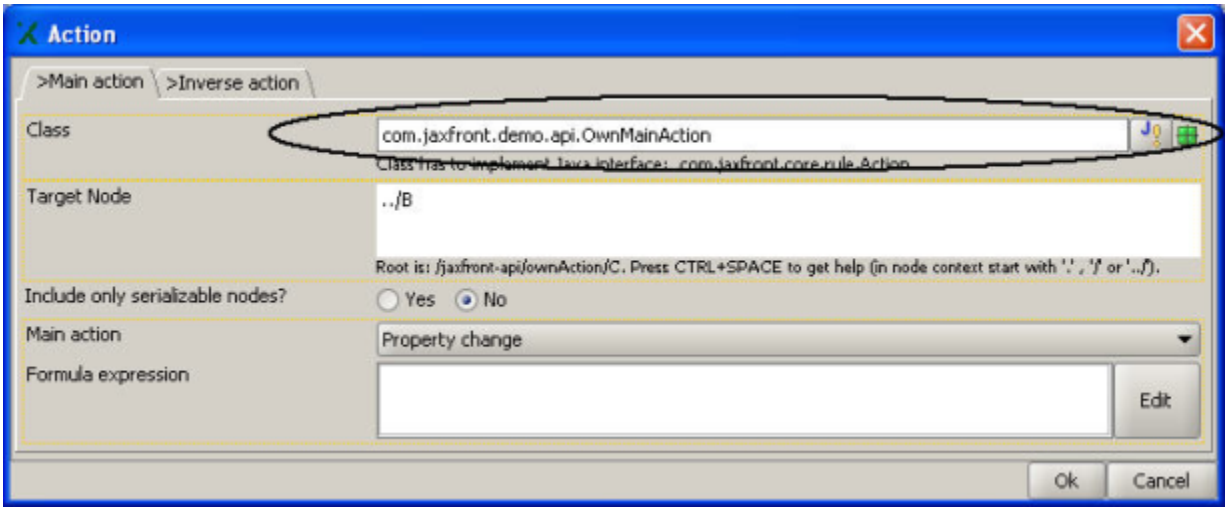
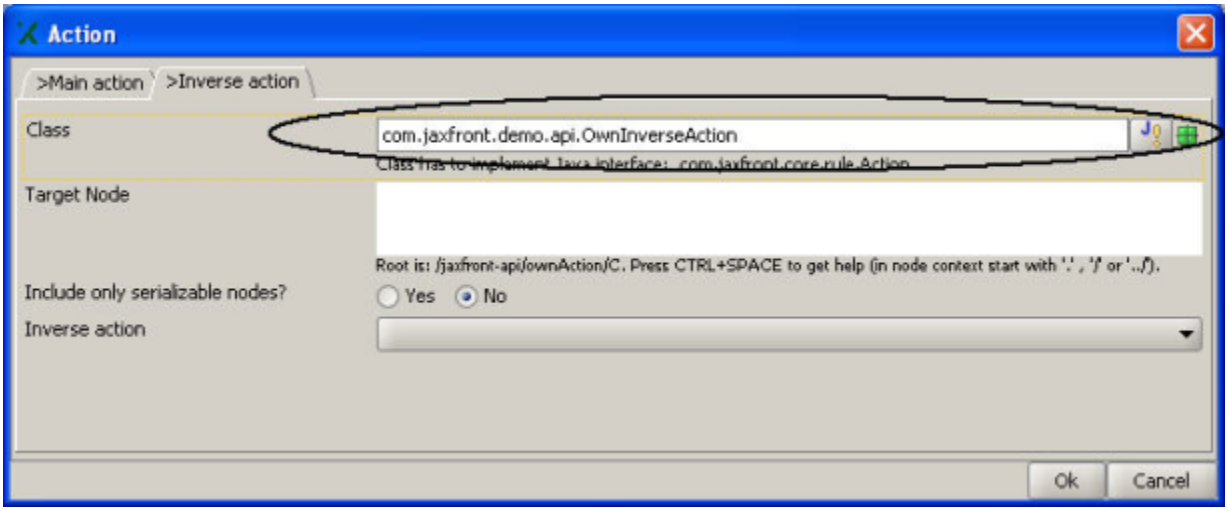
(4) Write your own action

Content

Overview

Within a XUI rule it's possible to define own actions to get fired. Conditions get checked if an event occurs, if the condition is true all corresponding actions are fired(ECA, Event-Condition-Action).
Instead of using the built in actions (uiActions), one can provide an own action implementation. If a predefined UI action is choosen in combination with an custom action implementation, both actions will be fired. The inverse action is usually a inverse implementation of the main action.





Implementation Info

The class need to implement the Action interface: ***com.jaxfront.core.rule.Action***

Name	Description
perform (Type, Type , EventObject)	This method gets called if the condition is true. The passing type is the target type of the action as defined in the xui rule (Target Node). If the target node is not specified, the source of the rule definition will become the target. If no target node is defined in the inververse action, the target node from the main action will be choosen.

MainAction

⚠ See class com.jaxfront.demo.api.OwnMainAction

```
/**
 * This Action marks the visualizer of the target node with a green border.
 */
public class OwnMainAction extends Object implements Action {

    @Override
    public void perform(final Type eventSource, final Type target, final EventObject event) {
        AbstractSimpleTypeView component = (AbstractSimpleTypeView) TypeVisualizerFactory.getInstance().
getVisualizer(target);
        component.getEditorField().setBorder(BorderFactory.createEtchedBorder(Color.green, Color.green));
    }
}
```

InverseAction

⚠ See class com.jaxfront.demo.api.OwnInverseAction

```
/**
 * This Action resets the border of the target node with an empty line border.
 */
public class OwnInverseAction implements Action {

    @Override
    public void perform(final Type eventSource, final Type target, final EventObject event) {
        AbstractSimpleTypeView component = (AbstractSimpleTypeView)TypeVisualizerFactory.getInstance().
getVisualizer(target);
        component.getEditorField().setBorder(BorderFactory.createLineBorder(Color.black));
    }
}
```