navbar"
- showpopupselectedhoverstyle (optional): "Style of selected href when a submenu is displayed inside the Navbar while hovering"
- cellbgcolor (optional): "BackGround Color of Cell when not Hovering"
- groupcellbgcolor (optional): "BackGround Color of a Group Item Cell when not Hovering"
- hoverbgcolor (optional): "BackGround color of cell while hovering"
- selectedbgcolor (optional): "bg color of selected href"
- selectedhoverbgcolor (optional): "bg color of selected href while hovering"
- showpopupbgcolor (optional): "Default bg color of submenu when displayed inside the Navabr"
- submenubgcolor (required): "Default bg color for a submenu"
- markerimage (required): "Image marker to the left of href"
- markeroverimage (required): "Image marker to the left of href while hovering"
- selectedmarkerimage (required): "Image marker to the left of selected href"
- textimage (required): "Image marker to the left of Text"
- vspaceimage (required): "Image spacer between groups"
- hspaceimage (required): "Image spacer between marker Image and href/text"
- groupspacerimage (optional):: Image to appear below the group
- horizontal (optional): True to change the orientation of the navbar to horizontal

### Nested Tags

**navbargroup:** This tag creates a navigation menu that can be controlled programmatically.

Attributes:

- name (required) The name of the component
- headerimage (required): The image over the group"
- vspace (optional) Vertical spacing between groups
- href (optional) The Href for a Group Item
- grouptitle (optional) The Text Displayed for a Group Item
- bgcolor (optional) bg color , Will override the Default group cell bg color for A Group Item

**navbaritem:** This tag creates a navigation menu that can be controlled programmatically.

Attributes:

- name (required) The name of the component
- title (required) The Text displayed for a item
- href (optional) The href for a item
- target (optional)
- hspace (optional) Horizontal padding
- submenuname (optional) Name of Sub Menu
- showpopup (optional) True/False , whether the item submenu will be displayed inside the Navbar
- bgcolor (optional) bg color , Will override the Default cell bg color for A Item

**navbarsubitem:** This tag creates a navigation menu that can be controlled programmatically.

Attributes:

- name (required) The name of the component
- title (required) The Text displayed for a item
- href (optional) The href for a item
- target (optional)
- hspace (optional) Horizontal padding
- submenuname (optional) Name of Sub Menu
- submenugroup (optional) Name of Sub Menu a subitem should be grouped under
- bgcolor (optional) bg color , Will override the Default submenu bg color for A subItem

## page

This tag is the top-level tag in each JSP using the tag library. It must appear once in each page. This tag is an empty tag in that it doesn't have a closing tag (this is done to allow jsp includes to be nested within which isn't allowed in a body tag). For the page to work properly, the <prefix:endpage/> tag should be used to indicate that there are no more framework tags in the page.

Attributes:

- controller (optional): This indicates the framework class that will handle the user gestures and control the actions on the page.
- secondarycontroller (optional): A page can have two controllers. This tag indicates the class that will be the second one.

- application (optional): This indicates which application the page lives in for the purpose of identifying the correct property file. If the attribute is left out, the application will be identified by the last package in the full class name of the controller attribute. (ex: com.fuji.applications.documentManager.PageController would cause the application to be documentManager).
- keepsession (optional): This is a true/false attribute that indicates whether or not the page should stay on the session after it is first created. For pages that don't have form components to submit, setting this attribute to true will eliminate the memory footprint of the page but reduce the response time of subsequent page hits. For pages that are interactive or contain form components, the flag should be set to true (default).
- sessionkeepaliveminutes (optional): A numeric property, if present will override the System.properties property SessionKeepAliveMinutes. This will cause the browser to ping the server at intervals, keeping the server session alive while the browser remains open and on that page. Valid values are: -1, don't keep session alive, 0=keep session alive as long as the browser stays open and on this page or n=keep the session alive for n minutes as long as the browser stays open and on this page.
- contenttype (optional): This is a string attribute indicating the returning contenttype of the JSP.

### param

This body tag's purpose is to aid in the creation of generic controllers. Parameters can be set in the view which customize the behavior of a generic controller. Generic controllers can be coded to expect a parameters with particular names in any view that uses them.

Attributes:

- name (required): The name of the component.
- Body content is the value of the parameter.

### property

This tag creates a property expression for a component in the page.

