

# Creatio IDE

Replace configuration elements

Version 8.0



This documentation is provided under restrictions on use and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this documentation, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

# Table of Contents

Replace configuration elements

4

# Replace configuration elements



Beginner

Creatio development is based on the main principles of object-oriented programming. In particular, the Creatio extension model is based on the **open-closed principle**: the major Creatio logic is open for extension but closed for modification. This means that new features must be developed by introducing new entities rather than modifying the existing entities.

The configuration elements in pre-installed packages are closed for modification on the system level. Develop and modify the functionality in custom [packages](#) using the **replacement mechanism**. In Creatio, the replacement implementation is based on the concepts of replacing and replaced configuration elements.

The **replacing configuration element** is a configuration element that replaces another configuration element of the corresponding type.

The **replaced configuration element** is a configuration element that is replaced by another configuration element of the corresponding type.

Creatio IDE lets you replace the following **configuration elements**:

- **The client module that defines the view model.**

Client modules implement the front-end part of Creatio. To create a replacing client module, use the replacing view model schema. To do this, follow the guide in a separate article: [Client module](#).

- **Object.**

Objects implement the back-end of Creatio. To create a replacing object, use the replacing object schema. To do this, follow the guide in a separate article: [Object](#).

- **Source code.**

Source code implements the back-end of Creatio. Classes serve as replacing configuration elements. To create a replacing class, use the schema of the [ *Source code* ] type. To do this, follow the guide in a separate article: [Source code \(C#\)](#).

After you implement the replacing configuration element, Creatio will execute the logic of the replacing configuration element when accessing the element.

Creatio IDE lets you replace a single configuration element in multiple custom packages. The [hierarchy of packages](#) that contain the replacing configuration elements defines the resulting implementation of the replacing configuration element in the compiled configuration.