Using the XAML Style tag to format UI controls on Windows Phone

This article shows how to make use of Style tag in various types of controls.

Introduction

A style tag is used to define style information for a control. If we talk about a Text Block Control, then we can define the font style, font size, alignment etc. parameters. Similarly, **we can define style tags for other controls as well like List Picker control, Image Control, Text Box etc.**

Starting a New Project

- Create a new project in Visual Studio 2010 from **File>>New Project>>Windows Phone Application - Visual C#**. Give a desired name to the project.

Target UI

In this article, our target UI should be like:

Adding Some Text Box Controls

In the **MainPage.xaml** file, we will drop down a few Text Block & Text Box controls. We will set the desired properties like shown below:

```
<Grid x:Name="ContentPanel" Grid.Row="1" Margin="12,0,12,0">
    <Grid.RowDefinitions>
        <RowDefinition Height=".10*"/>
        <RowDefinition Height=".30*"/>
        <RowDefinition Height=".12*"/>
        <RowDefinition Height=".30*"/>
        <RowDefinition Height=".12*"/>
        <RowDefinition Height=".30*"/>
        <RowDefinition Height="*"/>
    </Grid.RowDefinitions>

    <TextBlock Grid.Row="0" Text="Name : " FontFamily="Arial" FontSize="26" FontWeight="SemiBold" Margin="10,0,0,0"/>
    <TextBox Grid.Row="1" Height="75" TextWrapping="Wrap" Margin="0,-10,0,0"/>
    <TextBlock Grid.Row="2" Text="Designation : " FontFamily="Arial" FontSize="26" FontWeight="SemiBold" Margin="10,0,0,0"/>
    <TextBox Grid.Row="3" Height="75" TextWrapping="Wrap" Margin="0,-10,0,0"/>
    <TextBlock Grid.Row="4" Text="Department : " FontFamily="Arial" FontSize="26" FontWeight="SemiBold" Margin="10,0,0,0"/>
    <TextBox Grid.Row="5" Height="75" TextWrapping="Wrap" Margin="0,-10,0,0"/>
</Grid>
```


(C) Copyright Nokia 2013. All rights reserved.

Now we can see that we have just used 3 or 4 properties but without code reusability. If we could have made a single style tag like we used to do in HTML then it would be much more easier and readable code to work with. The style tag surely reduces the code maintenance time as well.

Making the controls using Style Tag

A style tag can be defined inside the static resource section. Resource can be defined in an individual page's XAML file, in the App.xaml file, or in a separate resource dictionary XAML file. A resource dictionary XAML file can be shared across apps, and more than one resource dictionary can be merged in a single app. Where the resource is defined determines the scope in which it can be used. Page-level resources are available only in the page where they are defined. If resources with the same key are defined in both App.xaml and in a page, the resource in the page overrides the resource in App.xaml. Like wise, a resource defined in App.xaml overrides a resource defined with the same key in a separate resource dictionary file.

There are two ways in which we can apply the *style* in our control

- Implicit : by specifying only a TargetType for the Style.The Style is applied to all elements whose type matches the target type of the style.

```
        <Style TargetType="TextBlock">
            <Setter Property="FontSize" Value="26"/>
            <Setter Property="FontFamily" Value="Arial"/>
            <Setter Property="FontWeight" Value="SemiBold"/>
            <Setter Property="TextWrapping" Value="Wrap"/>
            <Setter Property="Margin" Value="10,0,0,0"/>
        </Style>
```

- Explicit : by specifying a TargetType and an x:Key attribute for the Style and then by setting the target control's Style property to this key.

```
        <Style x:Key="TextBlockStyle" TargetType="TextBlock">
            <Setter Property="FontSize" Value="26"/>
            <Setter Property="FontFamily" Value="Arial"/>
            <Setter Property="FontWeight" Value="SemiBold"/>
            <Setter Property="TextWrapping" Value="Wrap"/>
            <Setter Property="Margin" Value="10,0,0,0"/>
        </Style>
```

Note: If a style contains the x:Key attribute,we can apply it to a control by setting the Style property of the control to the keyed style. In contrast, a style without an x:Key attribute is automatically applied to every control of its target type, if the control doesn't have an explicit style setting.

