# Complete wpf tutorial:

<https://www.wpf-tutorial.com/dialogs/the-messagebox/>

# MessageBoxViewModel pouzity v Changingu

<Window

x:Class="MIR.Media.Changing2.Gui.Views.MessageBoxView"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:gui="clr-namespace:MIR.Media.Changing2.Gui"

xmlns:local="clr-namespace:MIR.Media.Changing2.Gui.Views"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:viewModels="clr-namespace:MIR.Media.Changing2.Gui.ViewModels"

Width="650"

Height="200"

d:DataContext="{d:DesignInstance viewModels:MessageBoxViewModel}"

d:DesignHeight="500"

d:DesignWidth="480"

MouseLeftButtonDown="MessageBox\_OnMouseLeftButtonDown"

WindowStartupLocation="CenterScreen"

WindowStyle="None"

mc:Ignorable="d">

<Window.Resources>

<BitmapImage x:Key="White" UriSource="/MIR.Media.Changing2.Gui;component/Icons/white.jpg" />

<Style x:Key="ColoredButton" TargetType="Button">

<Setter Property="BorderThickness" Value="1.2" />

<Setter Property="Background" Value="Transparent" />

<Setter Property="Width" Value="35" />

<Setter Property="Margin" Value="10" />

<Setter Property="Template">

<Setter.Value>

<ControlTemplate TargetType="{x:Type Button}">

<Border

Name="Border"

BorderBrush="{TemplateBinding BorderBrush}"

BorderThickness="{TemplateBinding BorderThickness}"

CornerRadius="5">

<Border

Name="InnerBorder"

Background="{TemplateBinding Background}"

CornerRadius="5">

<ContentPresenter

Margin="{TemplateBinding Padding}"

HorizontalAlignment="Center"

VerticalAlignment="Center"

RecognizesAccessKey="True" />

</Border>

</Border>

<ControlTemplate.Triggers>

<EventTrigger RoutedEvent="MouseEnter">

<EventTrigger.Actions>

<BeginStoryboard>

<Storyboard>

<ColorAnimation

Storyboard.TargetProperty="(Button.Background).(SolidColorBrush.Color)"

From="Transparent"

To="Snow"

Duration="0:0:0.3" />

</Storyboard>

</BeginStoryboard>

</EventTrigger.Actions>

</EventTrigger>

<EventTrigger RoutedEvent="MouseLeave">

<EventTrigger.Actions>

<BeginStoryboard>

<Storyboard>

<ColorAnimation

Storyboard.TargetProperty="(Button.Background).(SolidColorBrush.Color)"

From="Snow"

To="Transparent"

Duration="0:0:0.3" />

</Storyboard>

</BeginStoryboard>

</EventTrigger.Actions>

</EventTrigger>

<Trigger Property="IsPressed" Value="True">

<Setter Property="RenderTransform">

<Setter.Value>

<ScaleTransform ScaleX=".9" ScaleY=".9" />

</Setter.Value>

</Setter>

<Setter Property="RenderTransformOrigin" Value=".5,.5" />

</Trigger>

</ControlTemplate.Triggers>

</ControlTemplate>

</Setter.Value>

</Setter>

</Style>

</Window.Resources>

<Window.Background>

<ImageBrush ImageSource="{StaticResource White}" />

</Window.Background>

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="Auto" />

<RowDefinition Height="Auto" />

<RowDefinition Height="Auto" />

</Grid.RowDefinitions>

<TextBox

Grid.Row="0"

Grid.Column="0"

Height="80"

Margin="20,20,20,0"

HorizontalContentAlignment="Center"

VerticalContentAlignment="Center"

AcceptsReturn="True"

Background="{x:Null}"

BorderThickness="0"

FontSize="16"

FontWeight="DemiBold"

IsReadOnly="True"

Text="{Binding Path=Message}"

TextAlignment="Center"

TextWrapping="Wrap" />

<!-- Buttons -->

<Grid

Grid.Row="1"

HorizontalAlignment="Center"

VerticalAlignment="Bottom"

Visibility="{Binding Path=ButtonsVisibility}">

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*" />

<ColumnDefinition Width="\*" />

</Grid.ColumnDefinitions>

<Button

Grid.Column="0"

Width="200"

Height="27"

Command="{Binding SaveCommand}"

Content="{x:Static gui:GuiResources.Yes}"

Style="{StaticResource ColoredButton}" />

<Button

Grid.Column="1"

Width="200"

Height="27"

