Castle Windsor

* Castle pomaha v programovani tim, ze nemusime vytvaret instance nekterych trid. Dobre vysvetleni byla stavba drevostavby na youtube. Stavi se zed ktera ma stejne prvky. Kdyz prijde na radu okno, musime zmenit postup a to pokazde kdyz chceme pridat okno. Toto okno zaregistrujeme do castlu a uz ho nemusime znovu a znovu vyrabet. Castle ho zna, a jeho instanci nam da pokazde kdyz ji potrebujeme. Instance umi vyrobit I z xml souboru ContainerConfig, kdyz mu v bootstrapperu rekneme, kde ma tohle xml najit.

m\_globalContainer = new WindsorContainer(new XmlInterpreter(Path.Combine(m\_applicationDirectory, "MIR.PrintStorage.Scanning2.Shell.Container.config")));

* Pak muze resolvovat instance trid ktere jsou definovane v xml. Viz: [ContainerConfig](https://d.docs.live.net/b22fb0fb09218bf0/Nielsen%20%20prace/Moje%20poznamky%20Nielsen/ContainerConfig%20aneb%20Jak%20vymenovat%20obrazek%20na%20splash%20screenu%20podle%20zeme%20a%20verze.docx)
* Tridy ktere chce resolvovat a nejsou v xml, musi byt zaregistrovane. Instance tridy se da vytvorit rucne a pak zaregistrovat. Tim rikame, ze ma castle pouzivat tuto instanci kdyz si rekneme o instanci takoveto tridy.

IWindowManager windowManager = new WindowManager();

m\_globalContainer.Register(Component.For<IWindowManager>().Instance(windowManager));

* Aby bylo mozne vytvaret instance naseho conteineru, musi si container zaregistrovat sam sebe.

m\_globalContainer.Register(Component.For<IWindsorContainer>().Instance(m\_globalContainer).LifestyleSingleton());

* Pro pouziti Installeru (trida ve ktere registrujeme jine tridy do castlu) potrebujeme jednu prazdnou tridu, kterou si pridame do assembly ze ktere chceme instalovat. A zavolame metodu Instal na containeru. V tomto pripade rikame ze se instaluje z Core a z Shellu.

m\_globalContainer.Install(FromAssembly.This(), FromAssembly.Containing<ScanningCoreAssemblyIdentificator>());

**Installery**

**Zavolanim metody** m\_globalContainer.Install se pravdepodobne zavolaji všechny Install metody z těchto assembly. Nejspise je to tim ze instance windsorContaineru je singleton. Metody instal jsou ve tridach :

Shell:

1. ScanningInstaller : IwindsorInstaller
2. PrintStorageDbAccessInstaller : IwindsorInstaller
3. MediaDataDbAccessInstaller : IwindsorInstaller

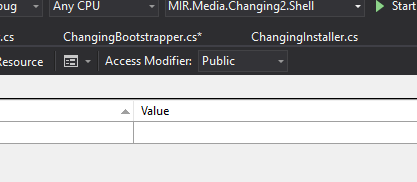
Core:

1. MappingInstaller : IWindsorInstaller

V těchto Install metodach jsou zaregistrovane všechny objekty které ma Castle znat, aby nam mohl poskytovat jejich instance.

Castle poprve vyzkouseno na projektu **MediaPlayer2.** Caliburn uz tam byl drive.

* Nejprve se postarame o Caliburn podle poznamek. (Start from bootstrapper using caliburn)
* Dale v Bootsrapperu vytvorime slozku Installers.
* Pridame projekty Common, Core a do nich tridy CoreAssemblyIdentificator a CommonAssemblyIdentifikator.
* V metode Bootstrapper.OnStartup si zaregistrujeme container, windowManagera a urcime z jakych assembly se bude instalovat. Postupovat podle kodu ve Skenovadle:
* Do Gui si pridame ResourcesFiles podle MojePoznamky -> JAZYK A CULTURE (nemusi se spustit generator staci zmenit AccesModifier na Public



