

Navigation

<https://learn.microsoft.com/en-us/windows/apps/design/controls/navigationview>

Types of navigation

Windows => **Activate**

Windows (Shell) => Frame as Container

Multiple Pages

Frame.**Navigate** (inside Pages)

Multiple windows example

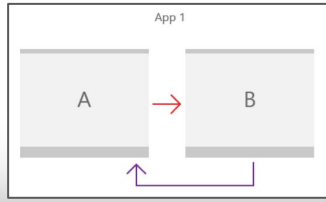
Logi(Window) > Dashboard (Window)

Activated: show **ProductsPage** > **AddProductPage**

GoBack

Data passing between windows & pages

Constructor's parameters



Basic 2 Pages

[Peer-to-peer navigation between two pages - Windows apps | Microsoft Learn](#)

Basic ideas

- ❑ Use **Frame** to navigate between **Pages**
- ❑ Navigation methods
 - a. **Navigate**
 - b. **GoBack / GoForward**
- ❑ Properties
 - a. **BackStack / ForwardStack**
 - b. **BackStackDepth**

Enable navigation - App.xaml.cs

```
protected override void OnLaunched(LaunchActivatedEventArgs args) {  
    m_window = new MainWindow();  
  
    Frame rootFrame = new Frame();  
    rootFrame.Navigate(typeof(MainPage), args.Arguments);  
  
    m_window.Content = rootFrame;  
    m_window.Activate();  
}
```

Navigate to another Page using string

```
Frame rootFrame = new Frame();  
var baseNamespace = typeof(App).Namespace;  
var type = Type.GetType($"{baseNamespace}.MainPage");  
rootFrame.Navigate(type, args.Arguments);
```

Basic ideas: **Using full namespace name**

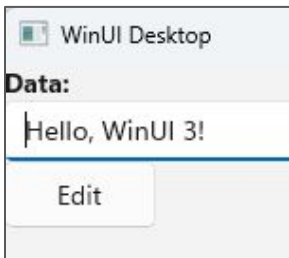
For example:

If the base namespace is “**MyProject**”

Then the full name of the Page **MainPage** is

“**MyProject.Mainpage**”

Prepare the MainPage

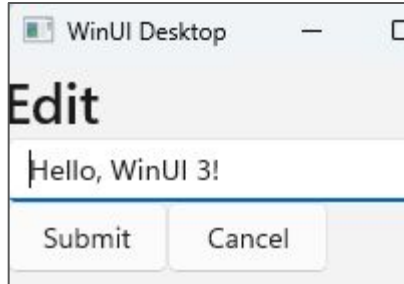


```
<StackPanel>
    <TextBlock Text="Data:" FontWeight="Bold"/>
    <TextBox Text="{x:Bind ViewModel.Info,
        Mode=TwoWay, UpdateSourceTrigger=PropertyChanged}"/>
    <Button Name="editButton" Content="Edit"
        Width="80" Height="35"
        Click="editButton_Click"/>
</StackPanel>
```

```
public class MainPageViewModel {
    1 reference
    public string Info { get; set; } = "Hello, WinUI 3!";
}

1 reference
MainPageViewModel ViewModel { get; set; }
    = new MainPageViewModel();
```

Prepare the EditPage



```
ckPanel>  
<TextBlock Text="Edit" Style="{StaticResource TitleTextBlockStyle}">  
<TextBox Text="{x:Bind ViewModel.Info, Mode=TwoWay,  
    UpdateSourceTrigger=PropertyChanged}"/>  
  
<StackPanel Orientation="Horizontal">  
    <Button Name="okButton" Content="Submit"  
        Width="80" Height="35"  
        Click="okButton_Click"/>  
    <Button Name="cancelButton" Content="Cancel"  
        Width="80" Height="35"  
        Click="cancelButton_Click"/>  
</StackPanel>  
</StackPanel>
```

Passing data to another Page - Basic ideas

- ❑ Use the **second parameter** of the function **Navigate**
- ❑ In the destination page, handle **OnNavigatedTo**

