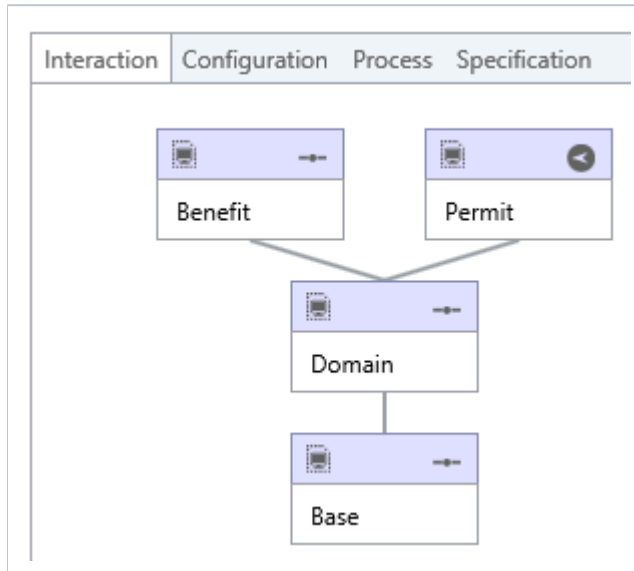


# How to copy modules

## Introduction

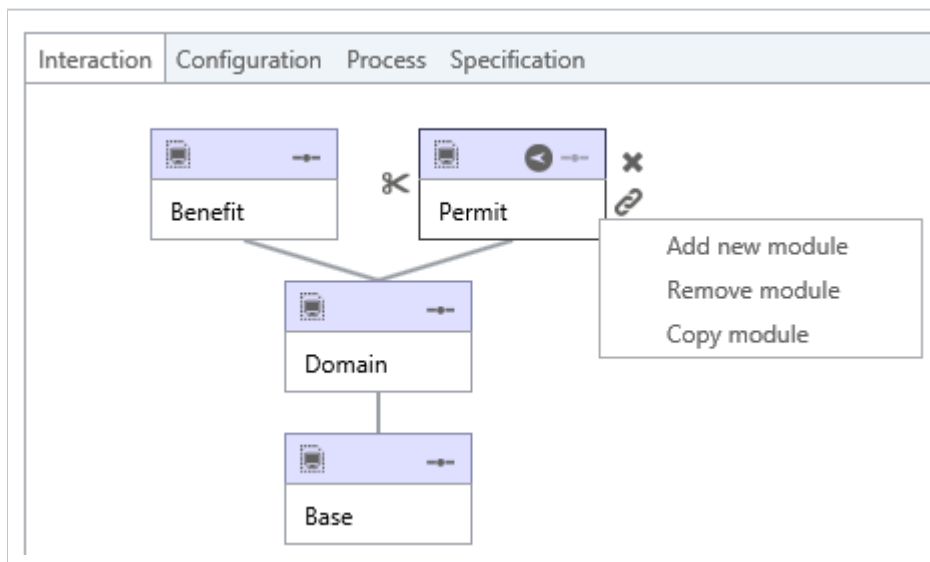
At some point in the development process, a business engineer might want to *copy a module*. This is most likely the case with interaction modules, therefore the functionality to copy modules is only available for modules of this type. To demonstrate this functionality, the example project shown below is used. Only the four interaction module are relevant in this example.



