

Refactoring model elements

To refactor an element in a model, use the [com.nomagic.magicdraw.uml.Refactoring](#) class.

Example #1: Converting an element to an interface

```
Element elementToConvert = ...;
Project project = ...;
SessionManager sessionManager = SessionManager.getInstance();
sessionManager.createSession(project, "Convert");

// Converts the element to an interface.
ConvertElementInfo info = new ConvertElementInfo(Interface.class);

// Preserves the old element ID for the new element.
info.setPreserveElementID(true);
Element conversionTarget = Refactoring.Converting.convert
(elementToConvert, info);
sessionManager.closeSession(project);
```

Related pages

- [Session management](#)
- [Checking element editing permissions](#)

Example #2: Replacing an element with another element

```
Element elementToReplace = ...;
Project project = ...;
SessionManager sessionManager = SessionManager.getInstance();
sessionManager.createSession(project, "Replace");

ConvertElementInfo info = new ConvertElementInfo(elementToReplace.
getClassType());
info.setConvertOnlyIncomingReferences(true);

Refactoring.Replacing.replace(element, elementToReplace, info);

sessionManager.closeSession(project);
```

Example #3: Extract refactoring

Use the [com.nomagic.magicdraw.uml.Refactoring.Extracting](#) class to create the extract manager for a symbol which you want to extract. Configure the extract refactoring by changing [com.nomagic.magicdraw.uml.refactor.extract.ExtractSource](#) and [com.nomagic.magicdraw.uml.refactor.extract.ExtractTarget](#). Invoke the refactoring with [com.nomagic.magicdraw.uml.refactor.extract.ExtractManager.extract\(\)](#). Review refactoring results by inspecting [ExtractSource](#) and [ExtractTarget](#).

```

    // Creates an extract refactor manager.
    ExtractManager extractManager = Refactoring.Extracting.
createExtractManager(symbols);
    if (extractManager != null)
    {
        Project project = ...;
        // A session has to be started before refactoring.
        SessionManager sessionManager = SessionManager.getInstance();
        sessionManager.createSession(project, "Extract Refactor Symbols");

        // We may control the extract refactor result by modifying extract
target.
        ExtractTarget extractTarget = extractManager.getExtractTarget();

        // Create a namespace to which we are going to refactor.
        Project project = Project.getProject(symbols[0]);
        Package packagel = project.getElementsFactory().
createPackageInstance();
        packagel.setOwner(project.getPrimaryModel());

        // Set the namespace to which the extract result should go.
        extractTarget.setTargetNamespace(packagel);

        // Choose target diagram type from allowed diagram types if the
default type does not suite.
        List<String> allowedTargetDiagramTypes = extractTarget.
getAllowedTargetDiagramTypes();
        extractTarget.setTargetDiagramType(allowedTargetDiagramTypes.get
(0));

        // Modify reference names which link the extract refactor source
to the target.
        List<? extends ExtractReference> references = extractTarget.
getReferences();

        for (int i = 0; i < references.size(); i++)
        {
            ExtractReference reference = references.get(i);
            reference.setName(Integer.toString(i));
        }

        // We may control the extract refactor source by modifying the
extract source.
        ExtractSource extractSource = extractManager.getExtractSource();
        extractSource.setElementName("sourceElementName");

        // Perform actual refactoring.
        extractManager.extract();
        sessionManager.closeSession(project);

        // The element which was created in the source during refactoring.
        Element sourceElement = extractSource.getElement();

        // The element which was created in the target during refactoring.
        Element targetElement = extractTarget.getElement();

        // The diagram which was created in the target during refactoring.
        DiagramPresentationElement targetDiagram = extractTarget.
getDiagram();
    }

```

Example #4: Reverse relationship refactoring

The relation reverse refactoring can be done using the [com.nomagic.magicdraw.uml.Refactoring.RelationReversing.reverseRelationDirection\(Element\)](#) method.

```
// We have an arbitrary element or symbol which represents a
relationship.
BaseElement baseElement = ...;
Project project = ...;
// Relationship reversing should be wrapped with session create/close
calls.
SessionManager sessionManager = SessionManager.getInstance();
sessionManager.createSession(project, "Reverse relation");

// Reverse the relationship.
Refactoring.RelationReversing.reverseRelationDirection(baseElement);

// Close the session.
sessionManager.closeSession(project);
```

Example #5: Moving element with connected relationships to other owner

Use [com.nomagic.magicdraw.uml.Refactoring.Moving.moveElementsWithRelation\(java.util.Collection<Element>, Element\)](#) to move [com.nomagic.uml2.ext.magicdraw.classes.mdkernel.Element](#) and all connected [com.nomagic.uml2.ext.magicdraw.classes.mdkernel.Relationship\(s\)](#) to a new owner.



You can find the code examples in `<modeling tool installation directory>\openapi\examples\refactor`