UML PROFILING AND DSL

18.1

user guide

No Magic, Inc.
2015
## CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customization Element Attributes</td>
<td>44</td>
</tr>
<tr>
<td>- Meta Property 44</td>
<td></td>
</tr>
<tr>
<td>- Property Group 46</td>
<td></td>
</tr>
<tr>
<td>- Auto Numbering 46</td>
<td></td>
</tr>
<tr>
<td>- Derived Property Specification 47</td>
<td></td>
</tr>
<tr>
<td>Rules of Stereotypes that cannot be Allowed to Apply</td>
<td>47</td>
</tr>
<tr>
<td>Creating Inheritance of DSL Customization</td>
<td>48</td>
</tr>
<tr>
<td>Using Customization Data</td>
<td>49</td>
</tr>
<tr>
<td>- Customizing Specification Window</td>
<td></td>
</tr>
<tr>
<td>- Using standard UML properties</td>
<td>49</td>
</tr>
<tr>
<td>- Always visible properties</td>
<td>50</td>
</tr>
<tr>
<td>- Meta property substitution (changing name of UML property)</td>
<td>54</td>
</tr>
<tr>
<td>Customizing Element Shortcut Menu</td>
<td>59</td>
</tr>
<tr>
<td>- Quick property editor</td>
<td>59</td>
</tr>
<tr>
<td>- Quick stereotype applying</td>
<td>59</td>
</tr>
<tr>
<td>Customizing Drag and Drop</td>
<td>60</td>
</tr>
<tr>
<td>- Properties used in drag-and-drop customization</td>
<td></td>
</tr>
<tr>
<td>- Selecting one of the few drag-and-drop rules</td>
<td>63</td>
</tr>
<tr>
<td>Creating Custom Rules for Relationships</td>
<td>64</td>
</tr>
<tr>
<td>Defining Preferred Metatype</td>
<td>65</td>
</tr>
<tr>
<td>Initializing Custom Model</td>
<td>65</td>
</tr>
<tr>
<td>- Default values</td>
<td>66</td>
</tr>
<tr>
<td>- Stereotypes on relationship ends</td>
<td>66</td>
</tr>
<tr>
<td>- Required Generalization or Interface Realization</td>
<td>66</td>
</tr>
<tr>
<td>Creating DSL Element from Customized Category in Shortcut Menu</td>
<td>67</td>
</tr>
<tr>
<td>Customizing Possible Owned Elements</td>
<td>68</td>
</tr>
<tr>
<td>- Customizing owned types</td>
<td>69</td>
</tr>
<tr>
<td>- Customizing owned diagrams</td>
<td>69</td>
</tr>
<tr>
<td>- Customizing suggested relationships</td>
<td>70</td>
</tr>
<tr>
<td>Creating Element-Specific Help Topics</td>
<td>70</td>
</tr>
<tr>
<td>Creating DSL Element from Shortcut Menu and Other Locations</td>
<td>71</td>
</tr>
<tr>
<td>Customizing Symbols</td>
<td>73</td>
</tr>
<tr>
<td>- Setting default symbol size</td>
<td>73</td>
</tr>
<tr>
<td>- Setting custom path style</td>
<td>74</td>
</tr>
<tr>
<td>Creating Property Groups and Subgroups</td>
<td>75</td>
</tr>
<tr>
<td>- Type restriction for custom table in Specification window</td>
<td>80</td>
</tr>
<tr>
<td>- Merging property groups and subgroups</td>
<td>81</td>
</tr>
<tr>
<td>- Default visibility of property groups and subgroups</td>
<td>81</td>
</tr>
<tr>
<td>Creating Numbering Customizations</td>
<td>82</td>
</tr>
<tr>
<td>- Basic steps for creating numbering customization</td>
<td>83</td>
</tr>
<tr>
<td>- Case study: Creating your first numbering customization</td>
<td>84</td>
</tr>
<tr>
<td>- Relevant property values</td>
<td>92</td>
</tr>
<tr>
<td>Extending Metamodel with Derived Properties</td>
<td>93</td>
</tr>
<tr>
<td>- Specifying derived properties</td>
<td>94</td>
</tr>
<tr>
<td>- Defining expressions</td>
<td>97</td>
</tr>
<tr>
<td>- Recursive properties</td>
<td>98</td>
</tr>
<tr>
<td>- Expressions merge</td>
<td>98</td>
</tr>
<tr>
<td>- Derived properties visibility</td>
<td>99</td>
</tr>
<tr>
<td>DISTRIBUTING RESOURCES</td>
<td>102</td>
</tr>
<tr>
<td>Exchanging Profiles via XMI</td>
<td>102</td>
</tr>
<tr>
<td>Distributing Resources by Using Resource Manager</td>
<td>102</td>
</tr>
</tbody>
</table>
UML is a general purpose visual modeling language for specifying, constructing and documenting the artifacts of systems that can be used with all major application domains and implementation platforms. It has been widely adopted by both industry and academia as the standard language for describing software systems.

However, the fact that UML is a general-purpose notation may limit its suitability for modeling some particular specific domains (e.g. web applications or business processes), for which specialized languages and tools may be more appropriate.

OMG defines two possible approaches for defining domain-specific languages. The first one is based on the definition of a new language, an alternative to UML, using the mechanisms provided by OMG for defining object-based visual languages (i.e., the same mechanisms that have been used for defining UML and its metamodel). In this way, the syntax and semantics of the elements of the new language are defined to fit the domain's specific characteristics. The problem is that standard UML tools will not be able to deal with such a new language.

The second alternative is based on the particularization of UML by specializing some of its elements, imposing new restrictions on them but respecting the UML metamodel, and without modifying the original semantics of the UML elements (i.e., the properties of the UML classes, associations, attributes, etc., will remain the same, but new constraints will be added to their original definitions and relationships).

UML provides a set of extension mechanisms (stereotypes, tag definitions, and constraints) for specializing its elements, allowing customized extensions of UML for particular application domains.

These customizations are sets of UML extensions, grouped into UML profiles. However UML profiles for specific domain cannot play role of specialized tool, they are just specialized metamodels.

MagicDraw UML tool provides ability to use DSL engine for adapting domain specific profiles to create your own custom diagrams, custom specification dialogs, custom real-time semantic rules and other. In other words user is able to create low-budget specialized domain-specific tool and hide UML underneath.

DSL customization is model-driven approach, based on UML profiling.

<table>
<thead>
<tr>
<th>DEFINITIONS</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DSL</strong></td>
<td>Domain Specific Language</td>
</tr>
<tr>
<td><strong>Stereotyped element</strong></td>
<td>An element with not customized stereotype applied.</td>
</tr>
<tr>
<td><strong>Customization element</strong></td>
<td>An element in which the customization data is specified. You can create the customization element by using a Profile diagram.</td>
</tr>
<tr>
<td><strong>Customization data</strong></td>
<td>A collection of properties and attributes of the Customization element.</td>
</tr>
<tr>
<td><strong>Customization model</strong></td>
<td>A collection of customization elements.</td>
</tr>
</tbody>
</table>
MagicDraw introduces several customization engines, based on UML Profiles:

- **Advanced UML Profiling**
  allows using some profiling enhancements that are not defined in UML, but that helps to solve some common problems like tag grouping, unwanted stereotypes and tags hiding and others. For more information, see "Working with Profiles" on page 8.

- **Customizing Diagrams**
  allows creating your own diagram types for custom profile, with your own toolbars, stereotyped elements, symbol styles and custom smart manipulators.
  Such customization is saved in the special "diagram descriptor", that could be exchanged between users, allowing them use your own custom diagrams.
  For more information, see "Customizing Diagrams" on page 16.

- **Domain Specific Language (DSL) Customization Engine**
  allows “tuning” domain specific profiles, customizing multiple GUI, model initialization, and semantic rules, creating your own Specification dialogs.
  DSL customization is model-driven approach, based on UML profiling. Customization is saved as a UML model.
  For more information, see "Domain Specific Language Customization" on page 27.

Below is the list of MagicDraw areas that are possible to customize using MagicDraw customization engine.

<table>
<thead>
<tr>
<th>Customization</th>
<th>Customization engine</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Custom diagrams</strong></td>
<td></td>
</tr>
<tr>
<td>Diagram type name</td>
<td>Custom diagram wizard</td>
</tr>
<tr>
<td>Base diagram type</td>
<td>Custom diagram wizard</td>
</tr>
<tr>
<td>Diagram abbreviation</td>
<td>Custom diagram wizard</td>
</tr>
<tr>
<td>Used profiles</td>
<td>Custom diagram wizard</td>
</tr>
<tr>
<td>Diagram toolbar customization</td>
<td>Custom diagram wizard</td>
</tr>
<tr>
<td>Symbol styles</td>
<td>Custom diagram wizard</td>
</tr>
<tr>
<td><strong>Custom shapes</strong></td>
<td></td>
</tr>
<tr>
<td>Scalable icon</td>
<td>Advanced UML Profiling</td>
</tr>
<tr>
<td>Custom draw and layout rules</td>
<td>Open API</td>
</tr>
<tr>
<td>Smart manipulators</td>
<td>Custom diagram wizard</td>
</tr>
<tr>
<td>Context actions for property changing</td>
<td>DSL engine</td>
</tr>
<tr>
<td>Keywords</td>
<td>DSL engine</td>
</tr>
<tr>
<td>Representation text</td>
<td>DSL engine</td>
</tr>
<tr>
<td>Custom shape size</td>
<td>DSL engine</td>
</tr>
<tr>
<td><strong>Custom paths</strong></td>
<td></td>
</tr>
<tr>
<td>Line style</td>
<td>Advanced UML Profiling</td>
</tr>
<tr>
<td>Line width</td>
<td>Advanced UML Profiling</td>
</tr>
<tr>
<td>Arrow types or end icons</td>
<td>Advanced UML Profiling</td>
</tr>
<tr>
<td>Customization</td>
<td>Customization engine</td>
</tr>
<tr>
<td>---------------------------------------</td>
<td>---------------------------------</td>
</tr>
<tr>
<td>Custom model initialization and constraints</td>
<td></td>
</tr>
<tr>
<td>Default values for primitive UML properties.</td>
<td>Custom diagram wizard</td>
</tr>
<tr>
<td>Default values for tags with primitive types</td>
<td>Advanced UML Profiling</td>
</tr>
<tr>
<td>Default super classes and interfaces</td>
<td>DSL engine</td>
</tr>
<tr>
<td>Numbering</td>
<td>DSL engine</td>
</tr>
<tr>
<td>Connection rules</td>
<td></td>
</tr>
<tr>
<td>Allowed metaclasses or stereotypes for end</td>
<td>DSL engine</td>
</tr>
<tr>
<td>Stereotypes that must be applied on relationship ends</td>
<td>DSL engine</td>
</tr>
<tr>
<td>Custom Specification dialogs</td>
<td></td>
</tr>
<tr>
<td>Grouping into nodes</td>
<td>DSL engine</td>
</tr>
<tr>
<td>Native UML properties on/off</td>
<td>DSL engine</td>
</tr>
<tr>
<td>Native UML property name changing</td>
<td>DSL engine</td>
</tr>
<tr>
<td>Custom abbreviation</td>
<td>DSL engine</td>
</tr>
<tr>
<td>Always visible custom properties</td>
<td>DSL engine</td>
</tr>
<tr>
<td>Custom Derived properties</td>
<td>DSL engine</td>
</tr>
</tbody>
</table>
2 WORKING WITH PROFILES

Profile in UML is a special kind of Package, used to contain collections of stereotypes, domain specific data types and libraries.

This chapter includes the following sections on working with profiles:

- "Creating Profiles" on page 8
- "Using Profile, Stereotypes and Tags" on page 14
- "Modifying Profile" on page 15

Related concepts
Domain Specific Language Customization

Creating Profiles

Profile can be created in any project, however often the same profiles are used among many projects. In order to reuse the same profile, it must be created as an independent file with shared data called module.

To create a Profile as a module

1. Create a new project.
2. In the Model Browser, right-click the root “Data” model and create a new Profile.
3. Create a new Profile diagram inside the Profile.
4. Create stereotypes in the diagram.
5. Use Associations, and Generalizations between them if needed.
6. In the Model Browser, from the created Profile shortcut menu, choose Modules > Share Packages.
7. Select the Profile as shared, enter its description, and click OK.
8. Save the project.
Now you may use this profile in other projects.

To use the created profile in other projects

- From the File menu, choose Use Module and select the module file. In such way, only shared packages of the module will be loaded into the project.

If you don’t want to see particular information, move elements from shared packages into the root model or other non-shared packages in your module.

Related external resources

---

1. Starting from version 18.1, referred as “used project” in MagicDraw UI. This user guide mentions the old keyword, which will be replaced in the documentation of the next MagicDraw version.
Specifying Invisible Stereotypes, Tags, and Constraints

Normally tags are used to hold some additional specific information that helps to interpret basic elements somehow different by some external tools, like transformations, code engineering, etc.

In these cases users don’t want to make impact on existing diagrams, only mark model.

Tag definitions are properties of stereotype in UML2, so UML doesn’t allow using tagged values without assigning of stereotype. That means your diagrams will be changed when some tags are added into model.

MagicDraw provides ability to make chosen stereotypes or tags not visible in diagrams.

To make stereotype invisible

The stereotype, which you want to make invisible, must be inherited from «InvisibleStereotype» (from MagicDraw profile).

1. Create a custom stereotype.
2. Create Generalization between your stereotype and InvisibleStereotype

You custom stereotype will be not displayed on symbols where it will be applied. Note that all tags also will be invisible on symbols.

It is not desirable to see an author tag in diagrams. To accomplish this, follow these steps:

1. Create stereotype «copyrighted» that extends Element metaclass.
2. Create property “author: String” for «copyrighted» stereotype.
3. Inherit this stereotype from «InvisibleStereotype» (create generalization in the model or diagram).

Now you may enter tagged value “author” for any element in your model. Stereotype «copyrighted» will be applied automatically but will be invisible in diagrams.

To create an invisible tag

• Apply «InvisibleStereotype» on your tag definition (property of stereotype).

All such hidden tags will be invisible on symbols of stereotyped elements.

Stereotype «identification» has two tags: ID and Description. The user would like to see Description value on elements in diagrams, but ID should be hidden as it is used in some external model transformation tools or similar activities.

Apply «InvisibleStereotype» on ID property of «identification» stereotype and it will be not visible on stereotyped elements on diagrams.

To make a constraint invisible

To make a constraint invisible, apply the «InvisibleStereotype» stereotype to the constraint. Invisible constraints may be used while creating validation constraints, in DSL customization, or in other modeling cases.
**Initializing opposite association end**

If two stereotypes are connected with Association and there are roles at both ends, role at one end will be initialized automatically when opposite role is specified.

