MisconfiguredComponents Unit Test

# Unit Test kod:

## Kodovadlo test kod

using System;

using System.Configuration;

using System.IO;

using System.Linq;

using System.Reflection;

using System.Text;

using System.Windows;

using Castle.Facilities.TypedFactory;

using Castle.MicroKernel.Handlers;

using Castle.MicroKernel.Registration;

using Castle.MicroKernel.Resolvers.SpecializedResolvers;

using Castle.Windsor;

using Castle.Windsor.Diagnostics;

using Mediaresearch.Framework.Utilities.Ares;

using Mediaresearch.Framework.Utilities.Castle.Proxy;

using Mediaresearch.Framework.Utilities.Configuration;

using MIR.Media.Coding.Common.Configuration;

using MIR.Media.Coding.Configurations;

using MIR.Media.Coding.Container.Installers;

using Moq;

using NUnit.Framework;

using Configuration = Castle.Windsor.Installer.Configuration;

namespace MIR.Media.Coding.Tests

{

[TestFixture]

public class MisconfiguredComponentsTest

{

[SetUp]

public void SetUp()

{

const string xmlPath = @"\MIR.Media.Coding\InstitutionalSectorToOwnershipMapping\InstitutionalSectorToOwnershipMapping.xml";

const string appConfigPath = @"\MIR.Media.Coding.Shell\app.config";

m\_windsorContainer = new WindsorContainer();

m\_windsorContainer.Kernel.ProxyFactory = new CustomWindsorProxyFactory();

m\_windsorContainer.AddFacility<TypedFactoryFacility>();

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

m\_windsorContainer.Kernel.Resolver.AddSubResolver(new CollectionResolver(m\_windsorContainer.Kernel));

var configurationProviderMock = new Mock<IConfigurationProvider>();

if (!UriParser.IsKnownScheme("pack"))

{

new Application();

}

// ShellConfig tohle funguje jen u me ale na TFS uz ne

var fileMap = new ExeConfigurationFileMap

{

ExeConfigFilename = GetProjectPath() + appConfigPath

};

var shellExeConfiguration = ConfigurationManager.OpenMappedExeConfiguration(fileMap, ConfigurationUserLevel.None);

// Configurations

var databaseConfiguration = shellExeConfiguration.GetSection("DatabaseConfiguration") as DatabaseConfiguration;

var aresConfiguration = shellExeConfiguration.GetSection("AresConfiguration") as AresConfiguration;

var addressConfiguration = shellExeConfiguration.GetSection("AddressConfiguration") as AddressConfiguration;

var codingConfiguration = shellExeConfiguration.GetSection("CodingConfiguration") as CodingConfiguration;

var propertiesConfiguration = shellExeConfiguration.GetSection("PropertiesConfiguration") as PropertiesConfiguration;

var motiveConfiguration = shellExeConfiguration.GetSection("MotiveConfiguration") as MotiveConfiguration;

var fileSystemConfiguration = shellExeConfiguration.GetSection("FileSystemConfiguration") as FileSystemConfiguration;

var advertisementTypesConfiguration = shellExeConfiguration.GetSection("AdvertisementTypesConfiguration") as AdvertisementTypesConfiguration;

var motiveNameFilterConfiguration = shellExeConfiguration.GetSection("MotiveNameFilterConfiguration") as MotiveNameFilterConfiguration;

var propagationConfiguration = shellExeConfiguration.GetSection("PropagationConfiguration") as PropagationConfiguration;

configurationProviderMock.Setup(d => d.GetConfig<DatabaseConfiguration>()).Returns(databaseConfiguration);

configurationProviderMock.Setup(d => d.GetConfig<AresConfiguration>()).Returns(aresConfiguration);

configurationProviderMock.Setup(d => d.GetConfig<AddressConfiguration>()).Returns(addressConfiguration);

configurationProviderMock.Setup(d => d.GetConfig<CodingConfiguration>()).Returns(codingConfiguration);

configurationProviderMock.Setup(d => d.GetConfig<PropertiesConfiguration>()).Returns(propertiesConfiguration);

configurationProviderMock.Setup(d => d.GetConfig<MotiveConfiguration>()).Returns(motiveConfiguration);

configurationProviderMock.Setup(d => d.GetConfig<FileSystemConfiguration>()).Returns(fileSystemConfiguration);

configurationProviderMock.Setup(d => d.GetConfig<AdvertisementTypesConfiguration>()).Returns(advertisementTypesConfiguration);

configurationProviderMock.Setup(d => d.GetConfig<MotiveNameFilterConfiguration>()).Returns(motiveNameFilterConfiguration);

configurationProviderMock.Setup(d => d.GetConfig<PropagationConfiguration>()).Returns(propagationConfiguration);

// InstitutionalSectorToOwnershipXmlPath

var institutionalSectorToOwnershipXmlPath = GetProjectPath() + xmlPath;

var factory = new InstitutionalSectorToOwnershipCacheFactory(institutionalSectorToOwnershipXmlPath);

// IsDefault() instances

m\_windsorContainer.Register(Component.For<InstitutionalSectorToOwnershipCacheFactory>().Instance(factory).IsDefault().Named("CodingTestFactory"));

m\_windsorContainer.Register(Component.For<IConfigurationProvider>().Instance(configurationProviderMock.Object).IsDefault().Named("CodingTestImplementation"));

// Installery

m\_windsorContainer.Install(new SplashScreenInstaller());

m\_windsorContainer.Install(Configuration.FromXmlFile(@"MIR.Media.Coding.Container.config"));

m\_windsorContainer.Install(new ConnectionInstaller());

//m\_windsorContainer.ResolveAll<IDaoSource>(); resolvnuto v connectionInstalleru.

m\_windsorContainer.Install(new CommonInstaller());

m\_windsorContainer.Install(new PricingServiceInstaller());

m\_windsorContainer

.Install(

new MotiveInstaller(),

new CreativeInstaller(),

new MessageInstaller(),

new PricingInstaller(),

new ReclamationInstaller(),

new PictureMatchingInstaller(),

new PropagationInstaller(),

new ProductionCodingModeInstaller(),

new ShortcutsInstaller(),

new WorkflowInstaller(),

new ShellInstaller(),

new RuntimeResolvedComponentsInstaller()