Pro vysvetleni Bootstrapperu viz <https://d.docs.live.net/b22fb0fb09218bf0/Nielsen%20%20prace/Moje%20poznamky%20Nielsen/Bootstrapper.docx>

Vytvorime tridu Installer : IWindsorContainer ve slozce Shell.Installers a implementujeme metodu Install

##### Do slozky Bootstrapper si pridame dve tridy : InternalServicePublisher a NotificationProcessor

using System;

using Castle.Windsor;

using Mediaresearch.Framework.Communication.Common;

namespace Shell.Bootstrapper

{

public class InternalServicePublisher : ClientToServicePublisherBase

{

private readonly IWindsorContainer m\_kernel;

public InternalServicePublisher(IWindsorContainer kernel, IServiceActionSubscriber serviceActionSubscriber) : base(serviceActionSubscriber)

{

m\_kernel = kernel;

}

public override IServiceAction GetAction(Type actionType)

{

var action = m\_kernel.Resolve(actionType);

IServiceActionCallback actionCallback = (IServiceActionCallback)action;

actionCallback.ExecutionFinished += ActionCallbackOnExecutionFinished;

return (IServiceAction)action;

}

private void ActionCallbackOnExecutionFinished(object sender, EventArgs eventArgs)

{

IServiceActionCallback action = (IServiceActionCallback)sender;

action.ExecutionFinished -= ActionCallbackOnExecutionFinished;

m\_kernel.Release(action);

}

}

}

using System;

using Castle.MicroKernel.Registration;

using Castle.Windsor;

using Mediaresearch.Framework.Communication.Common;

namespace MIR.PrintStorage.Scanning2.Shell.Bootstrapper

{

public class ScanningNotificationProcesor : NotificationReceiverProcessor

{

public ScanningNotificationProcesor(INotificationsReceiversAssemblyProvider notificationsReceiversAssemblyProvider, IWindsorContainer container) : base(notificationsReceiversAssemblyProvider, container)

{

}

protected override void DoAfterSubscribe<TNotification>(Type receiverType)

{

if (!m\_container.Kernel.HasComponent(receiverType))

{

m\_container.Register(Component.For(receiverType).ImplementedBy(receiverType).LifestyleSingleton());

}

}

}

}

##### Vytvoreni WindsorConteineru a jeho konfigurace (Container.config) (Je tu I konfigurace pripojeni k databazi)

**Tim, ze containeru vyrobime konfiguracni xml soubor mu muzeme nastavit ruzne parametry**

Kod v Bootstrapperu: (Timto bude container vedet ze si ma brat property z xml souboru) :

m\_globalContainer = new WindsorContainer(new XmlInterpreter(Path.Combine(m\_applicationDirectory, "MIR.Media.Changing2.Shell.Container.config")));

Do Shellu se prida NewItem -> ApplicationConfigurationFile.

Bude se jmenovat MIR.Media.Changing2.Shell.Container.config

Muze byt na zacatku klidne prazdny ale v properties musi mit nastaveno **CopyAlways** jinak vyhazuje vyjimku : $exception {"Error processing node resource FileResource: [] []"}

Kdyz aplikaci rozchodime aby zobrazovala okno, muzeme si pridat do Commonu rozhrani. Ve skenovadle to bylo IscanningConfiguration. Bylo ve slozce :

*C:\Pool\Admosphere\src\MIR.PrintStorage\MIR.PrintStorage.Scanning2\MIR.PrintStorage.Scanning2.Common\Scan\IScannigConfiguration.cs*

Rozhrani jen definuje property ktere musi mit trida, implementujici toto rozhrani. Trida ktera ho implementuje je: ScanningConfiguration : IScannigConfiguration

*C:\Pool\Admosphere\src\MIR.PrintStorage\MIR.PrintStorage.Scanning2\MIR.PrintStorage.Scanning2.Shell\Installers\ScanningConfiguration.cs*

Tato trida musi mit vsechny property ktere definuje rozhrani a zaroven ma konstruktor ve kterem tyto property inicializuje. Tady probehne magie protože hodnoty se nikde nezadavaji. Tim, ze ma trida zavislosti na tyto property, si je Castle resolvne prave z configu.