When «dog» Rex is set as a pet for «human» Tom, Tom will be automatically set as “master” for Rex:

![Diagram showing the association between a human and a dog]

**Applying Different Icons for the Same Stereotype**

MagicDraw provides ability to create stereotype, which uses different icons, depending on stereotype “kind”. For example, PseudoState is one in UML, but it uses different notation, depending on PseudoStateKind. The similar situation could appear in your model.

To apply different icons on the same stereotype

1. Create Enumeration (in our example supercarKind). Enumeration will define all stereotype “kinds”.
2. Apply «iconHolder» stereotype on enumeration.
3. Create as many EnumerationLiterals as different kinds you would like to have (in our example it is “ferrari” and “audi”).
4. Apply stereotypes with different icons to every EnumerationLiteral (in our example «ferrari» and «audi»).
5. Create Stereotype (in our example supercar) with property (tag definition), which type is such enumeration (supercarKind). Default value could be selected.
6. Assign the created stereotype to an element and select one of enumeration value in tags. Stereotyped element will change its icon regarding enumeration value.

![Diagram showing application of different icons on the same stereotype]
Grouping Tags inside Stereotype

When some stereotype has many properties, it is very useful to group logically these properties into named groups. For this purpose every property could have tagged value with its group name.

To group tags

1. Open specification dialog of stereotype property.
2. Go to “Tags” panel.
3. Type group name as “groupName” value, «hasGroupName» stereotype will be assigned automatically.
4. Repeat these step for every stereotype property.
Case study

See an example of grouped properties from the UseCase Description Profile.

Creating Tags with Default Values

1. Create tag definition (Property).
2. Specify default value for this tag.
3. Specify multiplicity larger than 0 (1, 1..*, 2, etc.).
Such multiplicity means that this tag is mandatory.

When stereotype will be applied, mandatory tags will be automatically created and default values will be assigned.

Creating Structured Expression Tags

If you need to have a tag value, which can be defined by using the dialog for specifying criteria, you must create a structured expression tag definition.

To create a structured expression tag definition

1. Create a new stereotype.
2. Right-click the stereotype and select Create Element > Property to create a new tag definition.
3. Open the Specification window of the tag definition and select the StructuredExpression primitive type in the cell of the Type property value.

Once the tag definition is created, you can apply the stereotype with this tag definition on an element in your model. Then in the Specification window of the element, click Tags on the left, select the structured expression tag, and click the Create Value button to define the value.

The qualified name of the StructuredExpression primitive type is as follows: `UML Standard Profile::MagicDraw Profile::datatypes::StructuredExpression`

The structured expression tag can also be represented as a regular property in the Specification window and Properties panel of the element. For this, you have to create a customization data for the stereotype with this tag definition.

Related concepts

Creating Customization Data

Related external resources


Customizing Style of Stereotyped Elements

There are several ways to customize symbol style of stereotyped elements.
To extend UML symbol styles by creating new styles for stereotyped elements

1. In the **Project Options** dialog, the **Symbols Properties Styles** tree, right-click the **Stereotypes** branch. The list of stereotypes appears.
2. Select the check box near the stereotype and click the **Apply** button. Stereotype will be included into the **Stereotypes** property group. Set the stereotype style properties in the right pane of the **Project Options** dialog.

The best practice is to create styles for stereotypes in the Profile, loading it as a project.

When some project uses Profile, styles for stereotypes will also be loaded and used. These styles will be applied right after applying stereotype on some element and will be used in all diagrams.

Style for stereotype in custom diagram

Custom diagram wizard allows specifying styles for stereotyped elements.

Note, that this style will be used only in diagrams of this type.

For more details of how to work with Custom Diagram Wizard, see the next chapter.

### Using Profile, Stereotypes and Tags

The description of how to use profiles in your project can be found in *MagicDraw UserManual.pdf*.

**Related external resources**


### Applying Stereotypes

A stereotype defines how an existing metaclass may be extended, and enables the use of platform or domain specific terminology or notation in place of, or in addition to, the ones used for the extended metaclass.

When a stereotype is applied to a model element, the values of the properties may be referred to as tagged values.

**Related external resources**


### Specifying Tagged Values

Special Stereotypes and Tags tree is used to reflect all stereotypes and tags that are applied or could be applied to this element.

**Related external resources**

Modifying Profile

If you want to make any changes in the profile, you may modify it by opening profile as a regular project or modify it in the read-write mode (see below).

Opening Profile as Project

This is the most natural and safe way to modify your profiles.

It helps to see all not shared infrastructure including diagrams, tests, examples and etc.

To open a Profile as a project, do one of the following

• In the Browser, from the used Profile shortcut menu, choose Modules > Open Module as Project.
• On the File menu, click Open to open the Profile as a simple Project.

Modifying Profile in Read-Write Mode

Using profile in the read-write mode is the fastest way to add changes when you are not in teamwork and do not care of the possible editing conflicts.

It helps to perform the basic refactoring of your modules by moving the elements directly from one module to the other.

You may choose read-write usage mode when trying to use some profile as a module.

To use Profile in the read-write mode

1. Do one of the following:
   • If the profile is not in use yet, on the File menu, click Use Module. Select a Profile and click Next >.
   • If the profile is already in use, from its shortcut menu, choose Modules > Module Options.
2. In the Module Accessibility area, click Read-Write.
   By setting this option you will be able to change the profile directly in your project.

Related external resources

3 CUSTOMIZING DIAGRAMS

You can create your own diagram types for specific domain, platform, technology, or other purposes using the Custom Diagram Wizard. This is a powerful engine that allows for creating your own custom elements in diagram toolbar, custom symbol styles, and other customizations.

To open the Customize Diagrams dialog

- On the main menu, click Diagrams > Customize.

You can change properties of existing diagrams (Edit function) or create your own brand new diagram type (Create function). Diagram customization descriptors are saved into separate file for every diagram, so you are able to exchange these customizations with your partners or colleagues (use Import or Export function).

The Reset to default button in Customize Diagrams dialog restores default configuration for diagrams bundled with MagicDraw installation (it does not work with user defined diagrams).

To open the Customize Diagram Wizard

Do either:

- In the Customize Diagrams dialog, select a diagram type and click Edit.
- Click Create and then select one of the following commands:
  - Diagram Type, if you need to create a new diagram type.
  - Dependency Matrix Type, if you need to create a new matrix type.

![Figure 3 -- Selecting to create new dependency matrix type](image)

Related procedures

- Creating New Diagram Type
- Creating New Dependency Matrix Type

Creating New Diagram Type

The Customize Diagram Wizard contains the following steps for creating a new diagram type or modifying a chosen one:

- Step #1: Specify Diagram Type and Icon
Step #1: Specify Diagram Type and Icon

All Custom Diagrams are based on some standard UML diagrams (like Class, Use Case, Sequence, and so on).

Start to create your own diagram from these steps:

- Diagram type name (for example, Robustness Diagram)
- Base Diagram Type – standard UML diagram which is extended. All configurations, semantics, and other settings will be inherited from this diagram type.
- Abbreviation – short form of diagram name. It will be used in Diagram Frames header or Diagram shapes used in Content diagrams.
- Category - your specific category in the Diagmams menu or in the Create Diagram command list is created. In this category you can store all your customized diagrams. If you leave this field empty, the customized diagrams will be added into the Custom Diagrams category.
- Icons – several icons for your custom diagram representation in MagicDraw GUI.

![Figure 4 -- Customize Diagram Wizard - Step #1]
Step #2: Specify Modules

Custom Diagrams are oriented to some new specific domain, technology, or platform and are often based on some custom profile.

![Custom Diagram Wizard](image)

In the second Wizard step (Specify modules), select the required modules or profiles.

**IMPORTANT:** Do not remove from the list the UML Standard Profile which is by default selected. It must be used by any custom diagram type.

Custom diagram could use stereotyped elements, so profiles that define these stereotypes must be used by the custom diagram.

The selected modules/profiles will be loaded when the custom diagram will be created in user projects.

You can choose stereotype for the diagram itself by clicking the **Diagram Stereotype** button.

Step #3: Specify Toolbars

Every diagram differs by elements that are used in these diagrams. In the **Specify toolbars** step, you can group standard or custom elements.

You can:

- Create your own custom diagram toolbar.
- Create your own toolbar, name it, and select an icon.
- Choose standard toolbars that will be visible in your diagram.
- Select existing toolbars inherited from base diagram type.
- Order toolbars using “Up” and “Down” buttons.
- Select which toolbars will be expanded or collapsed by default (use the **Open** check box).
In this step you can manipulate with toolbars, in the next step you will be able to customize buttons inside selected toolbars.
Step #4: Specify Toolbar Buttons

In the Specify toolbar buttons step, select standard or stereotyped elements for your own diagram toolbars.

Click Add and select standard UML elements or click Add and then New Button to create your own buttons for your own customized or stereotyped elements creation.

Customize the following properties when creating a new button:

- Model element type.
- Custom icon for button.
- Shortcut that activates this button.
- Description that will be visible in the tooltip.
- Stereotype(s) that will be applied for created element.
- Symbol style. The selected symbol properties will be used only when element will be created using this button.
- Default values for some primitive model element properties (like isAbstract = true or similar).
CUSTOMIZING DIAGRAMS
Creating New Diagram Type

Step #5: Specify Symbol Properties

In the Specify symbols properties step, you can customize style for any element that appears in your custom diagram (e.g. class dropped in your diagram should be suppressed and red).

You can change the appearance of standard symbols, symbols of stereotyped elements, and custom diagram itself.

Customized element style will be used only in the appropriated custom diagram type.

---

Figure 8 -- Edit Button dialog

- List as element. Select this check box to add this button as command in the Create Element command list. Then you are able to create as an ordinary element.
- Opposite. This check box exists for relationship buttons only. Select the check box to add the opposite relationship button.
Step #6: Specify Smart Manipulators

Smart manipulators are special small buttons that appear in the popup window when a shape or path is selected on a diagram.

![Figure 9 -- Example of smart manipulator](image)

In the **Specify smart manipulators** step, you can configure what kind of Relationships will be suggested when a custom shape or path is selected on a diagram.

There are three major steps:

1. Create a new configuration (or modify an existing one). Select the element you want to be customized. The smart manipulators configuration can be related to:
   - Element (displayed as [Element name])
   - Element + stereotype(s)
   - Symbol (displayed as {Symbol name})
   - Symbol + stereotype(s)
   - Stereotype(s) (displayed as «Stereotype name»)

2. Select suggested relationships for the selected configuration.
3. Select target elements for the selected relationship.

If few configurations could be applied for the same selected element on the diagram, only one configuration will be used according to such priority:

<table>
<thead>
<tr>
<th>#</th>
<th>Configuration</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Symbol + stereotype(s)</td>
<td>When stereotype(s) for symbol applied</td>
</tr>
<tr>
<td>2</td>
<td>Stereotype</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Symbol</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Element + stereotype(s)</td>
<td>When stereotype(s) for element applied</td>
</tr>
<tr>
<td>5</td>
<td>Element</td>
<td></td>
</tr>
</tbody>
</table>

**NOTE**

If there are created configurations for few stereotypes (for example, Stereotype1 and Stereotype2) in the same diagram, after both stereotypes will be applied for one Symbol on that diagram pane, first configuration (in this example, Stereotype1) will be used, if there are no symbol+ stereotype(s) configuration specified.

**Inheritance of configurations**

All diagrams have the base diagram from which they inherit configurations.

For example, there is such hierarchy: Any diagram > Static diagram > Class diagram > Generic DDL diagram.
If you add a new configuration to a Class diagram (for example, Symbol A + stereotype B), this configuration will be used in the Generic DDL diagram as the Class diagram is its base diagram. Any and Static diagrams will not have such configurations.

To change configuration from the base diagram type:

- From the **Configurations** list, select the configuration by the element type and choose the **Specify own configuration** button. The inherited configuration will be overwritten.

### Creating New Dependency Matrix Type

**Creation of the new Dependency Matrix type is available in Standard, Professional, Architect, and Enterprise editions.**

In case of creating a new dependency matrix type, the **Customize Diagram Wizard** contains the following steps:

- **Step #1: Specify Name for a New Matrix Type and Icons**
- **Step #2: Specify Modules**
- **Step #3: Specify Dependency Matrix Properties**

### Step #1: Specify Name for a New Matrix Type and Icons

Start creating a custom matrix type by defining the following properties, which store the basic information about the new matrix type:

- In the **Type** box, enter a new matrix type name (for example, “UseCase - Actors”).
- In the **Abbreviation** box, enter a short form of the dependency matrix type name. It will appear on the diagram overview shape of the custom matrix, when the **Show Abbreviated Type** property is set to **true**.
- In the **Category** box, enter a name for a new category wherein the new matrix type will be stored. The category will appear on the main menu under **Diagrams** and in the command list of **Create Diagram** on the Model Browser shortcut menu. You can leave the default value, if you need the new matrix type to be stored in the **Analysis Diagrams** category.
• In the **Icons** area, select icons for the new dependency matrix type representation in MagicDraw GUI

![Customize Dependency Matrix Wizard](image)

**Figure 10 -- Specifying basic information about new dependency matrix type**

**Step #2: Specify Modules**

Select required modules or profiles in this step of the wizard.

**IMPORTANT:** Do not remove from the list the UML Standard Profile which is by default selected. It must be used by any custom dependency matrix type.
A custom dependency matrix can be used for representing relationships among stereotyped elements, so profiles that define these stereotypes must be used by the custom dependency matrix type.

All the selected modules/profiles will be loaded in a project when a dependency matrix of the custom matrix type is created in the project.

To choose the stereotype for the custom dependency matrix type

- Click the **Diagram Stereotype** button.

**Step #3: Specify Dependency Matrix Properties**

In this step of the wizard, you can specify the appearance of the new dependency matrix type, for example, predefine the default column text direction, row and column scope, and so forth.

