How to use Cimbalino Windows Phone Toolkit - MessageBoxService

This code example shows how display message boxes in your app using `MessageBoxService` from the Cimbalino Windows Phone Toolkit.

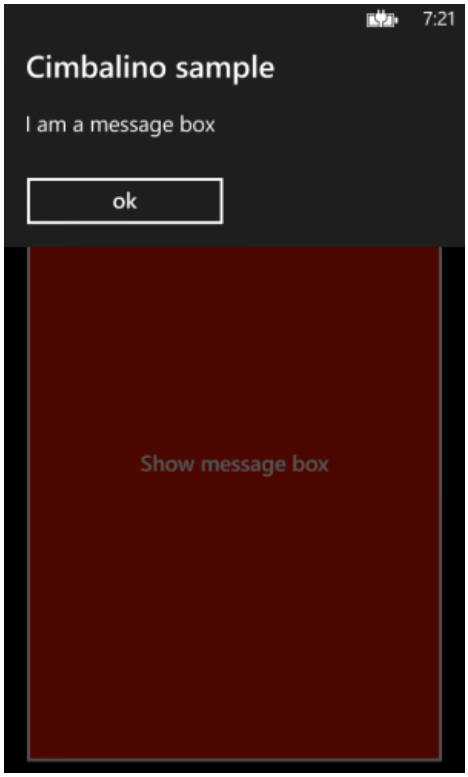
Introduction

The *Cimbalino Windows Phone Toolkit* delivers a set of useful and powerful MVVM-compatible tools and services to help developers build Silverlight applications for Windows Phone. The Toolkit is divided in projects which deliver different features, ranging from base MVVM services and helpers, through to code for background agents, and for accessing media library, location services etc. The base project (Cimbalino.Phone.Toolkit) contains base MVVM services, some very useful converters, helper classes and extension methods.

The Cimbalino Toolkit's Message Box service is used to display message boxes from the model of an MVVM-light app. The kit provides both the IMessageBoxService interface and its implementation MessageBoxService required to register the service in MVVM Light (note that MVVM and the MVVM Light Toolkit are not "preconditions" to use this service).

This article explains how to use the service in an MVVM Light app. Developers should first read How to use MVVM Light Toolkit for Windows Phone for a general overview of MVVM Light and how to use services.

A screenshot of the example app displaying a message box is shown below.



Building the example code

The source code for the code example is available here: MessageBoxService Example (github).

To build the source code you will also need the MVVM Light Toolkit and the *Cimbalino Windows Phone Toolkit*. Their packages are available in the Nuget Package Manager. For more details see:

- How to install MVVM Light Toolkit for Windows Phone
- How to install Cimbalino Windows Phone Toolkit packages.

The sample is built using MVVM Light and the article does not go into great detail about the plumbing/infrastructure provided by the toolkit. For more information see:

- How to use MVVM Light Toolkit for Windows Phone

(C) Copyright Nokia 2014. All rights reserved.

Registering the service

We should register each service in ViewModelLocator, as following:

```
using Cimbalino.Phone.Toolkit.Services;

    /// This class contains static references to all the view models in the
    /// application and provides an entry point for the bindings.
    public class ViewModelLocator
    {
        /// Initializes a new instance of the ViewModelLocator class.
        public ViewModelLocator()
        {
            ServiceLocator.SetLocatorProvider(() => SimpleIoc.Default);

            if (!SimpleIoc.Default.IsRegistered<IMessageBoxService>())
            {
                SimpleIoc.Default.Register<IMessageBoxService, MessageBoxService>();
            }

            SimpleIoc.Default.Register<MainViewModel>();
```


(C) Copyright Nokia 2014. All rights reserved.

```
        }

        public MainViewModel MainViewModel
        {
            get
            {
                return ServiceLocator.Current.GetInstance<MainViewModel>();
            }
        }

        public static void Cleanup()
        {
            // TODO Clear the ViewModels
        }
    }
```

Implementing the ViewModel

Then we should implement the MainViewModel as following:

```
using System.Collections.Generic;
    using System.Windows;
```


(C) Copyright Nokia 2014. All rights reserved.

```
using System.Windows.Input;

using Cimbalino.Phone.Toolkit.Services;

using GalaSoft.MvvmLight.Command;

/// This class contains properties that the main View can data bind to.
public class MainViewModel : ViewModelBase
{
    /// The message box service.
    private readonly IMessageBoxService _messageBoxService;

    /// Initializes a new instance of the MainViewModel class.
    public MainViewModel(IMessageBoxService messageBoxService)
    {
        _messageBoxService = messageBoxService;
        ShowMessageBoxCommand = new RelayCommand(ShowMessageBox);
    }
```


(C) Copyright Nokia 2014. All rights reserved.

```

        /// Shows the message box.
        private void ShowMessageBox()
        {
            _messageBoxService.Show("I am a message box", "Cimbalino sample", MessageBoxButton.OK);
        }

        /// Gets the show message box command.
        public ICommand ShowMessageBoxCommand { get; private set; }
    }
```

Implementing the View

Add the binding in main page like:

```
DataContext="{Binding MainViewModel,
                      Source={StaticResource Locator}}"
```

The **MainPage.xaml** can be the following:

```
<phone:PhoneApplicationPage x:Class="CimbalinoSample.MainPage"

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"
```

 (C) Copyright Nokia 2014. All rights reserved.

```
                              xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
                              xmlns:phone="clr-namespace:Microsoft.Phone.Controls;assembly=Microsoft.Phone"
                              xmlns:shell="clr-namespace:Microsoft.Phone.Shell;assembly=Microsoft.Phone"
                              FontFamily="{StaticResource PhoneFontFamilyNormal}"
                              FontSize="{StaticResource PhoneFontSizeNormal}"
                              Foreground="{StaticResource PhoneForegroundBrush}"
                              Orientation="Portrait"
                              DataContext="{Binding MainViewModel,
                                                    Source={StaticResource Locator}}"
                              SupportedOrientations="Portrait"
                              shell:SystemTray.IsVisible="True"
                              mc:Ignorable="d">

    <!--  LayoutRoot is the root grid where all page content is placed  -->
    <Grid x:Name="LayoutRoot" Background="Transparent">
        <Grid.RowDefinitions>
            <RowDefinition Height="Auto" />
```

 (C) Copyright Nokia 2014. All rights reserved.

```
            <RowDefinition Height="*" />
        </Grid.RowDefinitions>

        <!--  TitlePanel contains the name of the application and page title  -->
        <StackPanel x:Name="TitlePanel"
                    Grid.Row="0"
                    Margin="12,17,0,28">
            <TextBlock Margin="12,0"
                       Style="{StaticResource PhoneTextTitle2Style}"
                       Text="Cimbalino Sample" />
            <TextBlock Margin="9,-7,0,0"
                       Style="{StaticResource PhoneTextTitle2Style}"
                       Text="MessageBoxService" />
        </StackPanel>

        <!--  ContentPanel - place additional content here  -->
        <Grid x:Name="ContentPanel"
              Grid.Row="1"
```

 (C) Copyright Nokia 2014. All rights reserved.

```
                Margin="12,0,12,0">

             <Button Command="{Binding ShowMessageBoxCommand}" Content="Show message box" />

         </Grid>

     </Grid>

</phone:PhoneApplicationPage>
```

Source Code

The source code is available here.

(C) Copyright Nokia 2014. All rights reserved.