Attributes:

- name (required): The name of the expression
- component (required): The name of the component to set the property.
- propertyname (required): The name of the property to set. The component must have a set method for that property
- expression (required): The expression to evaluate to set the property
- datasource (required): The datasource to use to evaluate the expression

### raw

Insert some raw Html in the component (HTML is in the body content of the tag). This will allow the programmatic change of the HTML in the components body.

- name (required): The name of the component
- visible (optional): Whether or not the item is initially visible
- filename (optional): Html File Name

- datasource (optional): An expression used to fill the contents of the datasource.

### rowselector

Creates a radio button component that has one radio button for each row in a datatable. It is used to select a single row.

- name (required): The name of the component
- onclick (optional): Javascript to execute

### rowhighlighter

Creates a hidden component that allows rows in a datatable to select by clicking on them. The selected rows will be highlighted in the specified color. This component must be placed inside a <salmon:td> in the rows band of a datatable to work properly and only one should be placed inside any given datatable.

- name (required): The name of the component
- datasource: The datasource of the column or bucket bound to this component. The bucket will contain truevalue for selected rows and falsevalue for unselected ones when the page is submitted.
- truevalue: A value indicating the row is selected
- falsevalue: A value indicating the row is not selected
- highlightcolor: The color to make the selected rows
- multiple: A true/false value indicating whether or not multiple rows can be selected. If true, the shift and control keys can be pressed during the click to select multiple rows.
- theme: The theme to use for the component visuals.

### searchformdisplaybox

Provides a display box component with embedded controller logic for running a database search.

Attributes:

- name (required): Identifier for the component
- caption (required): Caption on the title bar of the display box
- bgcolor (optional): Background Color
- headerbgcolor (optional): Header Background Color
- headerfont (optional): font for the header caption
- border (optional): The table border
- cellpadding (optional): The table cell padding
- cellspacing (optional): The table cell spacing
- width (optional): A string with the width of the table expressed in percent or pixels
- theme (optional): A string indicating the theme to use
- visible (optional): true/false whether the component is visible
- enabled (optional): true/false whether the component is enabled
- classname (optional): Style sheet class name
- buttondisplaylocation (optional): Location of the buttons. Valid values are IN\_HEADER, BELOW\_TABLE, IN\_HEADER\_AND\_BELOW\_TABLE

- buttonbgcolor (optional): Background color for the buttons
- buttonfontstyle (optional): A style sheet style for the buttons
- addbuttoncaption (optional): String representing the caption for the add button
- addbuttonvisible (optional): A true/false value indicating whether or not the button is visible
- addbuttonaccesskey (optional): A single character from a to z indicating the access key for the add button
- searchbuttoncaption (optional): String representing the caption for the search button
- searchbuttonvisible (optional): A true/false value indicating whether or not the button is visible
- searchbuttonaccesskey (optional): A single character from a to z indicating the access key for the search button
- cancelbuttoncaption (optional): String representing the caption for the cancel button
- cancelbuttonvisible (optional): A true/false value indicating whether or not the button is visible
- cancelbuttonaccesskey (optional): A single character from a to z indicating the access key for the cancel button
- qbebuilder (optional): The QBEBuilders used for searches for this component
- qbevalidator (optional): The validator component for the qbe
- listformdisplaybox (optional): The list form display box to use in conjunction with this searchformdisplaybox
- supresheading (optional): A true/false indicating whether or not the heading should display
- searchformclass (optional): The name of the Java class to use instead of the generic JspSearchFormDisplayBox. The class must extend com.salmonllc.jsp.JspSearchFormDisplayBox

## style

Provide a style tag that can be programmatically manipulated.

Attributes:

- type: (required) The type of item the style is for (A, TD etc...)
- style: (required) A Style String (See Html Docs)
- name: (required) The name of the component.

## subreportcontainer

A special type of container tag that when embedded inside a datatable can be used to generate sub reports for each main row in the datatable.

Attributes:

- name (required): The name of the component.
- visible (optional): Whether or not the item is initially visible
- enabled (optional): Whether or not the components in the container are enabled.
- subreportcontainerclass (optional): A classname of a subclass of JspSubReportContainer that will be used in place of JspSubReportContainer for this component.

## table

This tag mimics the Html Table tag, but provides programmatic control of the table from within the controller class.

