

Creatio IDE

SQL script

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

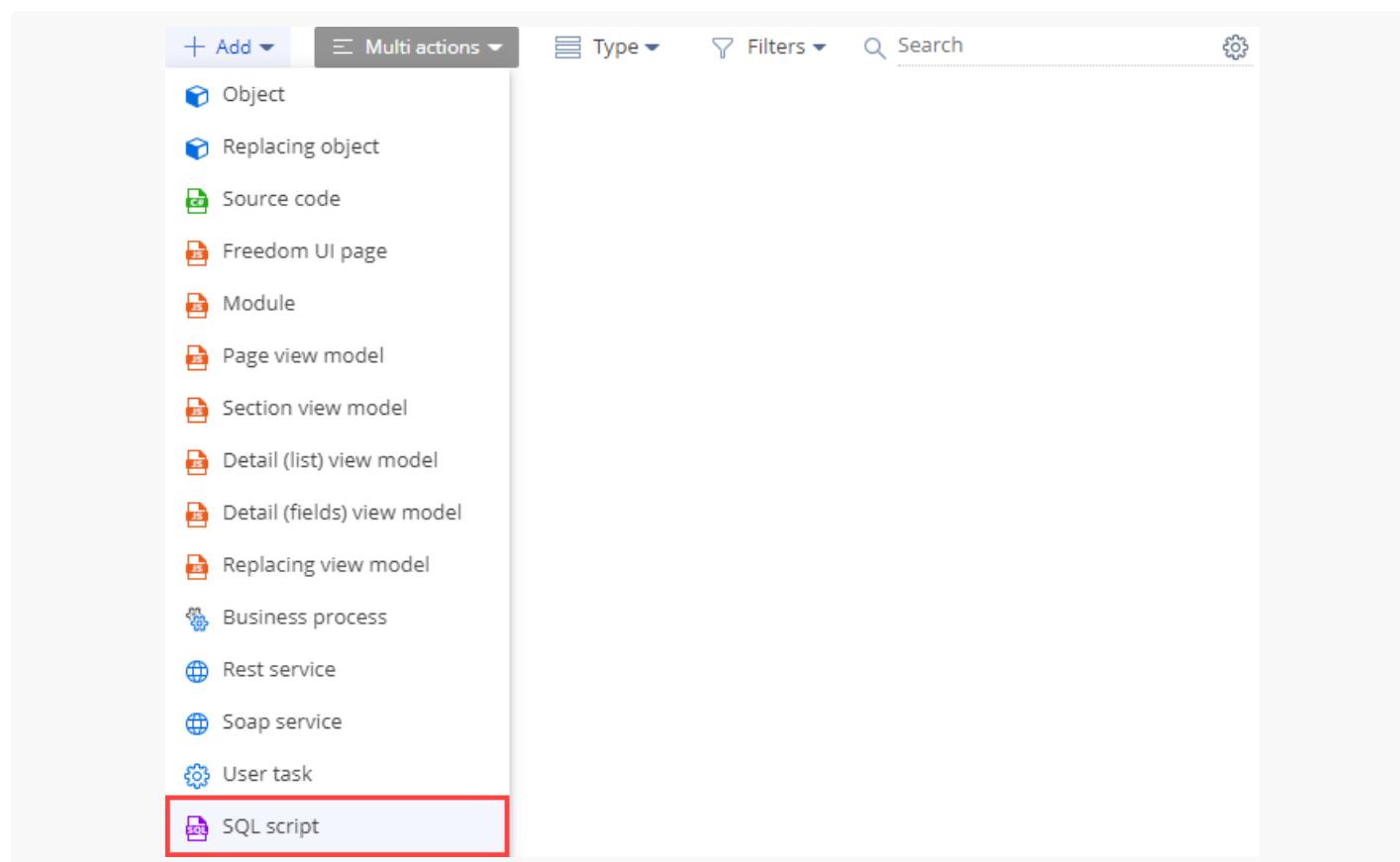
SQL script	4
Implement an SQL script	5

SQL script

 Beginner

Configuration element of the [*SQL script*] type is an entity that lets you implement database queries written in SQL. The **purpose** of an SQL script is to create databas objects, for example, views, procedures, functions, or execute other Creatio database queries.

