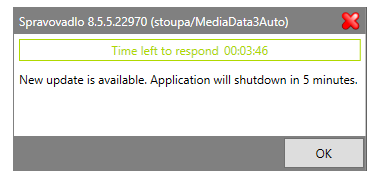
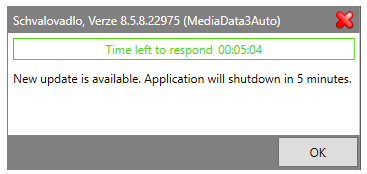
**Vycet Aplikaci ktere maji ClickOnce:**

1. ~~Admin~~   Je jen anglicky



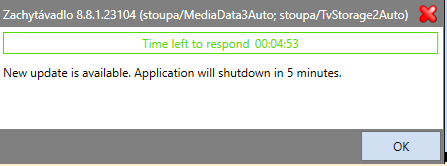
1. ~~Approving~~  External Program Failed: C:\Program Files (x86)\Microsoft Visual Studio\2017\Professional\MSBuild\15.0\Bin\MSBuild.exe (return code was 1) **Reseni: zbuildit pomoci new-build a pak publish**



**Anglicky je to proto, ze stare aplikace jedou na stare verzi .NetFrameworku a nove jedou na 4.7.1 Tim, ze deploy je v jinem threadu se to nepocesti**

1. ~~Catching~~  Nefunguje => zkusit nareferencovat rovnou a zalogovat => zacinam deploy .. atd.

Catching se netvori na i:\ ale na c:\ [C:\Pool\Admosphere\Publish\MIR.Media.Catching\BG\Test\Catching\_BG\_Test.application](file:///C:\Pool\Admosphere\Publish\MIR.Media.Catching\BG\Test\Catching_BG_Test.application) proto po spusteni z i:\ nefunguje.



[‎6/‎26/‎2018 2:18 PM] Filip Čálek:

trida Properties

 public TimeSpan DeploymentServiceCheckTimeSpan { get; }

           public TimeSpan DeplymentServiceShutdownTimeSpan { get; }

dal sis breakpoint tam, kde initializujes deploymentService?

[‎6/‎26/‎2018 2:19 PM] Peter Hlavenka:

ano tam mi to vypsalo co potrebuju

[‎6/‎26/‎2018 2:19 PM] Filip Čálek:

mas to ted otevreny?

[‎6/‎26/‎2018 2:20 PM] Peter Hlavenka:

otviram

[‎6/‎26/‎2018 2:20 PM] Filip Čálek:

jdu

 Chybi settery

Zalogoval jsem to , zkusil jsem nareferencovat naprimo , ne pres balicek, info.UpdateAvailable je false , tim padem to do tela metody nevleze, rovnou ji to opusti a vypise end

## Framework.DeploymentService:

using System;

using System.Deployment.Application;

using System.Reflection;

using System.Threading;

using System.Windows;

using System.Windows.Threading;

using log4net;

using MessageBox = Mediaresearch.Framework.Gui.MessageBoxControl.MessageBox;

namespace Mediaresearch.Framework.Gui.Deploy

{

public interface IDeploymentService

{

/// <summary>

/// Metoda nastavi timer, ktery v danem intervalu hleda nove aktualizace a po nalezeni aplikaci ukonci dle parametru.

/// </summary>

/// <param name="checkInterval">V jakem intervalu se maji hledat nove aktualizace</param>

/// <param name="shutdownTimeout">Po nalezeni nove aktualizace je uzivatel informovan a aplikace je ukoncena po uplynuti timeoutu</param>

void InitializeAutoUpdater(TimeSpan checkInterval, TimeSpan shutdownTimeout, string applicationName = null);

}

public class DeploymentService : IDeploymentService

{

private static readonly ILog Log = LogManager.GetLogger(MethodBase.GetCurrentMethod().DeclaringType);

//Pokud uzivatel nereaguje na upozorneni, je aplikace i tak ukoncena behem jedne hodiny (prozatim zadano v podobe konstanty, pokud se ukaze, ze je potreba to menit, muzeme parametrizovat)

private const int ResponseTimeout = 360000;

private readonly Timer m\_updateTimer;

private readonly Timer m\_shutdownTimer;

private bool m\_initialized;

private TimeSpan m\_shutdownTimeout;

public DeploymentService()

{

m\_updateTimer = new Timer(d => CheckForUpdate(), null, TimeSpan.FromMilliseconds(-1), TimeSpan.FromMilliseconds(-1));

m\_shutdownTimer = new Timer(d => ShutdownApp(), null, TimeSpan.FromMilliseconds(-1), TimeSpan.FromMilliseconds(-1));