Attributes:

- align (optional): "LEFT|RIGHT"
- bgcolor (optional): background color for the table
- border (optional): border size in pixels
- cellpadding (optional): cell padding for each cell in pixels
- cellspacing (optional): cell spacing for each cell in pixels
- height (optional): weight of the table
- width (optional): width of the table in pixels or percent
- cols (optional): number of columns in the table
- hspace (optional): horizontal margin
- vspace (optional): vertical margin
- name (required): the name of the component.
- visible (optional): whether or not the item is initially visible
- enabled (optional): Whether or not the components in the container are enabled.

## text

Insert some text into the Html. The text contents, font and visibility can be programmatically controlled.

Attributes:

- name (required): The name of the component
- font (optional): The name of the font as specified in the properties file
- text (optional): The text to display
- visible (optional): Whether or not the item is initially visible
- classname (optional): The HTML Style to use
- fixhtml (optional): A true / false value indicating whether special html characters, like >, < or " should be converted to &gt; &lt; &quot; (default = true).
- datasource (optional): A expression that specifies the text used to populate this column.
- textlocalekey (optional): The key used to replace the text in this component when used with internationalization.
- displayformatlocalekey (optional): The key used to change the display format for the text in this component when used with internationalization.
- displayformat (optional): The display format used for the text in this component. It is superceded by displayformatlocalekey if present.

## td

This tag mimics the HTML tag with the same name, but allows for programmatic control of the attributes via an JspTableCell component in the controller. Can be used within the tags library tags datatable and table

Attributes:

- align (optional) "CENTER|LEFT|RIGHT".
- valign (optional) "BASELINE|BOTTOM|MIDDLE|TOP"

- bgcolor (optional) "*color*". If left out, the color will be filled in from the default theme of the table or row color of the data table.
- colspan (optional) The number of columns this item will span
- rowspan (optional) The number of rows this item will span
- height (optional) The minimum height of the column
- width (optional) The minimum width of the column
- nowrap (optional) True or False indicating whether the column should wrap.
- onmouseover (optional) JavaScript to execute when the mouse passes over.
- onmouseout (optional) JavaScript to execute when the mouse passes out of the tag.
- onclick (optional) JavaScript to execute when the cell is clicked on.
- classname (optional) The Style to use for the component
- onmouseover (optional) Javascript to indicate what happens when the user moves the mouse over the text;
- onmouseout (optional) Javascript to indicate what happens when the user moves the mouse off the text

### tr

This tag mimics the HTML tag with the same name, but allows for programmatic control of the attributes via a JspTableRow component. Can be used within the tags library tags datatable and table.

Attributes:

- align (optional) "CENTER|LEFT|RIGHT"
- valign (optional) "BASELINE|BOTTOM|MIDDLE|TOP"
- bgcolor (optional) "*color*". If left out, the color will be filled in from the default theme of the table or row color of the data table.
- classname (optional) The Html Style to use for the row.

### tree

Creates an HtmlTree Component. The tree has nodes which can expand/contract when clicked.

Attributes:

- name: (required) The name of the component
- expandimage: (required) The URL of the image that when clicked will cause a tree node to expand
- contractimage: (required) The URL of the image that when clicked will cause a tree node to contract.
- nullimage: (required) The URL of an empty image to display in place of the expand or contract image when a node has no children.
- target: (optional) The frame to open a target page in.
- selectmode (optional) valid values are NONE (No items on the tree can be selected), ONE (One item on the tree can be selected) or, MANY (Many items on the tree can be selected).
- showroot (optional) true or false value indicating whether or not to display the root of the tree.
- treemode (optional) valid values are: LINK (clicking on a link in the tree will cause the browser to follow a link) or SUBMIT (clicking a link will cause the page to be submitted before following a link).

### Nested Tags

**treeitem:** Indicates an item on the tree. Tree items can be nested within other tree items to create a hierarchy.

Attributes

- text (required): The text for the entry
- href (optional): The URL to go to if the user clicks the text.
- imgsource (required): The URL for the default image for the node
- imgexpsource (optional): The URL for the image to use when the node is expanded or leave out to use the same image as the source.
- imghref (optional): The URL to go to if the user clicks the image.

**treeaddimage:** Adds an additional clickable image to the right side of a tree node. Used in the case where a tree node needs to be able to activate multiple pages instead of one.

Attributes

- src (required): The SRC for Image file
- text (optional): The alt text displayed on mouse over
- href (optional): The URL to browse when image clicked
- imghref (optional): The URL to go to if the user clicks the image.

### validator

This body tag indicates where a validation text component should be placed on the page.