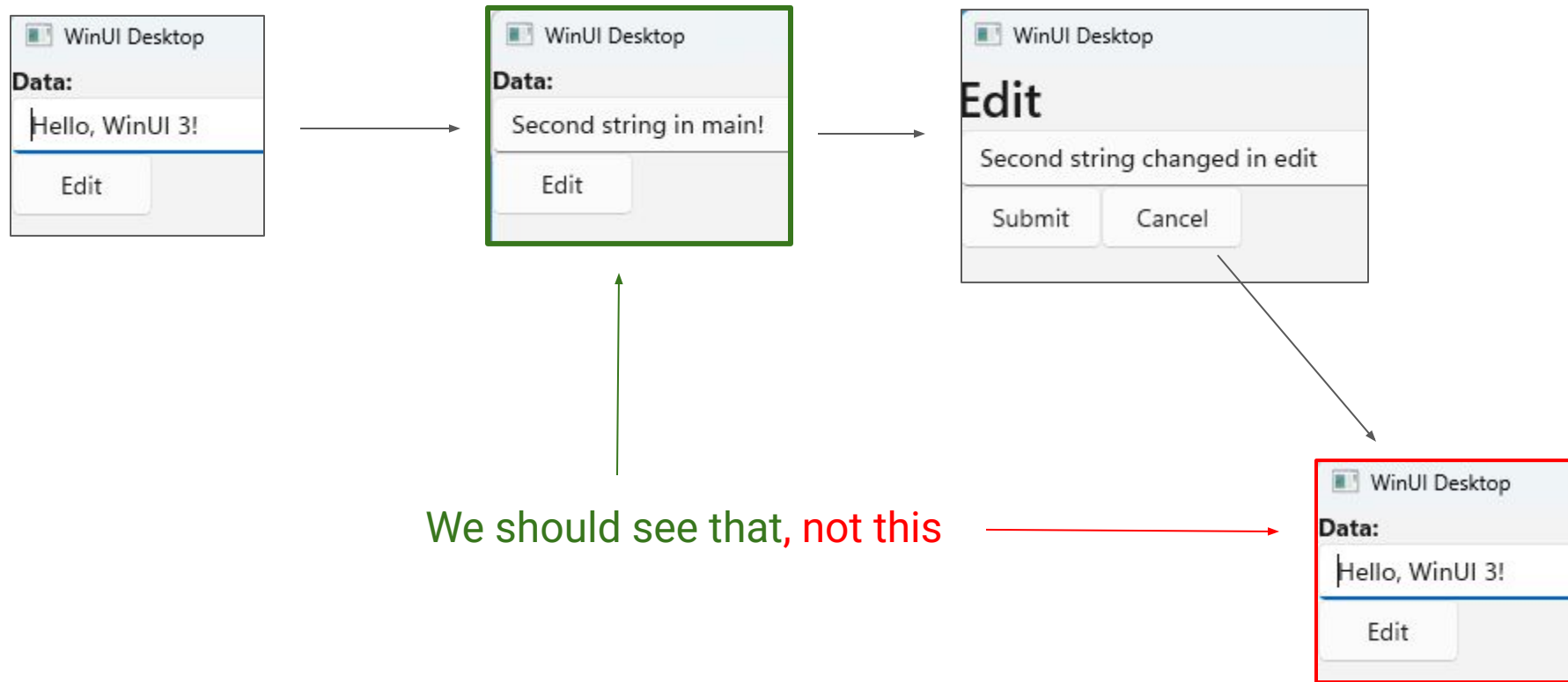
1 reference

```
private void editButton_Click(object sender, RoutedEventArgs e)
{
    Frame.Navigate(typeof(EditPage), ViewModel.Info);
}
```

```
protected override void OnNavigatedTo(NavigationEventArgs e)
{
    ViewModel.Info = e.Parameter as string;

    base.OnNavigatedTo(e);
}
```

Simple flow (not good)



Passing data back & forth

```
private string _oldString = "";
```

0 references

```
protected override void OnNavigatedTo(NavigationEventArgs e) {  
    _oldString = e.Parameter as string;  
    ViewModel.Info = _oldString;  
  
    base.OnNavigatedTo(e);  
}
```

1 reference

```
private void okButton_Click(object sender, RoutedEventArgs e) {  
    Frame.Navigate(typeof(MainPage), ViewModel.Info);  
}
```

1 reference

```
private void cancelButton_Click(object sender, RoutedEventArgs e) {  
    Frame.Navigate(typeof(MainPage), _oldString);  
}
```

How about passing an object rather than a string?

