

# Listening to property change events

The *PropertyChangeEvents* support provides the ability to listen to the event by different scopes. The scope depends on how the listener is registered:

- If the listener is registered at the whole repository, it receives events about the changes in all the elements.
- If the listener is registered at the specific element, it receives events about the changes in any property of this element.
- If the listener registered with the specific property at the element, it receives events about the changes in this property.
- If the listener is registered with the specific property for the element type, it receives events about the changes in all the elements of the specific type.

The *java.beans.PropertyChangeListener* should be registered to receive these events. There are several different ways to register a listener:

- To listen for the whole repository, use:

```
project.getRepositoryListenerRegistry().addPropertyChangeListener  
(listener, (RefObject)null);
```

This listener will get all the property change events from any element.

- To listen for any delete operation in the repository and get notified before the delete action is performed, use:

```
project.getRepositoryListenerRegistry().  
addPropertyChangeListener(listener, UML2MetamodelConstants.BEFORE_DELETE);
```

This listener will be notified, when any model element is set to be deleted.

- To listen for any property changes of the specific element, use:

```
element.addPropertyChangeListener(listener);
```

This listener will be notified when any element property is changed.

- To listen for any property changes of the specific element type, use:

```
project.getSmartEventSupport().registerConfig(aClass,  
configs, listener);
```

This listener will be notified, when any property of any element in the project with this type is changed. If "configs" is NULL, the listener will get all property change events.



*EventSupport* could be disabled from the event firing:

- To check if it is enabled, use:

```
project.getRepository().getEventSupport().  
isEnabledEventFiring()
```

- To stop/ start event firing, use:

```
project.getRepository().getEventSupport().  
setEnabledEventFiring(...)
```

These examples show, how to create the property change listeners to listen to the different kind of properties.

**Listener for listening to the specific element's any property changes:**

```
element.addPropertyChangeListener(new PropertyChangeListener()
{
    public void propertyChange(PropertyChangeEvent evt)
    {
        // evt.getPropertyName() is changed
    }
});
```

**Listener for listening to the specific property (NAME) of the specific element:**

```
element.addPropertyChangeListener(new PropertyChangeListener()
{
    public void propertyChange(PropertyChangeEvent evt)
    {
        if (PropertyNames.NAME.equals(evt.getPropertyName()))
        {
            // name is Changed
        }
    }
});
```