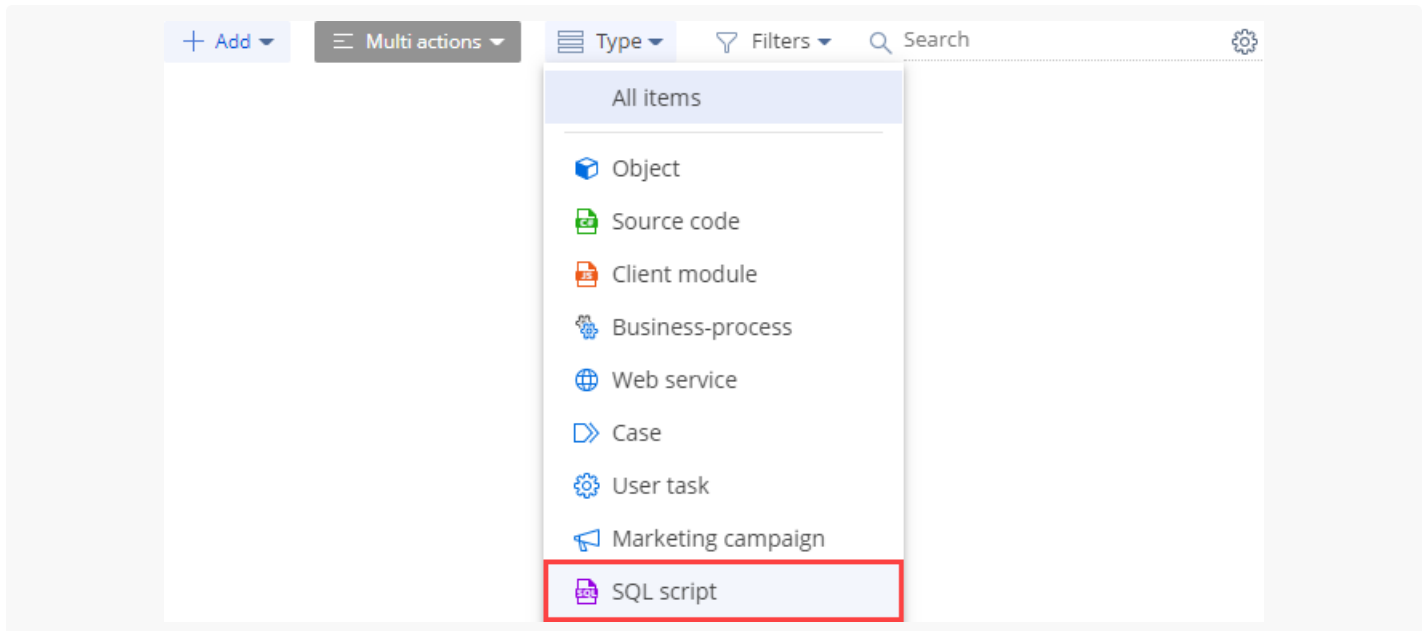
The item of the [*Add*] drop-down list in the toolbar of the [*Configuration*] section workspace represents the configuration element of the [*SQL script*] type you can add in Creatio IDE.



Learn more about configuration element types in a separate article: [Operations in Creatio IDE](#).

The [*SQL script*] item in the [*Type*] drop-down list in the toolbar of the [*Configuration*] section workspace represents the configuration element of the [*SQL script*] type.

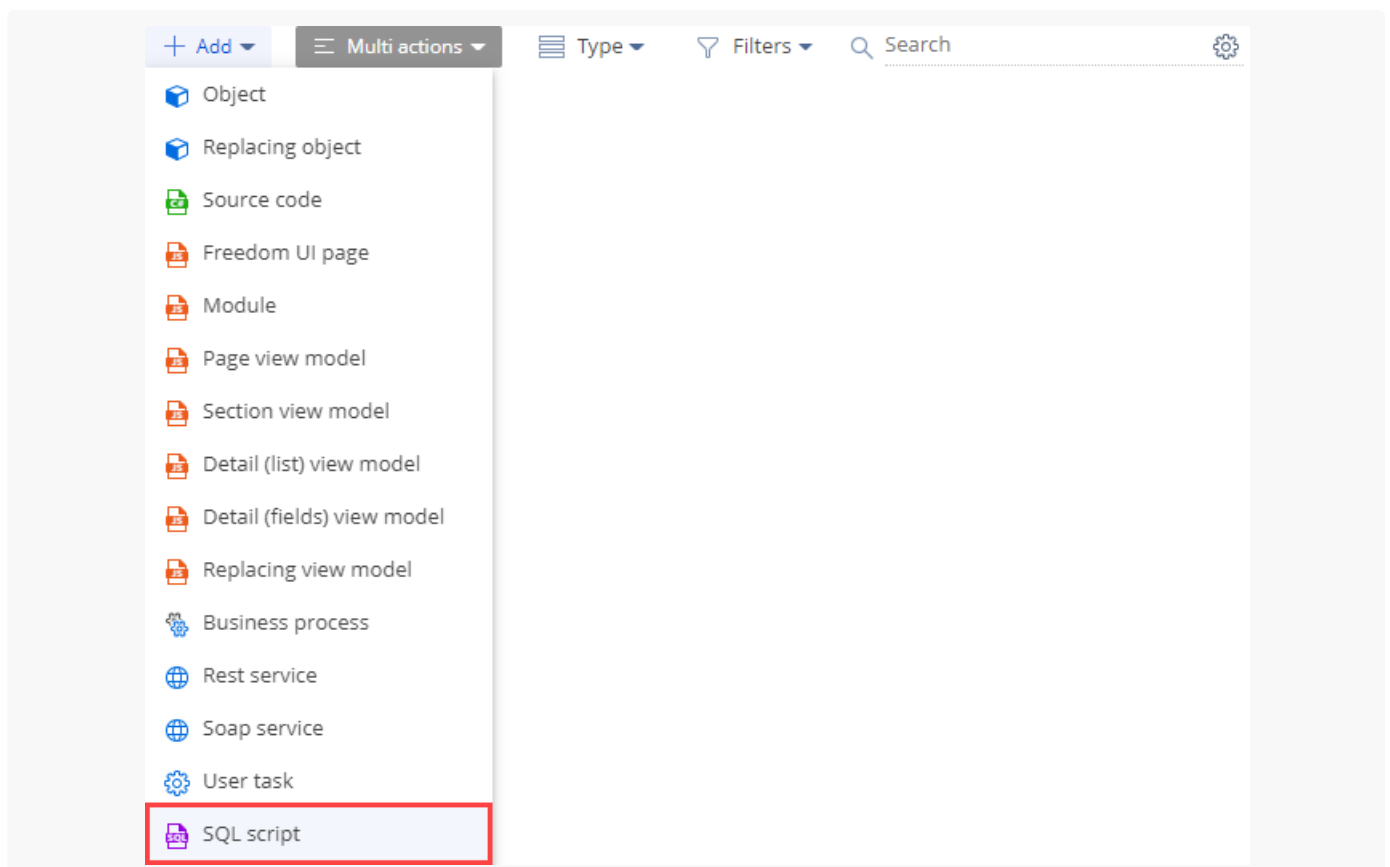
View the SQL script **type** in the figure below.



