Creatio Academy

Creatio IDE

Object

Version 8.0

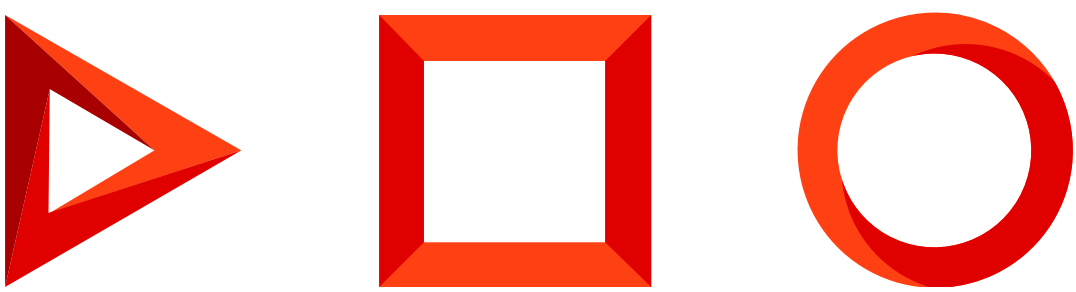


Table of Contents

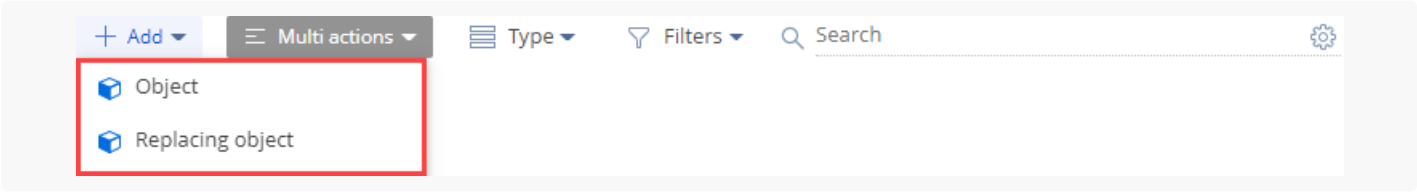
Object

Beginner

A **configuration element** of the [*Object*] type is a business entity that lets you declare a new ORM model class on the server core level. If you create an object, a new table with the same name and the same set of columns as the object is created on the database level. In other words, a Creatio object is a system view of a physical database table in most cases. The **purpose** of the object is to enable back-end Creatio development.

The items of the [*Add*] drop-down list in the toolbar of the [*Configuration*] section workspace represent the object types you can add in Creatio IDE.

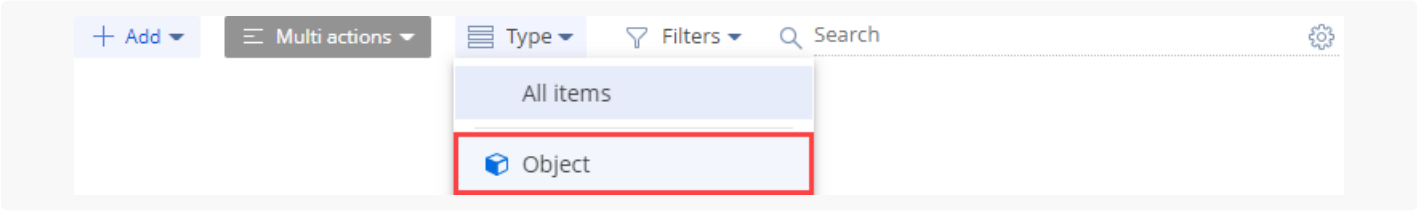
View the object **types** in the figure below.



Learn more about configuration element types in a separate article: Operations in Creatio IDE.

The [*Object*] type schema in the [*Type*] drop-down list in the toolbar of the [*Configuration*] section workspace represents the object. The object schema describes the list of object columns, indexes, and methods. Creatio platform does not limit the number of object columns. The number of object columns is limited by the maximum number of columns in the tables of the client database.

View the **type** of the object schema in the figure below.



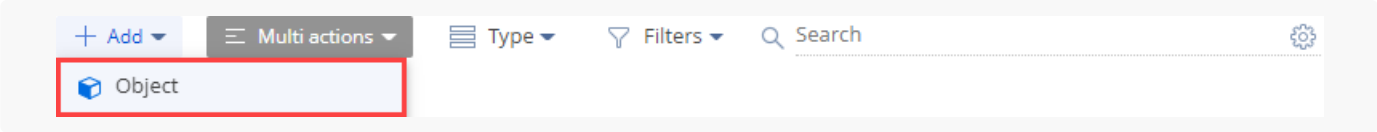
Learn more about configuration element types in a separate article: Operations in Creatio IDE.

Implement an object

The [*Object*] schema **type** represents the object.

