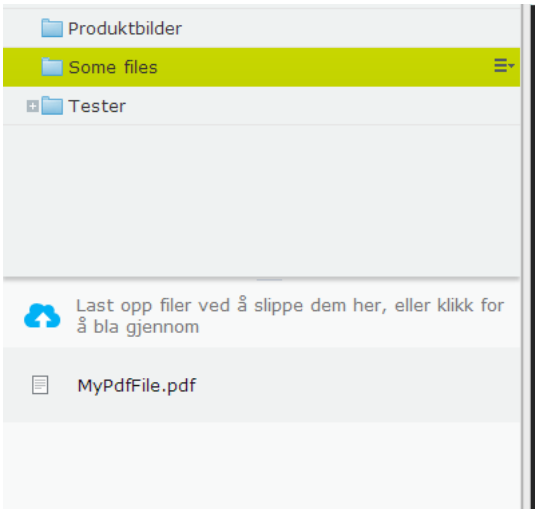
Custom icons for content types

A normal convention when listing files is to give them different icons, to easily distinguish between the different types. EPiServer's new media system does not do this by default, but there's a simple fix for this.

by: Arve Systad 05 May 2014

Usually when you're uploading files to the new media system, you will get thumbnails for images and video, and just a default icon for other file types. This can be somewhat confusing for editors that has a lot of different files (Word-files, Excel spreadsheets, PDFs or the like). It will typically look like this:



Produktbilder
Some files
Tester
Last opp filer ved å slippe dem her, eller klikk for å bla gjennom
MyPdfFile.pdf

In CMS7 and onwards, one can use a `UIDescriptor` to describe the edit mode presentation of a content type. For example, one can set the `IconClass` property, which is what we're going to do here. However, I didn't want to make one implementation of the logic for each and every media type I have. I'd rather rely on a super simple convention. My Media classes will be named `<Extension>File`, for instance `PdfFile`. This makes it super simple to add more icons when needed.

So first, my generic UIDescriptor:

```
public class FileIconDescriptor<T> : UIDescriptor<T> where T : ContentData
{
    public FileIconDescriptor()
    {
        Type type = GetType();
        string fileTypeName = type.BaseType.GetGenericArguments()[0].Name;
        IconClass = fileTypeName.Replace("File", "").ToLower() + "Icon";
    }
}
```

view raw gistfile1.cs hosted with ♥ by **GitHub**

Well, that was boring. Nothing really happened. But at least everything's ready to use. You now have to add an actual descriptor for each of your types - and I will be using our beloved PDF files as an example. So I added this (I keep it in the same file as the generic type to keep reduce the amount of near empty files):

```
[UIDescriptorRegistration]
public class PdfFileDescriptor : FileIconDescriptor<PdfFile> { }
```

view raw gistfile1.cs hosted with ♥ by **GitHub**

Along with this, of course, is a super simple media class called `PdfFile` like shown below. If you now inspect the markup in the media browser, you might see that your files has gotten a `span` with a class pdfIcon on it.

```
[ContentType(GUID = "34D69D83-83DF-4739-9132-8805BBC82196")]
[MediaDescriptor(ExtensionString = "pdf")]
public class PdfFile : MediaData
{
}
```

view raw gistfile1.cs hosted with ❤ by GitHub

Now, to tie it all togheter, you need to add some custom styles to make an actual icon show up. To do this, add a file module.config to the root of your web application (or modify your existing one), and fill it with the following:

```
<?xml version="1.0" encoding="utf-8" ?>
<module>
  <clientResources>
    <add name="epi-cms.widgets.base" path="/Styles/EpiserverUIExtensions.css" resour
  </clientResources>
</module>
```

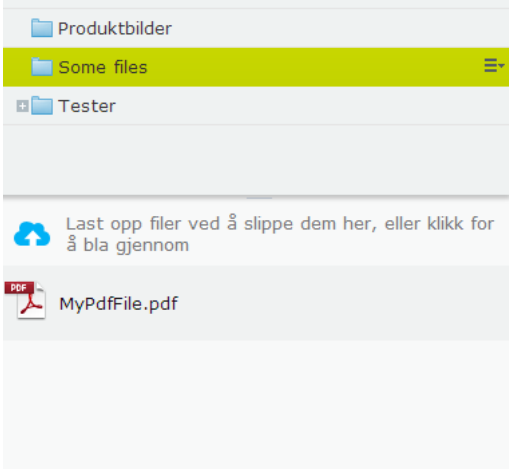
view raw gistfile1.xml hosted with ❤ by GitHub

...and then add a stylesheet on the correct path in which you can add some styling. Like this:

```
.pdfIcon { background-image: url(/UI/Icons/pdfIcon.png);width: 32px;height: 32px;mar
```

view raw gistfile1.css hosted with ❤ by GitHub

And now, it should look like something like this:



Last opp filer ved å slippe dem her, eller klikk for å bla gjennom

Neat. That looks quite a bit better. And hopefully your editors will enjoy it too.

So, for those of you who did not pay attention, copypaste the code above, and follow these conventions:

- Add CSS-classes called <extension>Icon that applies an actual icon. For example docxIcon and pptxIcon.
- Add UIDescriptor classes for the media types you want to give icons, just like my PdfFileDescriptor above. Examples are DocxFile and PptxFile.
- As my UIDescriptor needs T where T : ContentData, you should be able to add custom icons to your page tree with this too.

Note: This is partially based on stuff from the following articles. My aim was simply to have the entire step-by-step-guide for such a practical thing in one place:

- Customizing the look and behavior in the UI for your content types
- Adding a new device to the view resolution drop-down
- Changes to the module.config for EPiServer 7.1

TAGS

EPiServer
Edit-mode
EPiServer-7.5

Comments:

Thanks for sharing! Works like a charm, this should be default in EPiServer CMS.

Permanent link

12 May 2014 10:35 by: Daniel

Works great :) Although 32x32 icons looks nice in the assets pane, ContentArea/LinkItemCollection editors prefer 16x16. One way to target them individually is: https://gist.github.com/lonnkvist/173b7b14adabc42de33f

Permanent link

22 October 2014 13:59 by: Espen Lønnkvist

Neat. Clever thinking.

Permanent link

22 October 2014 14:03 by: Arve Systad

Name:

Email:

Comment:

□
Receive notification by email

Send

About me

Epinova profile
Twitter (@arvesystad)
EPiServer World profile

Archive

- 2015
  - July
  - June
  - March
- 2014
  - December
  - November
  - October
  - September
  - August
  - July
  - May
- 2013
  - August
  - June
  - January
- 2012
  - October

- July
- May
- March

- 2011

- September

Tag cloud

AccessControlLists build-process Checklist CMS Commerce CSS Edit-mode **EPiServer** EPiServer-7.5 HTML5 HttpModule IE IInitializableModule MIME MVC plugins PlugInSettings QA SEO teamcity UrlRewriting user-experience UX Virtual-Roles VPP vs2015 Web-standards Windows8 Yellow-screen-of-death

Oslo:
Arbeidersamfunnets plass 1,
0181 Oslo (kart)

Trondheim: Innherredsveien 7
7014 Trondheim (kart)

Epinova AS

Org.nr. 995 215 748
post@epinova.no