Attributes:

- name (required): the name of the component.
- font (optional): The type of font to use (usually ErrorFont).
- datasource (optional): The datastore used to perform the validations. Required if there are any rules in the validator.
- submitcomponents (optional): The name of the submit button or submit image this validator should listen for. Required if there are any rules in the validator.
- autopopulatelookups (optional): An optional value indicating that when the specified datastore is autoretrieved, the populate lookups function should also be called.
- dbprofile (optional): The dbprofile to use for lookup validations.
- breaksbefore (optional): The number of line breaks to place before the error message
- breaksafter (optional): The number of line breaks to place after the error message
- breaksbetween (optional): The number of line breaks to place in between error messages if multiple are allowed
- allowmultipleerrors (optional): A true/false attribute indicating whether or not multiple error messages can be displayed
- fixhtml (optional): A true/false flag indicating whether or not to substitute special html characters like < or >. This flag should be set to false to embed Html in a message.
- validatorclass (optional): Use a specific validator class that extends com.salmonllc.html.HtmlValidatorText instead of the generic one.

- alerts (optional) a true false value indicating whether or not Javascript alert dialog boxes should be used to display error messages.
- focuslinks (optional) a true false value indicating whether or not error messages should be under line links that when clicked on set focus to the field in error
- rulejs (optional) a true false value indicating whether or not certain types of rules (range, regular expression, and required) should be run on the browser in Javascript instead of on the server in Java. Individual types can be specified as well with the (rangerulejs, requiredrulejs and regexrulejs attributes).
- javascriptforvalidationonly (optional) A True/False value indicating whether or not validations should occur on both the client in javascript and on the server in java. True means only execute in javascript on the browser. Default is False.
- autoimportrules(optional) a true false value indicating whether or not the validator should automatically import rules from the datastore it is bound to.

#### Nested Tags

**validatortrule:** This empty tag is used to add a validation rule to the validator. It must be nested in a validator tag.

#### Attributes:

- expression (required): A boolean expression. If true, the rule succeeds and if false it doesn't.
- message (optional): A string indicating what message to display if the error occurs
- focuscomp (optional): The name of the component to set focus to if this rule is violated
- messagelocalekey (optional): For internationalization, the key that represents the error message.

**validatorlookup:** This empty tag is used to add a lookup rule to the validator. It must be nested in a validator tag. A lookup rule is valid if the value in the component has a corresponding row another table (foreign key).

#### Attributes:

- lookupfromtable (required): The table to lookup the value of in. Used for the from clause of the select that will do the lookup.
- lookupwhereexpression (required): An expression that will be used to build a where clause for the select performing the lookup.
- message (optional): A string indicating what message to display if the error occurs
- focuscomp (optional): The name of the component to set focus to if this rule is violated
- messagelocalekey (optional): For internationalization, the key that represents the error message.
- descriptionsource (optional): The column in the lookup table that contains a description for the code in the lookup rule.
- descriptiontarget (optional): The bucket in the datastore used to hold the description.

**validatortypecheck** This empty tag is used to add a type check rule to the validator. A type check is for date or number columns where the user must enter a string that can be converted to the string or date type used.

Attributes:

- message (optional): A string indicating what message to display if the error occurs
- focuscomp (required): The name of the component to set focus to if this rule is violated
- messagelocalekey (optional): For internationalization, the key that represents the error message.

**validatorrequired** This empty tag is used to add a rule that will require the user to enter a value in a field.

Attributes:

message (optional): A string indicating what message to display if the error occurs  
focuscomp (required): The name of the component to set focus to if this rule is violated  
messagelocalekey (optional): For internationalization, the key that represents the error message.

**validatorjavascript** This empty tag is used to add a javascript rule to the validator

Attributes:

javascript (required): A javascript Boolean expression. If it returns false the rule will be violated. Note: to get SOFIA form component names into the rule they can be indicated in the form %comName.  
message (optional): A string indicating what message to display if the error occurs  
focuscomp (required): The name of the component to set focus to if this rule is violated  
**messagelocalekey (optional): For internationalization, the key that represents the error message.**

**validatorrange** This empty tag is used to add a rule that will require the user to enter a value in a field between a minimum and maximum.