To **implement an object**:

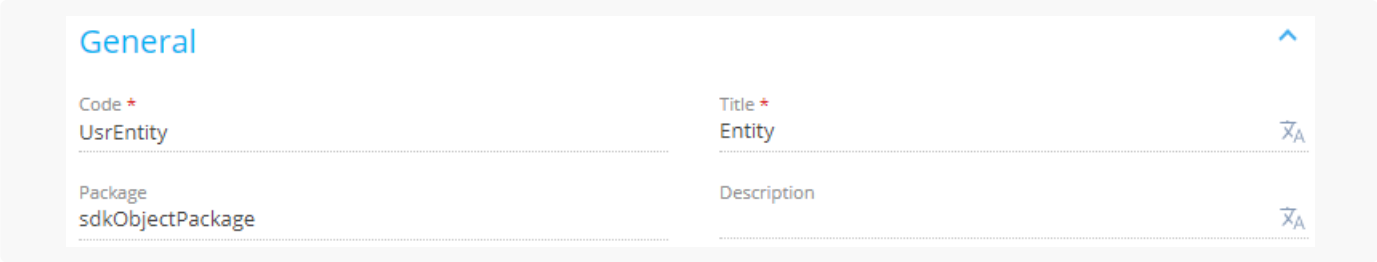
1. Go to the [*Configuration*] section and select a custom package to add the schema.
2. Click [*Add*] → [*Object*] in the section list toolbar.



3. Fill out the schema properties in the Object Designer.

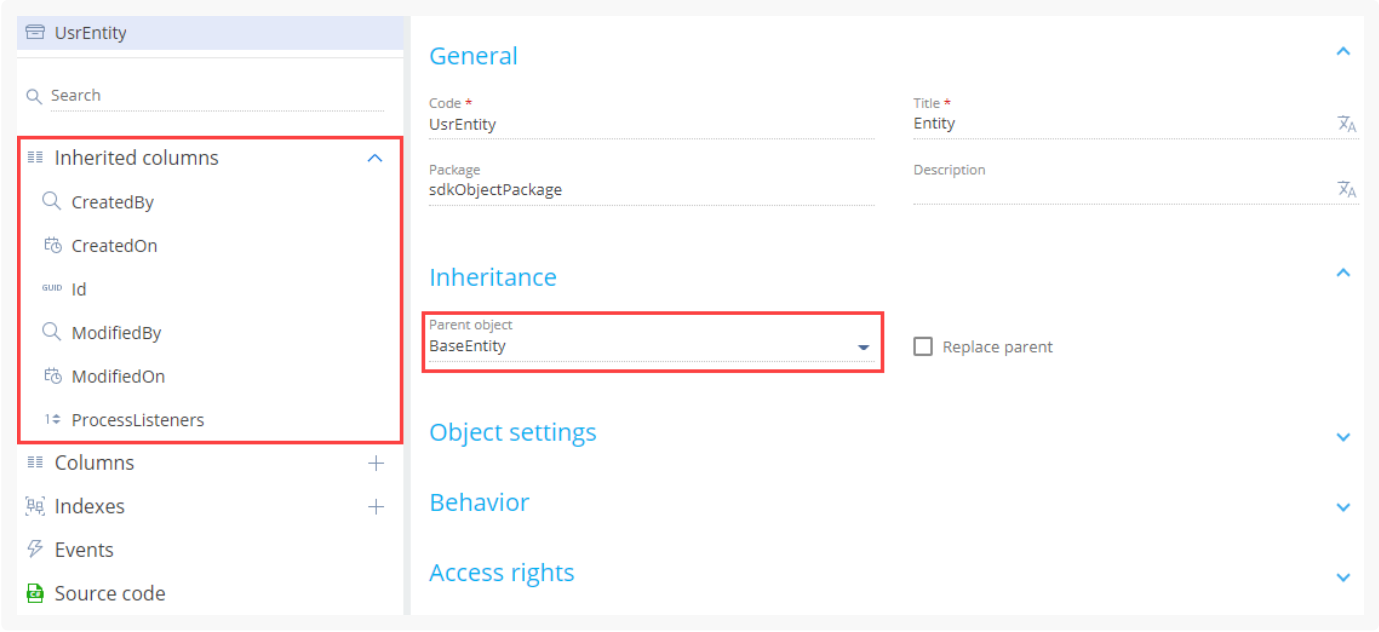
   The **main schema properties** are as follows:

   - [*Code*] is the schema name. Required. Starts with the prefix specified in the [*Prefix for object name*] (`SchemaNamePrefix` code) system setting, `Usr` by default. Can contain Latin characters and digits. The object name can contain up to and including 128 characters. Oracle databases version 12.2 and earlier do not accept objects that have names longer than 30 characters. When a configuration element schema is created, the prefix specified in the [*Prefix for object name*] (`SchemaNamePrefix` code) system setting is added to the current field automatically. Creatio checks for the prefix and whether it matches the system setting value when saving the schema properties. If the prefix is missing or does not match, the user receives a corresponding notification.
   - [*Title*] is the localizable schema title. Required. The title of the configuration element schema is generated automatically and matches the value of the [*Code*] property without the prefix.
   - [*Package*] is the custom package where you create the schema. The property is populated automatically and non-editable.
   - [*Description*] is the localizable schema description.