Learn more about configuration element types in a separate article: [Operations in Creatio IDE](#).

Implement an SQL script

1. [Go to the \[Configuration \] section](#) and select a custom [package](#) to add the configuration element.
2. Click [Add] → [SQL script] on the section list toolbar.



3. Fill out the configuration element properties in the Script Designer.

Fill out the **main properties** of the configuration element:

- [*Code*] is the schema name. Required. Name the SQL script using the **template** below.

Template of the SQL script name

[Prefix][Operation][Object][DBMS]

Example of the SQL script name

UsrUpdateActivityDateMSSQL

[Prefix] is the prefix of the configuration element name (by default, `Usr`) specified in the [*Prefix for object name*] (`SchemaNamePrefix` code) system setting. Can contain alphanumeric characters. Creatio checks whether the prefix exists and matches the system setting when you go to a different configuration element property. If the prefix does not exist or does not match, Creatio sends a corresponding user notification.

[Operation] is the operation the SQL script executes Available values: `Insert`, `Update`, `Delete`. Optional for SQL scripts that create objects (the `Create` value).

[Object] is the object with which the SQL script interacts. Required.

[DBMS] is the database type for which you are developing the SQL script. Required. Must match the [*DBMS type*] property value of the SQL script.

- [*DBMS type*] is the database type for which you are developing the SQL script. Required. Available values: "MSSql," "Oracle," "PostgreSql."
- [*Installation type*] is the script execution order when installing the package. Required.

Available **values**:

- Select "BeforePackage" to execute the SQL script before the package installation.
- Select "AfterPackage" to execute the SQL script after the package installation.
- Select "AfterSchemaData" to execute the SQL script after the package data (configuration elements of the [*Data*] type) installation.
- Select "UninstallApp" to execute the SQL script when deleting the package to which the script is bound.
- [*Package*] is the custom package where you create the configuration element. The property is populated automatically and non-editable.
- [*Backward compatible*] specifies whether the script is backward compatible. Backward compatible SQL scripts are available in Creatio version 8.0.3 and later. Learn more in a separate article: [Backward compatible SQL scripts](#).

Sql script [X]

Code *
UsrUpdateContactMSSql

DBMS type *
MSSql

Installation type *
AfterPackage

Package
sdkSqlScriptPackage

Backward compatible

Backwards compatible SQL scripts are those that comply with recommendations [specified in a separate article](#). Select this checkbox only if your script complies with every recommendation

CANCEL APPLY

Click [*Apply*] to apply the properties.

The **properties area** of the Script Designer lets you:

- edit the main configuration element properties (✎ button)
- specify the additional configuration element properties (+ button)

The **additional properties** of the configuration element are as follows:

- [*Depends on SQL Scripts*]. Lets you select the SQL scripts to execute before executing the current script.
- [*Dependent SQL Scripts*]. Contains the SQL scripts to execute after executing the current script. The property is populated automatically and non-editable.

To **execute SQL scripts in a set order**:

- a. Select the corresponding installation type (the value of the [*Installation type*] property).
 - b. Set the needed dependencies between scripts ([*Depends on SQL Scripts*] and [*Dependent SQL Scripts*] properties).
4. Click [*Validation*] on the Script Designer's toolbar to validate the syntax of the SQL script.
 5. Click [*Save*] on the Script Designer's toolbar to save the changes to configuration element metadata.