- Problem?
- Solution: `IClonable`

Make the Employee class Cloneable

```
public class Employee : INotifyPropertyChanged, ICloneable
{
    3 references
    public int ID {get; set;}
    11 references
    public string Name { get; set;}
    1 reference
    public object Clone() {
        var result = new Employee()
        {
            ID = this.ID,
            Name = this.Name
        };
        return result;
    }

    public event PropertyChangedEventHandler PropertyChanged;
}
```


Store for later retrieval

```
protected override void OnNavigatedTo(NavigationEventArgs e) {  
    _oldData = e.Parameter as Employee;  
    var newData = _oldData.Clone() as Employee;  
    ViewModel.Info = newData; //( _oldData.Clone() as Employee);  
  
    base.OnNavigatedTo(e);  
}
```

1 reference

```
private void okButton_Click(object sender, RoutedEventArgs e) {  
    Frame.Navigate(typeof(MainPage), ViewModel.Info);  
}
```

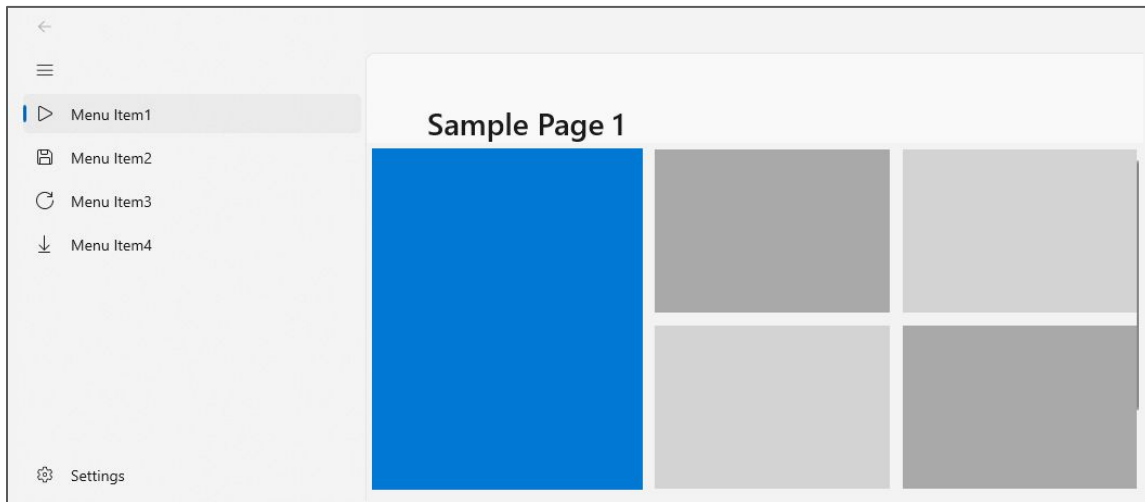
1 reference

```
private void cancelButton_Click(object sender, RoutedEventArgs e) {  
    Frame.Navigate(typeof(MainPage), _oldData);  
}
```