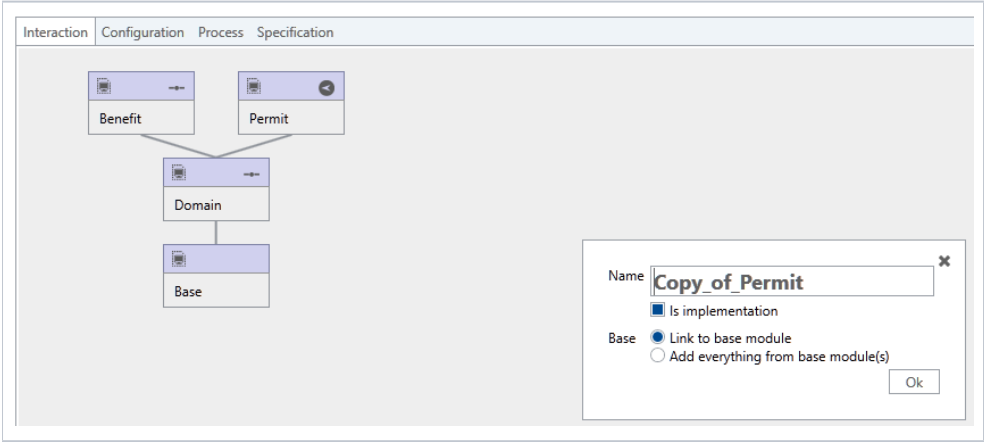
To copy a module, make sure you are in the *Edit project structure* mode.

- To get there when a project is open simply follow:
  - File > Close project > Single-click the project
  - Now follow one of these paths:
    - View Structure > Edit structure
    - Edit project > Edit project structure
- From the Home screen (Recent projects | Getting started) follow one of these paths:
  - Branch selection > Single-click the desired Repository (left) > Single-click the desired branch (right) > Open branch > Single-click the desired project
  - Now follow one of these paths:
    - View Structure > Edit structure
    - Edit project > Edit project structure

Select the module by clicking left and then right click on a module to see the menu. Choose Copy module:



When Copy module is pressed, the following dialogue appears:



Type in the new name, this name has to be unique within the project. Choose whether the module is an implementation or not. By default each copied module is an implementation module, regardless of the fact whether the source module is an implementation module or not.

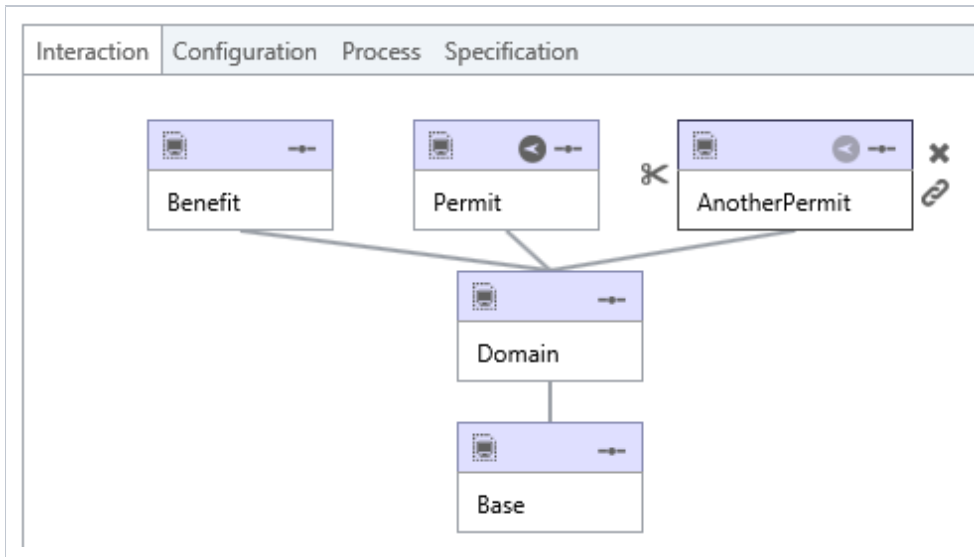
The radio button **Base** that lets you choose between linking the new module to a lower module and adding elements from lower modules. This will be discussed in the next two chapters, with the following example:

	Original		
Permit	↓, Customer (A,B',C)	↓ Address	= Permit
Domain	↓, Customer (A,B)	= Address	↑ Permit
Base	=, Customer (A,B)	↑ Address	↑ Permit

In the example above the focus is on the module Permit, but since it is linked to Domain and vicariously to Base, these modules are in scope as well. In the example, the module Base has an element (in this case an entity) Customer with attributes A and B. In the module Domain, this entity is visible and in Permit it has been specialized. This specialization comprised an adjustment of attribute B and the addition of attribute C. Entity Address originates in module Domain and has not been specialized further. Entity Permit originates in module Permit.

