

All pages on Kentico websites, with the exception of pages based on [content only](#) page types, are based on [page templates](#). Templates define the basic structure and components of pages without the content added by editors.

To access the main management interface for all page templates in the system, open the **Page templates** application. The system organizes templates into categories. You can edit individual templates by selecting them in the category tree.

To edit the properties of the template assigned to a specific page in the **Pages** application:

1. Select the page in the content tree
2. Open the **Properties -> Templates** tab.
3. Click **Edit template properties**.

– OR –

1. Open the page's **Design** tab.
2. Right-click the green template header and click **Edit template**.

When editing templates, you can set the following page template properties on the **General** tab:

Page template property	Description
Template display name	The name of the template displayed to users in the administration interface.
Template code name	A unique identifier of the template.
Category	Sets the category where the template is stored in the page template catalog.
Template description	Here you can enter a description for the template. Users can see this description when selecting page templates in the catalog (for example when creating new pages).
Thumbnail	<p>Allows you to set the image that represents the template in the page template selection catalog.</p> <p>You can choose between two types of images:</p> <ul style="list-style-type: none"> <li>• <b>Image</b> - upload a standard image file (for example a png).</li> <li>• <b>Font icon class</b> - select a CSS class that defines a <a href="#">font icon</a>.</li> </ul>
Clone as ad-hoc for new pages	<p>Only available if the <b>Template type</b> is <i>Portal page</i>.</p> <p>When a user creates a new page with a template that has this option enabled, the system automatically creates an ad-hoc copy of the template and assigns it to the page. This allows users to immediately modify the design of the new page without affecting the default re-usable template.</p>

Template type	<p>The following types of page templates are available:</p> <ul style="list-style-type: none"> <li>• <b>Portal page</b> - functions as a <a href="#">portal engine</a> page template.</li> <li>• <b>ASPX page</b> - uses the <a href="#">ASPX page template</a> development model. The content is based on a standard ASPX web form. If selected, you need to specify the path to the source file in the <b>File name</b> property.</li> <li>• <b>ASPX + Portal page</b> - works the same way as the <i>ASPX page</i> option, but also supports portal engine functionality — managing web parts or widgets in predefined zones on the <i>Design</i> tab of the <i>Pages</i> application.</li> <li>• <b>Dashboard page</b> - provides a template for <a href="#">widget dashboard</a> sections of the administration interface. This type of page template cannot be used for standard content tree pages.</li> <li>• <b>MVC</b> - template for creating pages defined through an MVC controller and action. Only choose this option when using the obsolete CMSApp_MVC project to develop your MVC sites. You need to specify the controller and action via the <b>Default controller</b> and <b>Default action</b> properties. Pages based on MVC templates automatically use the live URL and view in editing mode (i.e. on the <i>Page</i> tab of the <i>Pages</i> application).</li> <li>• <b>UI page</b> - functions as a <a href="#">portal engine</a> template for pages of the Kentico administration interface. You cannot assign UI page templates to normal website pages, only to <a href="#">UI elements</a>.</li> </ul>
Master template	<p>Only available if the <b>Template type</b> is <i>Portal page</i>.</p> <p>Indicates if the pages that use the template are designated as <a href="#">Master pages</a>. Enabling this option also causes the template to be selectable as the root master page template in the <b>New site wizard</b>.</p>
Page nesting	<p>Only available if the <b>Template type</b> is <i>Portal page</i>.</p> <p>Determines how pages based on the template display their content inside ancestor pages in the content tree.</p> <ul style="list-style-type: none"> <li>• <b>All ancestor pages</b> - nests within all ancestor pages in the content tree up to the nearest master page.</li> <li>• <b>None</b> - behaves as a standalone page without any nesting.</li> <li>• <b>Only the nearest master page</b> - nests only within the website's master page. If the website uses multiple master pages, the pages based on this template nest only within the closest master page in the content tree hierarchy.</li> <li>• <b>Specific content tree levels</b> - allows you to enable or disable nesting within specific levels of the content tree (represented by the checkboxes below). The actual nesting depends on the structure of the website where the template is used.</li> </ul> <p><b>See:</b> <a href="#">Inheriting portal engine page content</a></p>
File name	<p>Only available if the <b>Template type</b> is <i>ASPX page</i> or <i>ASPX + Portal page</i>.</p> <p>Specifies the path to the <b>.aspx</b> file that serves as the page template's source. You can either choose the using the <b>Select</b> button, or enter the path manually. The tilde character (~) represents the root directory of the project folder, e.g. <code>~/CMSTemplates/CorporateSite/Blog.aspx</code></p>
Default controller	<p>Only available if the <b>Template type</b> is <i>MVC</i>.</p> <p>Sets the name of the controller class containing the MVC action that the system performs when pages using this template are accessed.</p> <p>Do not type the <i>Controller</i> suffix of the class name. For example, if the class is called <i>NewsMVCController</i>, enter <i>NewsMVC</i>.</p> <p>The system first searches for the specified class in the <i>CMS.Controllers.&lt;current site code name&gt;</i> namespace. If the class cannot be found there, the <i>CMS.Controllers.Global</i> namespace is searched.</p>
Default action	<p>Only available if the <b>Template type</b> is <i>MVC</i>.</p> <p>Specifies the exact action defined within the controller class that the system performs when loading pages based on the template.</p>

