

## AF3 Phoenix - Change Request #3877

### [UI] Move type checks and casts from LWFxef visuals to a common base class within SystemFocus

11/29/2019 03:39 PM - Tiziano Munaro

<b>Status:</b>	New	<b>Start date:</b>	11/29/2019
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	Florian Hölzl	<b>% Done:</b>	0%
<b>Category:</b>		<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>	AF3 2.18 - Feature Freeze		
<b>Description</b>			
Checks of the type			
<pre>Object model = mvcb.getModel(); if(!(model instanceof ILayoutedModelElement)) {     throw new IllegalArgumentException(         "Expected model of type ILayoutedModelElement, but was " +         model.getClass().getSimpleName() + "."); }</pre>			
shall be moved to a common base class within the SystemFocus kernel. MVCBundlePartBase would be the most common class to place such a check. This way the class itself does not have to be parameterized while still avoiding code duplication. The same base class could be accommodate a generic method for returning the casted model element:			
<pre>protected &lt;T&gt; T getModelElement(Class&lt;T&gt; elementType)</pre>			
Of course calling this method with a wrong parameter leads to class-cast exceptions, but the expected type is already checked in the constructor. Also, a misuse is limited to derived classes, where a contributor should know the context.			
Moreover, the getName method of the ContentVisualBase is currently returning an empty string, leaving it to its children to implement a component-specific override. This method is usually identical in all visuals, except for the class-cast. By using the getName method of the KernelUIUtils directly in the ComponentVisualBase, this code duplication could be avoided:			
<pre>/** {@inheritDoc} */ @Override protected String getName() {     return KernelUIUtils.getName(getModel()); }</pre>			

#### History

##### #1 - 12/06/2019 09:40 AM - Tiziano Munaro

- Description updated

##### #2 - 12/09/2019 12:00 PM - Tiziano Munaro

- Description updated

##### #3 - 12/23/2019 04:01 PM - Florian Hölzl

LWFxef does only depend on JavaFX and is located in the kernel.common plugin. It therefore knows neither about ILayoutedElement (kernel.base) nor KernelUIUtils (kernel.core.ui). Abstract base classes should therefore go there, but not into MVCBundlePartBase.

##### #4 - 05/11/2020 10:47 AM - Johannes Eder

- Target version changed from AF3 2.17 (Feature Freeze) to AF3 2.18 - Feature Freeze