

How to implement link sharing using the Cimalino Windows Phone Toolkit

This article shows how to launch a dialog that enables the user to share a link on the social networks of their choice using `ShareLinkService` from the [Cimalino Windows Phone Toolkit](#).

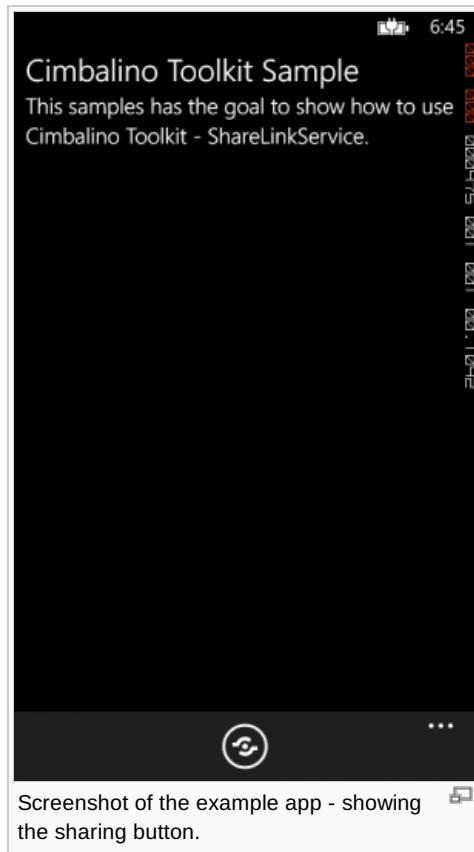
Introduction



The *Cimalino 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 Cimalino Toolkit's "Share Link" service is an MVVM compatible wrapper around the system `ShareLinkTask`, which can be used to launch a dialog to share links via social networks configured on the device. The kit provides both the `IShareLinkService` interface and its implementation `ShareLinkService` 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.



Building the example code

The source code for the code example is available here: [ShareLinkService Example](#) (Github)

To build the source code you will also need the [MVVM Light Toolkit](#) and the *Cimalino 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 Cimalino 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

Registering the service

Register the service in the `ViewModelLocator` constructor as shown below.

```
/// <summary>
/// Initializes a new instance of the ViewModelLocator class.
/// </summary>
public ViewModelLocator()
{
    ServiceLocator.SetLocatorProvider(() => SimpleIoc.Default);

    if (!SimpleIoc.Default.IsRegistered<IShareLinkService>())
    {
        SimpleIoc.Default.Register<IShareLinkService, ShareLinkService>();
    }

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

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

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

Implementing the ViewModel

Implement the `MainViewModel` as shown below. The highlighted sections show the `MainViewModel` constructor taking the `IShareLinkService` parameter and assigning it to the private variable. Later on the private member is used to show the link service.

```
public class MainViewModel : ViewModelBase
{
    /// <summary>
    /// The share link service.
    /// </summary>
    private readonly IShareLinkService _shareLinkService;

    /// <summary>
    /// The public application url.
    /// </summary>
    private readonly string _appUrl;

    /// <summary>
```

```

///> Initializes a new instance of the <see cref="MainViewModel"/> class
///> 
///> <param name="shareLinkService">
///>   The share Link Service.
///> </param>
public MainViewModel(IShareLinkService shareLinkService)
{
    _shareLinkService = shareLinkService;

    ShareSocialNetworkCommand = new RelayCommand(ShareSocialNetwork);
    _appUrl = string.Concat("http://windowsphone.com/s?appid=8df00038-1b7a-406b-
b33f-37a78b17348c");
}

///> 
///> Gets the share social network command.
///> 
public ICommand ShareSocialNetworkCommand { get; private set; }

///> 
///> The share social network.
///> 
private void ShareSocialNetwork()
{
    const string Message = "This application is amazing, should try it! See in";
    _shareLinkService.Show("Cimbalino Toolkit Sample", Message, new Uri(_appUrl,
UriKind.Absolute));
}
}

```

Implementing the view

The rest of the app is "plumbing" to hook up the `MainViewModel` to the View and send commands back to the invoke the service.

[How to use MVVM Light Toolkit for Windows Phone](#) explains most of what is going on.

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:cimbalino="clr-
namespace:Cimbalino.Phone.Toolkit.Behaviors;assembly=Cimbalino.Phone.Toolkit"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:i="clr-
namespace:System.Windows.Interactivity;assembly=System.Windows.Interactivity"
    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">
    <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="1"
            Margin="0,5,12,396">
        <TextBlock Margin="12,0"
                    Style="{StaticResource PhoneTextTitle2Style}"
                    Text="Cimbalino Toolkit Sample" />
    </StackPanel>
    <TextBlock Grid.RowSpan="2"
                Margin="12,50,-3,487"
                Style="{StaticResource PhoneTextTitle3Style}"
                TextWrapping="Wrap">
        This samples has the goal to show how to use Cimbalino Toolkit -
        ShareLinkService.
    </TextBlock>
    <!-- ContentPanel - place additional content here -->
<Grid x:Name="ContentPanel"
        Grid.Row="1"
        Margin="12,0,12,0" />
    <i:Interaction.Behaviors>
        <cimbalino:ApplicationBarBehavior>
            <cimbalino:ApplicationBarIconButton Command="{Binding
ShareSocialNetworkCommand,
                                            Mode=OneTime}"
                                            IconUri="/Images/appbar.share.png"
                                            Text="Share it" />
        </cimbalino:ApplicationBarBehavior>
    </i:Interaction.Behaviors>
</Grid>

</phone:PhoneApplicationPage>
```

Related samples

- Advanced About Page for Windows Phone apps

Source Code

The source code is available [here](#).