Attributes:

- minimum (optional): A minimum value for the field or null for no minimum. Date constants should be entered in the form yyyy-mm-dd
- maximum (optional): A maximum value for the field or null for no minimum. Date constants should be entered in the form yyyy-mm-dd
- message (optional): A string indicating what message to display if the error occurs

- focuscomp (required): The name of the component to set focus to if this rule is violated
- messagelocalekey (optional): For internationalization, the key that represents the error message.

**validatorregexp:** This empty tag is used to add a regular expression validation rule to the validator. The rule will be violated if the value entered doesn't match the regular expression.

Attributes:

- expression (required): A regular expression. The rule succeeds if the entered value matches the regular expression.
- message (optional): A string indicating what message to display if the error occurs
- focuscomp (required): The name of the component to set focus to if this rule is violated
- messagelocalekey (optional): For internationalization, the key that represents the error message.

## WML Tags

To use SOFIA WML tags, include wmltaglib.tld in your page in a JSP taglib tag. Also use set the contenttype attribute of the SOFIA page tag to “text/vnd.wap.wml”.

### a

Provides an anchor tag in WML. Used to identify an area on the page or to make an area on the page an underline link. This will create a WmlLink component.

Attributes:

- name: (Required) The name of the component
- href: (Required) The url this link will zoom to if the user clicks anything in the component.
- classname: (Optional) The class attribute of the tag
- title: (Optional) The title attribute of the component. This may be used for display purposes, depending on the WAP device.
- datasource: (Optional) The datastore expression to compute the href attribute. If used it will overwrite the attribute with the results of the expression.

### card

Provides a card tag in WML. This will create a WmlCard component.

Attributes:

- name: (Required) The name of the component
- title: (Optional) The title attribute is text that may appear on the WAP device depending device implementation.
- classname: (Optional) The class attribute of the tag
- backwarddatasource: (Optional) The datastore expression to compute the onenterbackward attribute. If used it will overwrite the attribute with the results of the expression.
- forwarddatasource: (Optional) The datastore expression to compute the onenterforward attribute. If used it will overwrite the attribute with the results of the expression.
- timerdatasource: (Optional) The datastore expression to compute the ontimer attribute. If used it will overwrite the attribute with the results of the expression.
- newcontext: (Optional) Indicates that the browser context should be re-initialized upon entry to this card.
- ordered: (Optional) Specifies a hint to the browser about the organization of the card content. May be used to organize the content presentation or to otherwise influence layout of the card. Default is true.
- onenterbackward: (Optional) The href that is loaded when the user navigates using a go task.
- onenterforward: (Optional) The href that is loaded when the user navigates using a go task.
- ontimer: (Optional) The href that is loaded when the timer expires.

### datasource

This tag allows access to relational database data from other components on the page. The tag set describes a select statement or model class in which the resulting columns can be bound to GUI components.

Attributes:

- **name:** (required) The name of the datasource. Used to reference the datasource in the controller and in other tags.
- **type:** (required) Valid Values: CONTENTCLASS, XML, SQL, MODEL. If type=XML the contents of the datasourcedef will be an XML representation of the datasource structure. If type=CONTENTCLASS the FatWire Update Engine contentclass attribute must be filled in and the datasourcedef is only required if the user wants to limit the number of columns the datasource retrieves. If type=SQL the datasourcedef must completely describe the datastore using the tags described below. If Type=MODEL, the class name of the model datastore or Java bean is required. In this case the class that gets the data for this datasource must be pre-built and placed in the class path for the web application.
- **contentclass:** (optional) For type="CONTENTCLASS". The name of the Fatwire Update Engine Content class to retrieve data from.
- **xmlfile:** (optional) For type="XML". The name of the xml file to be loaded containing the datastore definition. The complete datastore xml definition may optionally be embedded inside the datasourcedef tag and this attribute left out.
- **model:** (optional) For type="MODEL". The class name of the datastore or Java Bean that implements this model.
- **defaulttable:** (optional) The default table to use for all columns in the datastore where a table isn't described.
- **autobind:** (optional) Whether or not to automatically bind columns in the datasource to components on the page (default = yes)
- **autoretrieve:** (optional) When to retrieve the data from the database. NEVER: don't do an autoretrieve, "ONCHANGE" retrieve when the page is requested and some selection criteria changes, "ALWAYS" retrieve every time the page is requested. (not required, default="ONCHANGE")
- **distinct:** (optional) A true/false flag indicating whether a "distinct" should be added to the select statement (for SQL datasources)
- **maxrows:** (optional) The maximum number of rows the datastore should retrieve.
- **updatemethod:** (optional) Describes the SQL technique used to update the database from the datastore. Valid values are: UPDATE (use a SQL update statement) and DELETEINSERT (use SQL delete to first remove changed rows and then SQL insert the put them back with changed values).
- **application:** (optional) For SQL datastores, this will describe the application that the datastore is part of. The datastore will then get its database connection from the default database profile for that application. If the attribute is left out, it will be assumed from the application of the page.
- **dbprofile:** (optional) For SQL datastores, this will describe the name of the database profile to use. Database connection information will be retrieved from the property files using the application and dbprofile specified. If left out, it will use the default profile for the application.