);

//var configurator = new CastleDependencyMappingConfigurator(m\_windsorContainer, m\_windsorContainer.ResolveAll<MappingConfiguratorBase>());

//configurator.Configure();

}

private WindsorContainer m\_windsorContainer;

private const char Separator = '/';

private static string GetProjectPath()

{

var codeBase = Assembly.GetExecutingAssembly().CodeBase;

var pathItems = codeBase.Split(Separator);

var start = pathItems.ToList().FindIndex(x => string.Equals("C:", x));

var end = pathItems.ToList().FindIndex(x => string.Equals("MIR.Media.Coding", x)) + 1;

var projectPath = string.Join(Path.DirectorySeparatorChar.ToString(), pathItems.Skip(start).Take(end - start));

return projectPath;

}

[Test]

public void CheckForMisconfiguredComponents()

{

var diagnostic = new PotentiallyMisconfiguredComponentsDiagnostic(m\_windsorContainer.Kernel);

var handlers = diagnostic.Inspect();

if (handlers?.Any() == true)

{

var builder = new StringBuilder();

builder.AppendFormat("Misconfigured components ({0})\r\n", handlers.Count());

foreach (var handler in handlers)

{

var info = (IExposeDependencyInfo) handler;

var inspector = new DependencyInspector(builder);

info.ObtainDependencyDetails(inspector);

}

Assert.Fail(builder.ToString());

}

}

}

}

## Vystrihovadlo test kod

using System.Linq;

using System.Text;

using Caliburn.Micro;

using Castle.Facilities.TypedFactory;

using Castle.MicroKernel.Handlers;

using Castle.MicroKernel.Registration;

using Castle.Windsor;

using Castle.Windsor.Configuration.Interpreters;

using Castle.Windsor.Diagnostics;

using Castle.Windsor.Installer;

using Mediaresearch.Framework.DataAccess.BLToolkit.Dao;

using Mediaresearch.Framework.Mapping;

using Mediaresearch.Framework.Mapping.Castle;

using Mediaresearch.Framework.Utilities.Configuration;

using MIR.Media.Cutting2.Gui.Configuration;

using MIR.Media.Cutting2.Shell.Configuration;

using Moq;

using NUnit.Framework;

namespace MIR.Media.Cutting2.Core.Tests

{

[TestFixture]

public class MisconfiguredComponentsTest

{

[SetUp]

public void SetUp()

{

const string containerConfig = "MIR.Media.Cutting2.Shell.Container.config";

const string installersConfig = "MIR.Media.Cutting2.Shell.Installers.config";

var provider = new Mock<IConfigurationProvider>();

provider.Setup(d => d.GetConfig<CuttingConfiguration>()).Returns(new CuttingConfiguration {ApplicationName = "MIR.Media.Cutting2"});

provider.Setup(d => d.GetConfig<ColorRecognitionConfiguration>()).Returns(new ColorRecognitionConfiguration {GaietyGrayThreshold = 5});

provider.Setup(d => d.GetConfig<ColorRecognitionConfiguration>()).Returns(new ColorRecognitionConfiguration {GaietyColorThreshold = 0});

m\_windsorContainer = new WindsorContainer(new XmlInterpreter(containerConfig));

m\_windsorContainer.Register(Component.For<IConfigurationProvider>().Instance(provider.Object).IsDefault().Named("TestImplementation"));

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

m\_windsorContainer.Register(Component.For<IWindowManager>().ImplementedBy<WindowManager>());

m\_windsorContainer.Install(Configuration.FromXmlFile(installersConfig));

m\_windsorContainer.AddFacility<TypedFactoryFacility>();

m\_windsorContainer.ResolveAll<IDaoSource>();

var configurator = new CastleDependencyMappingConfigurator(m\_windsorContainer, m\_windsorContainer.ResolveAll<MappingConfiguratorBase>());

configurator.Configure();

}

private WindsorContainer m\_windsorContainer;

[Test]

public void CheckForMisconfiguredComponents()

{

var diagnostic = new PotentiallyMisconfiguredComponentsDiagnostic(m\_windsorContainer.Kernel);

var handlers = diagnostic.Inspect();

if (handlers?.Any() == true)

{

var builder = new StringBuilder();

builder.AppendFormat("Misconfigured components ({0})\r\n", handlers.Count());

foreach (var handler in handlers)

{

var info = (IExposeDependencyInfo) handler;

var inspector = new DependencyInspector(builder);

info.ObtainDependencyDetails(inspector);

}

Assert.Fail(builder.ToString());

}

}

}

}

## Admin test kod

using System;

using System.Linq;

using System.Text;

using System.Threading;

using System.Threading.Tasks;

using System.Windows;

using Castle.MicroKernel.Handlers;

using Castle.MicroKernel.Resolvers.SpecializedResolvers;

using Castle.Windsor;

using Castle.Windsor.Diagnostics;

using Mediaresearch.Framework.DataAccess.BLToolkit.Dao;

using Mediaresearch.Framework.Utilities.Castle.ModelsInteractions.Castle;

using MIR.Media.Admin.Container.Installers;

using NUnit.Framework;

namespace MIR.Media.Admin.Tests

{

[TestFixture]

[Apartment(ApartmentState.STA)]

public class MisconfiguredComponentsTest

{

[SetUp]

public void SetUp()

{

m\_windsorContainer = Container.Container.Current;

m\_windsorContainer.AddFacility(new ListenerRegistrationFacility());

m\_windsorContainer.Kernel.Resolver.AddSubResolver(new CollectionResolver(Container.Container.Current.Kernel));

m\_windsorContainer.Install(Castle.Windsor.Installer.Configuration.FromXmlFile("MIR.Media.Admin.Container.config"));