}

private string ApplicationName { get; set; } = string.Empty;

public void InitializeAutoUpdater(TimeSpan checkInterval, TimeSpan shutdownTimeout, string applicationName = null)

{

if (!m\_initialized)

{

m\_updateTimer.Change(checkInterval, checkInterval);

m\_initialized = true;

}

ApplicationName = applicationName ?? string.Empty;

m\_shutdownTimeout = shutdownTimeout;

}

private void CheckForUpdate()

{

try

{

if (!System.Diagnostics.Debugger.IsAttached)

{

Log.DebugFormat("Starting deploymentService");

ApplicationDeployment updateCheck = ApplicationDeployment.CurrentDeployment;

UpdateCheckInfo info = updateCheck.CheckForDetailedUpdate(); Tudy to jeste projde

Log.DebugFormat($"DeploymentService - info.UpdateAvailable = {info.UpdateAvailable}"); False

//Log.DebugFormat($"DeploymentService - info.AvailableVersion = {info.AvailableVersion}" ); Nedostupne => vleze do vyjimky a vypise Update is not available

//Log.DebugFormat($"DeploymentService - info.IsUpdateRequired = {info.IsUpdateRequired}");

//Log.DebugFormat($"DeploymentService - info.MinimumRequiredVersion = {info.MinimumRequiredVersion}");

if (info.UpdateAvailable == false)

{

Log.DebugFormat("DeploymentService - info.UpdateAvailable = true");

var dispatcher = Application.Current.Dispatcher;

if (info.IsUpdateRequired)

{

Log.DebugFormat("DeploymentService - info.IsUpdateRequired = true");

m\_updateTimer.Change(TimeSpan.FromMilliseconds(-1), TimeSpan.FromMilliseconds(-1));

dispatcher.Invoke(DispatcherPriority.Send, (Action)(() =>

{

try

{

MessageBox.Show(string.Format(DeployResources.UpdateAvailable, m\_shutdownTimeout.TotalMinutes),

ApplicationName, MessageBoxButton.OK, MessageBoxImage.Warning, ResponseTimeout);

}

catch (TimeoutException)

{

ShutdownApp();

}

}));

m\_shutdownTimer.Change(m\_shutdownTimeout, TimeSpan.FromMilliseconds(-1));

}

else

{

Log.DebugFormat("DeploymentService - info.IsUpdateRequired = false");

dispatcher.Invoke(DispatcherPriority.Send, (Action) (() =>

{

if (MessageBox.Show(DeployResources.UpdateAvailableQuestion, ApplicationName, MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.Yes)

{

ShutdownApp();

}

}));

}

}

}

Log.DebugFormat("DeploymentService - The end");

}

catch (Exception ex)

{

Log.ErrorFormat("{0}: {1}", DeployResources.ExceptionMessage, ex.Message);

}

}

private void ShutdownApp()

{

Log.DebugFormat("DeploymentService - ShutdownApp()");

var dispatcher = Application.Current.Dispatcher;

if (!dispatcher.CheckAccess())

{

dispatcher.Invoke(DispatcherPriority.Send, (Action) (() => Application.Current.Shutdown()));

}

else

{

Application.Current.Shutdown();

}

}

#region IDispose Implementation

private bool m\_disposed;

~DeploymentService()

{

Dispose(false);

}

public void Dispose()

{

//pass true indicating managed resources can be freed as well e.g. our code called

//dispose instead of the .NET framework

Dispose(true);

GC.SuppressFinalize(this);

}

private void Dispose(bool disposing)

{

if (!m\_disposed)

{

m\_updateTimer.Dispose();

m\_shutdownTimer.Dispose();

}

m\_disposed = true;

}

#endregion

}

}

1. ~~Changing~~ 2 ~~Ok~~
2. ~~Coding~~  Na defaultu nejde spustit => Pricing uz je zbalickovany, je potreba odstranit stare reference a nainstalovat z balicku

Odstranit reference, nejlepe cely subrepozitar a pak nainstalovat Pricing.Ads

1. Cutting2

[ConfigurationProperty("UpdateShutDownTimeOutInMinutes", IsRequired = true)]

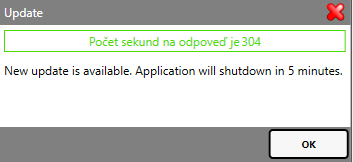
public int UpdateShutDownTimeOutInMinutes