4. Select the parent object in the Object Designer.

   For the object to inherit the functionality of a different object, select the schema of the object to inherit in the [*Parent object*] property's drop-down list. For example, to inherit the functionality of the `BaseEntity` object's base schema, specify the `BaseEntity` schema as the parent object. Creatio adds columns inherited from the parent object to the [*Inherited columns*] property of the object schema automatically.



5. Select the column that corresponds to the object ID in the Object Designer. [*Id*] is the system column that serves as the primary key in the database table. Required. To do this, follow the guide in a different section: Specify the object ID.
6. Add an object index in the Object Designer (optional). To do this, follow the guide in a different section: Add an object index.
7. Set up cascade connection in the Object Designer (optional) To do this, follow the guide in a different section: Set up the cascade connection.
8. Click [*Publish*] on the Object Designer's toolbar to create a corresponding table in the database.

   You can use the [*Publish*] button to generate the static content and update the database structure. In this case, the configuration is not compiled. This accelerates the development of objects and replacing objects. The object compilation on publishing is required if the embedded process of the object was saved while editing but not published in the Process Designer. To compile the configuration, generate the static content, and update the database structure, select [*Publish and compile*] in the drop-down list of the [*Publish*] button.

Specify the object ID

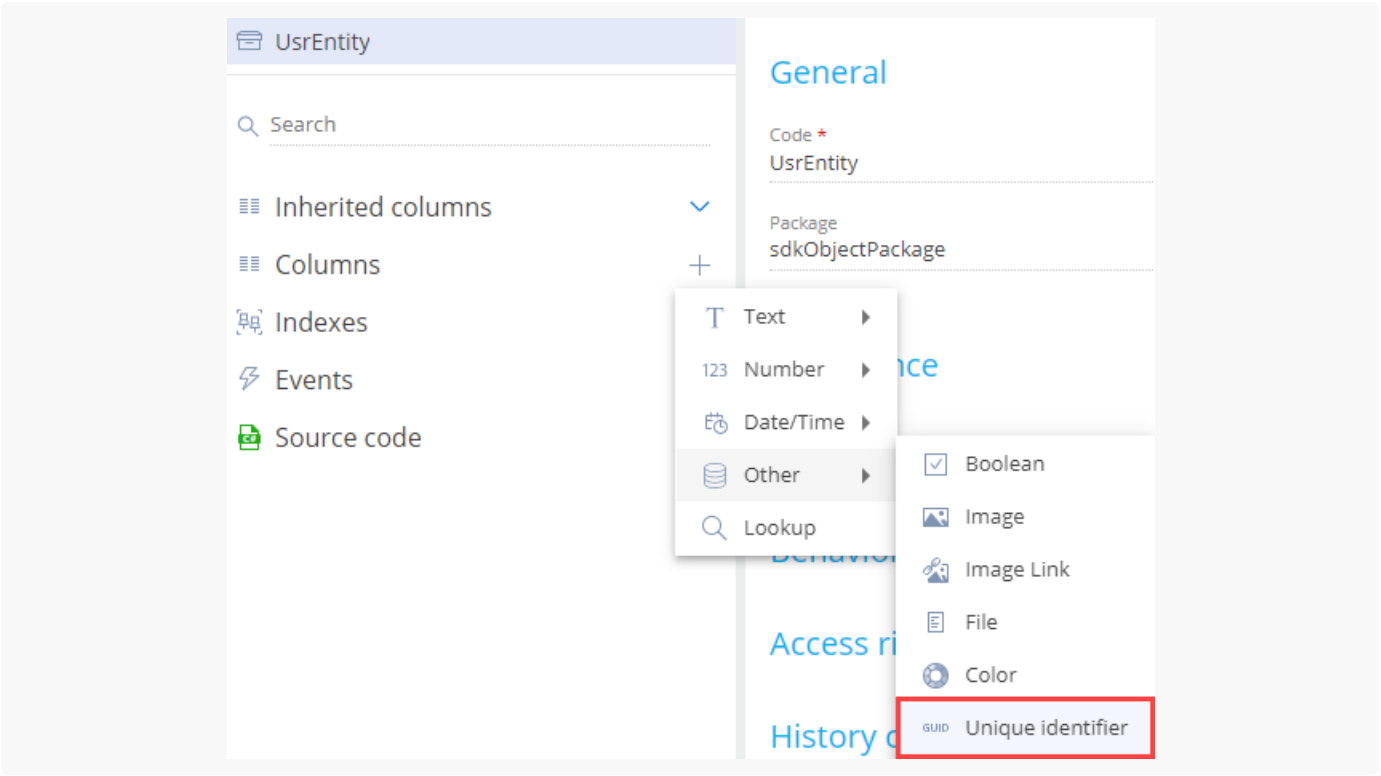
Since a Creatio object is a database table view, it must contain the ID column.

