**Jak vymenovat obrazek na splash screenu podle zeme a verze (test rc) zachytavadlo nebo stare scanovadlo ContainerConfig**

Enviroments

Kazdy environment ma svoji ikonu , tu nastavit jako Resource

Resolvnout v bootstrapperu a vlozit jako parametr do kostruktoru splash

Mam otevreny Shell.Enviroment, tady je slozka pro kazde prostredi -> CZ, BG, EN . Pridal jsem nahore do properties svoji novou propertu iconPath a do ni cestu k ikone kterou potrebuju.

<castle>

<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>

<iconPath>MIR.PrintStorage.Scanning2.Shell\Environments\CZ.Test\Scanning.ico"</iconPath>

<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>

</properties>

<components>

Ve stejne tride dole jsou parametry rozhrani IscanningConfiguration. Tady se rika ze jako parametr se ma pouzit properta

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

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>

* Pridame propertu do rozhrani IscanningConfiguration:

namespace MIR.PrintStorage.Scanning2.Common.Scan

{

public interface IScannigConfiguration

{

int PdfConversionDpi { get; }

int PdfConversionQuality { get; }

string PrimaryScannerName { get; }

bool PrimaryScannerDuplex { get; }

string SecondScannerName { get; }

int ScannerDpi { get; }

int PdfSourceOrder { get; }

int ScannerSourceOrder { get; }

int FilesSourceOrder { get; }

bool UseTwain2 { get; }

bool UseDoubleFeedDetection { get; }

string IconPath { get; }

}

}

* A zaroven doplnime chybejici propertu ve tride ktera implementuje rozhrani:

namespace MIR.PrintStorage.Scanning2.Shell.Installers

{

public class ScanningConfiguration : IScannigConfiguration

{

public string IconPath { get; }

}

* Ted si potrebujeme nejak resolvnout string v Bootstrapperu, aby se dal predat konstruktoru SplashScreenu. V ScanningInstalleru mame uz scanningConfiguration Resolvnuty:

public class ScanningInstaller : IWindsorInstaller

{

public void Install(IWindsorContainer container, IConfigurationStore store)

{

BLToolkit.Common.Configuration.NullableValues.String = null;

container.Kernel.Resolver.AddSubResolver(new CollectionResolver(container.Kernel, true));

container.AddFacility<TypedFactoryFacility>();

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

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

IScannigConfiguration scanningConfiguration = container.Resolve<IScannigConfiguration>();

* Toto si potrebujeme resolvnout I v Bootstrapperu:

namespace MIR.PrintStorage.Scanning2.Shell.Bootstrapper

{

public class ScanningBootstrapper : Bootstrapper<MainViewModel>

{

protected override void OnStartup(object sender, StartupEventArgs e)

{

ConfigureLog4Net();

ConfigureLanguage();

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

var configuration = m\_globalContainer.Resolve<PrintStorageDbConfiguration>();

IScannigConfiguration scanningConfiguration = m\_globalContainer.Resolve<IScannigConfiguration>();

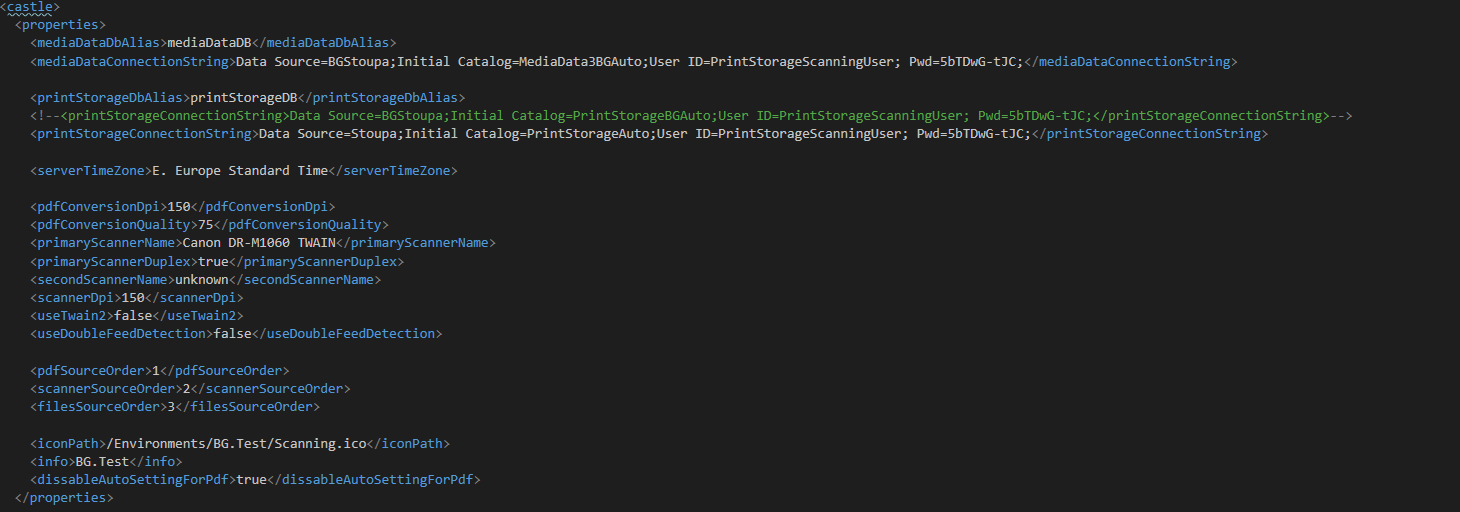
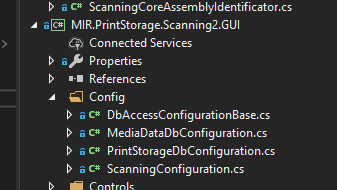
string path = scanningConfiguration.IconPath;

