XAML - RESOURCES

Resources are normally definitions connected with some object that you just anticipate to use more often than once. It has the ability to store data locally for controls or for the current window or globally for the entire applications.

Defining an object as a resource allows us to access it from another place. Hence, it allows reusability. Resources are defined in resource dictionaries and any object can be defined as a resource effectively making it a shareable asset. A unique key is specified to XAML resource and with that key, it can be referenced by using a StaticResource markup extension.

Let’s have a look at a simple example again in which two text blocks are created with some properties and their foreground color is defined in **Window.Resources**.

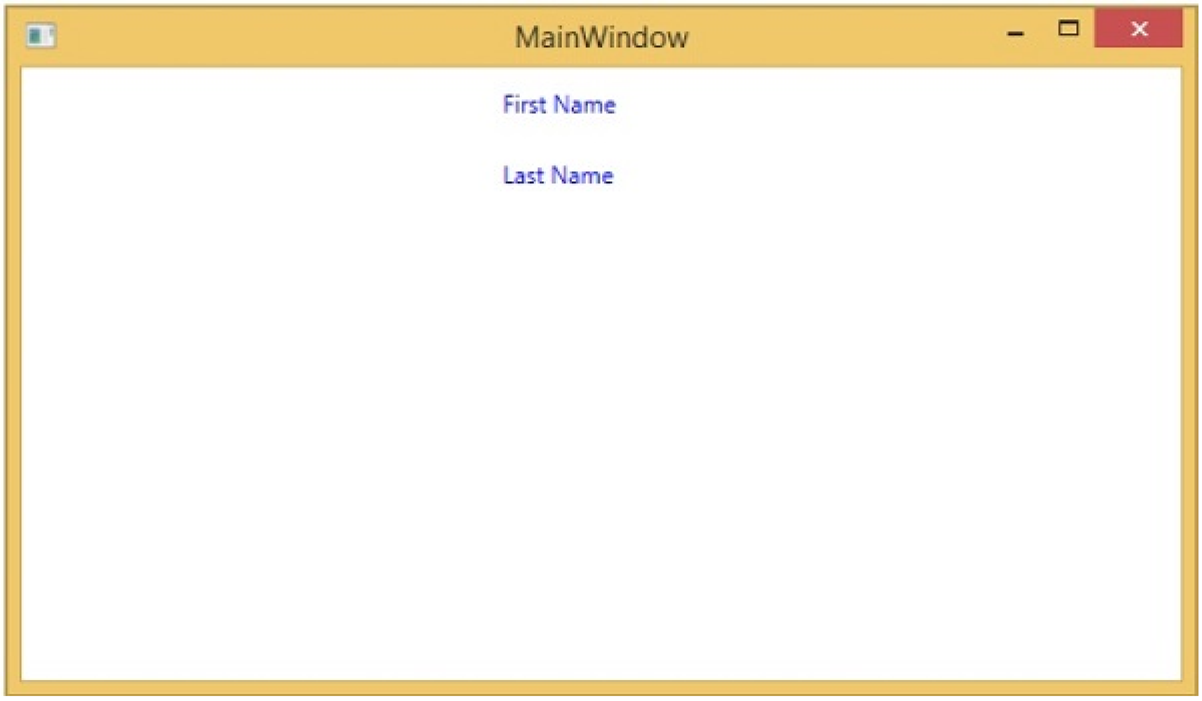
```
<Window x:Class = "XAMLResources.MainWindow"
   xmlns = "http://schemas.microsoft.com/winfx/2006/xaml/presentation"
   xmlns:x = "http://schemas.microsoft.com/winfx/2006/xaml"
   Title = "MainWindow" Height = "350" Width = "604">

   <Window.Resources>
      <SolidColorBrush Color = "Blue" x:Key = "myBrush"></SolidColorBrush>
   </Window.Resources>

   <StackPanel Orientation = "Vertical">
      <TextBlock Foreground = "{StaticResource myBrush}"
         Text = "First Name" Width = "100" Margin = "10" />
      <TextBlock Foreground = "{StaticResource myBrush}"
         Text = "Last Name" Width = "100" Margin = "10" />
   </StackPanel>

</Window>
```

When the above code is compiled and executed, it will produce the following MainWindow. You can see two text blocks with blue foreground color. The advantage of the resource is that if there are multiple text blocks and you want to change their background color, then you will need just to change it in the resource dictionary.

