# Spørgsmål

# Database navngivning

Upac\_400 på dev6 (her udvikler man)

# Versionsnummer

Versionsnummer er <umbracp Major>.<umbraco Minor>.<umbraco Maintenance>.<upac major>.<upac minor>.<upac patch>\*

Det er først ved 3. gruppe at upac starter. Jeg starter ved 0, og bevæger mig opad for hver implementering/tilrettelse, som indeholder nævneværdige eller kritiske ændringer ift. umbraco strukturer eller UPAC API.

**Build setup**

NArange  
Castle Doc

**Macros**

Makroer, er simple ting som blandt andet kan embedes direkte i wysiwyg editoren eller bruges direkte i layout filerne.

Eksempler på makroer

* Iframe
* YouTube, Vimeo, Google video
* Sitemap
* RSS text feed (flickr, Delicious, slideshare)
* Google map
* Gallery
* Document liste (filer)
* Faktaboks
* Listevisning af børn – Billede, titel, manchet, dato

Makroer findes i sit eget projekt og website. De er dog afhængige af Upac.Core

Settings på makroer skal ske via appSettings angivet i filen /App\_Data/Upac.Macro.AppSettings.config  
<appSettings>  
<!-- Upac.Macro.IFrame -->  
<add key="Upac.Macro.IFrame.DefaultWidth" value="150px" />  
<add key="Upac.Macro.IFrame.DefaultHeight" value="150px" />

<!-- Upac.Macro.Sitemap -->

<add key="Upac.Macro.Sitemap.DefaultLevels" value="3" />

</appSettings>

Inkluderingen af vores appsettings sker via /web.config  
<appSettings configSource="/App\_Data/Upac.Macro.AppSettings.config">

Makroer installeres i løsninger via umbraco. Vi har fået vores eget custom package repository  
<repository name="1508 Repository" guid="69ebead07-8fcb-4777-972e-bd70a5dbc133" />

Vi skal lave vores egen packageInstallAction “ Upac.PackageActions.AddMacroSetting” som tilføjer til /App\_Data/Upac.Macro.AppSettings.config

Settings hives ud via Upac.Configuration

string width = Upac.Configuration.GetAppSetting("Upac.Macro.IFrame.DefaultWidth");

eller

string width = Upac.Configuration.AppSettings["Upac.Macro.IFrame.DefaultWidth"];

Skal også kunne retunere type stærke værdier

int width = Upac.Configuration.GetAppSetting("Upac.Macro.IFrame.DefaultWidth", 100);

De skal helst bestå af xslt, hvis man skal bruges C# skal det placers I /App\_Code/ mappen

Upac.Macro.IFrame.cs

Man kan eventuelt overveje om C# skal embedes direkte i xslt’en

CSS til en macroer skal placeres i /css/ Upac.Macro.IFrame.css

Alle stylesheet concates via eksternt modul  
PS måske skal makroer slet ikke have stylesheet? – men man skal nemt se eksempel

Hvis man ændrer i xslt’en, ascx’en eller cs’en skal det det logges i et sted? Måske svn kan bruges

Fil struktur for Upac.Macro.IFrame  
/App\_Code/Upac.Macro.IFrame.cs  
/css/Upac.Macro.IFrame.Example.css  
/usercontrols/Upac.Macros/Upac.Macro.IFrame.ascx

# Projecter

Upac.Core – Class lib

Upac.Site – Website (Content, Templates, Scripts, Stylesheets, Xslt files, Macroes, .NET User Control)

Upac. Foundation – Website (Document Types, Media Types, Languages, Data Types)

Upac.Macroes – Website (Med seneste Upac.Site og Upac. Foundation)

# Upac.Foundation

Man burger en local database UpacFoundation

<http://upac.foundation.local>

Når man opretter eller redigere Document Types, så skal man exporte document typen til \Upac.Foundation\PackageFiles\Document Types på den måde får vi versions styring samt bruges mappen af build scriptet til at lave pakken.