The object ID is specified the following **ways**:

- If the [*Parent object*] property contains a **base object**, Creatio populates the [*Id*] property automatically.
- If the [*Parent object*] property contains a **custom object**, fill out the [*Id*] property manually.

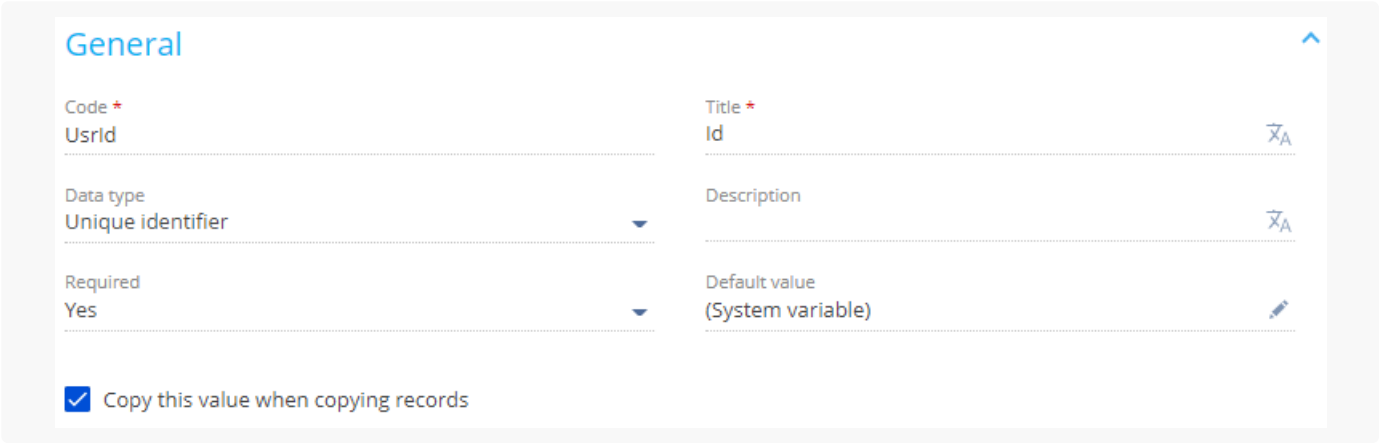
To **add a custom ID**:

1. Click + → [*Other*] → [*Unique identifier*] in the properties area of the [*Columns*] node's context menu.



2. Fill out the column properties in the Object Designer.

   The **main column properties** are as follows:



- [*Code*] is the column name. Required. Starts with the prefix specified in the [*Prefix for object name*] (`SchemaNamePrefix` code) system setting, `Usr` by default. Can contain Latin characters and digits. Make sure that the [*Code*] property is different from the [*Code*] of column's object. Otherwise, Creatio will display an error message when you attempt to publish the object.
- [*Title*] is the localizable schema title. Required.
- [*Data type*] is the column data type. Creatio populates the property automatically depending on the column type selected when adding a column. This is a non-editable field.
- [*Description*] is the localizable column description.
- [*Required*] specifies if the column is required. Select "`Yes`" since the column of the [*Unique identifier*] type is required for the object. If you attempt to save the object schema without the column of the [*Unique*

identifier] type, Creatio displays a corresponding message.

- [*Default value*] is the default column value. To **specify the default value**, follow the guide in a different section: [Specify the default value for the column of the [*Unique identifier*] type](#).
- [*Usage mode*] must be set to "`Advanced`."

  Creatio IDE supports the following column **usage modes**:

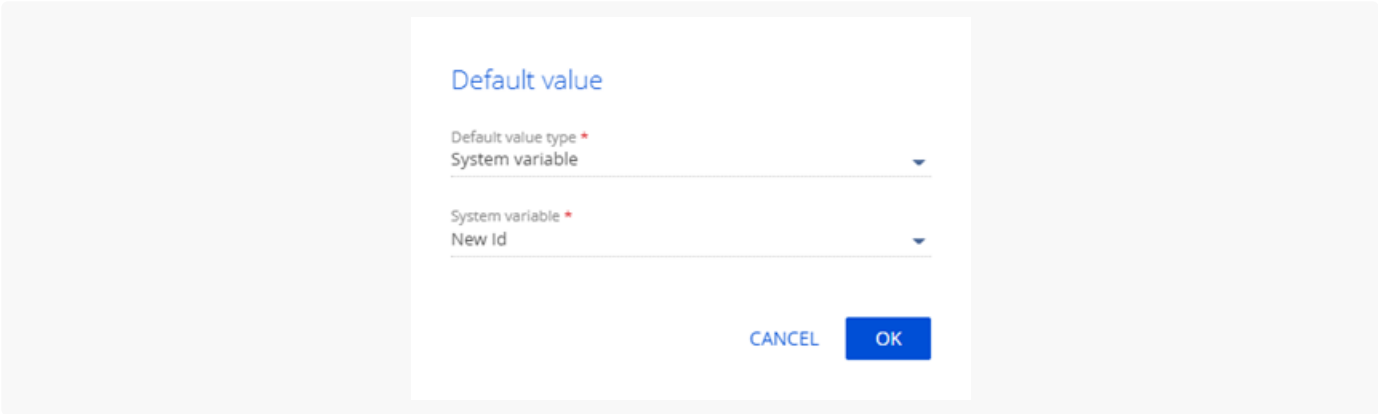
  - [*General*] is the standard column mode in the application.
  - Columns in the [*Advanced*] mode are displayed in the configuration and are available in the application.
  - Columns in the [*None*] mode are displayed as system columns in the configuration and are not available in the application.

Specify the default value for the column of the [*Unique identifier*] type

To **specify the default value** for the column of the [*Unique identifier*] type:

1. Click in the [*Default value*] property.
2. Fill out the **default value properties**:

   - Set [*Default value type*] to "`System variable`."
   - Set [*System variable*] to "`New Id`" since the IDs must be unique.

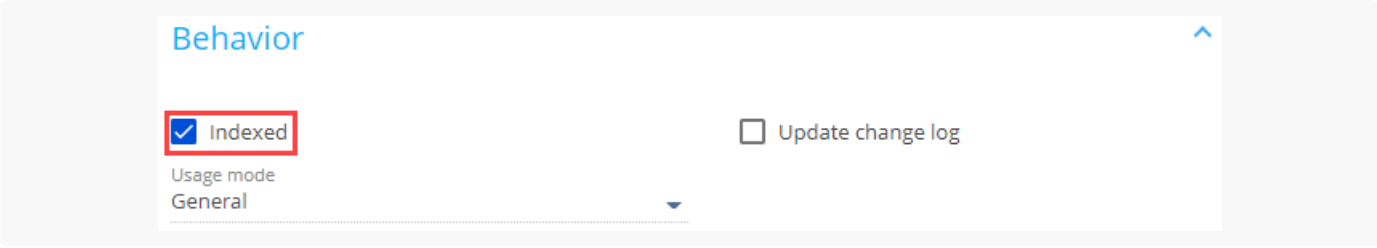


Add an object index

Besides columns, Creatio IDE lets you add indexes to the object. Indexes are created automatically in the database table when you publish an object.

You can add an object index in the following **ways**:

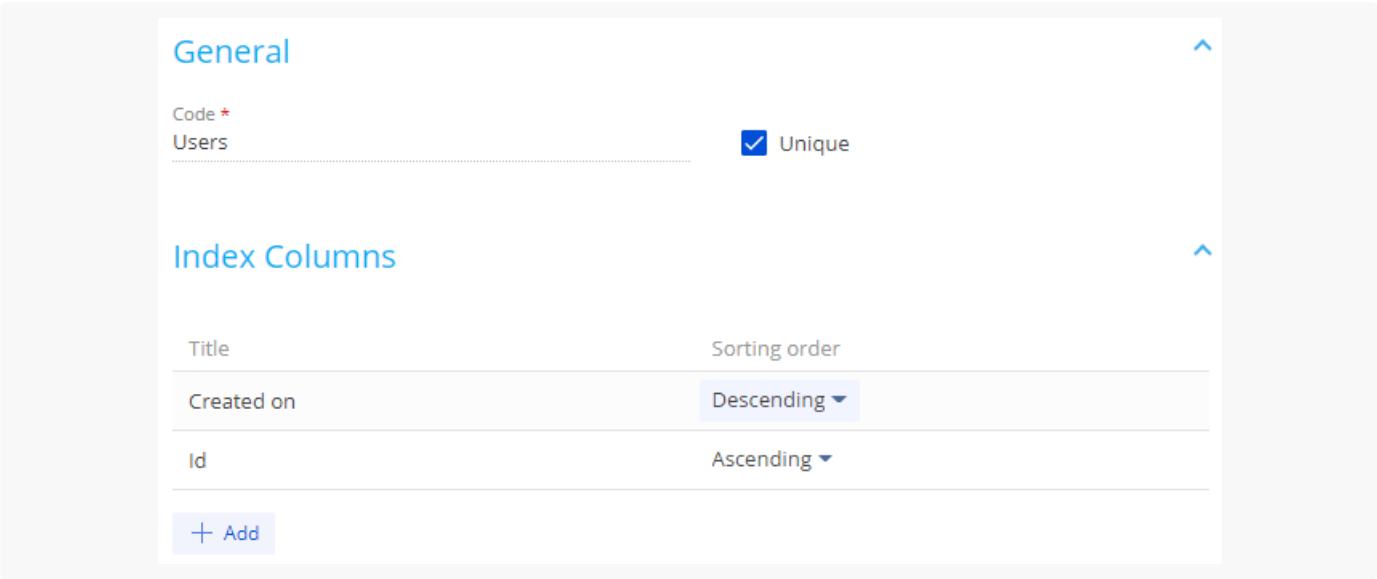
- **Single column index**. In this case, select the [*Indexed*] checkbox in the [*Behavior*] property block. Creatio indexes lookup columns by default.



