

The **Form controls** application allows you to view the catalog of available form controls. Click **New form control** to register custom form controls.

When you edit (✎) a form control, the following properties are available on the **General** tab:

| General | |
|---------------|--|
| Display name | Sets the name of the form control used in the administration interface, for example when selecting form controls in the field editor. |
| Code name | Serves as a unique identifier of the form control (for example in the API). |
| Type | Assigns the form control to a category. The type can be used to filter form controls. |
| Inherit from | Only available for inherited form controls . Displays the name of the parent form control. |
| Description | A text description of the control or comments for other administrators. |
| Thumbnail | An image file that represents the form control if it is displayed in the Form builder . |
| High priority | <p>When selecting a form control in the field editor, controls with high priority appear at the top of the list and are highlighted.</p> <p>Additionally, only high priority form controls are included among the form controls offered in the Form builder.</p> <p>Only enable this option for form controls that are most commonly used.</p> |

Control source



Note: The *Control source* properties are not available for [inherited form controls](#), which always use the same source as their parent.

The *Control source* properties determine how the form control is implemented. The following options are available:

- **Web user control** – a user control file in the main Kentico web project. The **File name** property must contain the relative path of the .ascx file. For example: ~/CMSFormControls/Basic/TextBoxControl.ascx
- **Assembly class** – a control class compiled in an assembly, specified by the **Control class** properties:
 - **Assembly name** – the name of the assembly where the control class is implemented.
 - **Class** – the exact class (including any namespaces) that defines the content and functionality of the form control.



Control source recommendations

When adding custom form controls, we recommend that you consider the source options in the following order:

1. [Inherited control](#) – allows you to add a specialized control based on one of the existing form controls (without needing to write custom code in most cases).
2. Assembly class – recommended in general for non-inherited form controls. Adding the form control class into a custom assembly provides cleaner separation of your code from the default Kentico web project, and does not require runtime compilation which can slow down the website.
3. Web user control – use for controls with extensive ASCX markup, or if you need a form control without dependencies on a custom code assembly.

Control extender

Extenders are only available for [inherited form controls](#).

By creating an extender class, developers can adjust the functionality of an inherited form control beyond the configuration options provided by the parent's [parameters](#). The extender adds custom code that runs in addition to the code of the parent control. For example, you can use extenders to dynamically set the form control's properties or assign handler methods to events that occur during the control's life cycle (e.g. form validation).

To add an extender, you need to specify the extender class using the corresponding properties:

- **Assembly name** – the name of the assembly where the extender is implemented.
- **Class** – the exact class (including any namespaces) that defines the extender's functionality.

Control scope

Use the properties in the **Use control for** section to choose the data types for which the form control is available. Possible data types for fields include standard options (Text, Long text, Decimal, Integer, Long integer, Boolean, Date-time), as well as several other options:

| | |
|--------------------------|--|
| Unique identifier (GUID) | Fields that store 32character strings used as globally unique identifiers for objects. |
| File | Fields that provide file management. |
| Page attachments | Fields that allow the management of page attachments. |
| Visibility | For form controls that modify the visibility options of other fields. See Allowing users to change the information they share for more details. |

The **Show control in** determines in which types of forms the control can be used:

| | |
|------------------------|---|
| Page types | Allows use in page type fields, which you can edit in Page types -> edit page type -> Fields . The editing forms are displayed when editing pages on the Form tab in the Pages application. |
| Forms | Allows the control to be used for the fields of forms , which you can edit in the Forms application. If you wish to make the form control available in the Form builder , you need to enable the High priority flag. If you allow a form control for forms, you also need to enter the following default values: <ul style="list-style-type: none"> • Default data type – the data type of the field used by default when a user chooses to create a new field of this type. • Column size – only applies to the <i>Text</i> data type. Sets the maximum size of the database column used to store the field, which also limits number of characters that may be entered into the field. |
| Custom tables | Allows the control in custom table fields, which you can edit in Custom tables -> edit custom table -> Fields . The corresponding data editing form is displayed in the Custom table data application. |
| System tables | Allows use in the fields of system objects. You can edit the fields in Modules -> Edit module -> Classes -> Edit class -> Fields . The editing forms of system objects are located in various sections of the administration interface or on the live site. |
| Reports | Allows the control for the user interface of report parameters , which you can define in Reporting -> edit a report -> Parameters . The reporting parameter form is displayed when viewing reports in the administration interface or on the live site. |
| Controls and web parts | Allows the control to be used for the properties of web parts and widgets , as defined in the Web parts or Widgets applications. This editing form provides the configuration dialog of individual web parts or widget instances. |

Defining form control parameters

On the **Properties** tab of the form control editing interface, you can define parameters. See [Defining form control parameters](#) for further information.

Previewing form controls

When editing form controls, you can use the **View** tab to check the appearance of the control in a sample 1-field form.

Finding where form controls are used

The **Usage** tab of the form control editing interface allows you to find all objects in the system that use the given control in their form. You can use the **Object type** filter to narrow down the list to specific types of forms (Page type editing forms, Web part property definitions, etc.).