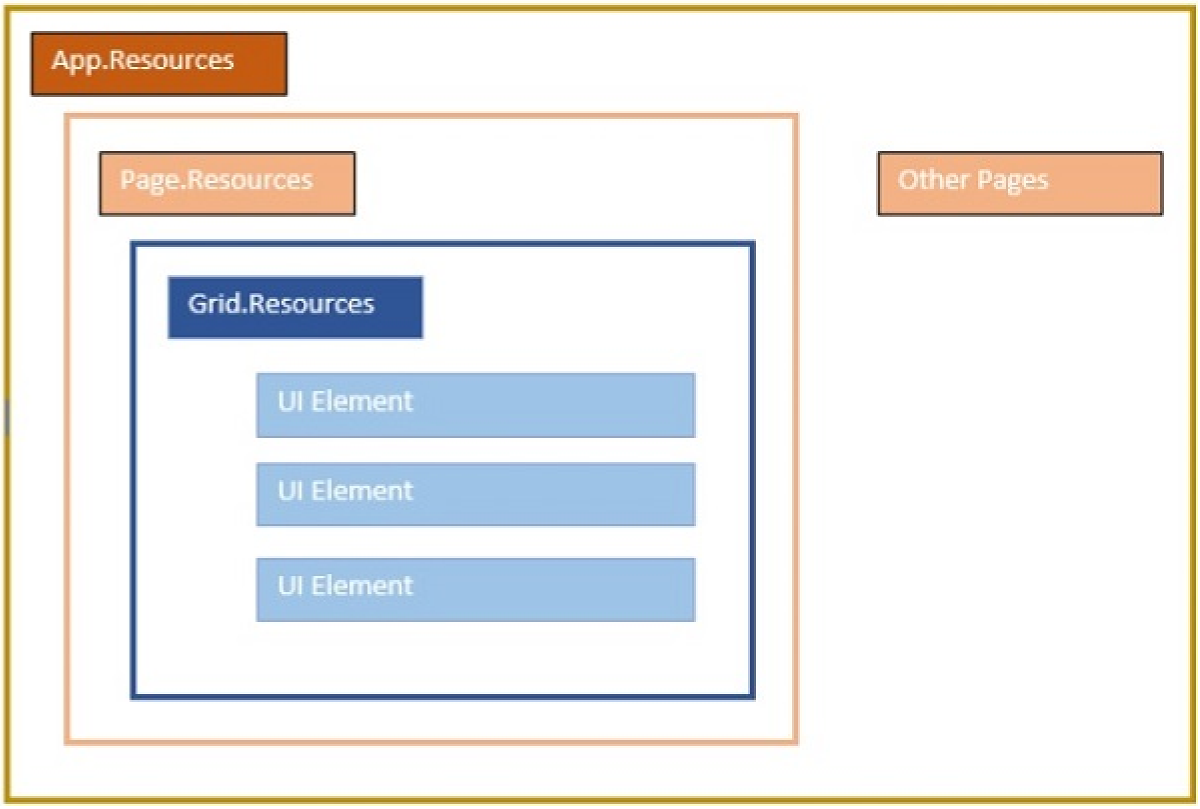


Resource Scope

Resources are defined in resource dictionaries, but there are numerous places where a resource dictionary can be defined. In the above example, a resource dictionary is defined on Window/page level. In what dictionary a resource is defined immediately limits the scope of that resource. So the scope, i.e. where you can use the resource, depends on where you've defined it.

- Define the resource in the resource dictionary of a grid and it's accessible by that grid and by its child elements only.
- Define it on a window/page and it's accessible by all elements on that window/page.
- The App root can be found in App.xaml resources dictionary. It's the root of our application, so the resources defined here are scoped to the complete application.

As far as the scope of the resource is concerned, the most often are application level, page level, and a specific element level like a Grid, StackPanel, etc.



Resource Dictionaries

Resource dictionaries in XAML apps imply resource dictionaries in separate files. It is followed in almost all XAML apps. Defining resources in separate files can have the following advantages −

- Separation between defining resources in the resource dictionary and UI related code.
- Defining all the resources in a separate file such as App.xaml would make them available across the App.

So, how we can define our resources in a resource dictionary in a separate file? Well, it is very easy, just add a new resource dictionary through Visual Studio by the following steps −

- In your solution, add a new folder and name it **ResourceDictionaries**.
- Right-click on this folder and select Resource Dictionary from Add submenu item and name it **DictionaryWithBrush.xaml**

Let’s have a look at the same application; just the resource dictionary is now defined in App level.