napr.: public string IconPath { get; }

public ScanningConfiguration(int pdfConversionDpi, int pdfConversionQuality, string primaryScannerName,

bool primaryScannerDuplex, string secondScannerName, int scannerDpi, int pdfSourceOrder,

int scannerSourceOrder, int filesSourceOrder, bool useTwain2, bool useDoubleFeedDetection, string iconPath)

{

PdfConversionDpi = pdfConversionDpi;

PdfConversionQuality = pdfConversionQuality;

PrimaryScannerName = primaryScannerName;

PrimaryScannerDuplex = primaryScannerDuplex;

SecondScannerName = secondScannerName;

ScannerDpi = scannerDpi;

PdfSourceOrder = pdfSourceOrder;

ScannerSourceOrder = scannerSourceOrder;

FilesSourceOrder = filesSourceOrder;

UseTwain2 = useTwain2;

UseDoubleFeedDetection = useDoubleFeedDetection;

IconPath = iconPath;

}

Konfiguracni soubor ze skenovadla:

*C:\Pool\Admosphere\src\MIR.PrintStorage\MIR.PrintStorage.Scanning2\MIR.PrintStorage.Scanning2.Shell\MIR.PrintStorage.Scanning2.Shell.Container.config*

<?xml version="1.0" encoding="utf-8" ?>

<castle>

**Properties -> sem zadame hodnoty pro property z rozhrani**

<properties>

<mediaDataDbAlias>mediaDataDB</mediaDataDbAlias>

<mediaDataConnectionString>Data Source=stoupa;Initial Catalog=Mediadata3Auto;User ID=PrintStorageScanningUser; Pwd=5bTDwG-tJC;</mediaDataConnectionString>

<printStorageDbAlias>printStorageDB</printStorageDbAlias>

<printStorageConnectionString>Data Source=stoupa;Initial Catalog=PrintStorageAuto;User ID=PrintStorageScanningUser; Pwd=5bTDwG-tJC;</printStorageConnectionString>

<serverTimeZone>Central Europe Standard Time</serverTimeZone>

<pdfConversionDpi>120</pdfConversionDpi>

<pdfConversionQuality>50</pdfConversionQuality>

<primaryScannerName>Panasonic KV-S4085C KV-S4065C</primaryScannerName>

<primaryScannerDuplex>true</primaryScannerDuplex>

<secondScannerName>unknown</secondScannerName>

<scannerDpi>150</scannerDpi>

<useTwain2>true</useTwain2>

<useDoubleFeedDetection>false</useDoubleFeedDetection>

<pdfSourceOrder>2</pdfSourceOrder>

<scannerSourceOrder>1</scannerSourceOrder>

<filesSourceOrder>3</filesSourceOrder>

<iconPath>/MIR.PrintStorage.Scanning2.GUI;component/ScanningIcons/Scanning.png</iconPath>

</properties>

<components>

<component id="MediaDataDbConfiguration" type="MIR.PrintStorage.Scanning2.Shell.Installers.MediaDataDbConfiguration, MIR.PrintStorage.Scanning2.Shell">

<parameters>

<mediaDataDbAlias>#{mediaDataDbAlias}</mediaDataDbAlias>

<mediaDataConnectionString>#{mediaDataConnectionString}</mediaDataConnectionString>

<serverTimeZone>#{serverTimeZone}</serverTimeZone>

</parameters>

</component>

<component id="PrintStorageDbConfiguration" type="MIR.PrintStorage.Scanning2.Shell.Installers.PrintStorageDbConfiguration, MIR.PrintStorage.Scanning2.Shell">

<parameters>

<printStorageDbAlias>#{printStorageDbAlias}</printStorageDbAlias>

<printStorageConnectionString>#{printStorageConnectionString}</printStorageConnectionString>

<serverTimeZone>#{serverTimeZone}</serverTimeZone>

</parameters>

</component>

**Definice rozhrani IscanningConfiguration. Parametry uvnitr rikaji, ze napr. hodnotu pro promennou IconPath definovanou v rozhrani IscanningConfiguration vezmeme z properties castlu**