//m\_windsorContainer.Install(FromAssembly.Containing<AdminAssemblyIdentificator>());

m\_windsorContainer.Install(new ConnectionInstaller());

m\_windsorContainer.Install(

new SplashScreenInstaller(),

new CommonComponentsInstaller(),

new RuntimeResolvedInstaller(),

new ScreensInstaller(),

new OtherComponentsInstaller()

);

m\_windsorContainer.ResolveAll<IDaoSource>();

}

private WindsorContainer m\_windsorContainer;

private void TestMe()

{

m\_windsorContainer.Install(

new SplashScreenInstaller(),

new CommonComponentsInstaller(),

new RuntimeResolvedInstaller(),

new ScreensInstaller(),

new OtherComponentsInstaller()

);

}

public Task<int> RunOnUiAsync(Func<int> f)

{

var dispatcherOperation = Application.Current.Dispatcher.InvokeAsync(f);

return dispatcherOperation.Task;

}

[Test]

public void CheckForMisconfiguredComponents()

{

var diagnostic = new PotentiallyMisconfiguredComponentsDiagnostic(m\_windsorContainer.Kernel);

var handlers = diagnostic.Inspect();

if (handlers?.Any() == true)

{

var builder = new StringBuilder();

builder.AppendFormat("Misconfigured components ({0})\r\n", handlers.Count());

foreach (var handler in handlers)

{

var info = (IExposeDependencyInfo) handler;

var inspector = new DependencyInspector(builder);

info.ObtainDependencyDetails(inspector);

}

Assert.Fail(builder.ToString());

}

}

}

}

## Zmenovadlo test kod

using System.Linq;

using System.Text;

using Castle.MicroKernel.Handlers;

using Castle.MicroKernel.Registration;

using Castle.Windsor;

using Castle.Windsor.Configuration.Interpreters;

using Castle.Windsor.Diagnostics;

using Castle.Windsor.Installer;

using Mediaresearch.Framework.DataAccess.BLToolkit.Dao;

using MIR.Media.Changing2.Core;

using MIR.Media.Changing2.Shell;

using NUnit.Framework;

namespace UnitTests

{

[TestFixture]

public class MisconfiguredComponentsTest

{

[SetUp]

public void SetUp()

{

const string path = "MIR.Media.Changing2.Shell.Container.config";

m\_windsorContainer = new WindsorContainer(new XmlInterpreter(path));

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

m\_windsorContainer.Install(FromAssembly.Containing<ShellAssemblyIdentificator>(), FromAssembly.Containing<CoreAssemblyIdentificator>());

m\_windsorContainer.Resolve<IMediaDataDaoSource>();

}

private WindsorContainer m\_windsorContainer;

[Test]

public void CheckForMisconfiguredComponents()

{

var diagnostic = new PotentiallyMisconfiguredComponentsDiagnostic(m\_windsorContainer.Kernel);

var handlers = diagnostic.Inspect();

if (handlers?.Any() == true)

{

var builder = new StringBuilder();

builder.AppendFormat("Misconfigured components ({0})\r\n", handlers.Count());

foreach (var handler in handlers)

{

var info = (IExposeDependencyInfo) handler;

var inspector = new DependencyInspector(builder);

info.ObtainDependencyDetails(inspector);

}

Assert.Fail(builder.ToString());

}

}

}

}

# Postup:

Je nekolik druhu postupu. Zmenovadlo napr tvori container kteremu dava pomoci xmlInterpretera cestu ke configu. Admin ma zase container.Current. Tam je bezparametricky konstruktor. Viz testy vyse.

## Admin:

V adminovi jsem mel vyjimku, ze MediumVersionDialogViewModel ma zavislost na IMediumVersion. Tato trida se ale tvori jen pomoci new . Nekdo ji chtel resolvovat pomoci factory, ktera se nepouzivala:

namespace MIR.Media.Admin.Screens.MediumAdministration.MediumVersions.MediumVersionDialog

{

public interface IMediumVersionDialogFactory

{

IMediumVersionDialogViewModel Create(IMediumVersion mediumVersion);

void Release(IMediumVersionDialogViewModel model);

}

Ve spravovadle byl RuntimeResolvedInstaller, kde bylo zaregistrovano a taky se to nikde nepouzivalo:

public void Install(IWindsorContainer container, IConfigurationStore store)

{

//container.Register(

// Component.For<IMediumVersionDialogViewModel>().ImplementedBy<MediumVersionDialogViewModel>().LifeStyle.Transient

// );

}

Odstranil jsem Factory i registraci. Zavislost na rozhrani IMediumVersionDialogViewModel neni v zadnem konstruktoru

# IServiceInstaller.ImplementedBy<DatabaseServiceInstaller>

[‎25.‎03.‎2019 14:00]  Petr Holubec:

nz. Co ten installer?

ten instaluje veci z databaze. Musis tam mit nejakou instanci sluzby.

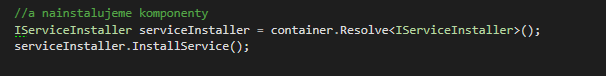
[‎25.‎03.‎2019 14:06]  Peter Hlavenka:

ten test chce vedet, jestli ma Castle vsechny potrebne zavislosti, aby mohl resolvnout jakoukoli zaregistrovanou komponentu. Tenhle installer jako registruje do castlu dalsi komponenty z db?  Mel bych teda vytvorit instanci VideoMatchingServiceWrapper a pro container v teto instanci pustit test ?

