



Content only page types don't have a presentation URL by default. This means that content editors cannot access content only pages in the **Preview** mode or when opening them on the live site via the administration UI. Content editors also aren't able to create page links using the [text editor](#). If you want your content only pages to support this functionality, you need to specify the **URL pattern** property of their page type. Specify the property when:

- Using a standalone [MVC application](#) which handles the presentation of your content only pages.
- Using the portal engine and make use of content only page types.

You will usually only want to specify the property for pages with a separate presentation. For example, pages that provide information that is only used when aggregated on a different page may not need to have their own *URL pattern* specified. Examples of these pages could be individual branches/offices/cafes of a company that you use on an aggregate page that displays all the branches/offices/cafes in one place.

Specifying the URL pattern property

1. If your live site runs on a different domain than the administration instance (e.g. for [MVC sites](#)), make sure your site's [Presentation URL](#) is specified.
2. Open the **Page types** application.
3. Edit a content-only page type.
4. Fill in the **URL pattern** under which you want the page to be accessible from the administration UI. You can use simple macro expressions for accessing [data of the current page](#) in the pattern, such as `{%NodeGUID%}`, `{%DocumentModifiedWhen.Month%}`, or `{%NodeAlias%}`. The `{%NodeAlias%}` macro retrieves a page alias that allows you to create more SEO-friendly URLs. For example:

```
/[%DocumentCulture%]/Articles/{%NodeAlias%}
```

resolves into URL `'http://www.example.com/en-US/Articles/On-Roasts'` for a page in the 'en-US' culture with *Node alias* 'On Roasts'.



Notes:

- Complex macro expressions including [macro methods](#) are not available in the URL pattern.
- We do not recommend using query string parameters in URL patterns. The system removes any parameters when generating *Preview* URLs for pages based on the pattern.

5. Click **Save**.

Editors can now access the pages of this type from the administration UI using the **Live site** button. They can also insert links to the pages using the [text editor](#).

The system may add a parameter to the resulting URL when displaying pages. For example, the *viewmode* parameter is added to the URL when you open a page on live site via the administration UI. The resulting URL may then look like this:

```
http://www.example.com/en-US/Articles/On-Roasts?viewmode=0
```

Adding support for preview mode

After you set the URL pattern for your content only page types, the system can automatically generate preview URLs for the given pages. You can use the preview mode in two ways within the **Pages** application:

- Switch to **Preview** mode using the main view mode selector.
- Generate preview links for pages in **Properties -> General -> Preview URL**, which you can then send to other users. See: [Sending links to unpublished pages](#)

**Note:**

Preview mode is not supported if you are using content only page types in combination with the portal engine and standard pages. You can instead use the preview of the pages that load and display the given content.

When developing an application that handles the presentation of content only pages (for example a [standalone MVC application](#)), you need to manually ensure that the preview mode displays the latest versions of pages. For more information, see [Adding preview mode support for MVC sites](#).

The preview URLs that the system generates for content only pages can be very long (they contain information about the virtual context of the page and validation hashes). We recommend increasing the maximum allowed URL length for the application used to present the content.

1. Edit the web.config of the application.
2. Set the **maxLength** attribute of the **httpRuntime** element in the **system.web** section (512 should be a large enough value):

```
<system.web>
...
    <httpRuntime targetFramework="4.6" relaxedUrlToFileSystemMapping="true"
requestValidationMode="2.0" maxLength="512" />
...
</system.web>
```

The application can now correctly process the preview URLs generated by the system.