

# SubmitPageOnValueChange

## How To Submit a page when an input control value changes (action)

Note: This task requires javascript.

The description below has been collected into a new component: [SubmitOnEvent](#)

To submit a page when an input control value changes, add an onchange attribute to call a javascript function which will click a commandButton. The commandButton submits the form which contains the control. The javascript function must reference the fully-qualified button name (in the example below, myForm:myUpdateButton)

Optionally, you can use a hidden commandButton to invoke an action other than the default form submit button. (display:none or visibility:hidden)

### Example

```
<script language="javascript">
    function pulldownChanged(){
        document.getElementById( "myForm:myUpdateButton" ).click();
    }
</script>

[ ... ]

<f:view>
    <h:form id="myForm">

        <h:selectOneMenu
            id="namedLocationInput"
            onchange="pulldownChanged()"
            value="#{page.physicalLocation}"
            title="No location selected"
            required="true">
            <f:selectItems
                value="#{page.locationItems}" />
            </h:selectOneMenu>

            <h:commandButton
                id="myUpdateButton"
                value="Submit"
                action="#{page.update}" />

        </h:form>
    </f:view>
```

Thanks to Slawek for this tip.

Note that for checkboxes and radio buttons you will want to use onclick instead of onchange as IE6 fires the onchange during 'blur' instead of the actual change of these input controls whereas onclick is fired when the value is actually changed for all browsers. Thanks to Andrew Robinson for this tip.

In the previous example, if you don't want your button to be seen, use the style style="visibility:hidden;", so the button is hidden and the effect is better 😊

## How To Submit a page when an input control value changes (valueChangeListener)

If you do not need navigation rules to apply, you can do this:

### Example

```

<f:view>
  <h:form>

    <h:selectOneMenu
      onchange="submit()"
      value="#{page.physicalLocation}"
      title="No location selected"
      required="true"
      valueChangeListener="#{page.menuChanged}">
      <f:selectItems
        value="#{page.locationItems}" />
      </h:selectOneMenu>

  </h:form>
</f:view>

```

1. less javascript
2. no need to use "id"s

If you need to update model values (e.g. to fill a second menu) in your valueChangeListener, please see here: <http://issues.apache.org/jira/browse/MYFACES-864>

## How To Submit a page when hitting "enter"

Here's another example provided by Bruno Aranda. It is doing a form submission with just a text field for something like a quick search. It works as follows:

1. Include this javascript in the <head>
2. Call the Javascript function from the attribute 'onkeypress' of your text field
3. Include the hidden button, which will submit the form and with the id used in the onkeypress event

```

<script language="javascript">
  function submitEnter(commandId,e)
  {
    var keycode;
    if (window.event)
      keycode = window.event.keyCode;
    else if (e)
      keycode = e.which;
    else
      return true;

    if (keycode == 13) {
      document.getElementById(commandId).click();
      return false;
    } else
      return true;
  }
</script>

[...]

<h:inputField value="Hello" onkeypress="return submitEnter('submitLogin',event)"/>

[...]

<x:commandButton id="submitLogin" forceId="true" action="#{yourBean.whateverAction}" style="visibility:hidden;"
/>

```

If you need to submit a commandLink instead of a commandButton, use the clickLink() function from [JavascriptWithJavaServerFaces](#). Then use the example above, but change the last "if" statement to the following:

```

  if (keycode == 13) {
    clickLink(commandId);
    return false;
  }

```