NavigationView

Basic NavigationView

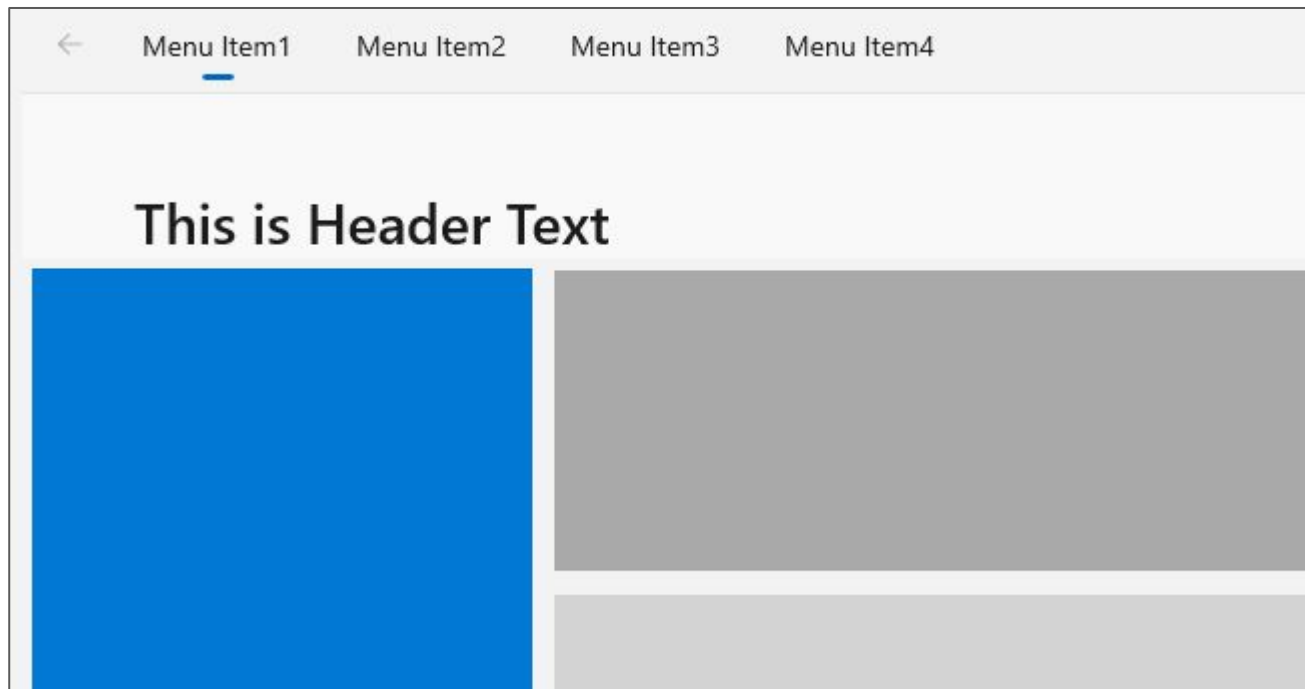
```
<NavigationView x:Name="nvSample">
  <NavigationView.MenuItems>
    <NavigationViewItem Icon="Play" Content="Menu Item1" Tag="SamplePage1" />
    <NavigationViewItem Icon="Save" Content="Menu Item2" Tag="SamplePage2" />
    <NavigationViewItem Icon="Refresh" Content="Menu Item3" Tag="SamplePage3" />
    <NavigationViewItem Icon="Download" Content="Menu Item4" Tag="SamplePage4" />
  </NavigationView.MenuItems>
  <Frame x:Name="contentFrame"/>
</NavigationView>
```



NavigationView_SelectionChanged

1. Extract the Page name from Tag
2. Create type from the Page name
3. Navigate using Frame

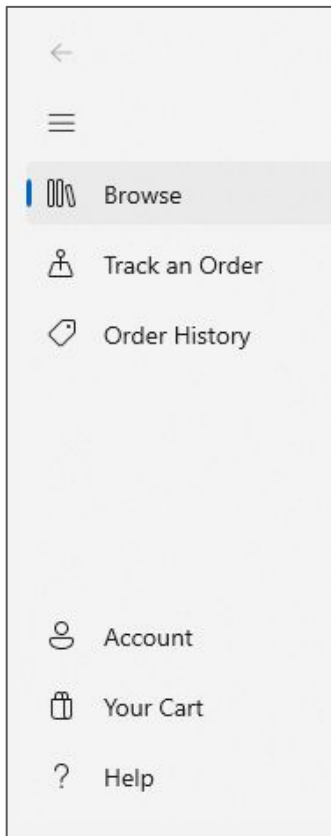
PaneDisplayMode="Top"



Switch panes based on window's width

```
<!-- Put the following VisualStateGroup(s) inside the first component of your Page -->
<VisualStateManager.VisualStateGroups>
  <VisualStateGroup>
    <VisualState>
      <VisualState.StateTriggers>
        <AdaptiveTrigger MinWindowWidth="{x:Bind nvSample.CompactModeThresholdWidth}" />
      </VisualState.StateTriggers>
      <VisualState.Setters>
        <Setter Target="nvSample.PaneDisplayMode" Value="Top" />
      </VisualState.Setters>
    </VisualState>
  </VisualStateGroup>
</VisualStateManager.VisualStateGroups >
```

Footer menu items



```
<NavigationView.FooterMenuItems>
  <NavigationViewItem Content="Account" Tag="SamplePage4" Icon="Contact" />
  <NavigationViewItem Content="Your Cart" Tag="SamplePage5" Icon="Shop" />
  <NavigationViewItem Content="Help" Tag="SamplePage5" Icon="Help" />
</NavigationView.FooterMenuItems>
```