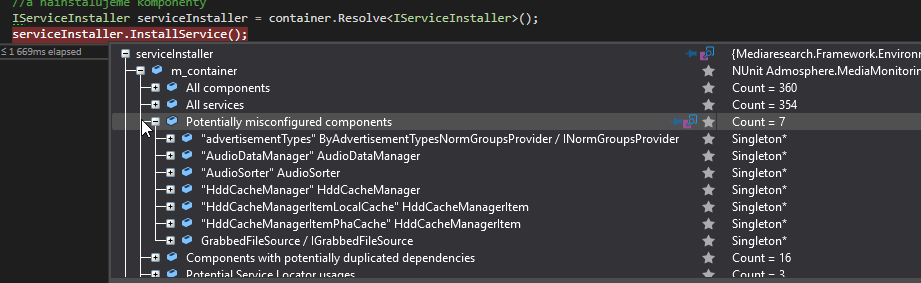
[‎25.‎03.‎2019 14:07]  Petr Holubec:

**jestlize to ma kontrolovat vsechny zavislosti VM, tak musis pustit i tento installer. Ten instaluje dalsi komponenty. Ale na zaklade databaze. Takze musis zajistit pro test databazi a mit v ni potrebna data.**

Admosphere.MediaMonitoring.VideoMatching. ComponentsInstaller:



Pred spustenim metody InstallService() ma castle 342 komponent a 342 sluzeb. Po provedeni metody ma 360 komponent a 354 sluzeb.:



# DependencyMappingConfigurator

**Trida dostane v konstruktoru vsechny tridy, ktere dedi od . Pokud zavolame Configure(), provede se to pro vsechny.**

public class DependencyMappingConfigurator

{

public DependencyMappingConfigurator(IEnumerable<MappingConfiguratorBase> configurators)

{

this.Configurators = configurators;

}

protected IEnumerable<MappingConfiguratorBase> Configurators { get; }

public virtual void Configure()

{

Mapper.Initialize((Action<IMapperConfigurationExpression>) (cfg => this.Configurators.ToList<MappingConfiguratorBase>().ForEach(new Action<MappingConfiguratorBase>(cfg.AddProfile))));

}

}

**Spravne reseni - zaregistrovat do containeru a resolvovat jako ve Zmenovadle:**

**Zmenovadlo ma tridu Changin2CoreMappingConfiguration : DependencyMappingConfiguration.** V bootstrapperu se ale tato trida nevola a to je mozna bug.

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

var mappingConfigurator = m\_windsorContainer.Resolve<DependencyMappingConfigurator>();

mappingConfiguratior.Configure();

**Vystrihovadlo : komponenta neni zaregistrovana, proto:**

(public class CastleDependencyMappingConfigurator : DependencyMappingConfigurator)

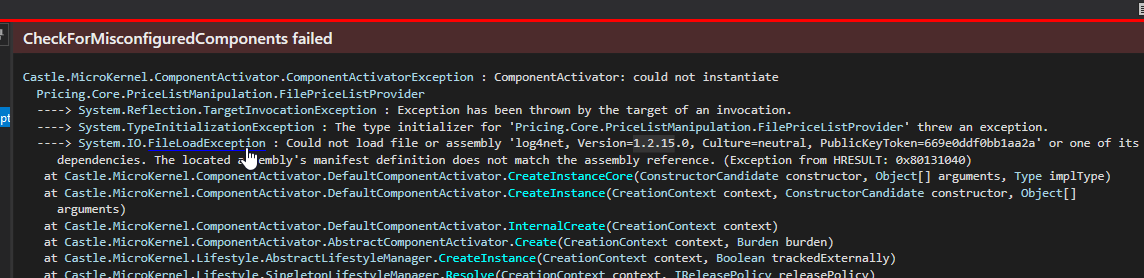
var configurator = new CastleDependencyMappingConfigurator(m\_windsorContainer, m\_windsorContainer.ResolveAll<MappingConfiguratorBase>());

configurator.Configure();

# Problemy:

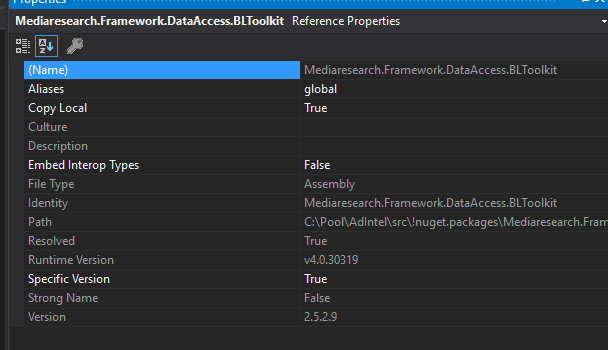
**Problem:**

Verze log4net 1.2.15 pricing



**Reseni:**

Vyresil Mitroz , je potreba zvednout balik Mediaresearch.DataAcces.BLToolkit na verzi



**Problem:**

Container.Current – kdyz pustim dva testCase se dvema configy tak jeden test projde a druhy rekne ,ze container uz takove komponenty ma.

**Reseni:**

Sice blbost ale test projde – dat setter ke containeru a v testu rict:

[TestCase(@"Environments\CZ.Production\MIR.Media.Catching.Container.config")]

[TestCase(@"Environments\BG.Production\MIR.Media.Catching.Container.config")]

public void CheckForMisconfiguredComponents(string path)

{

Core.Container.Container.Current = new Core.Container.Container();

var container = Core.Container.Container.Current;

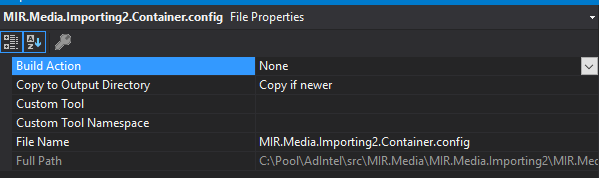
Nebo vykopat Currenta ze softu.

**Problem:**

Pri pousteni testu se dvema configy mi sel jen Importing2 CZ config ale BG ne. Takze reference byla ok

**Reseni:**

Config musi mit v properties nastaveno



**Problem:**

Potrebuju pustit stejny test vicekrat s jinym containter.configem (BG a CZ). Pise mi to, ze dao assemblies uz byly do castlu vlozeny resp item was allreadyadded.