Nested Tags

**datasourcedef** : Demarcates the section of the datasource where the structure of the datastore will be described. Alias, field, bucket, join and group tags are nested within.

**alias:** For datastores that require table aliases. It can be used when a table is required more than once (example a table joining to itself) or when a table is used in a join or as selection criteria, but no columns from that table are included in the column list.

Attributes

- tablename (required): The name of the table
- alias: (optional) The name of the alias. Leave it null to just include the table name as is in the from clause

**field:** If nested inside datasource def, each tag will specify one column to retrieve. If nested inside a group or order tag, it will specify a column to sort of group by respectively. The values required depend on the context of the tag. For example, sortdir has no meaning unless the tag is nested inside an order tag.

Attributes

- tablename (optional) The name of the table or alias. If left out, use the default table for the datasource
- fieldname (required) The name of the column.
- datatype(optional) The datatype of the column. If left out, the DataStore will figure out the datatype by examining database system tables. Valid values are STRING, INT, DATETIME, DOUBLE, BYTEARRAY, SHORT, LONG, FLOAT, DATE, TIME
- primarykey (optional) A true/false value indicating whether the column is a primary key. If left out, the DataStore will figure whether or not the field is part of the primary key by examining database system tables
- updateable (optional) A true false value indicating whether or not the column will be updated when the datastore update method is called.
- name (optional) The internal name of the column in the datasource (default is table.column)
- format (optional) A java.text string indicating how the column should be formatted for display
- sortdir (optional) When nested in an order tag, this attribute will indicate the direction of the sort (ASC or DESC);

**bucket:** For columns in the datastore that don't represent SQL columns

Attributes

- name (required) The name for the bucket
- datatype (required) The datatype for the bucket. Valid values are STRING, INT, DATETIME, DOUBLE, BYTEARRAY, SHORT, LONG, FLOAT, DATE, TIME
- format (optional) A java.text string indicating how the column should be formatted for display

**join:** Describes a join used in the datastore. It is used in conjunction with one or more joinfield tag which must be nested within it.

Attributes

- outer (required) : A true/false indicating whether the join is an outer join. The fields in fieldname1 in the join field would be the outer part of the join if this is true.

**joinfield:** Describes a column pair used in a join. Each column pair would require one of these tags.

Attributes

- fieldname1 (required) The name of the column in the first table
- fieldname2 (required) The name of the column in the second table

**order:** Demarcates the section of the datasource tag where the order of retrieved items is specified. It is used in conjunction with field tags (field tag indicates which field to sort by) which are nested within.

**group:** Demarcates the section of the datasource tag where the grouping of items is specified. It is used in conjunction with field tags which are nested within.

**selection** Demarcates the section of the datasource tag where selection criteria is specified (generate the where clause for the datasource). It is used in conjunction with the selectioncriteria tags which are nested within.

**groupselection:** Demarcates the section of the datasource tag where selection criteria for grouped columns are specified (generate the having clause for the datasource). It is used in conjunction with selectioncriteria tags which are nested within.

**selectioncriteria:** Describes a one element of selection criteria used in a query. Used in conjunction with the selection and groupselection tags

Attributes

- prefix (optional) If multiple criteria need to be grouped, the prefix may hold one or more open parenthesis.
- fieldname (required) The name of the field to use for selection criteria
- operator(required) The comparison operator used in the query (=,<>,>,<, like ) because some of these operators could cause problems with JSP engines (< and >), text can be substituted (EQUAL, NOT\_EQUAL, GREATER\_THAN, GREATER\_THAN\_OR\_EQUALS, LESS\_THAN, LESS\_THAN\_OR\_EQUALS)
- connector (optional) For multiple criteria, this describes how a query is logically connected to the one after it (and, or)
- suffix (optional) If multiple criteria need to be grouped, the suffix may hold one or more close parenthesis.