## Limiting the use of page templates

When editing page templates in the **Page templates** application, you can:

- Assign templates to specific websites on the **Sites** tab. Users can only work with the template on the selected websites.
- Add further limitations through *template scopes* (by default, page templates can be used anywhere on the websites to which they are assigned).

Scopes simplify the user experience by reducing the number of templates available for selection in specific parts of the website. To define scopes for a page template:



**Important:** Template scopes are *not* a security feature. After assigning a template to a page, users can move the page to a location that is not allowed by the given template's scopes.

1. Open the **Scopes** tab of the template's editing interface.
2. Select **Template can be used only within the following scopes**.
3. Choose the **Site** where the scope will apply.
  - You can only choose the sites to which the template is assigned.
  - Global scopes are valid on all sites in the system.
4. Click **New template scope**.
5. Set the parameters of the scope:
  - **Starting path** - defines a sub-section of the website where the template can be used. Allows the exact page with the entered path and all of its descendants (subpages).
  - **Page type** - allows the template only for pages of the selected [type](#).
  - **Culture** - allows the template for page versions in the selected [culture \(language\)](#).
  - **Levels** - you can allow the template only for pages on specific levels in the content tree hierarchy.
6. Click **Save**.

You can add any number of scopes for each template. Users can select the template for pages that fulfill the requirements of **at least one** of the scopes.

## Defining head content for templates

You can add custom head content for page templates, such as links to external CSS stylesheets or JavaScript files.

1. Open the page template's editing interface.
2. Switch to the **Header** tab.
3. Type in the required head content.
4. Click **Save**.

The system inserts the content into the `<head>` element in the output code of pages that use the given template. Note that head content is only rendered on pages whose [page type](#) has the **Behaves as Page (menu item) type** property enabled.

For pages that use [portal engine templates](#), the head content can also be inherited by subpages. The inheritance depends on the following options, which you can enable on the **Header** tab of each page template:

- **Allow descendant templates to inherit the header** - if enabled, the head content of the template can be inherited by pages on lower levels in the content tree.
- **Inherit headers from the templates of ancestor pages** - indicates if pages based on the template are allowed to inherit head content from the templates of pages that are above them in the content tree.





**Note:** The header inheritance works according to the [page nesting](#) settings of pages — pages can only inherit head content from ancestor pages where they are nested. This way, the system only adds the head content to pages that actually display the page template containing the header definition.

## Finding which pages use a given template

To view a list of all pages in the system that use a specific page template, open the **Pages** tab of the template's editing interface.

You can access individual pages in the list through the following actions:

-  **Edit page** - opens the page in the **Edit** mode of the **Pages** application.
-  **Navigate to page** - opens the page on the live site.

## Editing the XML source of a template's web parts

The system stores the content and configuration of [web parts](#) placed on page templates as XML data. If you cannot use the **Design** tab due to an error caused by an incorrectly configured web part, you can fix the issue on the **Web parts** tab of the given page template's editing interface. The tab allows you to edit the XML source directly.

- You can modify the values of properties for all web parts on the template.
- You can remove web parts from the template by deleting entire `<web part>` elements.