Command="{Binding StornoCommand}"

Content="{x:Static gui:GuiResources.No}"

Style="{StaticResource ColoredButton}" />

</Grid>

<Button

Grid.Row="1"

Width="200"

Height="27"

Command="{Binding StornoCommand}"

Content="{x:Static gui:GuiResources.Ok}"

Style="{StaticResource ColoredButton}"

Visibility="{Binding Path=OkButtonVisibility}" />

<TextBlock

Grid.Row="2"

Margin="10"

HorizontalAlignment="Center"

VerticalAlignment="Center"

FontSize="16"

FontWeight="DemiBold"

Foreground="{Binding Path=WarningForegroung}"

Text="{Binding Path=Warning}" />

</Grid>

</Window>

using System;

using System.Windows;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Threading;

using Caliburn.Micro;

using Mediaresearch.Framework.Gui;

namespace MIR.Media.Changing2.Gui.ViewModels

{

public class MessageBoxViewModel : Screen

{

private string m\_message;

private string m\_warning;

private Brush m\_warningForegroung = Brushes.Black;

public MessageBoxViewModel(string message, bool yesNoButtons, string warning)

{

SaveCommand = new RelayCommand(DoSave);

StornoCommand = new RelayCommand(DoStorno);

Message = message;

ButtonsVisibility = yesNoButtons ? Visibility.Visible : Visibility.Collapsed;

Warning = warning;

}

public MessageBoxViewModel(string message, bool yesNoButtons, bool okButton, string warning)

{

SaveCommand = new RelayCommand(DoSave);

StornoCommand = new RelayCommand(DoStorno);

Message = message;

ButtonsVisibility = yesNoButtons ? Visibility.Visible : Visibility.Collapsed;

OkButton = okButton;

Warning = warning;

}

public MessageBoxViewModel(string message, bool yesNoButtons, int duration, string warning)

{

SaveCommand = new RelayCommand(DoSave);

StornoCommand = new RelayCommand(DoStorno);

Message = message;

ButtonsVisibility = yesNoButtons ? Visibility.Visible : Visibility.Collapsed;

Timer = new DispatcherTimer();

RunningTime = 0;

Timer.Interval = TimeSpan.FromMilliseconds(100);

Timer.IsEnabled = true;

Timer.Tick += CloseMe;

Duration = duration;

Warning = warning;

Timer.Start();

}

private DispatcherTimer Timer { get; }

private int RunningTime { get; set; }

public ICommand SaveCommand { get; }

public ICommand StornoCommand { get; }

private int Duration { get; }

public string Message

{

get => m\_message;

set

{

m\_message = value;

NotifyOfPropertyChange();

}

}

public Visibility ButtonsVisibility { get; set; }

private bool OkButton { get; }

public Visibility OkButtonVisibility => OkButton ? Visibility.Visible : Visibility.Collapsed;

public string Warning

{

get => m\_warning;

set

{

m\_warning = value;

NotifyOfPropertyChange();

}

}

public Brush WarningForegroung

{

get => m\_warningForegroung;

set

{

m\_warningForegroung = value;

NotifyOfPropertyChange();

}

}

private void CloseMe(object sender, EventArgs e)

{

RunningTime += 100;

if (RunningTime > Duration)

{

Timer.Stop();

TryClose();

Timer.Tick -= CloseMe;

}

}

private void DoStorno()

{

TryClose(false);

}

private void DoSave()

{

TryClose(true);

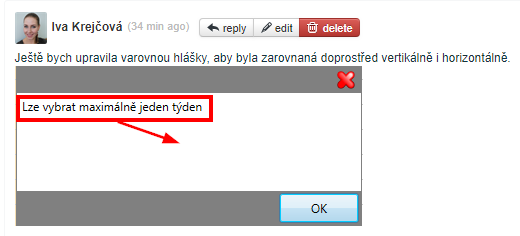
}

}

}

# MessageBox z MediaResearch.Framework vs System.Windows.MessageBox

Pokus z nejakeho duvodu nevyhovuje MB z Frameworku



Muzu pouzit ten z Windows :

using MessageBox = System.Windows.MessageBox;

# Jak poskladat message z kodu a Localisation

V kodu:

MessageBox.Show(string.Format(Localisation.ReallyDeleteMMsAndCreatives, messages.Length), Localisation.Deletion, MessageBoxButton.YesNo,

**V Localisation:**

Opravdu chcete odstranit označené {0} MediaMessage a kreativy ?