## do

Provides a do tag in WML. This will create a WmlDo component.

Attributes:

- name: (Required) The name of the component
- type: (Required) Provides a hint to the browser about the authors intended use of the tag and how the tag should be mapped to the physical user interface. Possible Values: 'accept' (positive acknowledgement); 'prev' (backward history navigation); 'help' (request for help); 'reset' (clearing or resetting state); 'options' (request for options or additional operations); 'delete' (delete item or choice); 'unknown' a generic do element.
- href: (false) The href that is loaded when this task is executed.
- sendreferer: (Optional) If true, the request for the URI will contain the URI of the deck containing this task – can be used for server based access control. Default is false.
- classname: (Optional) The class attribute of the tag

- method: (Optional) Specifies the HTTP submission method. POST or GET. Default is GET.
- acceptcharset: (Optional) Specifies a list of character encodings that the origin server must accept when processing the input.
- datasource: (Optional) The datastore expression to compute the href attribute. If used it will overwrite the attribute with the results of the expression.
- hrefparameters: (Optional) Indicates whether the form components values in the card will be appended to the href for transmission to the server. Default is true.
- postfields: (Optional) Indicates whether the form components values in the card will be passed to the server via postfield tags. Default is true.
- label: (Optional) Specifies a textual string for labeling the user widget. Recommend a string of six characters or less.
- optional: (Optional) If true the browser may ignore this element. Default is false.
- doprev: (Optional) If true the action that is taken when executed is a backward navigation of the browser.

## go

Provides a go tag in WML. This will create a WmlGo component.

Attributes:

- name: (Required) The name of the component
- href: (Required) The href that is loaded when this task is executed.
- sendreferer: (Optional) If true, the request for the URI will contain the URI of the deck containing this task – can be used for server based access control. Default is false.
- classname: (Optional) The class attribute of the tag
- method: (Optional) Specifies the HTTP submission method. POST or GET. Default is GET.
- acceptcharset: (Optional) Specifies a list of character encodings that the origin server must accept when processing the input.
- datasource: (Optional) The datastore expression to compute the href attribute. If used it will overwrite the attribute with the results of the expression.
- hrefparameters: (Optional) Indicates whether the form components values in the card will be appended to the href for transmission to the server. Default is true.
- postfields: (Optional) Indicates whether the form components values in the card will be passed to the server via postfield tags. Default is true.

## img

This tag creates an WmlImage component which in turn puts an image in the page.

Attributes:

- src (required): The URL of the location of the image file
- name (required): The name of the component
- localsrc (optional): Specifies an alternative internal representation for the image.
- alt (optional): An alternative textual representation of the image.
- align (optional): Position of the image top, middle, bottom
- height (optional): height of the image
- width (optional): width of the image
- hspace (optional): horizontal margin

- vspace (optional): vertical margin
- datasource (optional): An expression that computes the src attribute from data in a datasource.

### input

Allows for form components (from the com.salmonllc.wml package) that can be controlled programmatically by your controller and bound to datasource columns. To function properly, you should embed these inside an WML card tag.

- name (required): The name of the component
- type (required): The Type of component see below
- enabled (optional): Whether or not the item is initially enabled
- visible (optional): Whether or not the item is initially visible
- datasource (optional): The column to bind the input field to
- classname (optional): The name of the style class to use for this object
- maxlength: For text entry, the maximum number of characters to use.
- size: The number of characters to display
- value: The default value of the component.

### list

Repeats all nested tags once for every row in the datasource. Allows newspaper style display via the columns attribute or can render with no extra formatting (providing the HTML designer with complete control over the presentation) via the usetable attribute.

Attributes:

- name (required): The name of the component
- columns (optional): The number of columns to display in the list (default = 1)
- datasource (optional): The datasource used to enumerate the rows to display in the list
- type (optional): NEWSPAPER or STRAIGHT. Indicates how the columns should sort for multi-column lists. NEWSPAPER will sort down and then across, STRAIGHT will sort across and then down.
- visible (optional): True/False indicating whether or not the component is initially visible
- usetable (optional): A true false flag (default=true) to indicate whether the list will be rendered using an WML table. If false, table attributes (width, columnwidth, border, cellpadding and cellspacing) will be ignored)

### page

This tag is the top-level tag in each JSP using the tag library. It must appear once in each page. This tag is an empty tag in that it doesn't have a closing tag (this is done to allow jsp includes to be nested within which isn't allowed in a body tag). For the page to work properly, the <prefix:endpage/> tag should be used to indicate that there are no more framework tags in the page.