<table>
<thead>
<tr>
<th>Property name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Dependency Matrix</strong></td>
<td></td>
</tr>
<tr>
<td>Take Whole Model As Scope</td>
<td>Set to <em>true</em> to select the whole model as a scope for relationship analysis. The Data element will be by default selected in both the <strong>Row Scope</strong> and <strong>Column Scope</strong> boxes.</td>
</tr>
<tr>
<td>Direction</td>
<td>Select a direction of relationships for the relationship analysis and representation in the dependency matrix. Be aware that new relationships will be created with the selected direction.</td>
</tr>
<tr>
<td>Dependency Criteria</td>
<td>Specify what relationships between row and column elements you need to display in the dependency matrix cells.</td>
</tr>
<tr>
<td>Show Elements</td>
<td>Select to show only related (by a selected dependency criteria), only non-related, or all elements.</td>
</tr>
</tbody>
</table>
### Property name | Description
---|---
Show Inner Dependencies | Set to `true` to show number of relationships in every owning element cell.
Suppress Criteria Area | Set to `true` to suppress the Criteria area toolbar.
Read Only | Set to `true` to make the matrix read-only. You will not be able to create or delete relationships.
Description Area | Type the description for the dependency matrix. The text will be displayed in the description area of the matrix.
Hide Types | Set to `true` to hide the Row Element Type and Column Element Type boxes from Criteria area.
Hide Scope | Set to `true` to hide the Row Element Scope and Column Element Scope boxes from Criteria area.
Hide Dependency Criteria | Set to `true` to hide the Dependency Criteria and Direction boxes from Criteria area.
Column
Column Owner Display Mode | Select the Compact tree mode to display elements with their direct and common owners in the column header. The data will be represented as a tree.
| Select the Complete tree mode to display elements with all their owners in the column header. The data will be represented as a tree.
| Select the Hidden mode to display elements without any owners in the column header. The data will be represented as a list.
| Select the Full qualified name mode to display elements with their owners in the column header. The data will be represented as a list.
Column Text Direction | Specify the direction of the text in column header. Be aware that this property value can be applied only when the Column Owner Display Mode property value is Hidden or Full qualified name.
Column Element Type | Specify element types to show in the columns of the dependency matrix.
Column Property Filter | Select properties and their values to create more specific filter for column elements.
Column Header Height | Specify the height of column header in pixels.
Column Types Include Subtypes | Set to `true` to display subtypes of selected element types. For example, if a class is selected, then all its subtypes, such as component or custom subtypes like SysML block and requirement will be displayed.
Row
Row Owner Display Mode | Select the Compact tree mode to display elements with their direct and common owners in the row header. The data will be represented as a tree.
| Select the Complete tree mode to display elements with all their owners in the row header. The data will be represented as a tree.
| Select the Hidden mode to display elements without any owners in the row header. The data will be represented as a list.
| Select the Full qualified name mode to display elements with their owners in the row header. The data will be represented as a list.
Row Element Type | Specify element types to show in the rows of the dependency matrix.
Row Property Filter | Select properties and their values to create more specific filter for row elements.
Row Header Width | Specify the width of row header in pixels.
Row Types Include Subtypes | Set to `true` to display subtypes of selected element types. For example, if a class is selected, then all its subtypes, such as component or custom subtypes like SysML block and requirement will be displayed.
4 DOMAIN SPECIFIC LANGUAGE CUSTOMIZATION

Domain Specific Language Customization is available in Standard, Professional, Architect, and Enterprise editions.

Domain Specific Language (DSL) - creating customization

DSL (Domain Specific Language) customization is a model-driven approach, based on UML profiling. MagicDraw provides the ability to use DSL engine for adapting domain specific profiles to create your own custom specification windows, custom real-time semantic rules and others. In other words, you can create specialized domain specific tool and hide UML underneath.

The MagicDraw DSL customization engine is able to process user defined rules for DSL elements and reflect this in MagicDraw GUI and diagrams behavior.

The topics in this section show you how to create a new DSL element, how to customize its Specification window, shortcut menu, how to create customization rules for relationships drawing, how to customize possible owned elements, how to create numbering customization, how to extend metamodel with derived properties and others.

Related procedures

Creating Customization Data
Using Customization Data

Creating Customization Data

Topics in this section describe components of customization creation and show you how to create customization data.

• "Introduction" on page 27
• "Case Study: Creating Your First Customization" on page 29
• "Customization Element Properties" on page 32
• "Customization Element Attributes" on page 44
• "Rules of Stereotypes that cannot be Allowed to Apply" on page 47
• "Creating Inheritance of DSL Customization" on page 48

Introduction

In MagicDraw, customization creation begins of creating the following components:

• Profile diagram
• Stereotype
• Customization element.
Creating Customization Data

In the following figure, you can see an example on how customization data is passed to DSL elements through stereotypes.

![Diagram showing components of DSL customization creation](image)

**Figure 12 -- Components of DSL customization creation**

In the preceding figure, you can see the customization element, stereotype, class element, DSL element and relations between them. The **Company** stereotype element is set as customization target in the **Company** customization element. In addition, the **Company** stereotype is applied to the class element. The customization data from the **Company** customization element is passed to the class element. Thus, the class element becomes the DSL element. DSL element properties will appear in the Specification window and the **Properties** panel of the DSL element as regular properties.

**Related concepts**

- Rules of Stereotypes that cannot be Allowed to Apply
- Customization Element Properties
Case Study: Creating Your First Customization

We will demonstrate the recommended step-by-step instructions on how to customize an element - how to start to create customization, how to create customization data and how the customization data will be represented on the DSL element.

1. Create a new project and name it `Organization_Stereotypes.mdzip`.
2. Create a package and name it `stereotypes`.

   **TIP!**

   The purpose of the `stereotypes` package is to store all the stereotypes. Later we will share the package in order to use the stereotypes in the customization project.

3. Create a profile diagram and name it `Stereotypes`.

   **TIP!**

   The profile diagram pallet contains all the necessary for the customization creation buttons.

4. Create a stereotype, apply the `Class` metaclass, and then name the stereotype `Company`.

   • In the Figure 13 on page 28, you can see how customization rules will be passed to the DSL element by using the stereotype.
   • We selected the `Class` metaclass, because the «Company» stereotype will be applied to a class element. In other words, the type of the DSL element will be the class.
   • You can also assign an icon to the stereotype. The icon will be represented on the DSL element.

5. Share the `stereotypes` package. The `Organization_Stereotypes.mdzip` project becomes a module.

   **TIP!**

   Only the content of the shared `stereotypes` package will be visible in the project that will use the module.

6. Save the `Organization_Stereotypes.mdzip` project.
7. Create a new project and name it `Organization_Customizations.mdzip`.
8. Create a package and name it `customizations`.

   **TIP!**

   The purpose of the `customizations` package is to store all the customization data. Later we will share the package.

9. Create a profile diagram and name it `Customizations`.

   The `Customizations` diagram is intended for customization elements creation.
10. Create a customization element and name it Company.

On a diagram, you can see that the customization element is a class with the applied «Customization» stereotype:

11. Use the Organization_Stereotypes.mdzip module in the project.

12. Open the Company customization element Specification window, under the General category, click the Customization Target property and then specify the Company stereotype.

13. In the Company customization element Specification window, under the General category, set the Hide Metatype check box to true. If the value is true, the element acts like a new standard element type in MagicDraw.

14. In the Company customization element Specification window, under the Owned Elements category, click the Possible Owners property and then select the Package metaclass.

15. Share the customizations package. The Organization_Customizations.mdzip project with the shared customizations package becomes a module.

16. Save the Organization_Customizations.mdzip project.

17. Create a project and name it Organization.

18. Use the Organization_Customizations.mdzip module in the Organization project.

The Organization_Stereotypes.mdzip module will be used in the Organization project automatically, because the Organization_Customizations.mdzip module uses the Organization_Stereotypes.mdzip module.

19. In the Organization project, create a class element and name it NoMagic.

20. Apply the Company stereotype to the NoMagic class element.
21. Save and reopen the project.

In the preceding figures, you can see the Containment tree of the project with the used Organization_Stereotypes.mdzip and Organization_Customizations.mdzip modules.

In this example, the «Company» stereotype is applied to the NoMagic class element. You can see that the Company element is added to the package shortcut menu.

If you do not want the new DSL element to appear on the Create Element menu, you can hide the appropriate command when customizing your MagicDraw perspective.

Related concepts
- Introduction
- Rules of Stereotypes that cannot be Allowed to Apply
Customization Element Properties

The customization element properties are used for customizing:

- the DSL element Specification window
- the DSL element shortcut menu
- allowed to draw relationships
- allowed to drag-and-drop elements
- other

To specify a new property

1. In a profile diagram, create a customization element.
2. Open the **Customization** element Specification window.
3. In the right-side of the window, select a property and modify it depending on its type.

The properties in the **Customization** Specification window are listed in the following two locations:
1. General specification pane
2. **Tags** property group

You can specify properties either in one location or in other. Changes made in the general specification pane reflects in the **Tags** property group and vice versa.
In the following table, you can see the Customization element properties description and links to the related sections.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Customization” group</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Specifies the name of the Customization element.</td>
</tr>
<tr>
<td>To Do</td>
<td>Specifies the To Do text of the Customization element. You can use the To Do property for adding the notes about work you need to do. Then, you can quickly search for the elements that have the To Do properties defined. For more information, see the “Find TODO” section in MagicDraw UserManual.pdf.</td>
</tr>
<tr>
<td>“Connection Rules” group</td>
<td></td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Allowed Drag And Drops</td>
<td>Specifies elements that describes allowed drag-and-drop operations for customized elements.                                                                ----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>For more information, see &quot;Customizing Drag and Drop&quot; on page 60.</td>
</tr>
<tr>
<td>Allowed Relationships</td>
<td>Specifies types of relationships that are allowed to be connected to the DSL element.</td>
</tr>
<tr>
<td></td>
<td>For more detailed description, see &quot;Creating Custom Rules for Relationships&quot; on page 64.</td>
</tr>
<tr>
<td>Disallowed Relationships</td>
<td>Specifies types of relationships that are not allowed to be connected to the DSL element.</td>
</tr>
<tr>
<td></td>
<td>For more detailed description, see &quot;Creating Custom Rules for Relationships&quot; on page 64.</td>
</tr>
<tr>
<td>Types For Source</td>
<td>Specifies metaclasses or stereotypes that are allowed to be connected as source of the relationship. Types cannot conflict with UML permitted types for this relationship.</td>
</tr>
<tr>
<td></td>
<td><strong>NOTE:</strong> The Types For Source property is intended for relationships only. That is, the Customization Target property can be the stereotype that extends the metaclass of the relationship or the metaclass of the relationship.</td>
</tr>
<tr>
<td></td>
<td>For more detailed description, see &quot;Creating Custom Rules for Relationships&quot; on page 64.</td>
</tr>
<tr>
<td>Types For Target</td>
<td>Specifies metaclasses or stereotypes that are allowed to be connected as target of the relationship. Types cannot conflict with UML permitted types for this relationship.</td>
</tr>
<tr>
<td></td>
<td><strong>NOTE:</strong> The Types For Source property is intended for relationships only. That is, the Customization Target property can be the stereotype that extends the metaclass of the relationship or the metaclass of the relationship.</td>
</tr>
<tr>
<td></td>
<td>For more detailed description, see &quot;Creating Custom Rules for Relationships&quot; on page 64.</td>
</tr>
<tr>
<td>“Content” group</td>
<td></td>
</tr>
<tr>
<td>Additional Content</td>
<td>Stores a rule (in a form of derived property expression) for gathering the additional content of the DSL element.</td>
</tr>
<tr>
<td>Sub Element Contents Included</td>
<td>Specifies whether the content of the DSL element includes only directly related elements or both directly and indirectly related ones.</td>
</tr>
<tr>
<td>“General” group</td>
<td></td>
</tr>
<tr>
<td>Customization Target</td>
<td>Specifies a stereotype or a metaclass which you are going to customize.</td>
</tr>
<tr>
<td></td>
<td>See a case study, in &quot;Case Study: Creating Your First Customization&quot; on page 29.</td>
</tr>
<tr>
<td>Do Not Suggest As Type</td>
<td>Excludes customized elements from being suggested as a type in the list dialogs.</td>
</tr>
<tr>
<td>Hide Metatype</td>
<td>Indicates whether the DSL element will be recognized as a new type.</td>
</tr>
<tr>
<td>Preferred Metatype</td>
<td>Specifies a preferred metatype, if there is more than one Customization Target defined in the Customization Specification window.</td>
</tr>
<tr>
<td></td>
<td>For more information, see &quot;Creating Property Groups and Subgroups&quot; on page 75.</td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Quick Applying For</td>
<td>Specifies an element, to which the created customization can be applied directly from the element shortcut menu. For more information, see &quot;Quick stereotype applying&quot; on page 59.</td>
</tr>
<tr>
<td><strong>Customization Target</strong></td>
<td>Specifies if the name auto completion list is displayed when entering name for the DSL element on the diagram. If the Do Not Suggest Name Auto Completion property value is true, then the auto completion drop-down list is not displayed.</td>
</tr>
<tr>
<td><strong>“Model Initialization” group</strong></td>
<td>Specifies stereotypes that can be automatically applied on elements (source or target) after connecting those elements with this customized relationships. The condition is that customization target would be a stereotype for relationship. For more information, see &quot;Initializing Custom Model&quot; on page 65.</td>
</tr>
<tr>
<td>Apply To Source</td>
<td>Specifies stereotypes that will be applied on the source element of the relationship after connection. NOTE: The customization target should be a stereotype of relationship. For more information, see &quot;Initializing Custom Model&quot; on page 65.</td>
</tr>
<tr>
<td>Apply To Target</td>
<td>Specifies stereotypes that will be applied on the target element of the relationship after connection. NOTE: The customization target should be a stereotype of relationship. For more information, see &quot;Initializing Custom Model&quot; on page 65.</td>
</tr>
<tr>
<td>Super Types</td>
<td>Specifies the element that will be the super type of the DSL element. The generalization relationship will be created from the DSL element to the specified super type. For more information, see &quot;Required Generalization or Interface Realization&quot; on page 66.</td>
</tr>
<tr>
<td><strong>“Naming” group</strong></td>
<td></td>
</tr>
<tr>
<td>Category</td>
<td>Creates a category to group DSL elements in various shortcut menus, if the Hide Metatype property is true. For more information, see &quot;Creating DSL Element from Customized Category in Shortcut Menu&quot; on page 67.</td>
</tr>
<tr>
<td>Keyword</td>
<td>Defines a keyword to be displayed instead of the applied stereotype name.</td>
</tr>
<tr>
<td>Representation Text</td>
<td>Defines the text that will be used in status bars, dialogs, logs, and others. See an example of the Representation Text property usage in a drag-and-drop operation customization, &quot;Properties used in drag-and-drop customization&quot; on page 61.</td>
</tr>
<tr>
<td><strong>Do Not Suggest Name Auto Completion</strong></td>
<td>Specifies a short name for the DSL element. When creating the DSL element, the short name will be added automatically.</td>
</tr>
<tr>
<td><strong>“Owned Elements” group</strong></td>
<td></td>
</tr>
<tr>
<td>Hidden Owned Diagrams</td>
<td>Specifies the diagram types that will hidden for the DSL element. For example, if the property value is Any Diagram, then all diagrams will be hidden and not available from the DSL element shortcut menu, under the Create Diagram command in the Model Browser. For more information, see &quot;Customizing Possible Owned Elements&quot; on page 68.</td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Hidden Owned Types</td>
<td>Specifies the metaclass or the stereotype that will be hidden for the DSL element. For example, if the property value is the Element metaclass, then all standard UML elements will be hidden in the DSL element shortcut menu, under the Create Element command in the Model Browser. For more information, see &quot;Customizing Possible Owned Elements&quot; on page 68.</td>
</tr>
<tr>
<td>Possible Owners</td>
<td>Specifies the metaclass or stereotype that can own the DSL element or in other words, specifies types of elements that can be owners of the DSL element. For example, let’s say you would like to have a possibility to create a DSL element inside standard packages in the Model Browser, then you have to specify the Package metaclass as the Possible Owners property value. For more information, see an example in &quot;Case Study: Creating Your First Customization&quot; on page 29 and for more detailed description, see &quot;Customizing Possible Owned Elements&quot; on page 68.</td>
</tr>
<tr>
<td>Suggested Owned Diagrams</td>
<td>Specifies types of diagrams that will appear in the DSL element shortcut menu when creating possible inner diagrams. For more information, see &quot;Customizing Possible Owned Elements&quot; on page 68.</td>
</tr>
<tr>
<td>Suggested Owned Types</td>
<td>Specifies types of elements that can be owned by the DSL element. For example, if the property value is Section, then the Section elements can be owned by the customized element. For more information, see &quot;Customizing Possible Owned Elements&quot; on page 68.</td>
</tr>
<tr>
<td>&quot;Properties&quot; group</td>
<td></td>
</tr>
<tr>
<td>Check Spelling</td>
<td>Specifies the properties for which the spell checker is enabled.</td>
</tr>
<tr>
<td>Help ID</td>
<td>Defines a string value referring to an element-specific help topic, which can be opened for the DSL element. For more information, see &quot;Creating Element-Specific Help Topics&quot; on page 70.</td>
</tr>
<tr>
<td>In Shortcut Menu</td>
<td>Specifies properties that will be displayed in the DSL element shortcut menu. For more information, see &quot;Quick property editor&quot; on page 59.</td>
</tr>
<tr>
<td>Multi Line Text Properties</td>
<td>Specifies properties for which multi line text editor will be used. Note only properties of string type is listed in editor of this property. For more information about creating properties in the DSL element Specification window, see &quot;Customizing Specification Window&quot; on page 49. For more information about editing property values in the Specification window, see the “Editing Property Values” section in MagicDraw UserManual.pdf.</td>
</tr>
<tr>
<td>Show Properties When Not Applied</td>
<td>Customizes the stereotype properties to be visible in the element Specification window even if the stereotype is not yet applied on the element. For more information, see &quot;Always visible properties&quot; on page 50.</td>
</tr>
</tbody>
</table>
### Customization Target

