How to use Cimbalino Windows Phone Toolkit - ApplicationManifestService

This code example shows how to read information in the application manifest file (**AppManifest.xml**) using `ApplicationManifestService` 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. **Cimbalino.Phone.Toolkit.Background** (contains the `ApplicationManifestService`) is a MVVM compatible services for background agents.

The Cimbalino Toolkit's Application Manifest service is used to read information from the application manifest. This file includes, in particular, information that it can be useful to display in an app's about page - like the app version, author, product ID etc. The kit provides both the IApplicationManifestService interface and its implementation ApplicationManifestService 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 is shown below.

Application manifest file information displayed in app

Building the example code

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

(C) Copyright Nokia 2014. All rights reserved.

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

Registering the service

We should register each service in `ViewModelLocator`, as following:

```
/// 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<IApplicationManifestService>())
        {
            SimpleIoc.Default.Register<IApplicationManifestService, ApplicationManifestService>();
        }
        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:

```
    /// This class contains properties that the main View can data bind to.
    public class MainViewModel : ViewModelBase
    {
```

 (C) Copyright Nokia 2014. All rights reserved.

```
        /// The public application url.

        private readonly string _appUrl;


        /// The application manifest.

        private readonly ApplicationManifest _applicationManifest;


        /// Initializes a new instance of the Application Manifest Service.

        public MainViewModel(IApplicationManifestService applicationManifestService)

        {

            _applicationManifest = applicationManifestService.GetApplicationManifest();

            _appUrl = string.Concat("http://windowsphone.com/s?appid=", _applicationManifest.App.ProductId);

        }


        /// Gets the title.

        public string Title

        {

            get
```

 (C) Copyright Nokia 2014. All rights reserved.

```
            {
                return _applicationManifest.App.Title;
            }
        }

        /// Gets the author.
        public string Author
        {
            get
            {
                return _applicationManifest.App.Author;
            }
        }

        /// Gets the version.
        public string Version
        {
            get
```

 (C) Copyright Nokia 2014. All rights reserved.

```
            {
                return _applicationManifest.App.Version;
            }
        }

        /// Gets the description.
        public string Description
        {
            get
            {
                return _applicationManifest.App.Description;
            }
        }

        /// Gets the product ID.
        public string ProductID
        {
            get
```

 (C) Copyright Nokia 2014. All rights reserved.

```
            {
                return _applicationManifest.App.ProductId;
            }
        }

        /// Gets the publisher.
        public string Publisher
        {
            get
            {
                return _applicationManifest.App.Publisher;
            }
        }

        /// Gets the capabilities.
        public IList Capabilities
        {
            get
```

 (C) Copyright Nokia 2014. All rights reserved.

```
            {
                return _applicationManifest.App.Capabilities.ToList();
            }
        }
    }
```

Implementing the view

Add the binding in main page as shown:

```
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"
                            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
```

 (C) Copyright Nokia 2014. All rights reserved.

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

        <!--  TitlePanel contains the name of the application and page title  -->
        <StackPanel x:Name="TitlePanel"
```

 (C) Copyright Nokia 2014. All rights reserved.

```
                    Grid.Row="0"
                    Margin="12,17,0,28">
            <TextBlock Margin="12,0"
                       Style="{StaticResource PhoneTextTitle2Style}"
                       Text="Cimbalino Sample" />
        </StackPanel>

        <!--  ContentPanel - place additional content here  -->
        <Grid x:Name="ContentPanel"
              Grid.Row="1"
              Margin="12,0,12,0" >
            <TextBlock TextWrapping="Wrap"
                       VerticalAlignment="Top" Height="50">
                Title: <Run Text="{Binding Title}"/>
                </TextBlock>
            <TextBlock TextWrapping="Wrap"
                       VerticalAlignment="Top" Margin="0,50,0,0">
                Author: <Run Text="{Binding Author}"/>
```

 (C) Copyright Nokia 2014. All rights reserved.

```
            </TextBlock>

            <TextBlock TextWrapping="Wrap"

                       VerticalAlignment="Top" Margin="0,100,0,0">

                Version: <Run Text="{Binding Version}"/>

            </TextBlock>

            <TextBlock TextWrapping="Wrap"

                       VerticalAlignment="Top" Margin="0,150,0,0">

                Description: <Run Text="{Binding Description}"/>

            </TextBlock>

            <TextBlock TextWrapping="Wrap"

                       VerticalAlignment="Top" Margin="0,300,0,0">

                ProductID: <Run Text="{Binding ProductID}"/>

            </TextBlock>

            <TextBlock TextWrapping="Wrap"

                       VerticalAlignment="Top" Margin="0,350,0,0">

                Publisher: <Run Text="{Binding Publisher}"/>

            </TextBlock>

            <TextBlock TextWrapping="Wrap" Text="Capabilities:"
```

 (C) Copyright Nokia 2014. All rights reserved.

```
                        VerticalAlignment="Top" Margin="0,400,0,0"/>

            <ListBox DisplayMemberPath="Name" Margin="0,450,0,0"

                     ItemsSource="{Binding Capabilities}"/>

        </Grid>


    </Grid>


</phone:PhoneApplicationPage>
```

Related Samples

- Advanced About Page for Windows Phone apps
- How to use MVVM Light Toolkit for Windows Phone

Source Code

The source code is available here.

 (C) Copyright Nokia 2014. All rights reserved.