


Cache EL Expressions

 [ELResolver ordering](#)

 [Configuration of special features](#)

[Update all form javax.faces.ViewState hidden fields between ajax requests](#) 

Since MyFaces Core 2.0.8 / 2.1.2, a new optimization was added to prevent create EL Expressions unnecessary. Each time a page is build, facelets compiler create all EL expressions, and then evaluate them, so with this special configuration, you can reduce the time and memory resources required to build a view. Just add this to your web.xml file:

```
<context-param>
  <param-name>org.apache.myfaces.CACHE_EL_EXPRESSIONS</param-name>
  <param-value>always</param-value>
</context-param>
```

There are 4 valid modes for this param:

- always: Only does not cache when expressions are inside user tags or the expression contains a variable resolved using VariableMapper
- allowCset: Like always, but does not allow cache when ui:param was used on the current template context
- strict: Like allowCset, but does not allow cache when c:set with var and value properties only is used on the current page context
- noCache: All expression are created each time the view is built

To enable this optimization, you should check first some tips to see which option can be enabled in your application.

In theory the mode "always" does not work with the following case:

a.xhtml

```
<ui:composition template="c.xhtml">
  <ui:param name="var1" value="value1"/>
</ui:composition>
```

b.xhtml

```
<ui:composition template="c.xhtml">
  <ui:param name="var1" value="value1"/>
  <ui:param name="var2" value="value2"/>
</ui:composition>
```

c.xhtml

```
<ui:composition>
  <h:outputText value="#{var1}"/>
  <h:outputText value="#{var2}"/>
</ui:composition>
```

if a.xhtml view is constructed before b.xhtml, #{var2} will be cached, even if this is not wanted and then when b.xhtml is called, the expression will not work correctly.

So the first tip to use this param is check if your composition declarations (ui:include, ui:decorate, ui:composition) always use the same number of params.

The mode "allowCset" and "always" does not work with the following case too:

csetuse.xhtml

```
<c:if test="...">
  <c:set var="attribute1" value="somevalue" />
</c:if>

<!-- some use of attribute1 -->
```

The problem here consists in a value expression is created conditionally, but if the expression is not created the first time, other value expressions will not be marked as cachable.

The solution is use this syntax instead:

csetuse.xhtml

```
<c:set var="attribute1" value="#{somecondition ? 'somevalue' : null}"/>
```

In this way, any use of attribute1 will detect and handle the expression correctly.

In "strict" mode, any usage of the previous two tags (c:if and ui:param) will prevent cache inner EL expressions. The mode "noCache" (by default) will always recreate expressions each time the view is build.

Related issue [MYFACES-3160](#)

 [ELResolver ordering](#)

 [Configuration of special features](#)

[Update all form javax.faces.ViewState hidden fields between ajax requests](#) 