Specifies a stereotype or a metaclass which you are going to customize. The **Customization Target** property is the mandatory property in customization creation.

**To define a stereotype as customization target**

1. In the **Customization** Specification window, click the **Customization Target** property.
2. Click the ![button](image) button. The element Selection dialog appears.
3. Select the **Company** stereotype.
4. Click **OK** to close the dialog.
5. Click **Close**.
6. Apply the **Company** stereotype to the **NoMagic** class.
7. Reopen the project.

**To define a metaclass as customization target**

1. In the **Customization** Specification window, click the **Customization Target** property.
2. Click the ![button](image) button. The element Selection dialog appears.
3. Click the ![button](image) button. All the metaclasses will be included in the list.
4. Select the **Class** metaclass.

---

**Property** | **Description**
--- | ---
Show Properties When Not Applied Limited By Element Type | Customize the stereotype properties to be visible in the DSL element Specification window dependent on another stereotype or metaclass. For more information, see "Always visible properties" on page 50.

Show Properties When Not Applied Limited By Profile Application | Customize the stereotype properties to be visible in the DSL element Specification window dependent on the corresponding profile application. For more information, see "Always visible properties" on page 50.


Used UML Properties | Allows to select UML element properties that will be visible in DSL element Specification window. For more information, see "Using standard UML properties" on page 49.

**“Symbol” group**

Default Shape Size | Defines default size of the shape when the symbol is created on a diagram. For more information, see "Setting default symbol size" on page 73.

Symbol Standard Expert Configuration | Specifies properties visibility mode in DSL element **Symbol Properties** dialog, in the symbol shortcut menu, and in menus that opens after clicking the ⬇️ **Compartments** or ⚪ **Create Element** smart manipulators.

**Related external resources**


"Editing Property Values" in [MagicDraw UserManual.pdf](file)
5. Click **OK** to close the dialog.
6. Click **Close**.
7. Reopen the project.

Related properties

- **Hide Metatype**

Related procedures

- **Creating Customization Data**
- **Customization Element Properties**
- **Case Study: Creating Your First Customization**

**Do Not Suggest As Type**

Excludes elements from being suggested as a type in the list dialogs.

To exclude a DSL element from the list dialogs

1. In the **Customization** Specification window, click the **Customization Target** property.
2. Click the **+** button. The element Selection dialog appears.
3. Select a stereotype or a metaclass.
4. Click **OK** to close the dialog.
5. Set the **Hide Metatype** property to **true**.
6. Set the **Do Not Suggest As Type** property to **true**.
7. Click **Close**.
8. Reopen the project. Elements or metaclasses, which were selected in the 3st step of this procedure, are not listed in the list dialogs. For example, in the **Property** Specification window, click the **Type** property and then the **+** button. In the element Selection dialog, the elements or metaclasses, which were selected as customization targets, will not be displayed.

Related properties

- **Customization Target**
- **Hide Metatype**

Related concepts

- **Customization Element Properties**

Related procedures

- **Creating Customization Data**

**Hide Metatype**

Indicates whether the customized stereotype is recognized as a new type. If the **Hide Metatype** property value is **false**, the DSL element is *not* considered as a new type element - the DSL element will not be displayed in the shortcut menus (see the **Possible Owners** property), in the various lists and others.

To define the hide metatype property

1. In the **Customization** Specification window, click the **Customization Target** property and then select a customization target.
2. Set the **Hide Metatype** property to **true**.
3. Reopen the project.
Do Not Suggest Name Auto Completion

The property is useful when you do not want to display the name auto completion list for DSL elements that can typically be displayed only in one diagram or names of these DSL elements are full sentences or fragments of sentences.

To not display the name auto completion list

1. In the Customization Specification window, click the Customization Target property and then select a customization target.
2. Set the Do Not Suggest Name Auto Completion property to true.
3. Reopen the project.

Abbreviation

By using the Abbreviation property you can create a DSL element with already specified short name - abbreviation. In other words, you can enter the default name while specifying the customization element.

To define an abbreviation

1. In the Customization Specification window, click the Customization Target property and then select a stereotype or metaclass for which abbreviation will be used.
2. Click the Abbreviation property and then type the text.
3. Reopen the project.

Case Study

We will demonstrate how to add an abbreviation for the class attribute.

1. Create a profile diagram.
2. Create a customization element named PropertyAbbreviation.
3. In the PropertyAbbreviation customization Specification window, click the Customization Target property and then select the Property metaclass.

4. Under the Naming category, click the Abbreviation property and type a short name for the property, for example, prop.

5. Reopen the project.

6. Create a class in the class diagram.

7. Click the class shape on the diagram and then click the Insert New Attribute button.

8. The property with the predefined name (or abbreviation) is created.

Related properties
- Customization Target

Related concepts
- Customization Element Properties

Related procedures
- Creating Customization Data

Additional Content

Use the Additional Content property to specify that a desired element type can have a custom content in addition to the standard one that is defined by the UML ownership. The custom content of an element is a collection of elements that are related to this element according to some user-defined rule, for example, by some type of relationship.

To customize the content of a DSL element

1. Create a customization element for the DSL element or select an existing one.
2. Create a derived property for this customization element.
3. Specify a rule for gathering the additional content of the DSL element by creating an expression for the derived property.
4. Define the derived property as a value of the Additional Content property. Find the property in the general property group of the Specification window of the customization element, under the Content category.

Thereupon the custom content of the element will be visible in the following places:

- Containment tree
- Find dialog
- element Selection dialog
- Rows and columns of a dependency matrix

Case Study

Let’s say we need UML Packages to display not only their owned elements, but also the elements that are related to the Packages by the Element Import relationship. In order to obtain this, we have to specify the rule for gathering the custom content of the Package. Then we will analyze the contents of the customized Package in the Containment tree.
To specify the rule for gathering the custom content of the Package

1. Create a profile diagram and add to it a customization element named `PackageCustomization`.
2. Define the Package metaclass as the customization target of the customization element.
3. Create a derived property specification for the customization element, name it `importedElements`, and define the simple expression as is shown in the following figure.

![Simple Navigation](image)

<table>
<thead>
<tr>
<th>Relation Criterion</th>
<th>Is Applied</th>
<th>Direction</th>
<th>Include Subtypes</th>
<th>Filter by P</th>
</tr>
</thead>
<tbody>
<tr>
<td>Element Import</td>
<td>true</td>
<td>Source To Target</td>
<td>true</td>
<td></td>
</tr>
</tbody>
</table>

4. In the general property group of the Specification window of the customization element, under the **Content** category, find the **Additional Content** property and click the cell of the property value.
5. Define the `importedElements` derived property as a value of the **Additional Content** property.

- The **Additional Content** property can store a single value which can be only a derived property specification.
- You can select the derived property specification from either the same customization element or some other one with more abstract customization target. *In this case, for example, this could also be the customization element with the Element metaclass as a customization target.*

When you are done, the shape of the `PackageCustomization` element should look like the one that is depicted in the following figure.

![Additional Content](image)

6. Reopen the project.

To see how the **Additional Content** property affects the contents of the Package

1. Create a new diagram and draw in it a package named `P`.
2. Draw two classes named `A` and `B` in the package.
3. Outside the package, draw a class named C and connect it to the package with the Element Import relationship.

4. Now take a look at the Containment tree. The package P contains both owned classes A and B as well as the class C, which is not owned, but related to the package P by the Import Element relationship.

**Related properties**

- Customization Target

**Related concepts**

- Customization Element Properties

**Related procedures**

- Case Study: Creating Your First Customization
- Specifying derived properties
- Defining expressions

**Sub Element Contents Included**

Use the Sub Element Contents Included property to specify whether the content of a desired element type includes only directly related elements or both directly and indirectly related ones.

The property value can be either:

- **None**, which means that the content includes only directly related elements.
• *All*, which means that the content includes both directly and indirectly related elements.

![Diagram](image)

**Related concepts**
- Customization Element Properties

**Standard Expert Configuration**

In the DSL element Specification window, properties can be hidden or displayed by using *Standard, Expert* and *All* modes.

To customize Standard Expert mode

1. Create a customization element for the DSL element or select an existing one.
2. Click the **Standard Expert Configuration** property and then click the ![button](image)
3. In the **Customize Properties** dialog, click to select the mode in which the property will be displayed in the DSL element Specification window.
4. Reopen the project.

- **The Customize Properties** dialog contains the list of used UML properties and custom properties. Thus, you can select which UML properties and DSL specific properties will be displayed in the DSL element Specification window.
- You can customize property groups also. In the **Customize Properties** dialog, property groups are displayed at the end of the list.

**Related procedures**
- Creating Customization Data
- Customizing Specification Window
- Using Customization Data

**Related external resources**

**Symbol Standard Expert Configuration**

By using the **Symbol Standard Expert Configuration** property, you can specify properties visibility mode in:

- in the DSL element **Symbol Properties** dialog,
- in the symbol shortcut menu,
- in menus that opens after clicking the ![ compartments or ![ create element smart manipulators.

To customize Standard Expert mode

1. Create a customization element for the DSL element or select an existing one.
2. Click the **Symbol Standard Expert Configuration** property and then click the ![button](image).
3. In the **Customize Properties** dialog, click to select the mode in which the property will be displayed in the DSL element Symbol Properties dialog and in other places.

4. Reopen the project.

**Related procedures**

- Creating Customization Data
- Using Customization Data

**Related external resources**


### Customization Element Attributes

You can create an attribute in the customization element, then apply the particular stereotype and in this way to create customization rules.

To create an attribute in the customization element

1. In a profile diagram, create a customization element.
2. Select the customization element on the diagram.
3. Click the button. The menu with attributes appears.
4. Select the attribute.

In particular, attributes are properties. In this user guide we will use the attributes term, because the properties term is already used in the following context: "Customization Element Properties" on page 32.

In the following table, you can see the customization element attributes description and links to the related sections.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Applied Stereotype</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Meta Property</td>
<td>«metaProperty»</td>
<td>Used to change name of the default meta property.</td>
</tr>
<tr>
<td>Property Group</td>
<td>«propertyGroup»</td>
<td>Creates your own groups and subgroups to group the properties either of standard UML metaclasses or stereotypes.</td>
</tr>
<tr>
<td>Auto Numbering</td>
<td>«AutoNumber»</td>
<td>Creates property used to specify numbering rules for DSL element.</td>
</tr>
<tr>
<td>Derived Property Specification</td>
<td>«derivedPropertySpecification»</td>
<td>Creates a derived property that is the property, whose values are calculated automatically from the other properties' values</td>
</tr>
</tbody>
</table>

**Related procedures**

- Creating Customization Data

---

**Meta Property**

Meta Property attribute is used to change name of the default meta property

To create a meta property attribute

1. In a profile diagram, create a customization element.
2. Select the customization element on the diagram.
3. Click the button.
4. Click the Property command. The attribute is created.
5. Apply the «metaProperty» stereotype to the attribute.

![Customization - Company](image)

**Figure 18 -- Example of Meta Property attribute Specification window**

**Related concepts**
- Customization Element Attributes

**Related procedures**
- Meta property substitution (changing name of UML property)
- Creating Customization Data
Property Group

You can create your own groups and subgroups to group the properties either of standard UML metaclasses or stereotypes.

![Figure 19 -- Example of Property Group attribute Specification window](image)

**Related concepts**
- Customization Element Attributes

**Related procedures**
- Creating Customization Data
- Creating Property Groups and Subgroups

Auto Numbering

You can specify what numbering scheme should be used to number DSL element and where number should be stored in DSL element. For this, a numbering attribute must be created in the customization element.