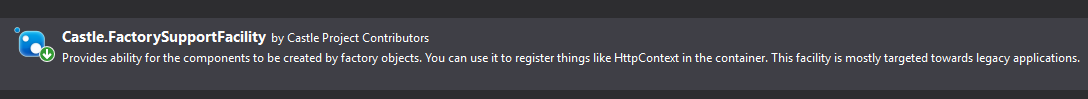
**Reseni:**

**Problem:**

****

**Reseni:**

Nuget do test assembly:



**Problem:**

**BadImageFormat Exception is almost always a problem between a 32bit dll trying to load up a 64bit dll or a DLL or executable is loaded as a 64-bit assembly, but it contains 32-bit features or resources.**

TFS nedela testy – pokud jsou test a shell assembly nastaveny na AnyCpu, testy projdou, pokud je Test = x64 tak se testy vubec neudelaji a kdyz Test = any tak failnou.System.BadImageFormatException

[‎19.‎03.‎2019 14:12]  Filip Čálek:

https://github.com/nunit/nunit3-vs-adapter/issues/122

jak to nastavit na tfs, fakt nevim

ach jo

[‎19.‎03.‎2019 14:13]  Peter Hlavenka:

 kdyz tomu zmenim target na x64, TFS ty testy vubec neprovede. Kdyz jim dam AnyCPU tak failnou. Zkousim nastavit i Shell na AnyCPU..

[‎19.‎03.‎2019 14:14]  Filip Čálek:

ne

shell musi byt x64

[‎19.‎03.‎2019 14:14]  Peter Hlavenka:

brr

[‎19.‎03.‎2019 14:14]  Filip Čálek:

nebo te Mira zabije

:D

[‎19.‎03.‎2019 14:14]  Peter Hlavenka:

takze zpatky ..

**Reseni:**

Zatim nic..

**Problem**

ipAddressInSubnetString bylo null v komponente :

container.Register(Component.For<ILocation>().ImplementedBy<Location.Location>().Named(CommonComponents.HradecLocation)

.DependsOn(

Dependency.OnValue("ipAddressInSubnetString", fileSystemConfiguration.HradecSubnetIpAddress),

Dependency.OnValue("identificator", Location.Location.Values.JH)

));

**Reseni:**

Puvodne tam bylo dependency on component a proto test hlasil zavislost na komponente 192.168.0.20 kterou nemuze najit. Spravne tam ma byt OnValue

**Problem:**

Dependency na typ kt, neni definovany u trid kterym se tento typ dodava pomoci factory.Create(dokladany typ)

**Reseni:**

Nahradit nullable typem v konstruktoru

**Problem: pack -** System.UriFormatException : Invalid URI: Invalid port specified.

V nekolika tridach je bazovka ScreenWithIcon. V registracich techto trid je definovana cesta k ikone takto:

Property.ForKey("IconImageSourcePath").Eq("pack://application:,,,/MIR.Media.Coding.Core;component/Resources/Images/Hint.ico"),

**Reseni:**

Je potreba zaregistrovat namespace Application do testu:

if (!UriParser.IsKnownScheme("pack"))

new System.Windows.Application();

Pokud nezna Application, je potreba pridat referenci na Presentation.Framework.

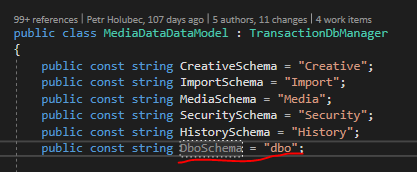
**Problem: IParamsDao**

Castle.MicroKernel.Resolvers.DependencyResolverException : Missing dependency.

Component MIR.Entities.Utilities.Params.Impl.DatabaseParamsSource has a dependency on MIR.Entities.MediaData.Dao.Dbo.IParamsDao, which could not be resolved.

**Spatne Reseni:**

MediaDataModel ma nejake schemata. Mezi nimi chybi schema “dbo” pod ktery je params tabulka.



**Reseni:**

asi jsou dve moznosti:

1. bud nemas DaoSource zaregistrovany jako DependencyDaoSource, nebo...
2. Nekde definujes DaoAssemblies pravdepodobne v tom configu, kterej si mockujes. Toto DaoAssemblies a EnumTableAssemblies je potreba DaoSourcu spravne vyplnit

F.

V Kodovadle v Connection installeru se registruje EntityDaoFactory. V ni se bere connstring a dalsi veci z app.configu, ze sekce databaseConfiguration. Tu ale pro test providera nemame definovanou.

var config = container.Resolve<IConfigurationProvider>().GetConfig<DatabaseConfiguration>();

container.Register(Component.For<EntityDaoFactory>().LifestyleSingleton().Named(Components.EntityDaoFactory)

.DependsOn(

Dependency.OnValue("defaultDbAlias", config.MediaData3DaoSource.DbAlias),

Dependency.OnValue("defaultConnectionString", config.MediaData3DaoSource.ConnectionString),

Property.ForKey("DaoAssemblies").Eq(config.MediaData3DaoSource.DaoAssemblies.ToArray()),

Property.ForKey("ServerTimeZoneId").Eq(config.MediaData3DaoSource.ServerTimeZone),

Property.ForKey("EnumTableAssemblies").Eq(config.MediaData3DaoSource.EnumTableAssemblies.ToArray()),

Dependency.OnValue<ITransactionManager>(new TransactionManager())

));

Reseni – viz [AppConfig](#AppConfig)

**Problem: IndexOutOfRangeException**

FileSystemConfiguration.LocalFileSystemPath vraci string.empty pro (string) this[nameof(LocalFileSystemPath)] pokud se spousti test.

**Reseni:**

Dat tam check na null a pripadne vratit cestu pro test explicitne:

public string LocalFileSystemPath

{

get

{

var folderPath = Environment.GetFolderPath(Environment.SpecialFolder.ApplicationData);

if (!string.IsNullOrWhiteSpace((string) this[nameof(LocalFileSystemPath)]))

{

var fileStoragePath = ((string) this[nameof(LocalFileSystemPath)]).Remove(0, "APP\_DATA".Length).Insert(0, folderPath);

return fileStoragePath;

}

return folderPath + "\\MIR.Media.Coding\\cache\\";

}

}

**Problem:**

Mam v Kodovadle AresConfiguration, kde neni atribut ConfigurationProperty, ale bere se tu **cele pole stringu** ConfigurationCollection. Potreboval jsem nasetupovat providera tak, aby mi na pozadani o AresConfiguration.SuffixiesToRemove.ToArray() vracelo nejakou prazdnou string kolekci. Namockovat se da ale jen rozhrani.