- **Composite index**. To do this, follow the guide below.

To **add a composite object index**:

1. Click + in the properties area of the [*Indexes*] node's context menu.
2. Fill out the **index properties**:
   - Set [*Code*] to the schema name. Required.
   - Select the [*Unique*] checkbox to enable integrity constraints for index columns, i. e., remove the possibility of adding repeating value combinations.
   - Select the columns to add to the index in the [*Index Columns*] property block. To do this, click [*Add*] in the [*Index Columns*] property block, select an object column, and specify the sorting order.

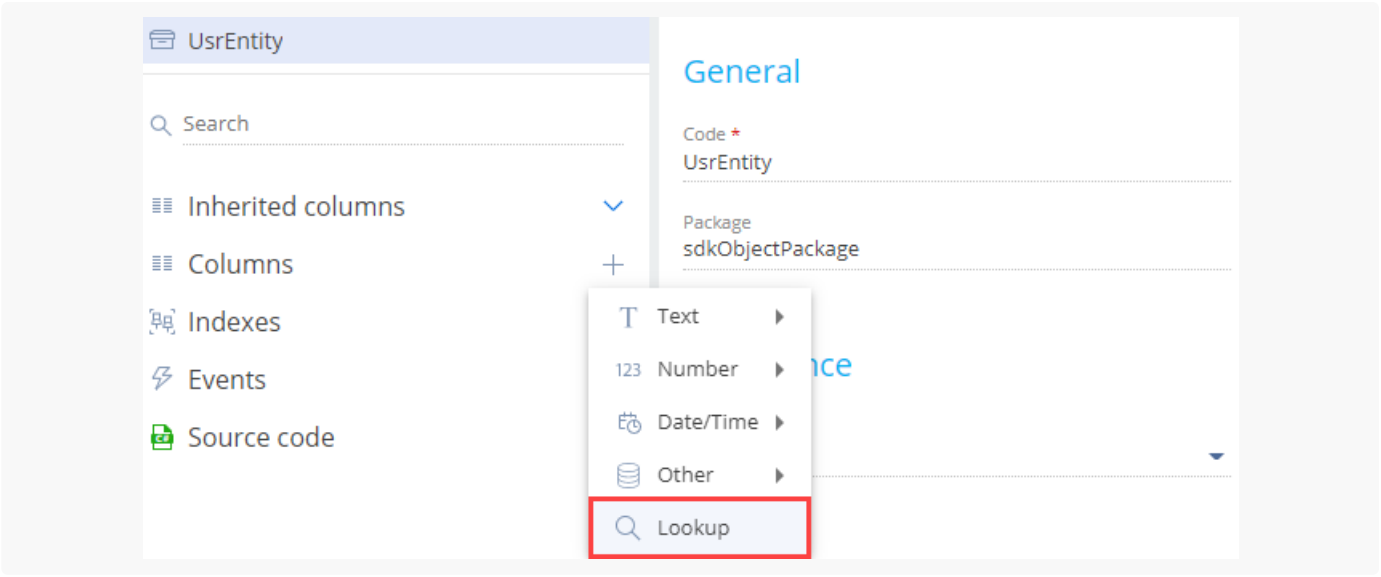


Set up the cascade connection

Creatio IDE lets you set up the cascade connection only for [*Lookup*] type columns.