### Link to base module

The radio button Base has two options. When the option **Link to base module** is chosen, the copied module will be linked to the same module that the original module was linked to. In essence, Permit and AnotherPermit are exactly the same module.



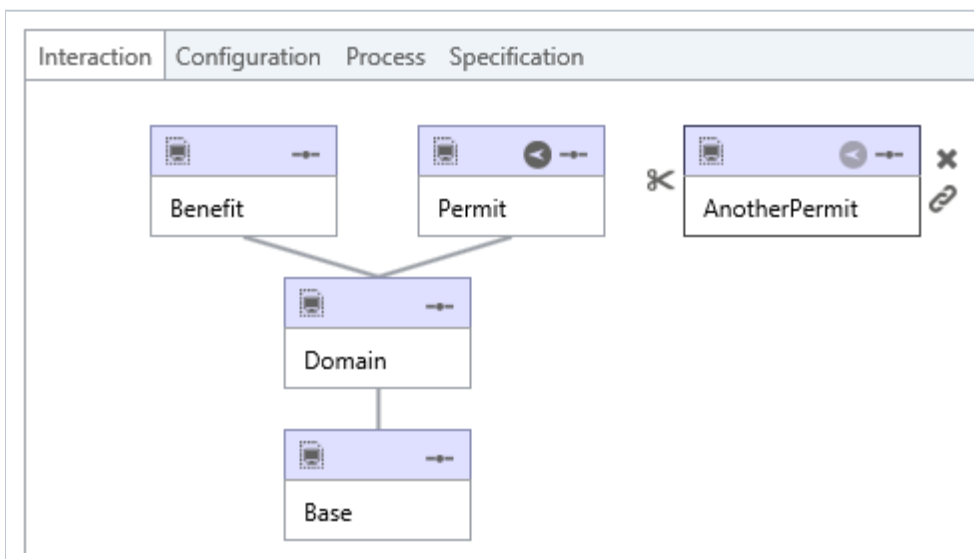
The AnotherPermit module consists of the parts as depicted below after copying the module Permit to AnotherPermit and linking it to its base module Domain:

Link to base module		
AnotherPermit	↓, Customer (A,B',C)	↓ Address = Permit
Domain	↓, Customer (A,B)	= Address ↑ Permit
Base	=, Customer (A,B)	↑ Address ↑ Permit

The content of the example above is exactly the same as the original, except of course the name of the module.

## Add everything from base module(s)

When the option **Add everything from base module(s)** from the radio button Base is chosen, the copied module will contain all elements from the original module and all modules that are directly and indirectly linked to the original module.



The AnotherPermit module consists of the parts as depicted below after copying the module Permit to AnotherPermit and adding everything from its base modules:



In the example shown above, everything from the modules Base, Domain and Permit is combined into the module AnotherPermit. This principle of combining is sometimes referred to as flattening ("platslaan" in Dutch) or rolling up ("oprollen" in Dutch). The result is that the module AnotherPermit contains Customer, Address and Permit and all of these elements are available in their newest, i.e. most specialist, manner. This is visualized by B' as part of the Customer entity in the copied module. The fact that B' is shown in the Customer entity does not mean that B is specialized in the copied module however. It merely means that the specialized version of B from the source is copied to AnotherPermit.

---

## Related documentation

[Reporting and sharing data with other systems](#)

---

## Related articles

- [How to setup a connection](#)
- [Reporting and sharing data with other systems](#)
- [Search in structured and unstructured information](#)
- [How to enable case-sensitive search with Oracle Text](#)
- [Task behavior](#)