**Related concepts**
- Customization Element Attributes

**Related procedures**
- Creating Customization Data
Derived Property Specification

You can create a derived property that is the property, whose values are calculated automatically from the other properties’ values.

![Derived Property Specification Window](image)

Figure 20 -- Example of Derived Property Specification window

Related concepts
- Customization Element Attributes

Related procedures
- Creating Customization Data
- Extending Metamodel with Derived Properties
- Specifying derived properties

Rules of Stereotypes that cannot be Allowed to Apply

When applying a stereotype to a DSL element, the list of the available to apply stereotypes is filtered according to the following rules:

- There is not allowed to apply two DSL stereotypes (with Hide Metatype=true) to the same element. For example, it should not show «Requirement» stereotype in the Block shortcut menu, if «Block» is already applied).
If the stereotype should be changed, the current stereotype should be unapplied first. (e.g. when changing from «Requirement» to «businessRequirement»).

You can quickly change the stereotype, by converting one element type to other (in the DSL element shortcut menu, click **Refactor** and then **Convert**).

- DSL ownership rules are checked and there is not allowed to apply a stereotype, which is not allowed in the parent element of DSL element.
- DSL Relationship rules are checked and there is not allowed to apply some stereotypes that source or target element is violating DSL rules for this DSL relationship (the same checking is used while drawing).

**Related concepts**
- Introduction
- Customization Element Properties
- Customization Element Attributes

**Related procedures**
- Creating Customization Data
- Case Study: Creating Your First Customization

**Related external resources**

**Creating Inheritance of DSL Customization**

DSL customization is inherited, when stereotype is inherited, so you don’t need to modify existing DSL customization when extending the (standard) profile.

For example, if you want to create your own subtype of SysML «Requirement», for example, named *Performance Requirement*, it is enough to define a new stereotype «Performance Requirement» and add a generalization between the SysML «Requirement» and «Performance Requirement».

All predefined symbol properties (style) and semantics rules of SysML «Requirement» will be reused on elements stereotyped by «Performance Requirement».

The subtype could have their own customization with some basic properties, for example, *human name*, or other. Customizations will be merged, and only additions should be defined.

Try to avoid the conflicting customizations, as result can be unpredictable.

Generalization between the two customization elements is also possible. In this case, specific (subclass) customization inherits customization rules, if these rules are not redefined in subclass (the particular properties are empty). If both customizations have the same property filled, the rules are not merged, the one from subclass will be used.

**Related procedures**
- Creating Customization Data
- Using Customization Data
Using Customization Data

Customizing Specification Window

The main purpose of DSL is to hide UML, so that DSL elements would look in MagicDraw as standard or as a new type elements.

DSL element properties will appear in the DSL element Specification window and the Properties panel as regular properties.

There are several customizations and grouping cases for properties:

- "Using standard UML properties" on page 49
- "Always visible properties" on page 50
- "Meta property substitution (changing name of UML property)" on page 54

Using standard UML properties

DSL element specification may contain custom properties defined in stereotype.

All DSL element properties will be used by default, but you may select which UML properties will be used in the DSL element Specification window.

To use standard UML properties

1. Open the customization element Specification window.
2. Click the Used UML Properties property and then click the button.
3. In the Items Filter dialog, select the standard UML properties that you want to be displayed in the DSL element Specification window.
4. Click OK when you are done.

**Always visible properties**

Generally stereotype properties become visible in an element's Specification window only after the stereotype is applied on the element. However, you may customize stereotype properties to be visible in the element Specification window even if the stereotype is not yet applied on the element.

This feature allows specifying some domain-specific properties for standard UML elements. It has already been used in MagicDraw to make stereotype properties, such as **Type Modifier**, **Active Hyperlink**, and **To Do** visible even if appropriate stereotypes are not yet applied on an element.

---

*Figure 21 -- Always visible properties in UML property Specification window*
To make stereotype properties visible, when the stereotype is not yet applied on an element

1. Open the customization element Specification window.
2. Click to change the Show Properties When Not Applied property to true. All properties of the customized stereotype will become visible in the DSL element Specification window even if the stereotype is not yet applied on the element.

Case study

Use the C# Profile (C#.Profile.mdzip) in your project and you will see an additional tab, the C# Language Properties one, appeared in a class Specification window. C# properties appear in the Specification window of a class even though no profile-specific stereotype is yet applied on the class.

![Specification of C# Language Properties](image)

*Figure 22 -- C# properties in class Specification window*
Specify a desired C# property value in the class Specification window and you will see the «C#LanguageProperty» stereotype automatically applied on the class.

Advanced cases of properties visibility

There are two cases of displaying not yet applied stereotype properties. You can customize the stereotype properties visibility to depend on:

- **Profile application.** Properties will be visible only when the owning package of an element has a corresponding profile application.

- **Another stereotype and/or metaclass.** Properties will be visible only when:
  - Some other stereotype is already applied on an element.
  - Element’s metaclass is one of the metaclasses extended by the stereotype.

To make the visibility of stereotype properties dependent on a corresponding profile application

1. Open the customization element Specification window.
2. Click to change the Show Properties When Not Applied property to true.
3. Click to change the Show Properties When Not Applied Limited By Profile Application property to true.
To make the visibility stereotype properties dependent on another stereotype or a metaclass

1. Open the customization element Specification window.
2. Click to change the **Show Properties When Not Applied** property value to **true**.
3. Click the **Show Properties When Not Applied Limited By Element Type** property and then select a stereotype or a metaclass.

You can select more than one value.

**Case study**

Let’s analyze the customization element of the «OraUser» stereotype in the *Oracle Customization* profile (*Oracle_Profile.mdzip*). This profile belongs to the *Cameo Data Modeler* plugin and can be used for database modeling.

If the «User» stereotype is selected as the **Show Properties When Not Applied Limited By Element Type** property value. This means that properties of the «OraUser» stereotype will be visible only for elements those have the «User» stereotype applied.

The **Show Properties When Not Applied Limited By Profile Application** property value is set to **true**. This means that properties of the «OraUser» stereotype will be visible only for elements, whose owning package has the *Oracle Customization* profile application. For example, for elements inside the Oracle schema.

In conclusion Oracle-specific properties will be visible only for users inside the Oracle schema.

---

*Figure 24 -- Specification window of user that is NOT INSIDE Oracle schema*
Meta property substitution (changing name of UML property)

Sometimes standard UML property needs to be customized, to represent more restricted types, or under different name (renamed).

Property name substitution

Case study

In the DSL element Specification window, the Name property will be renamed to the blockName.

1. Create a name attribute in the customization element.
2. Apply the «metaProperty» stereotype on the attribute.
3. In the «metaProperty» attribute Specification window, click the New Name property, and then type a new name blockName.
4. To apply the changes, reopen the project.

Figure 26 -- Example of property name substitution

Property description substitution

In the element Specification window, the description of the selected property is displayed below the properties list (see the preceding figure). You can customize, that is, change the property description. For example, after you have changed the property name (see "Property name substitution" on page 54), the property description...
could be changed too.

Figure 27 -- Example of property description area in element Specification window

Case study

To change the Name property description:

1. Create a new description attribute in the customization element.
2. Apply the «metaProperty» stereotype on the attribute.
3. In the customization element Specification window, under the description attribute, Documentation/Hyperlinks property group, type a new description (see Figure 27 on page 56).
4. To apply the changes, save and then reopen the project. The description of the Name property changes.
Figure 28 -- Example of Customization element, "description" attribute documentation
Type substitution (restriction)

To change the type

1. Create the type attribute for the customization element.
2. Apply the «metaProperty» stereotype.
3. Click the New Types property in the type attribute Specification window.
4. Specify the «Block» stereotype.
5. To apply the changes, save and then reopen the project.

Suggesting predefined values

There is an ability to create such properties, where list of predefined values will be suggested for the end user (as now with "type modifier").

1. Create an attribute for the customization element.
2. Name attribute as property which should be customized.
3. Apply the «metaProperty» stereotype.
4. Enter suggested values into the Suggested Values property.

If the new property description is not specified, by default the old description is used.
Customizing Element Shortcut Menu

You can show or hide the existing menus in shortcut menus by using the Perspectives functionality.

You can create shortcut menu items by using customization model.

Quick property editor

You can add the mostly used properties to the DSL element shortcut menu.

To specify properties that will be added in shortcut menu, use the following property:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>In Shortcut Menu</td>
<td>Specifies properties that will be displayed in the DSL element shortcut menu.</td>
</tr>
</tbody>
</table>

Only these types of properties can be edited from the shortcut menu:

- Boolean, boolean – you can click to select or clear a check box.
- Enumeration – you can select one of the listed enumeration literals.
- Reference to one element (subclass of Classifier) – you can use element list to select a type.

Properties of all other types will be ignored.

Quick stereotype applying

Stereotypes as «continuous», «buffer» and others act like a flag, so it would be nice to use quick check box to set these stereotypes from the element shortcut menu.

Use the following property for adding the stereotypes into the DSL element shortcut menu:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quick Applying For</td>
<td>Specifies a stereotype or a metaclass, to which the created customization can be applied directly from the element shortcut menu.</td>
</tr>
</tbody>
</table>

For example, if the Quick Applying For property value is the Class metaclass, the customization will be available for the class elements.
Customizing Drag and Drop

The drag-and-drop customization allows for creating relationships or assigning property values by dragging and dropping one element onto another right in the diagram pane. By using the DSL engine you may set up your own drag-and-drop specifications between elements.

Depending on what elements are dragged onto which, you can do the following:

- create relationships
- assign property values

There are also a number of predefined rules to create relationships or assign property values automatically.

You can customize the following drag-and-drop actions:

- drag-and-drop one element from the diagram pane onto the another element on the diagram pane
- drag-and-drop one element from the Model Browser onto the another element on the diagram pane
- drag-and-drop two or more elements from the Model Browser onto the element on the diagram pane

Related procedures

Creating Customization Data
Using Customization Data
Properties used in drag-and-drop customization

In the following table there are listed the properties that are used to customize the drag and drop rules.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Source Element</strong></td>
<td>Specify the element type that will be dragged to the target element. You can specify a class, metaclass, or stereotype. This is the mandatory property - if the Source Element property is undefined, then the drag and drop rule is inactive.</td>
</tr>
<tr>
<td><strong>Representation Text</strong></td>
<td>Specify the name of the drag and drop rule.</td>
</tr>
<tr>
<td><strong>Relation Action Result</strong></td>
<td>Specify a relationship that will be created between the source and target elements after drag-and-drop.</td>
</tr>
<tr>
<td><strong>Property Action Result</strong></td>
<td>Specify a property that will be assigned for the target element after drag-and-drop.</td>
</tr>
<tr>
<td><strong>Append Mode</strong></td>
<td>Change the append mode for the property assignment. The possible Append Mode property value is true or false. If true, then properties are assigned one by one. If false, then assigning the property the second time, the property is overwritten over the former property.</td>
</tr>
</tbody>
</table>

Case study #1: Adding a rule to create Generalization relationship on drag-and-drop

In this example you can see step-by-step instructions on how to customize a drag-and-drop rule, according to which the Generalization relationship will be created between classes after drag-and-drop.

1. Create a profile diagram.
2. Create a class, and name it Class onto Class.
3. For the Class onto Class class, assign the «DragAndDropSpecification» stereotype.

There is a new button in the Profile diagram pallet - Drag and Drop Specification. You can use this button to create a class element with the assigned «DragAndDropSpecification» stereotype.

4. Create a Customization element.
5. In the Customization Specification window, under the Connection Rules category, for the Allowed Drag and Drops property, assign the Class onto Class element.
6. In the **Customization** Specification window, under the **General** category, for the **Customization Target** property, assign the **Class** metaclass.

7. In the **Class onto Class** Specification window, assign the following property values:
   - For the **Source Element** property, assign the **Class** metaclass.
   - For the **Relation Action Result** property, assign the **Generalization** metaclass.
   - For the **Representation Text** property, assign the name of the rule, for example, *Create Generalization*.

8. Reopen the project. Drag the class to the other class, the Generalization relationship will be created between these classes.

**Case study #2: Adding a rule to assign a property type on drag-and-drop**

In this example you can see step-by-step instructions on how to customize a drag-and-drop rule, according to which a property type will be assigned after a class drag to the property.

1. Create a profile diagram.
2. Create a class, and name it *Classifier on Property*.
3. For the *Classifier on Property* class, assign the «DragAndDropSpecification» stereotype.
4. Create a customization element.
5. In the **Customization** Specification window, under the **Connection Rules** category, for the **Allowed Drag and Drops** property, assign the **Classifier on Property** element.
6. In the **Customization** Specification window, under the **General** category, for the **Customization Target** property, assign the **Property** metaclass.

![Customization Specification](image)

7. In the **Classifier on Property** Specification window, assign the following property values:
   - For the **Source Element** property, assign the **Classifier** metaclass.
   - For the **Property Action Result** property, assign **Type**.
   - For the **Representation Text** property, type the name of the rule, for example, **Assign Type**.

![Classifier on Property Specification](image)

8. Reopen the project. Now after drag a classifier to a property, the classifier will be assigned as a property type.

![Drag-and-drop](image)

**Related procedures**
- Creating Customization Data
- Creating Numbering Customizations
- Using Customization Data

**Selecting one of the few drag-and-drop rules**

There are cases, when more than one rule can be applied for one drag-and-drop action. In that case, you can select what should be created.

To select a rule

1. Do one of the following:
   - Drag a selected element to another, hold it for a few seconds till the tool-tip opens, and then release. A shortcut menu is displayed.
   - Right-click an element, drag it onto the another element, and then drop it. A shortcut menu is displayed. (Note that this procedure is valid on Windows OS only).
2. On the shortcut menu, click the rule you want to apply. The rule is applied.

1. Right-click

2. Drag-and-drop

3. Click on menu

<table>
<thead>
<tr>
<th>Customer</th>
<th>Company</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Create Generalization
Create new attribute

In MagicDraw there are a number of predefined drag-and-drop rules.

Related procedures
- Creating Customization Data
- Creating Numbering Customizations
- Using Customization Data

Related external resources

Creating Custom Rules for Relationships

DSL relationships may have special rules, describing what kind of elements can be connected.

