

[Attachments](#) represent files, for example images, stored in [page type](#) fields. Attachments can be stored in the following [data types](#) of page type fields:

- **Attachments** – multiple attachments in a single page type field.
- **File** – a single page attachment.

For working with page attachments in MVC applications, we recommend using the API of the classes that you [generate](#) for individual page types. The generated code uses the **Attachment** type (available in the *CMS.DocumentEngine* namespace) for *File* fields, and **IEnumerable<Attachment>** for *Attachments* fields.

The generated API gives you more control than the standard attachment API when creating URLs for page attachments. You can create strongly typed views, and the provided methods for creating page attachment URLs is independent of Kentico's settings. That is, the resulting URL is always in the same format and Kentico's image resizing settings are not applied.



To work with page attachments, you need to install the **Kentico.Content.Web.Mvc** [integration package](#) into your MVC project.

### Example of working with page attachments using [generated](#) page type code

```
// Retrieves an article
Article article = ArticleProvider.GetArticle(nodeGuid, "en-US", "MySite");

// Accesses a property of an attachment stored in the article's 'Teaser' field (File data type)
int teaserImageWidth = article.Fields.Teaser.ImageWidth;

// Iterates over attachments in the article's 'Images' field (Attachments data type)
foreach (Attachment image in article.Fields.Images)
{
    string imageMimeType = image.MimeType;
    ...
}
```

## Creating page attachment URLs

1. [Perform the initial set up](#) of your MVC application.
2. Call the **GetPath** extension method (available in the *Kentico.Content.Web.Mvc* namespace) to retrieve page attachment URLs:

```
string attachmentUrl = article.Fields.Teaser.GetPath();
```



### Image variants

To get URLs for [image variants](#) of page attachments, you need to specify the image variant definition identifier when calling the *GetPath()* method.

```
string attachmentUrl = article.Fields.Teaser.GetPath("Mobile");
```

The result is a relative URL in the following format:

```
~/getattachment/<file GUID>/<filename>
```

### Special characters

The system replaces all special characters in file names with a '-' (hyphen). This includes characters that are not 0-9, a-z (Latin lower case 'a' to 'z'), '.' (full stop) or '\_' (underscore).

Because the returned URLs are relative, you need to resolve them for your website by calling the **ImageUrl** or **FileUrl** extension method in your views.

#### Example

```

```

See also:

- [Attachment versioning](#)
- [Constraining the returned attachment size for images](#)
- [Prompting users for action when accessing attachments](#)

## Attachment versioning

The *GetPath()* extension method always returns the attachment that is attached to the version of the page you are currently working with. For example, using the *GetPath()* extension method on pages retrieved via a [generated provider class](#) retrieves the attachment belonging to the published version of the page (the providers work with published versions of pages by default).

## Constraining the returned attachment size for images

You can use the **Url.Kentico().ImageUrl** method to resize image attachments:

- Specify the maximum height of the retrieved image. This constraint retains the original aspect ratio.

```
Url.Kentico().ImageUrl(attachmentRelativeUrl, SizeConstraint.Height(200))
```

- Specify the maximum width of the retrieved image. This constraint retains the original aspect ratio.

```
Url.Kentico().ImageUrl(attachmentRelativeUrl, SizeConstraint.Width(200))
```

- Specify the maximum width and height that the retrieved image cannot exceed. This constraint retains the original aspect ratio.

```
Url.Kentico().ImageUrl(attachmentRelativeUrl, SizeConstraint.MaxWidthOrHeight(400))
```

- Specify the exact width and height of the retrieved image. This constraint does **not** retain the original aspect ratio. The resized image is never made larger than the original size.

```
Url.Kentico().ImageUrl(attachmentRelativeUrl, SizeConstraint.SizeConstraint.Size(400, 1200))
```

## Prompting users for action when accessing attachments

You can use the **Url.Kentico().FileUrl** method to create links that prompt users to save an attachment file to their disk:



```
@using Kentico.Content.Web.Mvc

@{
    var urlOptions = new FileUrlOptions() { AttachmentContentDisposition = true };
}


```