Here is the XAML code for MainWindow.xaml.

```
<Window x:Class = "XAMLResources.MainWindow"
   xmlns = "http://schemas.microsoft.com/winfx/2006/xaml/presentation"
   xmlns:x = "http://schemas.microsoft.com/winfx/2006/xaml"
   Title = "MainWindow" Height = "350" Width = "604">

   <StackPanel Orientation = "Vertical">
      <TextBlock Foreground = "{StaticResource myBrush}" Text = "First Name" Width =
"100" Margin = "10" />
      <TextBlock Foreground = "{StaticResource myBrush}" Text = "Last Name" Width =
"100" Margin = "10"/>
   </StackPanel>

</Window>
```

Here is the implementation in DictionaryWithBrush.xaml –

```
<ResourceDictionary
   xmlns = "http://schemas.microsoft.com/winfx/2006/xaml/presentation"
   xmlns:x = "http://schemas.microsoft.com/winfx/2006/xaml">

   <SolidColorBrush Color = "Blue" x:Key="myBrush">
   </SolidColorBrush>

</ResourceDictionary>
```

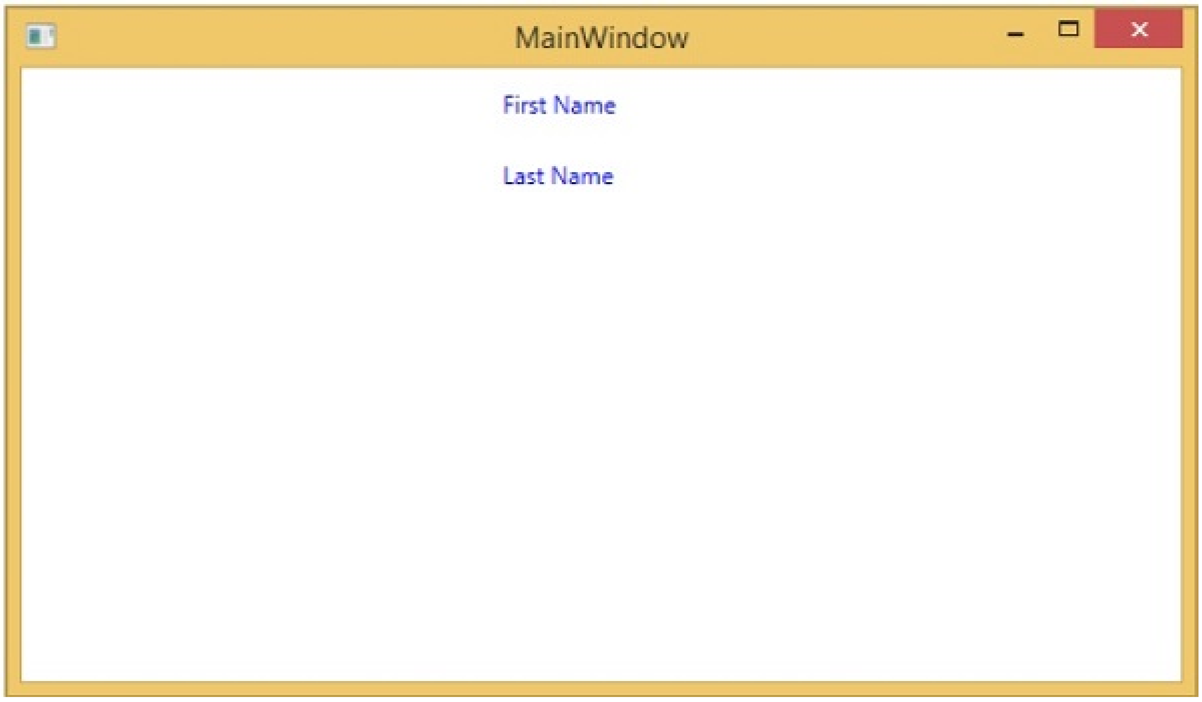
Here is the implementation in app.xaml –

```
<Application x:Class = "XAMLResources.App"
   xmlns = "http://schemas.microsoft.com/winfx/2006/xaml/presentation"
   xmlns:x = "http://schemas.microsoft.com/winfx/2006/xaml"
   StartupUri = "MainWindow.xaml">

   <Application.Resources>
      <ResourceDictionary Source = "
XAMLResources\ResourceDictionaries\DictionaryWithBrush.xaml" />
   </Application.Resources>

</Application>
```

When the above code is compiled and executed, it will produce the following output –



First Name

Last Name

We recommend you to execute the above code and experiment with some more resources such as background color, etc.