**Reseni:**

Musim vytvorit instanci arese a tu pak predat mockovanemu providerovi:

var ares = new AresConfiguration {SuffixesToRemove = new SuffixiesToRemoveCollection()};

configurationProviderMock.Setup(d => d.GetConfig<AresConfiguration>()).Returns( ares);

m\_windsorContainer.Register(Component.For<IConfigurationProvider>().Instance(configurationProviderMock.Object).IsDefault().Named("CodingTestImplementation"));

Aby bylo mozne setnout propertu SuffixiesToRemoveCollection, musim pridat setter. Jenze ne obycejny, jinak dostanu StackOverflow exception.

public SuffixiesToRemoveCollection SuffixesToRemove

{

get => (SuffixiesToRemoveCollection) base[nameof(SuffixesToRemove)];

set => base[nameof(SuffixesToRemove)] = value;

}

**Problem : App.congig**

App.config (viz nasledujici problem) Vystrihovadlo ma tridu CuttingConfiguration a v app.configu ma konfiguracni sekce. V jednom installeru si z takoveto sekce bere propertu. V aplikaci to normalne funguje, jenze v testu ne. App.config se containeru ani davat nema to je proste spatne.

**Spravne reseni:**

Testovaci app.config ze Shellu nakopiruju rucne do test assembly (nahradim stavajici). Je mi vlastne jedno jake hodnoty jsou v jednotlivych sekcich. Jde mi o to, otestovat Castle. Databaze musi byt testovaci – proto app.config z test environmentu.

**Spatne Reseni: (potrebuje settery v konfiguracich)**

Filip mi poradil v testu si zaregistrovat instanci IConfigurationProvidera jako default, ktery bude jen pro potreby testu a pokud se spusti test, container pouzije tohoto defaultniho providera.

var provider = new Mock<IConfigurationProvider>();

provider.Setup(d => d.GetConfig<CuttingConfiguration>()).Returns(new CuttingConfiguration { ApplicationName = "MIR.Media.Cutting2" });

m\_windsorContainer = new WindsorContainer(new XmlInterpreter(containerConfig));

m\_windsorContainer.Register(Component.For<IConfigurationProvider>().Instance(provider.Object).IsDefault().Named("TestovaciImplementace"));

m\_windsorContainer.Install(Castle.Windsor.Installer.Configuration.FromXmlFile(installersConfig)); // Spusti installery ve kterych se bere config z Castle

**Druhe spatne reseni viz Kodovadlo test:**

const string appConfigPath = @"\MIR.Media.Coding.Shell\app.config";

// ShellConfig

var fileMap = new ExeConfigurationFileMap

{

ExeConfigFilename = GetProjectPath() + appConfigPath

};

var shellExeConfiguration = ConfigurationManager.OpenMappedExeConfiguration(fileMap, ConfigurationUserLevel.None);

// Configurations

var databaseConfiguration = shellExeConfiguration.GetSection("DatabaseConfiguration") as DatabaseConfiguration;

var configurationProviderMock = new Mock<IConfigurationProvider>(); // namockuji providera

configurationProviderMock.Setup(d => d.GetConfig<DatabaseConfiguration>()).Returns(databaseConfiguration); // reknu mu co ma vracet jako databaseConfiguration (bere z shell app.configu)

m\_windsorContainer.Register(Component.For<IConfigurationProvider>().Instance(configurationProviderMock.Object).IsDefault().Named("CodingTestImplementation")); // dam ho windsoru jako Default(), tim nepouzije toho v aplikaci ale tohoto z testu.

**Problem:**

**(**Viz predchazejici problem**)**

System.Configuration.ConfigurationErrorsException : Could not find section 'castle' in the configuration file associated with this domain.

**Reseni:**

m\_windsorContainer = new WindsorContainer(new XmlInterpreter(config)); Tohle by melo dat app.config. Jenze v testovaci assembly mi to neda ani kdyz mu to dam explicitne (jako tam davam container.config).

**Problem:**

Neprojde installerem kt. je definovany v configu. Ve Vystrihovadle je Installers.config, ve kterem jsou definovany installery, ktere se maji nainstalovat. Neco jako je v Shedule.

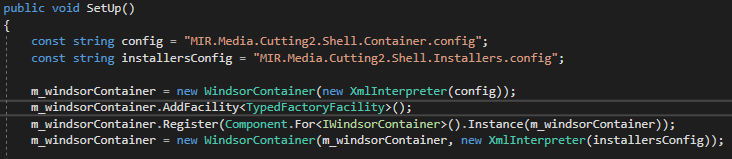
Tohle (je v kodu) nefunguje:

//m\_windsorContainer.Install(Castle.Windsor.Installer.Configuration.FromXmlFile(new XmlInterpreter(installersConfig).ToString()));

Oprava: funguje to! Viz test ve Vystrihovadle – musi byt reference na Shell

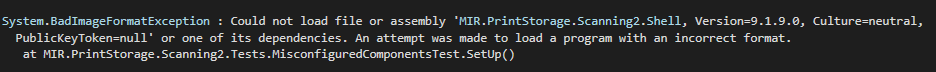
**Reseni:**

Pokud chci v unit testu pouzit tento installers.config, musim si vytvorit novy container, ktery vezme toho prvniho (ktery uz zna container.config) jako parenta a jako druhy argument dostane configuration interpretera.

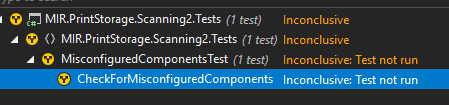


**Problem:**

BadImageFormatException

****

**Inconclusive:**

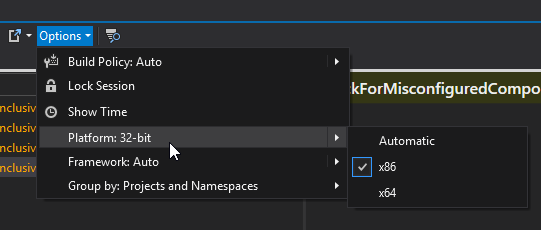
****

**Reseni:**

BadImageFormatException  :D

Filip: shel je 64 a testy ne co?