**Attributes:**

- controller (Optional): This indicates the framework class that will handle the user gestures and control the actions on the page.
- secondarycontroller (Optional): A page can have two controllers. This tag indicates the class that will be the second one.
- application (Optional): This indicates which application the page lives in for the purpose of identifying the correct property file. If the attribute is left out, the application will be identified by the last package in the full class name of the controller attribute. (ex: com.fuji.applications.documentManager.PageController would cause the application to be documentManager).
- keepsession (Optional): This is a true/false attribute that indicates whether or not the page should stay on the session after it is first created. For pages that don't have form components to submit, setting this attribute to true will eliminate the memory footprint of the page but reduce the response time of subsequent page hits. For pages that are interactive or contain form components, the flag should be set to true (default).
- contenttype (Optional): This is a string attribute indicating the returning contenttype of the JSP.

**param**

This body tag's purpose is to aid in the creation of generic controllers. Parameters can be set in the view which customize the behavior of a generic controller. Generic controllers can be coded to expect a parameters with particular names in any view that uses them.

**Attributes:**

- name (required): The name of the component.
- Body content is the value of the parameter.

**property**

This tag creates a property expression for a component in the page.

**Attributes:**

- name (required): The name of the expression
- component (required): The name of the component to set the property.
- propertyname (required): The name of the property to set. The component must have a set method for that property
- expression(required): The expression to evaluate to set the property
- datasource (required): The datasource to use to evaluate the expression

**raw**

Insert some raw Wml in the component (WML is in the body content of the tag). This will allow the programmatic change of the WML in the components body.

- name (required): The name of the component
- visible (optional): Whether or not the item is initially visible
- filename (optional): Wml File Name

- datasource (optional): An expression used to fill the contents of the datasource.

### **select**

This tag represents a Select tag in WML. Creates a WmlSelect component.

- name (required): The name of the component
- enabled (optional): Whether or not the item is initially enabled
- visible (optional): Whether or not the item is initially visible
- multiple (optional): Whether or not the multiple items can be selected
- datasource (optional): The column to bind the select field to
- classname (optional): The name of the style class to use for this object
- tabindex: Specifies the tabbing position of this element.
- iname: Specifies the name of the variable to set with the index of the selected item/items
- value: The default value of the component.
- ivalue: The default selected index value of the component.
- title: Specifies a title for this element, which may be used for display purposes.

Nested Tag:

**option:** Adds options to a select tag

Attributes:

- name (required): The name of the component
- value: The value of the option.
- visible (optional): Whether or not the item is initially visible
- datasource (optional): An expression that computes the onpick attribute from data in a datasource.
- onpick (optional): The URL of the location to go to if this item is picked
- title: Specifies a title for this element, which may be used for display purposes.
- classname (optional): The name of the style class to use for this object
- table: The table to use in the database for the option
- keycolumn: The column to use for key values for the option
- displaycolumn: The column to use for display values for the option
- nulloption: Add a blank at the top (normally used for a search field)
- toupper: Used to make sure all keys and display values are upper case
- criteria: the criteria to use to retrieve the data from the table

### **submit**

Provides an anchor tag to its own page which acts as a submit button. This will create a WmlSubmit component.

Attributes:

- name: (Required) The name of the component
- title: (Required) The title attribute is the text of the submit link which will appear on the WAP device.
- classname: (Optional) The class attribute of the tag

- datasource: (Optional) The datastore expression to compute the title attribute. If used it will overwrite the attribute with the results of the expression.

### **text**

Insert some text into the Wml. The text contents, font and visibility can be programmatically controlled.

#### Attributes:

- name (required): The name of the component
- text (optional): The text to display
- visible (optional): Whether or not the item is initially visible
- classname (optional): The HTML Style to use
- fixwml (optional): A true / false value indicating whether special html characters, like >, < or " should be converted to &gt; &lt; &quot; (default = true).
- datasource (optional): A expression that specifies the text used to populate this column.
- bold (optional): Whether or not the item is bold
- big (optional): Whether or not the item is big
- italic (optional): Whether or not the item is italic
- small (optional): Whether or not the item is small
- strong (optional): Whether or not the item is strong
- underline (optional): Whether or not the item is underline