* Ted uz si muzeme do konstruktoru SplashScreenu poslat Path:

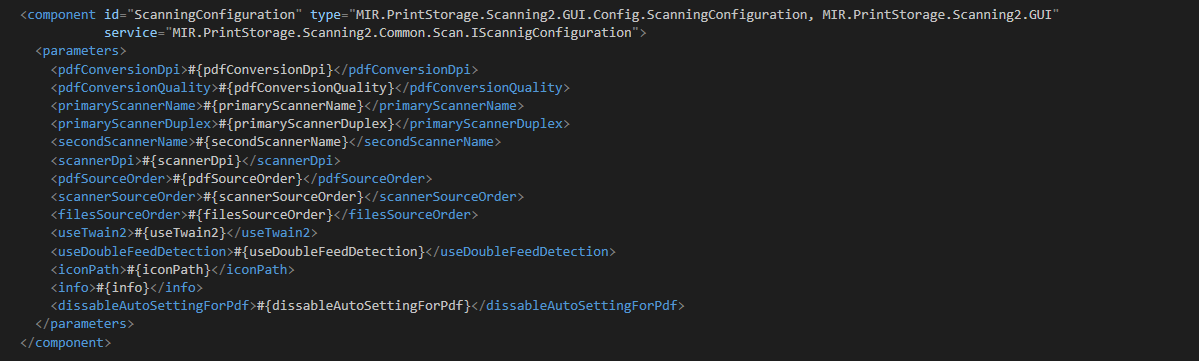
SplashScreenViewModel splashScreen = new SplashScreenViewModel(initAction, version, path);

* Dulezite: Container.config je hlavni v Shellu ten se spousti pri startu aplikace . Ty ktere jsou ve slozce Enviroments jsou kopie tohoto configu, ktere se pouziji jen pro ClickOnce. Jsou skoro stejne, jen maji pozmenene nektere hodnoty podle prostredi ve kterem se spousti. Tady treba jina cesta k ikone pro ceskou a bulharskou verzi. Pokud neni tento hlavni config vyplnen, vyjimka dependencies to be satisfied.

Chapu to tak, ze mame nejaky xml dokument ContainerConfig, ze ktereho se nacitaji vsechny v tomto dokumentu prednastavene promenne ktere v programu potrebujeme. Castle se sem podiva a nastavi property pro tridu scanningConfiguration, printStorageDbConfiguration a jine tridy definovane v containerConfigu. V horni casti ContainerConfigu je jakasi inicializace promennych (deklarace je dole => parameters).



V dolni casti parameters je deklarace promennych . Rika ze napr. ze hodnota promenne pro tridu ScanningConfiguration ktera se jmenuje #{iconPath} se ma vzit z casti properties. Type oznacuje cestu k teto tride. Pozor nachazi se v Gui



Service je rozhrani nad tridou ScanningConfiguration. Tohle je jeden priklad co muze byt v ContainerConfigu , dale je tam napr PrintStorageDbConfiguration atd.

**V Bootstrapperu se Windsoru nastavi cesta k tomuto xml ku, aby vedel kam se ma divat**

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

**A muzeme si resolvovat vse co je definovano v xml dokumentu**

var configuration = m\_globalContainer.Resolve<PrintStorageDbConfiguration>();

var scanningConfiguration = m\_globalContainer.Resolve<IScannigConfiguration>();

(ostatni veci resolvnute v bootstrapperu musi byt registrovane v installeru )

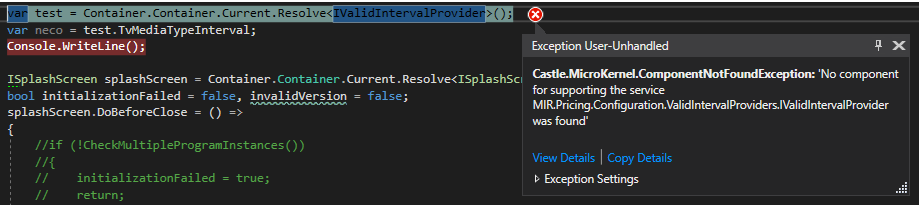
Kdyz teda potrebuju cestu k ikone ktera se ma zobrazit ve splashScreenu, tak ji vezmu ze scanningConfiguration, jelikoz je to properta teto tridy

var scanningConfiguration = m\_globalContainer.Resolve<IScannigConfiguration>();

var info = scanningConfiguration.Info;

var path = scanningConfiguration.IconPath;

**Nesnazit se resolvnout moc brzo nejdrive se musi xml ko nainstalovat.**

****

**To co je v configu, se nemusi zvlast registrovat (Component.Register)**

**Pokud je v configu za tridou klicove slovo** service **je mozne resolvnout jen rozhrani**

**Jde resolvnout rozhrani :**

<component id="TvIntervalProviderViewModel" type="MIR.Pricing.Configuration.ValidIntervalProviders.TvIntervalProviderViewModel, MIR.Pricing" service="MIR.Pricing.Configuration.ValidIntervalProviders.IValidIntervalProvider">

**Jde resolvnout trida:**

<component id="TvIntervalProviderViewModel" type="MIR.Pricing.Configuration.ValidIntervalProviders.TvIntervalProviderViewModel, MIR.Pricing" service="MIR.Pricing.Configuration.ValidIntervalProviders.IValidIntervalProvider">

**ViewModel se da definovat a resolvnout z configu viz Pricing**