For this purpose the following properties are used:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Types for Source</td>
<td>Specifies metaclasses or stereotypes that are allowed to be connected as source of the relationship. Types cannot conflict with UML permitted types for this relationship.</td>
</tr>
<tr>
<td></td>
<td>NOTE: The Types For Source property is intended for relationships only. That is, the Customization Target property can be relationship or metaclass of relationship.</td>
</tr>
<tr>
<td>Types for Target</td>
<td>Specifies metaclasses or stereotypes that are allowed to be connected as target of the relationship. Types cannot conflict with UML permitted types for this relationship.</td>
</tr>
<tr>
<td></td>
<td>NOTE: The Types For Source property is intended for relationships only. That is, the Customization Target property can be relationship or metaclass of relationship.</td>
</tr>
</tbody>
</table>

Subtypes of specified end type are allowed only if type is abstract (for example, the Classifier metaclass), otherwise only concrete types must be used.

EXAMPLE

- If the value of the Types For Source property is the Classifier metaclass, then the Class, Component and other classifiers will be allowed to connect as source of relationship.
- If the value of the Types For Source property is the Class metaclass, then the Class elements will be allowed to connect as source of relationship.
- If the value of the Types For Source property is the «Block» DSL element, then the «Block» elements will be allowed to connect as source of relationship.

In the following table, see properties that are used for customizing non-relationships:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Allowed Relationships</td>
<td>Specifies types of relationships that are allowed to be connected to the DSL element.</td>
</tr>
</tbody>
</table>
Defining Preferred Metatype

DSL customization engine allows for specifying a preferred metatype, if there is more than one metatype defined for the DSL element.

To define a preferred metatype for the DSL type

1. Select the appropriate customization element and open its Specification window.
2. Click the Preferred Metatype property and click the Create Value button.
3. In the element Selection dialog, select a preferred metatype and click OK. The selected metatype will be assigned as the Preferred Metatype property value (see the figure below).

```
```

Multiple selection is not allowed. Define a single value.

4. Click Close

```
```

After updating the customization element specification, either the DSL Customization profile must be reused in the model or the model must be reopen.

From now on all the DSL elements will be created with the preferred metatype, which has been specified in the customization.

Related procedures
Creating Customization Data
Creating Numbering Customizations
Using Customization Data

Initializing Custom Model

DSL customization provides behavior when some model element initialization could be done after applying stereotype on it.
Default values

Tags as any other properties can have default values. These values will be used when tag is created.

Tag definition must have multiplicity greater then 0, only in this case it will be created automatically when applying a stereotype.

Related external resources

Stereotypes on relationship ends

Sometimes DSL requires applying stereotypes on some elements after DSL relationship connection to those elements.

You can use the following properties for this purpose:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apply to Source</td>
<td>Specifies stereotypes that must be applied on the source element of this relationship after connection.</td>
</tr>
<tr>
<td>Apply to Target</td>
<td>Specifies stereotypes that must be applied on the target element of this relationship after connection.</td>
</tr>
</tbody>
</table>

“Source” and “Target” are respectively “source” and “target” of directed relationship.

The first connected element will act as “source” for non-directed relationship.

1. Create a stereotype «serve» and select the Dependency metaclass.
2. Create two stereotypes «master» and «servant» and select the Element metaclass.
3. Create a customization element
4. In the customization Specification window, specify the following property values:
   • for the Customization Target property, select the «serve» stereotype
   • for the Apply To Source property, select - the «master» stereotype
   • for the Apply To Target property, select the «servant» stereotype.
5. Reopen the project. Draw the «serve» dependency from one element to another, the «master» and «servant» stereotypes will be applied on these elements.

Required Generalization or Interface Realization

Sometimes DSL requires that elements should be subtypes of some general abstract class or interface. You can use the following property for this purpose:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Super Types</td>
<td>Specifies types that will be super types of the DSL element. Generalization or InterfaceRealization (If interface) will be created in the model after applying such customized stereotype.</td>
</tr>
</tbody>
</table>
See an example of customization, when every JAVA class will be a subclass of the Object class, or in other words, every Class marked with the «JAVAClass» stereotype will be inherited from Object.

- Create a customization element.
- In the customization Specification window, specify the following **Customization Target** property - the «JAVAClass» stereotype
- In the customization Specification window, specify the following **Super Types** property - the Object class.

**Related procedures**

- [Creating Customization Data](#)
- [Using Customization Data](#)

### Creating DSL Element from Customized Category in Shortcut Menu

You can group elements into the categories in the **Create Element** menu in the Model Browser or in various create menus in dialogs and specifications. The **Category** property of customization element is used for category creation.

**Case Study**

1. Assign the **Business Requirement** stereotype as **Customization Target** property in the customization element Specification window and specify the Package metaclass as **Possible Owners** property value.

   **TIP!** For more detailed description on how to assign the Possible Owners property value, see "Case Study: Creating Your First Customization" on page 29.

2. In the customization element Specification window, click the **Category** property and then type the name of the new category, for example, Requirements (see **Figure 31 on page 68**).
The Business Requirement element will be added to the package shortcut menu, under the Create Element command, in the Requirements category (see Figure 32 on page 68).

![Figure 31 -- Example of customization element](image)

![Figure 32 -- Example of new Requirements category](image)

**Related procedures**

- Creating Customization Data
- Using Customization Data

### Customizing Possible Owned Elements

You can define situations when elements of particular predefined stereotypes or metaclasses will be created only inside some other elements. For example:

- **Block** will contain **BlockProperty** property, instead of standard property.
- **Block** will own **InternalBlockDiagram**, instead of standard **CompositeStructureDiagram**.

All these cases can be customized by using properties listed in the customization element Specification window, under the Owned Elements category.

For more information, read:

- "Customizing owned types" on page 69
- "Customizing owned diagrams" on page 69
- "Customizing suggested relationships" on page 70
Customizing owned types

You can customize the list of owned types in the DSL element shortcut menu, under the Create Element command, in the Model Browser and in other lists. In the following table, see the properties used in customizing owned types:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Suggested Owned Types</td>
<td>Specifies the list of stereotypes and metaclasses that will be suggested when creating inner elements in the DSL element.</td>
</tr>
<tr>
<td>Hidden Owned Types</td>
<td>Specifies the list of stereotypes and metaclasses that will be not suggested as inner elements.</td>
</tr>
</tbody>
</table>

When some stereotyped elements will be suggested to create in standard UML element, for example, a package, it's not possible to customize standard (not DSL) element.

In this case customization must be done upside-down, by specifying possible owners. You can use the following property from the customization element Specification window: Related procedures

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Possible Owners</td>
<td>Specifies the list of stereotypes and metaclasses that will be suggest when creating owned elements.</td>
</tr>
</tbody>
</table>

Until MagicDraw version 17.0.3, the Suggested Owned Type and Hidden Owned Types properties were not functioning if customization target was metaclass. Now this problem is solved along with the implementation of the capability that metaclasses can be defined as values of Suggested Owed Types and Hidden Owned Types properties.

Customizing owned diagrams

You can customize the list of owned diagrams in the DSL element shortcut menu, under the Create Diagram command, in the Model Browser and in other lists. In the following table, see the properties used in customizing owned Diagrams:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Suggested Owned Diagrams</td>
<td>Specifies types of diagrams that will appear in the DSL element shortcut menu when creating possible inner diagrams.</td>
</tr>
<tr>
<td>Hidden Owned Diagrams</td>
<td>Specifies types of diagrams that can be owned by the DSL element. For example, if the property value is Section, then the Section elements can be owned by the customized element.</td>
</tr>
</tbody>
</table>

Until MagicDraw version 17.0.3, Suggested Diagram Types and Hidden Diagram Types properties were not functioning if customization target was metaclass. Now this problem is solved along with the implementation of the capability that metaclasses can be defined as values of Suggested Diagram Types and Hidden Diagram Types properties.

Related procedures

Creating Customization Data
Customizing Possible Owned Elements
Using Customization Data
Customizing suggested relationships

You can customize the list of relationships that are suggested to create directly in the model, from the Model Browser, starting from source or target element.

The list of the suggested relationships can be customized according to the following rules:

- If a DSL relationship has no connection rules for restricted end types and the **Hide Metatype** property value specified is **true**, then the DSL relationship will be added into all the lists where the extended UML type is suggested. For example, the «allocation» relationship in SysML model will be suggested everywhere Dependency is suggested.
- If a DSL relationship has connection rules, then it will be suggested to be created only from or to these restricted types.

Related procedures

Creating Customization Data
Customizing Possible Owned Elements
Using Customization Data

Creating Element-Specific Help Topics

For each DSL element, you can define a string value referring to an element-specific help topic, which can be opened for the DSL element in one of the following ways:

- By clicking the **Help** button in the element's Specification window.
- By pressing F1, when the element's symbol is selected.

To define a help topic ID for the DSL element

1. Select the appropriate customization element and open its Specification window.
2. In the general specification pane, select the **Help ID** property and type the help topic ID.
3. Click **Close** when you are done.

After updating the customization element specification, either the DSL Customization profile must be reused in the model or the model must be reopen.

In case an element represents two or more DSL elements, the first in order DSL element’s help topic ID will be taken.

In case the DSL type customization does not have the **Help ID** property defined, the help topic ID will be taken from his ancestor's (more general stereotype) customization the **Help ID** property. If there is no ancestor, the metaclass help will be opened.

Related procedures

Creating Customization Data
Using Customization Data

Related external resources

“Plug-in descriptor” in "MagicDraw OpenAPI UserGuide.pdf"
Creating DSL Element from Shortcut Menu and Other Locations

By using the **Possible Owners** property you can define the possible owner for the customized element and at the same time you may add the possibility to quickly create a customized element from the possible owner shortcut menu in the Model Browser and in other locations. For example, for a customized use case element, you can add a package as a possible owner.

By using the **Possible Owners** property, you can add a way to create a customized element quickly from the following locations:

- In the possible owner’s shortcut menu, under the **Create Element** command, in the Model Browser.
- In the possible owner’s Specification window, the **Inner Elements** property group.
- The **Select Element** dialog, when element creation mode is turned on.

**Case study**

See examples of the possibility to create the *Business Requirement* element, from the following locations:

1. In the package shortcut menu, in the Model Browser, under the **Create Element** command ([Figure 33 on page 72](#)).
2. In the possible owner’s Specification window, the **Inner Elements** property group (see [Figure 34 on page 72](#)).
3. In the element Selection dialog, when creating a new element (see [Figure 35 on page 73](#)).
Figure 33 -- Example of Possible Owners property implementation

Figure 34 -- Example of Possible Owners property implementation
Related procedures
Creating Customization Data
Using Customization Data

Customizing Symbols

Setting default symbol size

Default size can be defined in DSL customization, by using the following property from the Customization element Specification window:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default Shape Size</td>
<td>Defines default size of the shape when the symbol is created on a diagram.</td>
</tr>
<tr>
<td></td>
<td>- The first value is width (x) in pixels. Value= 0 is used if default MD width should remain, but height should be changed.</td>
</tr>
<tr>
<td></td>
<td>- The second value is height (y) in pixels.</td>
</tr>
</tbody>
</table>

Default shape size will be used everywhere when a new symbol is created - using a drag-and-drop, diagram toolbar button, diagram wizards, and other.
Setting custom path style

You can customize the custom path style and by changing:

- Line style (dashes, dots, solid line, etc).
- Arrow style at first end and arrow style at second end.
- Custom scalable icon for the first end and for the second style.

To create the custom path style

1. In a profile diagram pallet, click the Stereotype button and then click on the diagram pane.
2. In the Select Metaclass dialog, select the Relationships metaclass.
3. Apply the stereotype to a relationship.
4. In the stereotype Specification window, click the Icon property and then click the button.
5. Click Yes, to open the Path Icon Customization dialog.
6. Specify the path style.
7. Click OK, when you are done.
8. Apply the stereotype to a relationship.

See the following path icon usage rules:

- custom appearance will be used only after the stereotype is applied.
- end icons will by rotated according to the path position and direction.

Figure 36 -- Path Icon Customization dialog

You can change the line width and the rest of the path properties in the Symbol Properties dialog.
• end icon size will be scaled according to the font size (the same as for standard arrows)

Related external resources

Creating Property Groups and Subgroups

You can create your own groups and subgroups to display the properties either of standard UML metaclasses or stereotypes. You can also choose, where you want these property groups and subgroups to be visible. Property groups and subgroups can be visible in the following places:

• Element’s Specification window.

• Element’s Properties panel (at the bottom of the Model Browser).

• Element’s shortcut menu > Go To > submenu.

• Compartment Edit dialog.
• **Criterion Editor** dialog for editing the relation’s criteria in the Relation Map diagram.

To create a property group

1. In the Model Browser, right-click a Customization element and choose **Create Element** > **Property Group**.
2. Type the attribute’s name (it will be used to name the property group).
3. Open the attribute’s Specification window.
4. In the general specification pane, edit the property values (properties are described in the table below).
5. Click Close.

A stereotype «propertyGroup» has been automatically applied to the newly created attribute.

To create a property subgroup

1. Choose the customization element, for which you want to create a property subgroup.

   The customization element must have at least one attribute with the stereotype «propertyGroup» applied.

2. Right-click the attribute with a stereotype «propertyGroup» and choose Create Element > Property Group.
3. Type the attribute's name (it will be used to name the property subgroup).
4. Open the attribute’s Specification window.
5. In the general specification pane, edit the property values (properties are described in the table below).

   A subgroup does not inherit the properties of the group. You have to specify them for each subgroup separately.

6. Click Close.

A stereotype «propertyGroup» has been automatically applied to the newly created attribute.

