

# Configuring cell entries - dependencies

In order to get the dependencies between row and column elements in the dependency matrix, components of the [com.nomagic.magicdraw.dependencymatrix.datamodel.cell.DependencyExtractor](#) interface are used. These components allow to

- Define custom dependencies that can be created in a cell.
- Add shortcut menu commands for navigation (other than opening the [Specification window](#) of the element and selecting the element in the Containment tree).
- Construct smart listener configurations that define cases when some element dependencies should be updated instantly (without the full rebuild of the matrix).

Use the [com.nomagic.magicdraw.dependencymatrix.configuration.DependencyMatrixConfigurator.configureDependencyHandlers\(java.util.Collection<DependencyExtractor>, java.util.Collection<DependencyEditor>\)](#) method to register a custom dependency extractor. The method can remove the default dependency extractor from the collection and leave only a custom extractor.

The most important method is [com.nomagic.magicdraw.dependencymatrix.datamodel.cell.DependencyExtractor.getDependencies\(ElementNode, ElementNode\)](#). It is called to add dependencies in the matrix cell. By using custom algorithms you can create additional instances of the [com.nomagic.magicdraw.dependencymatrix.datamodel.cell.DependencyEntry](#) class for each cell.

## Configuring the shortcut menu of a cell

All commands of the shortcut menu of a cell, are available only from the components of the [com.nomagic.magicdraw.dependencymatrix.datamodel.editing.DependencyEditor](#) interface. These components allow for

- Defining if new dependency can be created in the cell.
- Defining if current cell value can be edited.
- Adding custom actions for creating or editing dependencies.

Use the [DependencyMatrixConfigurator.configureDependencyHandlers\(java.util.Collection<DependencyExtractor>, java.util.Collection<DependencyEditor>\)](#) method to register a custom dependency editor. The method implementation can remove the default dependency editor from the collection and leave only a custom editor.

Dependency matrix cells with the default renderer can be displayed either as editable or as read-only. If it is possible to create or delete at least one dependency in the cell, the cell is displayed as editable, that is, the cell's background is white.

When adding a new command, the [DependencyEditor.canCreate\(PersistenceManager, ElementNode, ElementNode, AbstractMatrixCell\)](#) method is first called and if it returns *true*, the [DependencyEditor.createAddActions\(PersistenceManager, ElementNode, ElementNode, AbstractMatrixCell, ActionsCategory, ActionsCategory, ActionsCategory\)](#) method is called to add the command.

When editing a command, the [DependencyEditor.canEdit\(PersistenceManager, ElementNode, ElementNode, AbstractMatrixCell\)](#) method is first called and if it returns *true*, the [DependencyEditor.createEditActions\(PersistenceManager, ElementNode, ElementNode, AbstractMatrixCell, ActionsCategory, ActionsCategory\)](#) method is called to edit the command.