Stalo se to u skenovadla. Tam je Shell projekt porad na x86. Timpadem musi byt x86 i testovaci projekt jinak dostanu inconclusive error. V nastaveni unit testu musim zmenit platformu a test projde. Jak se s tim popere TFS nevim.



[‎11.‎03.‎2019 13:14]  Peter Hlavenka:

Cau, je nejaky duvod proc je projekt MIR.Printstorage.Scanning2.Shell na  platform target x86 ?  Muzu zmenit na x64?

[‎11.‎03.‎2019 13:14]  Petr Mitrofan:

musi zustat x32 kvuli ovladacum scaneru

[‎11.‎03.‎2019 13:14]  Peter Hlavenka:

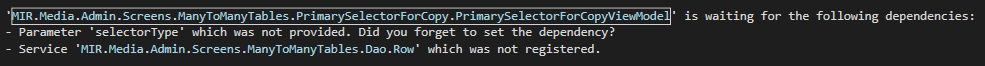
ok

Skenovadlo dostane svuj prvni unit test :D

[‎11.‎03.‎2019 13:16]  Petr Mitrofan:

zazraky e deji

**Problem:**

****

Pri resolvovani instance tridy (za rozhranim) se castlu predaji typy, na ktere si stezuje unit test:

IPrimarySelectorForCopyViewModel primarySelectorForCopyViewModel =

m\_container.Resolve<IPrimarySelectorForCopyViewModel>(new { selectorType = PrimarySelectorForCopyType.CopyTo, targetRow = m\_primaryTableViewModel.SelectedRow });

**Reseni:**

**Problem:**

STA Registrace ktere probihaji v aplikaci jsou v main threadu. Registrace v UnitTestu jsou ve worker threadu. Nekdy se v installeru vytvari instance trid a pak se zaregistruji jako .Instance() . Admin.ColumnHiderControl vola uz v konstruktoru NotifyOfPropertyChange(). Musi byt teda STA

**Reseni:**

Do Unit testu pridat atribut :

[TestFixture]

[Apartment(ApartmentState.STA)]

**Problem:**

Container.current => pri vytvareni currentu v testu se musi container vytvorit takto:

public void SetUp()

{

m\_windsorContainer = Container.Container.Current;

**Reseni:**

**Problem:**

Zhulena reference , nebo neni mozne pouzit nejaky typ.

**Reseni:**

Reference – Testovaci assembly musi mit reference na vsechny projekty, ktere chce pouzivat a to dokonce ve vyssi nebo rovne verzi .netu nez je pouzivana assembly

**Proble****m:**

Changing: Unit test se pousti nad appdata a proto nevidi na container.config. Tohle jsem prepral a skladam si cestu.

**Reseni:**

**Nejprve jsem se snazil poskladat stejny string jako dostavam v bootstrapperu. Na TFS to stejne padalo**

private static string GetProjectPath()

{

var codeBase = Assembly.GetExecutingAssembly().CodeBase;

var pathItems = codeBase.Split(Separator);

var start = pathItems.ToList().FindIndex(x => string.Equals("C:", x));

var end = pathItems.ToList().FindIndex(x => string.Equals("MIR.Media.Changing2", x)) + 1;

var projectPath = string.Join(Path.DirectorySeparatorChar.ToString(), pathItems.Skip(start).Take(end - start));

return projectPath;

}

m\_windsorContainer = new WindsorContainer(new XmlInterpreter(Path.Combine(GetProjectPath(), "MIR.Media.Changing2.Shell\\bin\\Debug\\MIR.Media.Changing2.Shell.Container.config")));

**Nakonec stacilo misto sloziteho skladani stringu jen tohle:**

const string path = "MIR.Media.Changing2.Shell.Container.config";

m\_windsorContainer = new WindsorContainer(new XmlInterpreter(path));

**Problem:**

Tim, ze je test v jiné assembly nez je config, nedokaze zaregistrovat komponenty z configu, protože ty mají definovanou relativni cestu : type="MIR.Media.Changing2.Shell.Installers.MediaDataDbConfiguration, MIR.Media.Changing2.Shell">

**Reseni:**

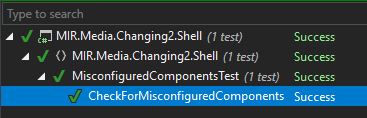
Resenim je presunout tento test do assembly, ve ktere jsou komponenty.

**Problem**:

Changing: Vytvorim instanci containeru, pustim installer, resolvnu dependencyDaoSource a dostanu error ze viewModel není registrovany. Pritom registrace VM a všech zavislosti probiha v installeru, kterym to urcite projde.

**Reseni:**

Musel jsem vytvorit novy installer – ShellInstaller. IWindowManager se totiz registroval v bootstrapperu, cimz ho unit test nenasel, protoze ten instaluje jen z installeru.



**Problem:**

Spusteni aplikace (ne testu) : ShellInstaller se mi instaloval dvakrat, protoze nejprve ho pustim v bootstrapperu kvuli IWindowManagerovi, ktereho potrebuju v main threadu, a pak se pousti jeste jednou asynchronne v akci ktera bezi na pozadi za splashscreenem:

m\_globalContainer.Install(new ShellInstaller());

void InitAction()

{

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

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

**Reseni:**

Vytvori si klic ktery se kontroluje pri instalaci. Dvakrat se vola jen instalace, instance ShellInstalleru se vytvari jen jednou.

public class ShellInstaller : IWindsorInstaller

{

private static string ShellInstallerKey { get; set; }

private static readonly object Lock = new object();

public ShellInstaller()

{

ShellInstallerKey = Guid.NewGuid().ToString();

}

public void Install(IWindsorContainer container, IConfigurationStore store)

{

if (!container.Kernel.HasComponent(ShellInstallerKey))

{

container.Register(Component.For<IWindowManager>().ImplementedBy<WindowManager>().Named(ShellInstallerKey));

}

}

}

**Problem:**

Zmenovadlo. U me to zbuildit slo a unit test prosel, u Karla taky, u Filipa a na TFS ne.

