

02-Regions and views

Credit: <https://www.youtube.com/watch?v=ZfBy2nfykqY>

How to create Prism regions, views and navigation

Repo: <https://github.com/cmeegamarachchi/wpfprism-02-RegionAndViews>

Please refer following doc(s) before starting

<https://github.com/cmeegamarachchi/wpfprism-01-Shell/blob/master/docs/01-Creatingshell.pdf>

1. Create two views (WPF UserControl), **FirstView** and **SecondView** in Views folder
2. Register the two views for navigation by overriding Bootstrapper's **ConfigureContainer** method
 - a. Override **ConfigureContainer** method on Bootstrapper
 - b. Use RegisterTypeForNavigation extension method to register two views for navigation using unique-names (keys)

```
using System.Windows;
using Prism.Unity;
using Microsoft.Practices.Unity;
using Prism6Demo2.Views;

namespace Prism6Demo2
{
    public class Bootstrapper : UnityBootstrapper
    {
        // rest of the code

        protected override void ConfigureContainer()
        {
            base.ConfigureContainer();

            Container.RegisterTypeForNavigation<FirstView>("FirstView");
            Container.RegisterTypeForNavigation<SecondView>("SecondView");
        }
    }
}
```

3. Using text blocks, display unique text to identify the views
4. Add a region to MainWindow. In MainWindow.xaml
 - a. Add prism name-space
 - b. Add a stackpanel and two buttons to act as view switches
 - c. Add a ContentControl and add declare it as "Content" region using **RegionManager.RegionName** attached property

```
<Window x:Class="Prism6Demo2.Views.MainWindow"
        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:prism="http://prismlibrary.com/"
        mc:Ignorable="d"
        Title="MainWindow" Height="350" Width="525">
    <Grid>
        <Grid.RowDefinitions>
            <RowDefinition Height="Auto" />
            <RowDefinition Height="*" />
        </Grid.RowDefinitions>

        <StackPanel Grid.Row="0" Orientation="Horizontal">
            <Button Margin="2">First view</Button>
            <Button Margin="2">Second view</Button>
        </StackPanel>

        <ContentControl Grid.Row="1" prism:RegionManager.RegionName="Content" />
    </Grid>
</Window>
```

5. Create a view model for MainWindow and host navigation logic
 - a. Prim ViewModelLocator in its default configuration expects the ViewModel to match the name of the View with suffix 'ViewModel'. It also expects to find the view model in ViewModels folder. So lets
 - i. Add **MainWindowViewModel.cs** to **ViewModels** folder
 - ii. Create a command for navigation. The navigation command is a delegate command and it expect to the view to navigate. The key is what we used to register the view for navigation in step 2.b

```
using System.Windows.Input;
using Prism.Commands;
using Prism.Regions;

namespace Prism6Demo2.ViewModels
{
    public class MainWindowViewModel
    {
        private readonly IRegionManager _regionManager;
        public ICommand NavigateCommand { get; private set; }

        public MainWindowViewModel(IRegionManager regionManager)
        {
        }
    }
}
```

```

    {
        _regionManager = regionManager;
        NavigateCommand = new DelegateCommand<string>(NavigateCommandExecute);
    }

    private void NavigateCommandExecute(string path)
    {
        _regionManager.RequestNavigate("ContentRegion", path);
    }
}

```

iii. Configure MainWindow view to auto discover viewmodel.

1. Add **ViewModelLocator.AutoWireViewModel="True"** to MainWindow.xaml. This will tell Prism to automatically locate and bind viewmodel to view

```

<Window x:Class="Prism6Demo2.Views.MainWindow"
    ...
    xmlns:prism="http://prismlibrary.com/"
    prism:ViewModelLocator.AutoWireViewModel="True"
    Title="MainWindow" Height="350" Width="525">
    <Grid>
    ...
    </Grid>
</Window>

```

iv. Update buttons by binding them to NavigationCommand

1. Use binding extensions to bind the buttons to NavigationCommand

```

<Window x:Class="Prism6Wpf2.Views.MainWindow"
    ...
    prsim:ViewModelLocator.AutoWireViewModel="True"
    mc:Ignorable="d"
    Title="MainWindow" Height="350" Width="525">
    <Grid>
    <Grid.RowDefinitions>
        <RowDefinition Height="Auto"/>
        <RowDefinition Height="*/>
    </Grid.RowDefinitions>

    <StackPanel Grid.Row="0" Orientation="Horizontal">
        <Button Command="{Binding NavigateCommand}" CommandParameter="ViewA">View A</Button>
        <Button Command="{Binding NavigateCommand}" CommandParameter="ViewB">View B</Button>
    </StackPanel>

    <ContentControl Grid.Row="1" prsim:RegionManager.RegionName="ContentRegion"/>
    </Grid>
</Window>

```

- v. Add view models for the other two views following the same convention **ViewAViewModel** for **ViewA** and **ViewBViewModel** for **ViewB**
- vi. Update xaml for each view to automatically locate and bind the view as we did with step iii