

Generic tables

A generic table is a diagram with a specific stereotype. It is used to represent the element properties in a table form. It can represent different element types in one table and show all their properties. Table rows represent elements, and columns represent element properties. For more information about generic tables and their managing, see [Generic Table](#).

The Open API of modeling tools provides the [com.nomagic.generictable.GenericTableManager](#) class for creating and manipulating generic tables.

This API is used for creating a generic table, adding, editing, and removing table columns and rows, as well as getting cell values.



Find the Generic Table Manager Example Plugin in <MagicDraw_installation_directory>\openapi\examples\generictablemanager.

Creating generic table

To create a generic table, use the [GenericTableManager.createGenericTable\(Project, java.lang.String\)](#) method.

```
Diagram createdDiagram = GenericTableManager.createGenericTable(project,
"Generic table name");
```

Setting generic table filter

To set a generic table element types, use the [GenericTableManager.setTableElementTypes\(Diagram, java.util.List<java.lang.Object>\)](#) method. The type of the element types list could be the Class or the Stereotype. See an example of the table element types:

```
//The table element types list
Set<Class> set = new HashSet<Class>();
List<Object> tableElementTypes = new ArrayList<Object>();

//Get all row elements
rowElements = GenericTableManager.getRowElements(createdDiagram);
for (int i = 0; i < rowElements.size(); i++)
{
    set.add(rowElements.get(i).getClassType());
}
tableElementTypes.addAll(set);
```

Getting generic table element types

To get the generic table element types, use the [GenericTableManager.getTableElementTypes\(Diagram\)](#) method:

```
List<Object> setElementTypes = GenericTableManager.getTableElementTypes
(createdDiagram);
```

Adding/getting columns

To add columns to the generic table, use the [GenericTableManager.addColumnsById\(Diagram, java.util.List<java.lang.String>\)](#) method.

To get all possible columns from the generic table, use the [GenericTableManager.getPossibleColumnIDs\(Element\)](#) method. To get set columns for the generic table, use the [GenericTableManager.getColumnIds\(Diagram\)](#) method.

```
List<String> columnIDs = GenericTableManager.getPossibleColumnIDs
(elementType);
```

[getColumnIds\(Diagram\)](#) returns columns from the generic table:

On this page

- [Creating generic table](#)
- [Setting generic table filter](#)
- [Getting generic table element types](#)
- [Adding/getting columns](#)
- [Getting cell value](#)
- [Getting column names](#)
- [Getting row elements](#)
- [Adding row element](#)
- [Removing row element](#)

Related pages

- [Session management](#)

```
List<String> columnIDs = GenericTableManager.getColumnIds(createdDiagram);
GenericTableManager.addColumnsById(createdDiagram, columnIDs);
```

How to get *columnIDs*, please look at the *com.nomagic.magicdraw.examples.generictablemanager.GenericTableManagerExamplePlugin.GenericTableManagerAction* class in the *GenericTableManagerExamplePlugin.java* file. The file can be found in *<program_installation_directory>\openapi\examples\generictablemanager*.

Getting cell value

To get a value of the generic table call, use the [GenericTableManager.getCellValue\(Diagram, Element, java.lang.String\)](#) method:

```
Property mdProperty = GenericTableManager.getCellValue(createdDiagram,
element, columnId);
```

Getting column names

To get column names from the generic table, use the [GenericTableManager.getColumnNames\(Diagram\)](#) method.

To get a single column name from the generic table, use the [GenericTableManager.getColumnNameById\(Diagram, java.lang.String\)](#) method.

```
List<String> columnNames = GenericTableManager.getColumnNames
(createdDiagram);
String columnName = GenericTableManager.getColumnNameById(createdDiagram,
columnID);
```

Getting row elements

To get row elements from the generic table, use the [GenericTableManager.getRowElements\(Diagram\)](#) method:

```
List<Element> rows = GenericTableManager.getRowElements(createdDiagram);
```

Getting visible row elements

To get visible row elements from the generic table, use the [GenericTableManager.getVisibleRowElements\(Diagram\)](#) method:

```
List<Element> rows = GenericTableManager.getVisibleRowElements
(createdDiagram);
```

Adding row element

To add a new row for the model element to the generic table, use the [GenericTableManager.addRowElement\(Diagram, Element\)](#) method:

```
GenericTableManager.addRowElement(createdDiagram, element);
```

The element you want to add must be of the same type as one of generic table filter types.

Removing row element

To remove a row from the generic table, use the [GenericTableManager.removeRowElement\(Diagram, Element\)](#) method:

```
GenericTableManager.removeRowElement(createdDiagram, element);
```