<component id="ScanningConfiguration" type="MIR.PrintStorage.Scanning2.Shell.Installers.ScanningConfiguration, MIR.PrintStorage.Scanning2.Shell"

service="MIR.PrintStorage.Scanning2.Common.Scan.IScannigConfiguration">

<parameters>

<pdfConversionDpi>#{pdfConversionDpi}</pdfConversionDpi>

<pdfConversionQuality>#{pdfConversionQuality}</pdfConversionQuality>

<primaryScannerName>#{primaryScannerName}</primaryScannerName>

<primaryScannerDuplex>#{primaryScannerDuplex}</primaryScannerDuplex>

<secondScannerName>#{secondScannerName}</secondScannerName>

<scannerDpi>#{scannerDpi}</scannerDpi>

<pdfSourceOrder>#{pdfSourceOrder}</pdfSourceOrder>

<scannerSourceOrder>#{scannerSourceOrder}</scannerSourceOrder>

<filesSourceOrder>#{filesSourceOrder}</filesSourceOrder>

<useTwain2>#{useTwain2}</useTwain2>

<useDoubleFeedDetection>#{useDoubleFeedDetection}</useDoubleFeedDetection>

<iconPath>#{iconPath}</iconPath>

</parameters>

</component>

</components>

</castle>

**Shrnuti:**

Kdyz chci aby byl container trochu chytry pridame mu rozhrani InecoConfig, Tridu ktera ho bude implementovat NecoConfig a xml konfiguracni soubor Container.config;

Kdyz chceme aby castle znal nejakou novou propertu at uz boolean, string nebo cokoli jineho, pridame si propertu do rozhrani, do tridy ktera ho implementuje, do properties v configu a do definice rozhrani v configu.

V installeru si musime rozhrani **Resolvnout** . : **IScannigConfiguration scanningConfiguration = container.Resolve<IScannigConfiguration>();**

**Mozna chyba:**

* Zkopiroval jsem cestu k souborum z Properties tridy ChangingConfiguration (pri zakladani zmenovadla) . Do cesty se mi ale misto tecek pridaly lomitka a Castlu se to nelibilo. Napsal ze nemuze vyrobit type from string .
* I kdyz jsem provedl uvedene kroky, string ktery jsem potreboval v Bootstrapperu byl porad null:

IChangingConfiguration changingConfiguration = m\_globalContainer.Resolve<IChangingConfiguration>();

var somethingUsefull = changingConfiguration.SomethingUsefull; **(porad null)**

Console.WriteLine(somethingUsefull);

Bylo to proto, ze :

Chybel konstruktor ve tride ChangingConfiguration se zavislosti na somethingUsefull. Ale je to stale null

Trida public class ScanningConfiguration : IscannigConfiguration musi mit propertu: public static string TempFilesDirectory { get; } = Path.Combine(Path.GetTempPath(), "Scanning"); d

V Bootstrapperu musi byt :

if (!Directory.Exists(ScanningConfiguration.TempFilesDirectory))

{

Directory.CreateDirectory(ScanningConfiguration.TempFilesDirectory);

}

Ted uz to funguje :D

##### Language

Dale jsem sel po cervene sviticich radcich a dopnoval to co aplikace potrebuje. Svitilo mi language var cultureInfo = CultureInfo.GetCultureInfo(Settings.Default.Language);

Ve slozce Shell je potreba rozklikat Settings.settings az dolu a doplnit :

public string Language

{

get

{

return ((string)(this["Language"]));

}

set

{

this["Language"] = value;

}

}

##### Pripojeni k databazi

Zacneme v uz znamem Configu. Na pripojeni k databazi potrebujeme mit v ContainerConfigu ConnectionString a komponentu PrintStorageDbConfiguration:

Definice ConnectionStringu

<properties>

<printStorageDbAlias>printStorageDB</printStorageDbAlias>

<printStorageConnectionString>Data Source=stoupa;Initial Catalog=PrintStorageAuto;User ID=PrintStorageScanningUser; Pwd=5bTDwG-tJC;</printStorageConnectionString>

