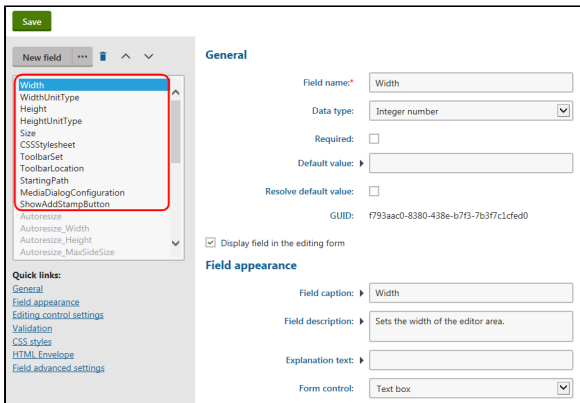


Parameters of [form controls](#) allow users to specify the behavior or appearance of the form fields generated by the controls. Parameters greatly increase the flexibility of form controls by providing configuration options for individual fields that use the control. Parameters allow you to reuse the same control without having to develop a new one for every small difference in functionality or design.

To access the parameters of a form control:

1. Open the **Form controls** application.
2. Edit (✎) the form control.
3. Switch to the **Properties** tab.






The screenshot shows the 'Form controls' application interface. On the left, a list of parameters is displayed, with 'Width' selected and highlighted by a red box. The main area is divided into two tabs: 'General' and 'Field appearance'. The 'General' tab is active, showing configuration options for the 'Width' parameter. The 'Field name' is 'Width', the 'Data type' is 'Integer number', and the 'Required' checkbox is unchecked. The 'Default value' is empty, and the 'Resolve default value' checkbox is also unchecked. The 'GUID' is 'f793aac0-8380-438e-b7f3-7b317c1cfd0'. The 'Display field in the editing form' checkbox is checked. The 'Field appearance' tab is visible below, showing 'Field caption' as 'Width', 'Field description' as 'Sets the width of the editor area.', 'Explanation text' as empty, and 'Form control' as 'Text box'.

You can manage the fields that represent individual parameters.

Create the parameters through the [field editor](#). The form used to edit the parameter values is based on the standard form engine. The value editing interface of every parameter is provided by a selected form control, which allows you to create completely custom parameters according to your own requirements.

The system displays the defined parameters when you select the given form control in the field editor of an object. You can view and configure the parameters under the **Editing control settings** section of the field.

Save

New field ...   

NewsID*

NewsTitle*

NewsReleaseDate*

NewsSummary*

NewsText

NewsTeaser

Page name source field:

NewsTitle

Page alias source field:

(Page name)

Quick links:

[General](#)

[Field appearance](#)

[Editing control settings](#)

[Validation](#)

[CSS styles](#)

[HTML Envelope](#)

[Field advanced settings](#)

Field appearance

Field caption: News Summary

Field description:

Explanation text:

Form control: Rich text editor

Editing control settings

Simplified

Width:

Width unit type: px

Height:

Height unit type: px

Maximum text length (without HTML tags):

CSS stylesheet: (default)

Toolbar set:

Toolbar location:

Starting path:

Media dialog configuration: [Configure](#)

Show Add stamp button:

There are two possible modes for this group of settings: *Advanced* and *Simplified*. You can switch between them using the link in the given section of the field editor. Simplified mode only offers the parameters that have the **Display in simple mode** option enabled (on the **Properties** tab of the given form control under the **Field advanced settings** section).

Warning!

Removing or modifying the parameters of the built-in form controls may cause the system to behave incorrectly in some cases, since the controls are used in various parts of the Kentico interface.

If you wish to create custom parameters, we recommend implementing a new form control or [inheriting](#) from an existing control.

<https://docs.xperience.io>

2

Parameters of inherited form controls

When editing [inherited](#) form controls on the **Properties** tab, the system automatically loads the parameters from the parent control.



You can:

- Set the **Default value** for parameters
- Determine whether parameters are visible in the field editor (**Display field in the editing form**)
- Modify the appearance and behavior of parameters
- Add additional parameters or categories to the inherited form control

You cannot:

- Remove inherited parameters
- Change the **Field name**, **Data type** or **Size** of inherited parameters

Inheritance allows you to create specialized form controls that may easily be used in the field editor without any configuration of their **Editing control settings**.

Handling parameters in the code of custom form controls

In order to be reflected by the field in the resulting form, each parameter must be processed in the code of the form control. The following methods can be used to access the value of a parameter:

- `GetValue("ParameterName")`
- `SetValue("ParameterName", Object value)`

Replace the *ParameterName* with the **Field name** of the parameter that you wish to get or set. These two methods are inherited from the **FormEngineUserControl** class, so they are available for every form control. For easy storage and usability, it is recommended to define a separate property for every parameter (as a member of the control's class). Once you have the value of the parameter, you can implement code that modifies the control accordingly.

See also: [Example - Developing custom form controls](#)