JSValueChangeListener

Description

This component replicates the 'Value Change Listener' functionality on the client side. It can be used when the user would like a change in the value of one control to trigger off changes in the states of other controls. One or more Javascript Listeners can be nested within the source control (a control belonging to the 'javax.faces.Input' family). When the value of the source control is modified, the listeners are triggered and the states of the target controls modified.

Screen Shot

Not a Visual Component

API

component-family	javax.faces.Output
renderer-type	org.apache.myfaces.JsValueChangeListener
component-class	org.apache.myfaces.custom.jslistener.JsValueChangeListener
renderer-class	org.apache.myfaces.JsValueChangeListener
tag-class	org.apache.myfaces.custom.jslistener. JsValueChangeListenerTag

Usage

- You can use this component when you want to assign a client side 'Value Change Listener'.
- This value change listener can have one source component, and one an optional target component.
- The listener is called when value of source component changes.
- The listener is an expression in `expressionValue` attribute which should evaluate to true or false.
- The listener can access the source and target component using '$srcElem' and '$destElem' keywords in its expression
- The `for` and `property` value of tag define target of listener the result of expression will be written into this target, note both attributes are optional.
- You can have multiple `jsValueChangeListener` tags with different attributes including targets and expressions in one tag.

Syntax

```
<t:jsValueChangeListener>
```

```
<t:jsValueChangeListener
   for="id"
   property="property"
   expressionValue="{true|false}"
   bodyTagEvent="eventName" />
```

Instructions

Attributes

name	'_required *	*_ 'description'"
bodyTagEvent	false	If specified this JavaScript event will be inserted in the body tag. JavaScript code will be the same like it is rendered in the parent component. Events are triggered by the 'onchange' event of the source control. Here, an additional event can be specified (onload?)
expressionValue	true	the javascript expression to evaluate. The keyword '$srcElem' resolves to the source control and the keyword '$destElem' resolves to the target control
for	false	The result of the evaluated expression is assigned to the specified property of the target control
property	false	the id of the target control

Configuration

Don't need any extra configuration.

Notes and Known issues

-

Examples

Example 1, Two inputText Components Connected

Suppose we have two text fields on a page. We would like to keep the value of the second text field in sync with the value of the first. This can be accomplised with the following code:

```
<h:inputText id="text1">
   <t:jsValueChangeListener for="text2" property="value" expressionValue="$srcElem.value" />
   </h:inputText>
<h:inputText id="text2"/>
```

When the value of text1 changes, the 'onchange' event is triggered. The javascript expression specified by 'expressionValue' is evaluated, and the result is assigned to the specified property (in this case, 'value') of the target control.

Example 2

Sometimes, the evalution of the javascript expression itself causes the desired side-effect. In this case, it is not necessary to specify the 'property' attribute for the target control. In this example, we have a combo-box, and we want the selection of a specific value in the combo-box to cause a text box to be hidden.

```
<h:selectOneMenu id="selone_menu_colors" value="red" styleClass="selectOneMenu">
   <f:selectItems value="#{carconf.colors}" />
   <t:jsValueChangeListener for="selone_menu_subcolors"
         expressionValue="($srcElem.options[$srcElem.selectedIndex].value=='black')?
                          $destElem.style.display='inline':$destElem.style.display='none';"/>
</h:selectOneMenu>
<h:inputText id="selone_menu_subcolors"/>
```

The evaluation of the expression causes the text box to be hidden when the appropriate value is selected.

FAQ

Additional Information