Here Iam using an Explicit approach , So, after defining individual style tags for a text box and a text control, our XAMLl code will look like

```
<phone:PhoneApplicationPage
    x:Class="SampleStyleTag.MainPage"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:phone="clr-namespace:Microsoft.Phone.Controls;assembly=Microsoft.Phone"
    xmlns:shell="clr-namespace:Microsoft.Phone.Shell;assembly=Microsoft.Phone"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    mc:Ignorable="d" d:DesignWidth="480" d:DesignHeight="768"
    FontFamily="{StaticResource PhoneFontFamilyNormal}"
    FontSize="{StaticResource PhoneFontSizeNormal}"
    Foreground="{StaticResource PhoneForegroundBrush}"
    SupportedOrientations="Portrait" Orientation="Portrait"
    shell:SystemTray.IsVisible="True">

    <UserControl.Resources>
        <Style x:Key="TextBlockStyle" TargetType="TextBlock">
            <Setter Property="FontSize" Value="26"/>
            <Setter Property="FontFamily" Value="Arial"/>
            <Setter Property="FontWeight" Value="SemiBold"/>
            <Setter Property="TextWrapping" Value="Wrap"/>
            <Setter Property="Margin" Value="10,0,0,0"/>
        </Style>

        <Style x:Key="TextBoxStyle" TargetType="TextBox">
            <Setter Property="Height" Value="75"/>
            <Setter Property="TextWrapping" Value="Wrap"/>
            <Setter Property="Margin" Value="0,-10,0,0"/>
        </Style>
    </UserControl.Resources>

   <!--LayoutRoot is the root grid where all page content is placed-->
   <!--Add Grid part here-->
</phone:PhoneApplicationPage>
```

- **Another example of a style tag for the *List Picker Control*** is shown below

```
        <Style x:Key="ListPickerStyle" TargetType="toolkit:ListPicker">
            <Setter Property="HorizontalAlignment" Value="Left"/>
            <Setter Property="Margin" Value="0,0,0,30"/>
            <Setter Property="FontFamily" Value="Segoe WP Bold"/>
            <Setter Property="Height" Value="Auto"/>
            <Setter Property="Width" Value="450"/>
            <Setter Property="FontSize" Value="32"/>
            <Setter Property="VerticalAlignment" Value="Top"/>
        </Style>
```

Using Style Tag in the Controls

We will have to make use of *Style* property of each control to link our defined corresponding style tags. At last, our code will look something like:

```
        <Grid x:Name="ContentPanel" Grid.Row="1" Margin="12,0,12,0">
            <Grid.RowDefinitions>
                <RowDefinition Height=".10*"/>
                <RowDefinition Height=".30*"/>
                <RowDefinition Height=".12*"/>
                <RowDefinition Height=".30*"/>
                <RowDefinition Height=".12*"/>
```

 (C) Copyright Nokia 2013. All rights reserved.

```
                <RowDefinition Height=".30*"/>
                <RowDefinition Height="*"/>
            </Grid.RowDefinitions>

            <TextBlock Grid.Row="0" Text="Name : " Style="{StaticResource TextBlockStyle}"/>
            <TextBox Grid.Row="1" Style="{StaticResource TextBoxStyle}"/>
            <TextBlock Grid.Row="2" Text="Designation : " Style="{StaticResource TextBlockStyle}"/>
            <TextBox Grid.Row="3" Style="{StaticResource TextBoxStyle}"/>
            <TextBlock Grid.Row="4" Text="Department : " Style="{StaticResource TextBlockStyle}"/>
            <TextBox Grid.Row="5" Style="{StaticResource TextBoxStyle}"/>
        </Grid>
    </Grid>
```

- Now build the project and run it using F5.

 (C) Copyright Nokia 2013. All rights reserved.

 (C) Copyright Nokia 2013. All rights reserved.