The following table gives the descriptions of the stereotype «propertyGroup» properties.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Show Group In Compartment Edit</td>
<td>This property allows for displaying the element properties in the Compartment Edit dialog &gt; the Element Properties tab and in a note on a diagram either within the property group/ subgroup or without it. If the value is true, the property name will be displayed as follows: &lt;property_group_name&gt;::&lt;property_name&gt;. If the value is false, the property name will be displayed as follows: &lt;property_name&gt;.</td>
</tr>
<tr>
<td>Show Group In Element Specification</td>
<td>If the value is true, the property group/ subgroup will be visible in the element’s Specification window.</td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Show Group In Go To</td>
<td>If the value is <code>true</code>, the property group/ subgroup will be visible in element’s shortcut menu as a submenu under the Go To menu.</td>
</tr>
<tr>
<td>Show Group In Quick Properties</td>
<td>If the value is <code>true</code>, the property group/ subgroup will be visible in the element’s Properties panel.</td>
</tr>
<tr>
<td>Show Group In Relation Map</td>
<td>If the value is <code>true</code>, the property group/ subgroup will be visible in the Criterion Editor dialog, when you create or edit a Relation Map diagram.</td>
</tr>
</tbody>
</table>
| Use As Node                    | If the value is `true`, the property group will be displayed in a separate tab in the element’s Specification window and/ or Properties panel.                                                                                                                                                                                                                                                                                                                                                   
|                                | If the value is `false`, the property group will be displayed as a group in the general (default) pane of the element’s Specification window and/ or Properties panel.                                                                                                                                                                                                                                                                                                                                                     
|                                | **NOTE:** Do not set this property value to `true` for the property subgroup.                                                                                                                                                                                                                                                                                                                                                                                                            |
| Properties                     | Stores a list of properties that will be visible in the property group or subgroup.                                                                                                                                                                                                                                                                                                                                                                                                               
|                                | **NOTE:** If the group has at least one subgroup, it may have no properties.                                                                                                                                                                                                                                                                                                                                                                                                                  |
| Columns                        | Stores the specified table columns for showing the values of a multivalued property in the element’s Specification window, for example, `ownedOperation` in Class.                                                                                                                                                                                                                                                                                                                                                                       
|                                | **NOTES:**                                                                                                                                                                                                                                                                                                                                                                                                                      
|                                | • Do not specify this property, when there is more than one property assigned to the properties group and/ or subgroup, i.e., when there is more than one value added to the Properties property.                                                                                                                                                                                                                                                                                                       
|                                | • The property group, to which the multivalued property is assigned, must be specified as a separate tab in the element’s Specification window, for example, the Use As Node property must be set to true.                                                                                                                                                                                                                                                                                                             |
| Filter                         | Stores the element types (UML metaclasses and custom stereotypes) that will be displayed as property values.                                                                                                                                                                                                                                                                                                                                                                             
|                                | For more information, see “Type restriction for custom table in Specification window” on page 80.                                                                                                                                                                                                                                                                                                                                                                                       |
Case study

Let’s create a group *My Group* in the element’s *Properties* panel. This group will be displayed as a new tab and its properties will be grouped into the subgroups *Subgroup1* and *Subgroup2*.

![Figure 40 -- The property group “My Group” in the element’s Properties panel](image)

The element’s customization element will have the attribute representing the property group.

![Figure 41 -- “MyGroup” attribute Specification window](image)
The property group will have another two properties representing the property subgroups.

![Property Group - Subgroup2](image)

*Figure 42 -- "Subgroup2" attribute Specification window*

**Related procedures**
- Creating Customization Data
- Creating Property Groups and Subgroups
- Using Customization Data

**Type restriction for custom table in Specification window**

DSL allows creating a new node in Specification window and showing values of some UML property in the table form.

It is possible to filter elements according to type. The following property is used for this purpose:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Filter</td>
<td>Stores the element types (UML metaclasses and custom stereotypes) that will be displayed as property values. DSL engine checks this value and displays only elements of these types in the table. The <strong>Create</strong> button allows creating a new elements of these restricted types only.</td>
</tr>
</tbody>
</table>
Rules for subtypes

- If type is stereotype, all subtypes (sub stereotypes) are accepted also (for example, if type is Requirement, then BusinessRequirement is accepted also).
- If type is abstract metaclass, all subclasses are accepted (e.g. if type is Classifier - Class, Component and other is accepted. DSL elements are accepted also.

Related procedures
Creating Customization Data
Creating Property Groups and Subgroups
Using Customization Data

Merging property groups and subgroups

You can specify property groups and subgroups for different customization elements of the same element. The element, to which these customizations are applied, will have groups and subgroups from all the customization elements.

Let’s say we have two customizations of the same element type, the actor. The first customization has the property group My Group with the subgroup Subgroup 1 containing the property Prop 1. The second customization has the same property group with the same subgroup specified, but the subgroup Subgroup 1 contains the property Prop 2.

As a result, the actor element type will have properties group My Group with the subgroup Subgroup 1 containing both properties Prop 1 and Prop 2.

Related procedures
Creating Customization Data
Creating Property Groups and Subgroups
Using Customization Data

Default visibility of property groups and subgroups

You can create property groups and subgroups for both standard UML metaclasses and stereotypes. Whether the group will be added to the metaclass or to the stereotype, depends on the customization target of the customization where the property group is created.

If the customization target is a stereotype and the Hide Metatype property is set to true, then the property group specified in the customization will be visible only for the elements with this particular stereotype applied.

If the customization target is a metaclass, then the group specified in this customization will be visible for the elements of this particular type. This group will also be visible for the elements with the stereotypes applied, except for the cases when the Hide Metatype property in this stereotype customization is set to true.

The table below lists the default property group visibility in the element’s Specification window depending on the DSL customization.

<table>
<thead>
<tr>
<th>Customization target</th>
<th>Property group visibility in element's Specification window and Properties panel</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Element of metaclass type</td>
</tr>
<tr>
<td>Stereotype</td>
<td>Invisible</td>
</tr>
</tbody>
</table>

Copyright © 2006-2015 No Magic, Inc.
Creating Numbering Customizations

Numbering customization as well as the numbering scheme, is used in the generic numbering mechanism. In other words, Generic Numbering Mechanism consists of Numbering Customization and Numbering Scheme, as it is depicted in the following figure.

You can create numbering schemes (formats) and apply them on DSL elements in MagicDraw or its add-on program, such as the SysML, Cameo Business Modeler, or UPDM plugin.

You can create one or more numbering schemes for the same DSL element. Moreover, the same numbering scheme can be applied on several DSL elements. In this case, instances of different DSL elements will be numbered in sequence. For example, all actors and use cases in a use case diagram will be numbered in sequence, if the same numbering customization is used for both actors and use cases. Another example of numbering different DSL elements in sequence is shown in the following figure. As you can see, start events,
end events, tasks, and gateways in the BPMN Process diagram are numbered using the same numbering scheme.

![Diagram of BPMN elements numbered in sequence](image)

In this section, you will find the instructions how to customize the generic numbering mechanism in order to create your own numbering scheme. For better understanding the further material, first of all read the following definitions.

**DEFINITIONS**

| Numbering scheme | Defines a numbering style as well as number parts that are used to compose a DSL element number. A numbering schema can have one or more number parts. A numbering scheme is represented as a class with the «NumberingScheme» stereotype applied. |
| Number part | Represents a rule for calculating an individual part of the whole element number, as the element number is composed of one or more individual number parts. It can be a number, character, separator, or other. Number part is represented as a numbering scheme property with the «NumberPart» stereotype applied. |
| Numbering property | Indicates a DSL element property wherein the element number will be stored and defines a numbering scheme that will be used for the DSL element numbering. Numbering property is represented as a customization element property with the «AutoNumber» stereotype applied. |

Learn more about creating numbering customizations in the following sections:

- "Basic steps for creating numbering customization" on page 83.
- "Case study: Creating your first numbering customization" on page 84.
- "Relevant property values" on page 92.

**Basic steps for creating numbering customization**

In order to create a numbering customization for an element, you have to perform the following steps:

1. Create a profile diagram.
2. Create a stereotype to customize a desired element.
3. Add a new attribute to the stereotype.

This attribute will be used to store the element number.
4. Create a numbering scheme to define a numbering style and number parts.
5. Create a customization element for the previously created stereotype.
6. Add a numbering attribute to the customization element. Reopen the project to apply changes.

Related concepts
- Customization Element Attributes

Related procedures
- Creating Customization Data
- Creating Property Groups and Subgroups
- Using Customization Data
- Creating your first numbering customization

Case study: Creating your first numbering customization

Let’s assume that we need to number packages as shown in the following figure.

![Figure 45 -- Task for numbering packages](image)

There should be three numbering levels. Rules for calculating package numbers in each level are described in the following table.

<table>
<thead>
<tr>
<th>Numbering level</th>
<th>Numbering format</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st</td>
<td>&lt;number&gt;</td>
<td>“1”, “2”, “3”</td>
</tr>
</tbody>
</table>
So we need to create a numbering scheme and use it in the numbering customization for package elements. For this purpose, we will perform the steps described in “Basic steps for creating numbering customization” on page 83.

1. Creating a stereotype to customize a desired element

To create a stereotype that customizes UML packages

- Create a profile diagram.
- On the diagram pallet, click the **Stereotype** button.
- Click a free space on the diagram pane. A shape of the new stereotype appears, and the **Select Metaclass** dialog opens.
- Select the **Package** metaclass and click **OK**.
- Name the stereotype **Numbered Package**.

![Figure 46 -- Stereotype for customizing UML packages](image)

You can make the «Numbered Package» stereotype invisible, if you want the stereotype and its properties to be hidden on a diagram.

**Related procedures**

- [Specifying Invisible Stereotypes, Tags, and Constraints](#)

2. Adding a new attribute for storing numbers

To add a new attribute for storing package numbers to the “Numbered Package” stereotype

- Select the shape of the **Numbered Package** stereotype on the profile diagram pane.
- Add a new stereotype attribute of the String type named **No**.

![Figure 47 -- Property for storing package numbers](image)
3. Creating a numbering scheme to define a numbering style and number parts

To create a numbering scheme

1. On the profile diagram pallet, click the **Class** button.
2. Click a free space on the diagram pane. A shape of a new class appears.
3. Name the class **Package Numbering Scheme**.

   The name of the numbering scheme will be displayed in the **Element Numbering** dialog. It is recommended to give it a short and meaningful name.

4. Apply the «NumberingScheme» stereotype on the class.

   ![Diagram](image)

   **Figure 48** -- Scheme for numbering UML packages

To define a numbering style

1. Open the Specification window of the just created numbering scheme.
2. Click the **Numbering Style** value cell to open the list of available numbering styles. Select the **Multi-level** numbering style in this list. We use the Multi-level numbering style because we need to have three numbering levels for numbering packages.
3. Click **Close**, when you are done.

   ![Diagram](image)

   **Figure 49** -- Numbering scheme with multilevel numbering style defined

The following table shows number parts that should be created to compose numbering formats described in the preceding table.

<table>
<thead>
<tr>
<th>Number part</th>
<th>Number part name</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;number&gt;</td>
<td>Level 1</td>
</tr>
<tr>
<td>&lt;separator&gt;</td>
<td>Dot 1</td>
</tr>
<tr>
<td>&lt;character&gt;</td>
<td>Level 2</td>
</tr>
<tr>
<td>&lt;separator&gt;</td>
<td>Dot 2</td>
</tr>
<tr>
<td>&lt;number&gt;</td>
<td>Level 3</td>
</tr>
</tbody>
</table>
Related concepts
Numbering Style property values

To create a number part

1. Select the shape of the numbering scheme on the profile diagram pane.
2. Add a new number part.

To specify a number part for the first numbering level

1. Create a number part named Level 1 (see the procedure "To create a number part" on page 87).
2. Open the Specification window of the number part.
3. Click the Sequence property value specification cell and select Numeric in the values list. We need to have numbers at the first numbering level.
4. Click the Initial Value property value cell and type the number 1 as its value, because the numbering at the first level should start from 1.
5. Click Close, when you are done.

To specify a number part for the first separator

1. Create a number part named Dot 1 (see the procedure "To create a number part" on page 87).
2. Open the Specification window of the number part.
3. Click the Sequence property value specification cell and select Separator in the values list.
4. Click the Initial Value property value cell and type the character (a dot) as its value, because the separator between the first and the second numbering level should be a dot.
5. Click Close, when you are done.

To specify a number part for the second numbering level

1. Create a number part named Level 2 (see the procedure "To create a number part" on page 87).
2. Open the Specification window of the number part.
3. Click the **Sequence** property value specification cell and select **Character** in the values list, because we need to have letters at the second numbering level.
4. Click the **Initial Value** property value cell and type the letter A as its value, because the numbering at the second level should start from A.
5. Click **Close**, when you are done.

![Figure 52 -- Numbering scheme with number parts defined for second numbering level](image)

The rest of number parts are created appropriately.

![Figure 53 -- Numbering scheme with all number parts defined](image)

As there are no number parts specified for the fourth and further numbering levels, appropriate packages will be numbered using number parts of the third numbering level. For example, “1.A.1.1”, “1.B.1.2”, and so forth.

### 4. Creating a customization element for the stereotype that customizes the desired element

In order to use the just created scheme for numbering packages, we need to define it in the numbering customization for package elements. For this, a customization element must be first created.

To create a customization element for the “Numbered Package” stereotype

1. On the diagram pallet, click the **Customization** button.
2. Click a free space on the diagram pane. A shape of the new customization appears.
3. Name the customization **Numbered Package Customization**.
4. Open the **Customization** Specification window.
5. Click the **Customization Target** property, then select the **Numbered Package** stereotype in the element Selection dialog, and then click **OK**.
6. Set the Show Properties When Not Applied property to true. All properties of the Numbered Package stereotype will be visible as properties of the Package metaclass even if the stereotype is not yet applied.
7. Click Close, when you are done.

Figure 54 -- Customization element for “Numbered Package” stereotype

5. Add a numbering attribute to the customization element

After the customization for the Numbered Package stereotype has been created, we can specify the stereotype attribute for storing element numbers as well as define a numbering scheme that will be used. For this, a numbering attribute must be created.

To add a numbering attribute to the customization element of the Numbered Package stereotype

1. Select the shape of the customization.
2. Add a new Auto Numbering Property named Package Numbering.
3. Open the Specification window of the created attribute.
4. Specify value for the Numbered Property property. Click the property value cell and then click the ... button. In the opened element Selection dialog, select the No. property of the Numbered Package stereotype and click OK.
5. Specify value for the Numbering Scheme property. Click the value cell and do one of the following:
   - Select the Package Numbering Scheme class in the opened list.
   - Click the ... button. In the opened element Selection dialog, select the Package Numbering Scheme class and click OK.
6. Change the Default Number property value to true. The true value means that this numbering attribute will be set as default, when numbering instances of the customized Package metaclass.
7. Specify the Use Prefix On Owner attribute value. The true value means that the element prefix will be used for its numbered owner as well.
8. Click **Close**, when you are done.

---

**Figure 55 -- Numbering attribute in customization element of the “Numbered Package” stereotype**

6. Changing the number display mode

In order to see the number before the element name, we need to change the number display mode.

To change the number display mode on packages:

1. Open the appropriate package diagram.
2. Create a package on which you want to apply the numbering scheme.
3. Name the package **Organization**.
4. Apply the **Numbered Package** stereotype on the **Organization** package.
5. Right-click the shape of the package on the diagram pane, and then click **Symbol Properties**.
6. Set the **Show Number Tag Name** property value to **false**. The **false** means that the element number tag name, that is, the No property of the **Numbered Package** stereotype, will not be displayed on the package shape.
7. Click the **Element Number Display Mode** value cell to open the list of available modes. Select the **Before the element name** mode in this list.

---

For more information on how to apply the numbering settings to all packages of the **Numbered Package** stereotype, refer to “Stereotype properties” in **MagicDraw UserManual.pdf**.
7. Reopen the project to apply changes

Open the appropriate package diagram and see the packages numbered using the just created numbering customization.