Når man eksportere skal filnavnet bestå af Alias på dokumentet typen hvor mellemrum er erstattet af – og filen skal end på .udt

Build inkluderer

* dokument typer
* File (media type) da vi har to ekstra egenskaber (Thumbnail og Description)
* Danish (Denmark) – Vi tager dansk sprog med som default

**Configuration**

SolutionSettings går på tværs af løsning

WebsiteSettings er på den enkelte løsning

Upac settings gemmes i App\_Data

Upac content basis: Image, Date, Titel, Manchet, Text

# Naming convention

## Document types

Document types består af Name og Alias. Name bruges af redaktører og Alias bruges af udviklere.

Efter Umbraco best practice bruges PascalCase til alias

### Eksempler på document typer som ikke tilhører et modul

#### Forside

Name: Frontpage  
Alias: Frontpage

#### Document

Name: Document  
Alias: Document

### Eksempler på document typer som tilhører et modul (gruppe)

#### Publikations beholder

Name: Publication Module  
Alias: PublicationModule

#### Publikations års beholder

Name: Publication Year Container  
Alias: PublicationYearFolder

#### Publikations måneds beholder

Name: Publication Month Container  
Alias: PublicationMonthFolder

#### Publikation

Name: Publication  
Alias: PublicationDocument

## Document Properties

Document types består af Name og Alias. Name bruges af redaktører og Alias bruges af udviklere.

Efter Umbraco best practice bruges camelCase til alias

### Eksempler

Name: Show to top  
Alias: showToTop

## Templates

## Macroes/Xslt/Ascx

## Javascript

## Stylesheets

## Dictionary

**Source Control**

http://ariejan.net/2006/11/24/svn-how-to-structure-your-repository/

**Trunk**

<http://svn/int900.0010upac/trunk> (her udvikler man)

The trunk contains the most current development code at all times. This is where you work up to your next major release of code.

**Branches**

<https://svn.example.com/svnroot/project/branches/>

**Release Branches**

<http://svn/int900.0010upac/branches/releases>

<http://svn/int900.0010upac/branches/releases/4.0.0.0>

When the trunk reaches the stage that it’s ready to be released (or when you want to freeze the addition of new features) you create a release. This release is just a copy of your current trunk code.

**Bug fix branches**

<http://svn/int900.0010upac/branches/bugs>

<http://svn/int900.0010upac/branches/bugs/BUG-3391>

Bug branches may be used to address the more serious bugs found in the trunk or a release branch. The bugs are of such a magnitute that you can’t fix them by yourself in a single commit. So, in order to focus on the problem of fixing this bug you should create a new bug branch for this purpose. This allows development in the trunk or your release branch to continue, and you won’t disturb them with new bugs or tests that break the current code.

Bug fix branches are named after the ID they are assigned in your bugtracking tool. Mostly this is a number: BUG-3391

**Experimental branches**

<http://svn/int900.0010upac/branches/try>

<http://svn/int900.0010upac/branches/try/TRY-3392-SilverLight>

Something that also happens a lot is the introduction of new technologies. This is fine, of course, but you don’t want to bet your entire project on it.

Experimental branches may be abandonned when the experiment fails. If they succeed you can easily merge that branch with the trunk and deliver your big new technology.

These branches are named after a ticket ID and what you’re experimenting with. I always prefix them with ‘TRY-’:

# Multi Developer Setup

Læsning: <http://www.farmcode.org/post/2009/09/09/How-to-sync-data-across-multiple-Umbraco-environments.aspx>

<http://our.umbraco.org/forum/core/general/3619-Source-control-and-multiple-developers?p=1>

<http://our.umbraco.org/forum/getting-started/installing-umbraco/2918-Update-an-Umbraco-website?p=0#comment11311>

# Richtext editor

Man kan sætte bredde på editoren via profiles. Her lægges ekstra bredde til, så det passer. Men det gør det ikke på default, her skal man selv lægge 35 pixel til i bredden!