To **set up the cascade connection**:

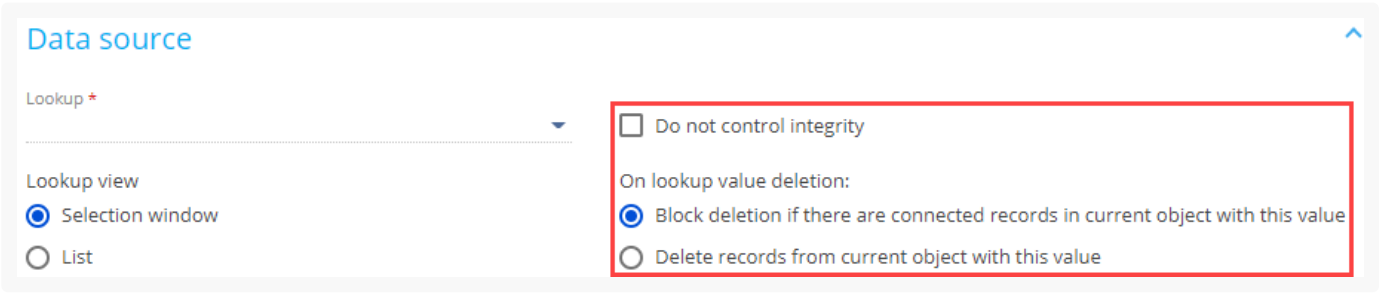
1. Add a [*Lookup*] type column (optional). To do this, click + → [*Lookup*] in the context menu of the [*Columns*] node.



2. Set up the **cascade connection** in the [*Data source*] property block.

   Use the following **controls** to set up the cascade connection:

   - [*Do not control integrity*] checkbox
   - [*On lookup value deletion*] items



For this example, set up a cascade connection to the [*Contact*] object connected to the [*Account*] object via the `[AccountId]` lookup column. To do this, select [*Account*] in the [*Lookup*] field.

The cascade connection **setup options** are as follows:

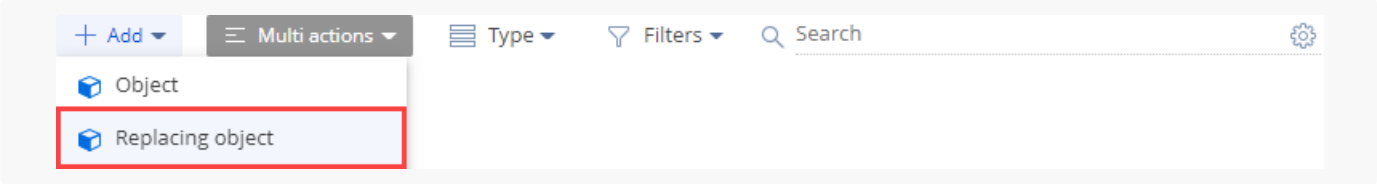
- If you **select** the [*Do not control integrity*] checkbox, Creatio deletes the account without deleting the contacts connected to the account.
- If you **do not select** the [*Do not control integrity*] checkbox and **select** the [*Block deletion if there are connected records in current object with this value*] item, Creatio checks for connected contacts. If Creatio detects them, it displays a warning message asking you to confirm the deletion. Confirm the deletion to delete the account without deleting the contacts connected to the account.
- If you **do not select** the [*Do not control integrity*] checkbox and **select** the [*Delete records from current object with this value*] option, Creatio deletes both the account and connected contacts.

Implement a replacing object

The schema of the [*Replacing object*] **type** represents the replacing object.

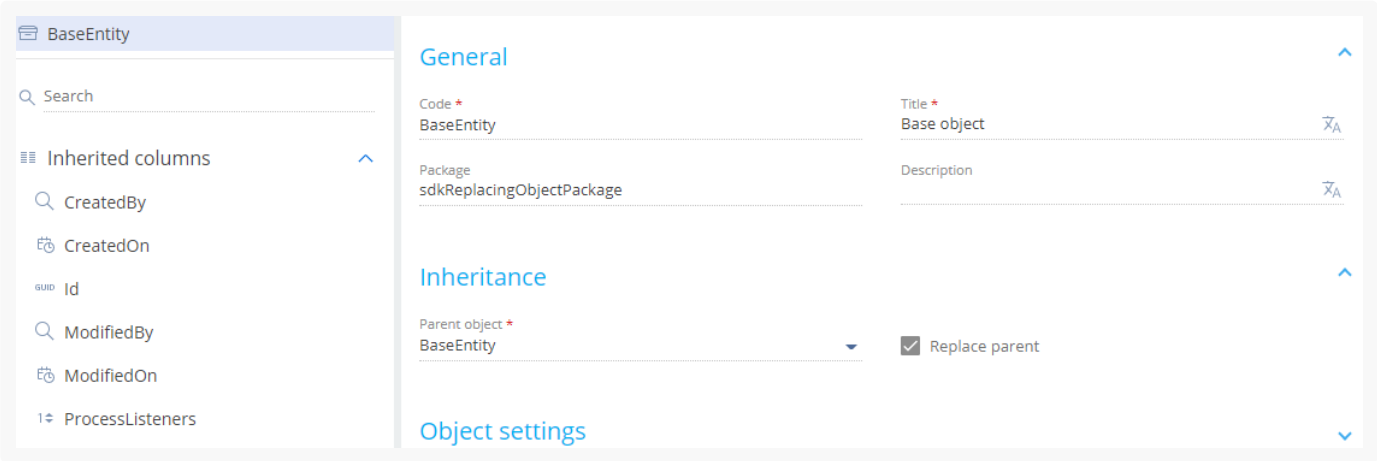
To **implement a replacing object**:

1. Go to the [*Configuration*] section and select a custom package to add the schema.
2. Set up the package dependencies. You must add the package that contains the replaced object to the dependency list.
3. Click [*Add*] → [*Replacing object*] on the section list toolbar.