![Figure 56 -- Packages numbered using numbering customization](image)

**Related procedures**

- Creating Customization Data
- Creating Property Groups and Subgroups
- Using Customization Data
## Relevant property values

### Numbering Style property values

<table>
<thead>
<tr>
<th>Property value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Multi-level</strong></td>
<td>Select this style, if you need to number elements in several numbering levels. Specify at least one number part for each numbering level. Top level elements will be numbered using the first number part. Elements in the subsequent levels (that is, those that are directly or indirectly contained in the top level element) will be numbered using successive number parts. <strong>NOTE:</strong> If you have specified less number parts than there are element levels, the last element number part of the separator type and the last element number part of the non-separator type (for example, numeric or character) will be reused to number a subsequent level elements. For instance, if number parts are specified only for two numbering levels, and elements in the second level are numbered as “1.A”, “1.B”, and “1.C”, elements in the third numbering level will be numbered as “1.A.A”, “1.A.B”, and so on.</td>
</tr>
<tr>
<td><strong>Consecutive</strong></td>
<td>Select this style, if you need to number elements in one numbering level. Specify only one number part. <strong>NOTE:</strong> If you have specified more than one number part, elements will be numbered using only the last number part. All other number parts in this case will be treated as static. For instance, if an element number includes five number parts (for example, “1.1.1”), and the consecutive numbering style is selected, then all elements will be numbered by changing only the last number part, that is, “1.1.1”, “1.1.2”, “1.1.3”, and so forth. The same happens after changing multi-level numbering style to consecutive.</td>
</tr>
</tbody>
</table>

### Sequence values

### Related procedures

<table>
<thead>
<tr>
<th>Property value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Numeric</strong></td>
<td>Select this type, if you need to use only positive numbers for calculating a value of the number part. <strong>NOTE:</strong> Elements will be numbered starting with “1” in ascending order, if no initial value is specified.</td>
</tr>
<tr>
<td><strong>Character</strong></td>
<td>Select this type, if you need to use only Latin letters for calculating a value of the number part. <strong>NOTE:</strong> Elements will be numbered starting with “A” in ascending order, if no initial value is specified.</td>
</tr>
<tr>
<td><strong>Expression</strong></td>
<td>Select this type, if you need to use an OCL or Binary (Reference to a code class) expression for calculating a value of the number part. Use the <strong>Expression Editor</strong> dialog to define the expression.</td>
</tr>
<tr>
<td><strong>Separator</strong></td>
<td>Select this type, if you need to specify a separator between two number parts. <strong>IMPORTANT!</strong> Do not forget to specify the separator value for the <strong>Initial Value</strong> property. If the property value is not specified, the separator will not be used.</td>
</tr>
<tr>
<td><strong>OwnerNumber</strong></td>
<td>Select this type, if you need to use the number of an owner that is numbered using another numbering customization for calculating a value of the number part.</td>
</tr>
</tbody>
</table>

Creating Customization Data  
Creating Property Groups and Subgroups  
Using Customization Data
Extending Metamodel with Derived Properties

DSL customization allows for extending the UML metamodel or its profile (such as SysML or UPDM) by creating new read-only derived properties.

A derived property is the one, whose values are calculated automatically from the other properties’ values.

It is important to notice that the derived properties are calculated on the fly. MagicDraw analyzes the derived properties and calculates them for the existing model elements, when loading a model, and dynamically updates them according to the model changes.

The derived properties can be added either to the standard UML elements, or to the ones that are extended with stereotypes. In order to extend the metamodel with the derived properties, there is no need to apply the stereotypes - the metamodel is extended in runtime.

You can define the derived properties in different expression types: Simple, Meta Chain, OCL, and Binary (reference to java code).

You can easily access derived properties of a select model element, navigate to their specifications, or use them in the model analysis tools, such as dependency matrices, report templates, an relation maps.
Figure 57 -- Derived properties specification metamodel

Related procedures
- Creating Customization Data
- Using Customization Data

Specifying derived properties

In order to be included in a MagicDraw project as a property extending a UML element, the attribute must be owned by a DSL customization element and must be stereotyped as «derivedPropertySpecification».
To create and specify a derived property

1. Select the customization element, for which you want to create a derived property.

   The derived property will be created for the stereotype or metaclass, which is defined as a customization target in the selected customization element specification.

2. From its shortcut menu select Create Element > Property. The attribute is created.

3. Type the attribute name.

   The attribute’s name corresponds the name of the derived property.
   It is recommended to type derived property names in camel case, for example, realizingActivities.
   A property name, that has been defined in camel case, will be displayed in separate words with the first word capitalized, for example, realizingActivities will be changed to Realizing Activities for showing in the user interface.

4. Apply the stereotype «derivedPropertySpecification» to this attribute. The result view of the Model Browser must be appropriate to the picture below.

   ![Figure 58 -- Example of customization element with the attribute for the derived property specification](image)

5. In the attribute’s Specification window, edit property values specifying the derived property.

   All the properties are specified in the general specification pane. Some of them can also be specified in the Tags specification pane. For detailed information, see the table below.

   If you do not see the newly created derived property in the element’s Specification window, read the following information. It may help you.
   - In the element's Specification window, the specified derived property is visible by default only in the All properties mode. So first of all try to switch to this mode. Furthermore, you can change the mode, in which you want the derived property to be visible (click the Customize button).
   - If you have specified the derived property for the element, which has the Used UML Properties property of DSL customization specified, you must add the newly created derived property to the element’s visible properties list. For the detailed information, please refer to the section "To make derived properties of customized elements which has the Used UML Properties property specified visible" on page 99.

The properties that are specified for the derived property are described in the table below.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Specified in...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>The name of the property. This name will be added and visible in the specification of the element, whose type is defined as a customization target.</td>
<td>General specification pane.</td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
<td>Specified in...</td>
</tr>
<tr>
<td>---------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-----------------------------------------------------</td>
</tr>
<tr>
<td>Type</td>
<td>The type of the derived property values. You may set a model element type or a data type (e.g., string, boolean, or integer from the UML Standard Profile) as a property value. Example: If you want classes and components to be displayed as property values, select the classifier as a type for the derived property value.</td>
<td>General specification pane.</td>
</tr>
<tr>
<td>Multiplicity</td>
<td>The multiplicity of the derived property values. Example: If the multiplicity is “0..1”, the derived property value will be shown as single. If the multiplicity is “0..*”, the derived property values will be shown as collection.</td>
<td>General specification pane.</td>
</tr>
<tr>
<td>Is Read Only/ isReadOnly</td>
<td>If true, the derived property is read-only, meaning that it is not allowed to edit the result elements for the derived property value. If false, the property is read-write, meaning that it is allowed to edit the result elements for the derived property according to the expressions defined. <strong>NOTE:</strong> Currently only the read-only derived properties are supported.</td>
<td>• General specification pane.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Tags specification pane.</td>
</tr>
<tr>
<td>Is Ordered</td>
<td>If true, the property values are always displayed in the same order. If false, the property values each time are displayed in non-predictable order.</td>
<td>General specification pane.</td>
</tr>
<tr>
<td>Is Unique</td>
<td>If true, the property values are unique, i.e., the same element is displayed only once. If false, the property values may be displayed more than once.</td>
<td>General specification pane.</td>
</tr>
<tr>
<td>Expression/ expression</td>
<td>One or many expressions of the derived property. An expression defines the criterion for selecting the result elements. There are four possible expression types: • Simple (UML relationships, properties, and tags) • Multi properties chain (Meta Chain) • OCL • Binary (Reference to code class) The criterion for selecting the result elements can also be any combination as union of the expression types above. For the detailed information, see &quot;Defining expressions&quot; on page 97.</td>
<td>• General specification pane.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Tags specification pane.</td>
</tr>
<tr>
<td>Documentation</td>
<td>The text that will be displayed as the derived property description in the element's Specification window and Properties panel.</td>
<td>Documentation specification pane.</td>
</tr>
<tr>
<td>customizationTarget</td>
<td>One or many element types (stereotypes and/or metaclasses), for which the specified derived property will be added. <strong>NOTE:</strong> The element type of the customization target is the one, from which the expression calculations are started.</td>
<td>Customization element Specification window</td>
</tr>
</tbody>
</table>
Figure 59 -- Derived property specification. General specification pane

Related procedures

Creating Customization Data
Defining expressions
Extending Metamodel with Derived Properties
Using Customization Data

Defining expressions

Each derived property must have one or more expressions defining the criteria according to which the result elements are searched.
To open the dialog for a new expression definition

It is strongly recommended to have the customization target specified in the customization element before defining the expression for its derived property.

1. Select the attribute with the stereotype «derivedPropertySpecification».
2. Open the attribute’s Specification window.
3. In the general specification pane (opened by default), click the Expression property and then click the “...” button.

As the dialog opens, you are ready to define one or more criteria for calculating the derived property values.

**Related external resources**


**Recursive properties**

A derived property gathering all property values recursively of the property that is specified for the result elements.

A recursive property is defined as a binary class

```
"com.nomagic.magicdraw.derivedproperty.RecursiveExpression()"
```

with parameters `<property-name>` and `<recursive-property-name>`.

With help from the recursive properties, you can collect and represent, for example:

- All the elements, which are directly or indirectly related to the specific one through the generalization relationships.

Refer to derived property customization expression, which is available in `<MagicDraw installation directory>\profiles\Traceability Customization.mdzip`. Go to Traceability customization::Properties descriptors::Realization::Classifier Realizing Classifiers::moreSpecificClassifiers.

- All the elements, which are directly or indirectly related to the specific one through the abstraction and other types of relations.

Refer to derived property customization expression, which available in `<MagicDraw installation folder>\profiles\Traceability Customization.mdzip`. Go to Traceability customization::Properties descriptors::Realization::Element All Realizing Elements::allRealizingElements.

**Related procedures**

Extending Metamodel with Derived Properties
Using Customization Data
Creating Customization Data

**Expressions merge**

Let’s suppose we have two customization elements extending the same UML element, for example, the same element is specified as a customization target in both customization elements. Each customization element has a derived property. If the derived properties’ names, types, and multiplicities are the same, then the expressions defined for these derived properties are merged. Consequently the extended UML element will have only one derived property instead of two and the property values will be calculated according to the union of expressions defined in both derived properties.
This situation occurs when the derived property, which is already specified in, for example, the UML Standard Profile, is also specified in a project. Note that in contrast to the example above the real-life expression merge can be performed in more complex situations with more than two derived properties having more than one expression defined. The expression merge can make unions of different types of expressions.

**Related procedures**
- Extending Metamodel with Derived Properties
- Using Customization Data
- Creating Customization Data

### Derived properties visibility

A newly created derived property by default is visible in the following places:

- General specification pane of the DSL element’s Specification window.
- *Properties* panel of the DSL element.
- *Compartment Edit* dialog.

To see derived properties of customized elements which has **Used UML Properties** property specified, make sure derived properties are assigned as values for the **Used UML Properties** property.

- *Criterion Editor* dialog for editing relation criteria in a Relation Map diagram.
- *Select Dependency Criteria* dialog for editing the dependency criteria in a Dependency Matrix diagram.

Moreover, DSL customization allows for being more precise, when specifying a derived property’s visibility as this can be done through the specification of the DSL element’s properties group or subgroup, to which the derived property is assigned. As a result, the derived property will be visible in the group or subgroup according to this group’s or subgroup’s visibility settings.

To make derived properties of customized elements which has the Used UML Properties property specified visible

1. Open the Specification window of customization element.
2. In the general specification pane, select the **Used UML Properties** property and assign one or more derived property values.
3. Save and reopen the project.

To assign the derived property to a property group

1. Select the customization element with the derived property.
2. Create a property group for the customization element (perform 1 to 4 steps of the procedure "To create a property group" on page 76). 
3. In the general specification pane, edit property values that specify the property group’s visibility.
4. In the property group Specification window, click the **Properties** property and then click the button.
5. In the *Item Filter* dialog, select the check box near the appropriate derived property and click **OK**. As it is depicted in the figure below, the selected derived property will be assigned to **Properties** as a new value.
6. Click **Close**.
The steps for assigning the derived property to a property subgroup is a parallel case. You just have to create a property subgroup in addition and assign the derived property not to the property group, but to the subgroup.

It is also allowed to assign a derived property to a property group, which is specified not in the same customization element, if both customization elements extend the same UML element.

Figure 60 -- Derived property assigned to property group
Moreover, all the derived properties, together with custom UML properties, are available for creating your own Report templates.

**Forward Traceability – Realization**

The forward traceability ensures that all specified artifacts are covered by elements from lower abstraction level.

1. **Class Realizing Components**

Shows the components representing the class realization in the implementation model.

```
#set($classes = $report.filterElement($elements, ['"Class"']))
#set($counter=0)
#set($coverage=0)
#set($percent=0)

$project.name
Date: $date.get("MMMM dd, yyyy")
Traceability Report
Revision: $Revisions.lastChild.name
```

```
<table>
<thead>
<tr>
<th>Class</th>
<th>Component</th>
</tr>
</thead>
<tbody>
<tr>
<td>#foreach ($Class in $sorter.sort($classes)) #set ($list = $Class.realizingComponent)</td>
<td></td>
</tr>
<tr>
<td>#set($void = &quot;&quot;) $report.setIconFor($Class) $Class.name</td>
<td></td>
</tr>
<tr>
<td>[Class.owner.qualifiedName]</td>
<td></td>
</tr>
</tbody>
</table>
| $project.name
Date: $date.get("MMMM dd, yyyy") |
Traceability Report
Revision: $Revisions.lastChild.name |
```

*Figure 61 -- Derived properties in Traceability Report template*

**Related procedures**

- Extending Metamodel with Derived Properties
- Using Customization Data
- Creating Customization Data
- Creating Property Groups and Subgroups
You can distribute profiles with included customization data in the following two ways:

- Exchanging Profiles via XMI
- Distributing Resources by Using Resource Manager

**Exchanging Profiles via XMI**

Using UML XMI interchange mechanisms it is possible to interchange profiles between tools, together with models to which they have been applied. A profile must therefore be defined as an interchangeable UML model. In addition to exchanging profiles together with models between tools, profile application should also be definable “by reference”; that is, a profile does not need to be interchanged if it is already present in the importing tool.

**Related procedures**

Creating Profiles

**Distributing Resources by Using Resource Manager**

You can distribute created custom resources (such as a sample, profile, DSL customization, custom diagram, or other) using Resource/Plugin Manager.

**Related external resources**

“Distributing Resources” in MagicDraw OpenAPI UserGuide.pdf