AliasBean

<x:aliasbean/>

The aliasbean tag is a pseudo tag, which allows the encapsulation of web parts into reusable pseudo components. Similar mechanisms exist currently in Webobjects, Struts, Tapestry and Webforms.

Usage and functionality

pseudo content of a subpage (file subpage.jsp)

```
...
<h:outputFormat value="#{holder.title}"/>

<h:outputFormat value="#{holder.description}"/>
... some other code
```

```
... main form using the alias bean
<x:aliasBean alias="#{holder}" value="#{databean1}">
 <f:subview>
   <jsp:include page="subpage.jsp"/>
 </f:subview>
</x:aliasBean>

<x:aliasBean alias="#{holder}" value="#{databean2}">
 <f:subview>
   <jsp:include page="subpage.jsp"/>
 </f:subview>
</x:aliasBean>

<x:aliasBean alias="#{holder}" value="#{databean3}">
 <f:subview>
   <jsp:include page="subpage.jsp"/>
 </f:subview>
</x:aliasBean>
```

The form subpage is reused on three different places in the main form. The include is not mandatory. Between the alias tags there can be any kind of jsf code.

*Note, trying to bind a controller delegate (binding) to the alias bean wont work, the alias bean is rendered in a later phase of the jsf rendering cycle.

Combining alias beans

In MyFaces 1.0.9 combining alias beans only works by nesting them, this restriction might be removed in later incarnations of MyFaces.

In the current nightly build, this is done by <t:aliasBeansScope>. <t:aliasBean> tags inside an alias scope define an alias in the scope of that tag, not in themselves. This can be used if the content has more then 1 variable:

```
... main form using the alias bean
<x:aliasBeansScope>
  <x:aliasBean alias="#{holderA}" value="#{databean1A}" />
  <x:aliasBean alias="#{holderB}" value="#{databean1B}" />
  <x:aliasBean alias="#{holderC}" value="#{databean1C}" />
  <f:subview>
    <jsp:include page="subpage.jsp"/>
  </f:subview>
</x:aliasBeansScope>

<x:aliasBeansScope>
  <x:aliasBean alias="#{holderA}" value="#{databean2A}" />
  <x:aliasBean alias="#{holderB}" value="#{databean2B}" />
  <x:aliasBean alias="#{holderC}" value="#{databean2C}" />
  <f:subview>
    <jsp:include page="subpage.jsp"/>
  </f:subview>
</x:aliasBeansScope>

<x:aliasBeansScope>
  <x:aliasBean alias="#{holderA}" value="#{databean3A}" />
  <x:aliasBean alias="#{holderB}" value="#{databean3B}" />
  <x:aliasBean alias="#{holderC}" value="#{databean3C}" />
  <f:subview>
    <jsp:include page="subpage.jsp"/>
  </f:subview>
</x:aliasBeansScope>
```

Note

Sadly Alias beans cannot be used as bindings for components. So the following will not work.

```
... In main page
<x:aliasBean alias="#{AliasTable}" value="#{FailedCustomerOrdersTable}">
  <f:subview>
    <jsp:include page="GenericDataTable.jsp"/>
  </f:subview>
</x:aliasBean>

... In GenericDataTable.jsp

<h:dataTable binding="#{AliasTable}"
             .... >
...
```

Component bindings are resolved at creation time of the component and are thus out of scope of the alias bean component. JSF allows you to create only one component instance for one jsf tag.

If you just need to access the component instance try to use a managed bean which acts as an interface between the component and the aliased bean. The component instance is stored in the non aliased bean and the aliased bean uses the non aliased bean to access the component instance.

For a workaround please look at the followng issue in JIRA concerning this. http://issues.apache.org/jira/browse/MYFACES-334