<serverTimeZone>Central Europe Standard Time</serverTimeZone>

</properties>

Definice tridy PrintStorageDbConfiguration

<components>

<component id="PrintStorageDbConfiguration" type="MIR.PrintStorage.Scanning2.Shell.Installers.PrintStorageDbConfiguration, MIR.PrintStorage.Scanning2.Shell">

<parameters>

<printStorageDbAlias>#{printStorageDbAlias}</printStorageDbAlias>

<printStorageConnectionString>#{printStorageConnectionString}</printStorageConnectionString>

<serverTimeZone>#{serverTimeZone}</serverTimeZone>

</parameters>

</component>

</components>

Musime si pridat tridu :

*C:\Pool\Admosphere\src\MIR.PrintStorage\MIR.PrintStorage.Scanning2\MIR.PrintStorage.Scanning2.Shell\Installers\PrintStorageDbConfiguration.cs*

Ktera potrebuje*:*

*C:\Pool\Admosphere\src\MIR.PrintStorage\MIR.PrintStorage.Scanning2\MIR.PrintStorage.Scanning2.Shell\Installers\DbAccessConfigurationBase.cs*

##### Tridy PrintStorageDbConfiguration a DbAccessConfigurationBase

namespace MIR.PrintStorage.Scanning2.Shell.Installers

{

public class PrintStorageDbConfiguration : DbAccessConfigurationBase

{

protected override string ConnectionString => PrintStorageConnectionString;

public string PrintStorageDbAlias { get; }

public string PrintStorageConnectionString { get; }

public string ServerTimeZone { get; }

public PrintStorageDbConfiguration(string printStorageDbAlias, string printStorageConnectionString, string serverTimeZone)

{

PrintStorageDbAlias = printStorageDbAlias;

PrintStorageConnectionString = printStorageConnectionString;

ServerTimeZone = serverTimeZone;

}

}

}

using System.Text.RegularExpressions;

namespace MIR.PrintStorage.Scanning2.Shell.Installers

{

public abstract class DbAccessConfigurationBase

{

protected abstract string ConnectionString { get; }

public string DefaultDatabaseName

{

get

{

Match match = GetConnectionStringMatch();

if (match == null || match.Groups.Count < 3)

return null;

return match.Groups[2].Value;

}

}

public string DefaultDataSource

{

get

{

Match match = GetConnectionStringMatch();

if (match == null || match.Groups.Count < 3)

return null;

return $"{match.Groups[1].Value}/{match.Groups[2].Value}";

}

}

private Match GetConnectionStringMatch()

{

if (string.IsNullOrEmpty(ConnectionString))

return null;

Regex regex = new Regex(@".\*Data Source=([a-z0-9\w\.]\*);.\*Initial Catalog=([a-z0-9\w\.]\*);.\*", RegexOptions.IgnoreCase);

Match match = regex.Match(ConnectionString);

if (!match.Success)

return null;

return match;

}

}

}

##### Pokracovani pripojeni k Db:

* Ted uz v Installeru muzeme rict ze:

PrintStorageDbConfiguration configuration = container.Resolve<PrintStorageDbConfiguration>();

* Abychom mohli v installeru rict ze:

container.Register(Component.For<IParamsSource>().ImplementedBy<DatabaseParamsSource>().LifestyleSingleton());

* Budeme potrebovat rozhrani a tridu IparamsSource a DatabaseParamsSource

**Tato cast uz s Castlem nema moc spolecneho ale budu tady pokracovat at to neni na deseti mistech**

Vyjimka kterou mi vyhazovalo protoze jsem nemel vyresenou zavislost na Usera vyresil Mitroz : no bylo potreba trocu uklidit, user nesel kvuli tomu ze tam chybelo m\_globalContainer.Resolve<IMediaDataDaoSource>();

* Otestovat spojeni s databazi se da pomoci Unit testu -> MojePoznamky -> Unit Testy -> ConnectionTest
* Projekt jsem ulozil na plochu do slozky ChangingProjekty