{

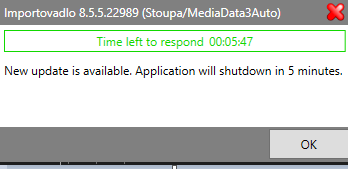
get { return (int)this["UpdateShutDownTimeOutInMinutes"]; }

set { this["UpdateShutDownTimeOutInMinutes"] = value; }

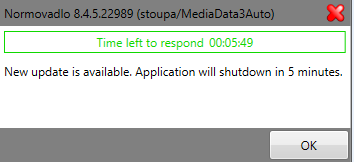
}

zatim v cuttingu

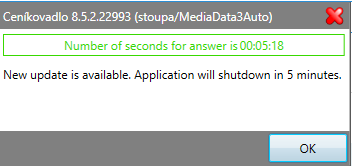
1. ~~Importing 2  Ok~~



1. ~~Norming~~



1. ~~Scanning  Ok~~
2. ~~Pricing~~



1. ~~Shedule  Nema spousteci projekt Shell => nereseno~~

1. SimLog  InversionOfControl is not initialized => resi Filip

[‎6/‎19/‎2018 3:35 PM] Filip Čálek:

SimLog pada, protoze v db chybi application name

Pada to tady :

protected SimLogTabPageConductorBase(IEnumerable<T> items, SimLogMembershipPermissionsProvider permissionsProvider, string title = null)

{

PermissionsProvider = permissionsProvider;

base.DisplayName = title;

Items.AddRange(items);

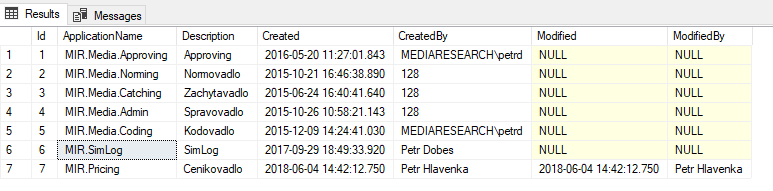
T toActivate = Items.FirstOrDefault(item => **permissionsProvider.UserPermissions**.FirstOrDefault(up => up.PermissionCode == item.DisplaySecurablePermission?.PermissionCode && up.SecurableCode == item.DisplaySecurablePermission?.SecurableCode) != null);

if (toActivate != null)

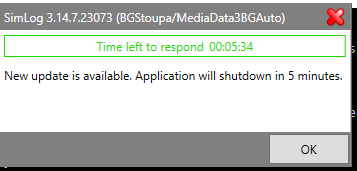
ActivateItem(toActivate);

}

**Reseni: SimLog chybi na MediaData3Auto v Membership.Application (obrazek je z MediaData3BG) => SimLog je jen na BG , v cechach neni potreba => vyrabet jen BGClickOnce**

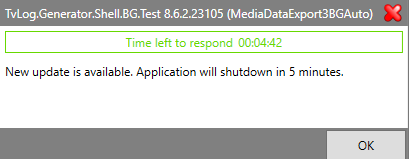
****

**Simlog se navic tvori na jinem miste a to :** [**c:\Pool\Admosphere\Publish\MIR.SimLog\BG\Test\SimLog\_BG\_Test.application**](file:///c:\Pool\Admosphere\Publish\MIR.SimLog\BG\Test\SimLog_BG_Test.application)



**Pozor SimLog se tvori jinde nez na i:\**

1. TvLogGenerator  Mel by fungovat test uz necham na Ivu. Zhuleno na defaultu ze ktereho jsem vysel a pak I u mne.

****

# Kod v bootstrapperu:

TableCache.Initialize(Container.Container.Current.Resolve<IEnvironmentDaoSource>());

ConfigureDeploymentService();

}

private static void ConfigureDeploymentService()

{

var holder = Container.Container.Current.Resolve<ConnectionStringsHolder>();

var version = $"{Localisation.AdminWindowTitle} {Assembly.GetEntryAssembly().GetName().Version}";

var appVersionInfo = $"{version} ({holder.DefaultDataSource})";

var properties = Container.Container.Current.Resolve<Container.Properties>();

var deploymentService = Container.Container.Current.Resolve<IDeploymentService>();

var checkTimeSpan = properties.DeploymentServiceCheckTimeSpan;

var shutDownTimeSpan = properties.DeplymentServiceShutdownTimeSpan;

deploymentService.InitializeAutoUpdater(checkTimeSpan, shutDownTimeSpan, appVersionInfo);

}

1)

var holder = Core.Container.Container.Current.Resolve<ConnectionStringsHolder>();

var version = $"{Localisation.Coding} {Assembly.GetEntryAssembly().GetName().Version}";

var appVersionInfo = $"{version} ({holder.GetDataSource(holder.MediaDataConnectionString)})";

2