Viz [Problem1](#Problem1)

**Reseni:**

Viz [Problem1](#Problem1)

**Problem:**

Assembly has no entry point. (nema main metodu statickou) Kdyz jsem ve vystrihovadle odstranil z assembly s testama tridu UnitTests , zacal problem

class UnitTest

{

static void Main(string[] args)

{

}

}

**Reseni:**

Vrati ho zpet.

# Hotove softy - checkboxy:

Seznam softu, aplikaci a sluzeb

Admin container.curent **pridal jsem setter ke container.current**

Approving **pridal jsem setter ke container.current**

Catching current **pridal jsem setter ke container.current**

Coding app domain, ale pod ifem

Cutting

Changing

Importing2

Norming container.current **pridal jsem setter ke container.current**

Pricing **pridal jsem setter ke container.current**

Scanning

SimLog **pridal jsem setter ke container.current**

TvLogGenerator **pridal jsem setter ke container.current**

Aaaaaaaaaaaaaaaaaaaaaaaaaaaa softy hotove

Sluzby

**Jak spusti sluzbu a zjistit kde zacina:**

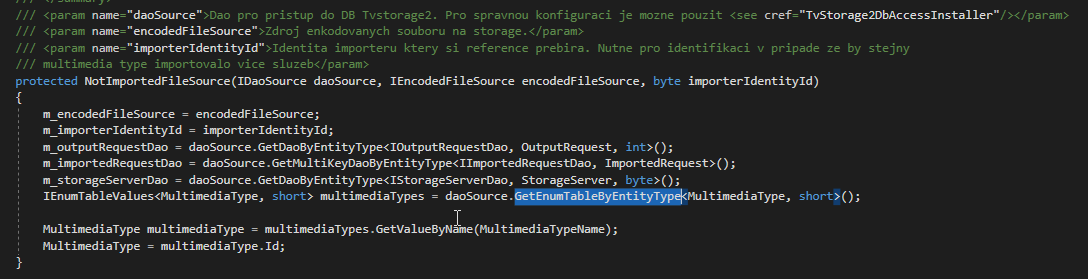
Natavim projekt ktery konci na sercvice jako start project. Macknu F11, dostanu se do tridy Program.cs do metody Main. Tady se obvykle vytvori nejaky serviceWrapper ve kterem se pousti installery a registruji se treba nejake komponenty. Nekdy se to deje uz v metode Main.

Mediaresearch.Framework.Services.ServiceBase.Program.Main(

() => new PostOfficeServiceWrapper(), sw => new PostOfficeService(sw), () => isTest ? "MIR.Export.PostOffice.Service.Container.TEST.config" : "MIR.Export.PostOffice.Service.Container.config");

**Jak zjistit, jestli nejaky konstruktor chce do db:**

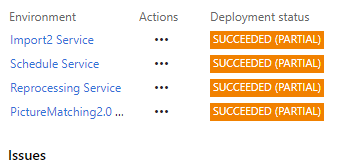
Pokud u sluzeb chci vedet, jestli nekde v konstruktoru je potreba pristup do db, poskodim nejak conn string a test mi upadne a ukaze mi misto kde to pada. Pujdu po zavislostech az se dostanu na misto kde chce konstruktor dao tridy, nebo nejaky enum z databaze (v lepsim pripade). Pokud chce jen tohle, muzu pouzit ostry config. Prikladem je videomatching, kde jsem poskodil tvStorageConnectionString a dostal jsem se az sem:



**Na Cz se vydavaji tyto sluzby:**

****

**Na BG se vydavaji jen:**

****

Videomatching

DuplicityHunter

Picturematching

Reprocessing registration service

Import service

Schedule service

Export thumbnail service tohle snad Castle ani nepouziva

PostOffice service

Export creative service neni v master sln mame jen export.creative.download

Pricing service neni v master sln – repo Hg/Pricing

# Tridy u kterych jsem vlozil zavislost jako volitelny parametr a pak to neresil – strasna blbost:

= null

1. public **~~VideoDuplicityHunterDialogViewModel~~**(IMediaDataDaoSource daoSource, ITvMessagesInfoViewModelFactory factory, IFileSystemSelector selector, IUser currentUser,
2. public **~~MotiveEditDialogViewModel~~**(MotiveVersionDao motiveVersionDao, IMessageViewer infoMessageViewer, IMessagingService messageService, int? motiveVersionId = null)
3. public **~~ChangeMotiveVersionDialogViewModel~~**(MotiveVersionDao motiveVersionDao, int? motiveId = null)
4. public **~~MotiveVersionsDialogViewModel~~**(MotiveGridViewModel motiveGridViewModel, EntityDaoFactory factory, ITaskQueue taskQueue, int? motiveId = null)
5. public **~~EditMediaMatchingDefinitionViewModel~~**(IMediumDao mediumDao, IMediumVersionDao mediumVersionDao, AdvertisementTypeDao advertisementTypeDao, VideoProcessStatusDao videoProcessStatusDao, IEnumTableValues<CodingPlausibility, byte> codingPlausibilities,
6. public **~~PrimarySelectorForCopyViewModel~~**( ManyToManyTablesDao manyToManyTablesDao, ManyToManyTableSelectorViewModel manyToManyTableSelectorViewModel, PrimarySelectorForCopyType selectorType = PrimarySelectorForCopyType.CopyFrom, Row targetRow = null)
7. **~~MessageHistoryDialogViewModel~~**

**bbbb91c94770f3239652dfbc2cbda12208a9359c**

**~~Broadcasting~~**

**~~ImageDialogViewerViewModel~~**

**~~BroadcastingDescriptionViewModel~~**

**L~~ogEditorViewModel~~**

**~~ProgrammeTranslationViewModel~~**

**~~ProgrammeViewModel~~**

**~~DownloadedVideoInfoViewModel~~**

**~~VideoPlayerViewModel~~**

**~~VideoPlayerViewModelWrapper~~**