4. Select the parent object in the Object Designer.

   To replace the base object functionality, specify the object schema to replace in the drop-down list of the [*Parent object*] required property. For example, to replace the functionality of the `BaseEntity` base object schema, specify the `BaseEntity` schema as the parent object. Creatio adds columns inherited from the parent object to the [*Inherited columns*] property of the object schema automatically. After you select the parent object, Creatio populates other object properties automatically.



5. Implement the functionality that distinguishes the replacing object from the replaced object in the Object Designer.
6. Click [*Publish*] on the Object Designer's toolbar to create a corresponding table in the database.

   You can use the [*Publish*] button to generate the static content and update the database structure. In this case, the configuration is not compiled. This accelerates the development of objects and replacing objects. The object compilation on publishing is required if the embedded process of the object was saved while editing but not published in the Process Designer. To compile the configuration, generate the static content, and update the database structure, select [*Publish and compile*] in the drop-down list of the [*Publish*] button.

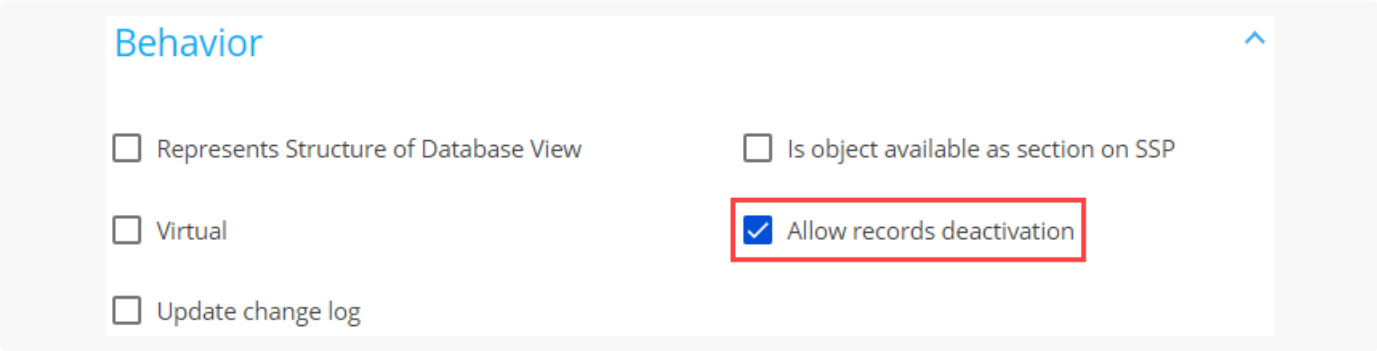
Learn more about replacing configuration elements in a separate article: Replace configuration elements.

Deactivate object records

Creatio lets you deactivate object records to exclude them from the business logic. For example, this can be useful if the data is outdated and no longer used. You can deactivate records of any object.

To **deactivate object records**:

1. Open the schema of the object whose records to deactivate.
2. Select the [*Allow records deactivation*] checkbox in the [*Behavior*] property block.



3. Click [*Publish*] on the Object Designer's toolbar.

Creatio can **filter inactive records** automatically for certain UI elements.

The **automatic record filtering** is available for the following UI elements:

- drop-down list
- lookup value selection box
- quick filter

The **automatic record filtering** is not available for the following UI elements:

- lookup content page
- section
- advanced filter

The `UseRecordDeactivation` parameter of the `EntitySchemaQuery` class lets you manage the filtering of inactive records. The default value is `false`. If you set the `UseRecordDeactivation` parameter to `true`, the select query to the object that has record deactivation enabled will contain the filter that excludes inactive record.

View the example that deactivates records in the front-end below.

Example that deactivates records (front-end)

```
var esq = Ext.create("Terrasoft.EntitySchemaQuery", {
   rootSchemaName: "MyCustomLookup",
   useRecordDeactivation: true
});
```

View the example that deactivates records in the back-end below.

Example that deactivates records (back-end)

```
var esq = new EntitySchemaQuery(userConnection.EntitySchemaManager, "ContactType") {
    UseRecordDeactivation = true
```

```
};
esq.PrimaryQueryColumn.IsAlwaysSelect = true;
```

View the resulting SQL query below.

Example of the SQL query

```
SELECT [ContactType].[Id] [Id]
FROM [dbo].[ContactType] [ContactType] WITH(NOLOCK)
WHERE [ContactType].[RecordInactive] = 0
```