How to use Cimbalino Windows Phone Toolkit Media Library - FMRadioService

This samples has the goal to show how to use Cimbalino Windows Phone Toolkit Media Library - FMRadioService.

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. **Cimbalino.Phone.Toolkit.MediaLibrary** is a MVVM compatible services for media library access.

- IFMRadioService - Represents a service capable of handling the device FM radio. The implementation is FMRadioService.

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.

Building the example code

The source code for the code example is available here: (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

Note: you must specify the following capabilities in the app manifest: ID_CAP_MEDIALIB

Registering the service

Register the service in the `ViewModelLocator` constructor as highlighted below

```
using Cimbalino.Phone.Toolkit.Services;
 using GalaSoft.MvvmLight;
 using GalaSoft.MvvmLight.Ioc;
 using Microsoft.Practices.ServiceLocation;

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

 (C) Copyright Nokia 2014. All rights reserved.

```
        {
            SimpleIoc.Default.Register<IFMRadioService, FMRadioService>();
        }
        SimpleIoc.Default.Register<MainViewModel>();
    }

    /// <summary>
    /// Gets the main view model.
    /// </summary>
    /// <value>
    /// The main view model.
    /// </value>
    public MainViewModel MainViewModel
    {
        get
        {
            return ServiceLocator.Current.GetInstance<MainViewModel>();
        }
    }

    public static void Cleanup()
    {
        // TODO Clear the ViewModels
        var viewModelLocator = (ViewModelLocator)App.Current.Resources["Locator"];
        viewModelLocator.MainViewModel.Cleanup();
    }
}
```

Implementing the ViewModel

Then we should implement the `MainViewModel` as following.

```
using System.Windows;
   using System.Windows.Input;
   using System.Windows.Threading;

   using Cimbalino.Phone.Toolkit.Services;

   using GalaSoft.MvvmLight.Command;
   using GalaSoft.MvvmLight;

   using Microsoft.Devices.Radio;

   /// <summary>
   /// This class contains properties that the main View can data bind to.
   /// </summary>
   public class MainViewModel : ViewModelBase
   {
       /// <summary>
       /// The FM radio service
       /// </summary>
       private readonly IFMRadioService _fmRadioService;

       /// <summary>
       /// The frequency
```

 (C) Copyright Nokia 2014. All rights reserved.

```
        /// </summary>
        private double _frequency;

        /// <summary>
        /// Initializes a new instance of the MainViewModel class.
        /// </summary>
        public MainViewModel(IFMRadioService fmRadioService)
        {
            _fmRadioService = fmRadioService;
            PropertyChanged += MainViewModel_PropertyChanged;
            Frequency = 91.7;
            _fmRadioService.CurrentRegion = RadioRegion.Europe;
            _fmRadioService.PowerOff();
            StopCommand= new RelayCommand(Stop);
            PlayCommand = new RelayCommand(Play);
        }

        /// <summary>
        /// Plays this instance.
        /// </summary>
        private void Play()
        {
          _fmRadioService.PowerOn();
        }

        /// <summary>
        /// Stops this instance.
        /// </summary>
        private void Stop()
        {
         _fmRadioService.PowerOff();
        }

        /// <summary>
        /// Gets or sets the frequency.
        /// </summary>
        /// <value>
        /// The frequency.
        /// </value>
        public double Frequency
        {
            get
            {
                return this._frequency;
            }
            set
            {
                Set("Frequency", ref _frequency, value);
            }
        }

        /// <summary>
        /// Gets the play command.
        /// </summary>
        /// <value>
        /// The play command.
        /// </value>
```

 (C) Copyright Nokia 2014. All rights reserved.

```
        public ICommand PlayCommand { get; private set; }

        /// <summary>
        /// Gets the stop command.
        /// </summary>
        /// <value>
        /// The stop command.
        /// </value>
        public ICommand StopCommand { get; private set; }

        /// <summary>
        /// Handles the PropertyChanged event of the MainViewModel control.
        /// </summary>
        /// <param name="sender">The source of the event.</param>
        /// <param name="e">The <see
cref="System.ComponentModel.PropertyChangedEventArgs"/> instance containing the event
data.</param>
        private void MainViewModel_PropertyChanged(object sender,
System.ComponentModel.PropertyChangedEventArgs e)
        {
            if (e.PropertyName == "Frequency")
            {
                _fmRadioService.Frequency = Frequency;
            }
        }
    }
```

Implementing the View

The **MainPage.xaml** is as shown below.

```
<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:converters="clr-
namespace:Cimbalino.Phone.Toolkit.Converters;assembly=Cimbalino.Phone.Toolkit"
                            xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
                            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"
                            DataContext="{Binding MainViewModel,
                                                  Source={StaticResource Locator}}"
                            FontFamily="{StaticResource PhoneFontFamilyNormal}"
                            FontSize="{StaticResource PhoneFontSizeNormal}"
                            Foreground="{StaticResource PhoneForegroundBrush}"
                            Orientation="Portrait"
                            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">
```

 (C) Copyright Nokia 2014. All rights reserved.

```
        <Grid.RowDefinitions>
            <RowDefinition Height="Auto" />
            <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="FMRadioService" />
        </StackPanel>

        <!--  ContentPanel - place additional content here  -->
        <Grid x:Name="ContentPanel"
              Grid.Row="1"
              Margin="12,0,12,0">
            <TextBlock TextWrapping="Wrap">This samples has the goal to show how to use
Cimbalino Windows Phone Toolkit Media Library - FMRadioService</TextBlock>

            <TextBlock Margin="0,90,0,-90" Text="Frequency:" />
            <TextBox Margin="0,122,0,451" Text="{Binding Frequency, Mode=TwoWay}" />
            <Button Margin="0,219,0,293"
                    Command="{Binding PlayCommand}"
                    Content="Play" />
            <Button Margin="0,351,0,161"
                    Command="{Binding StopCommand}"
                    Content="Stop" />
        </Grid>
    </Grid>
</phone:PhoneApplicationPage>
```

 (C) Copyright Nokia 2014. All rights reserved.