How to make the pivot control app title match inbuilt apps

This article explains how two approaches for making the application title match that of standard Windows Phone apps.

Introduction

If you create a "Windows Phone Application" or a "Windows Phone Pivot Application" in Visual Studio the application title is **not** displayed as it is the built-in metro applications of the Windows Phone 7. You can verify this by creating a "Windows Phone Pivot Application" and changing the title to match one of the in-built apps.

Compare screenshots of the Settings App and a user-created copy below. Note the differences in the "SETTINGS" application title - both in size and hue.

Settings application (built in)

User created Pivot application

How to match the native style

To approximate the "native app" title design you can use the following style for your pivot control:

```
<Style x:Key="PivotStyle" TargetType="controls:Pivot">
    <Setter Property="TitleTemplate">
        <Setter.Value>
            <DataTemplate>
                <TextBlock FontFamily="{StaticResource PhoneFontFamilySemiBold}" FontSize="{StaticResource PhoneFontSizeMedium}" Mar
            </DataTemplate>
        </Setter.Value>
    </Setter>
</Style>
```

Apply the style:

(C) Copyright Nokia 2013. All rights reserved.

```
<controls:Pivot Title="SETTINGS" Style="{StaticResource PivotStyle}">
```

The same issue happens with every page (`PhoneApplicationPage`). The next section provides an alternative to applying the style to all pages in the application.

The TitleControl

Instead of applying the fix to all the pages of an application you can instead use the `TitleControl` (code is available for download at the end of the article).

Two of the goals behind the `TitleControl` were:

1. Get rid of all the XAML code that displays the header.
2. Remove the unnecessary extra grid that displays the header and the content. Remember, loading speed improves if there are less containers like panels, grids, or stack panels in a page.

So, instead of having the following XAML code:

```
<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="0" Margin="12,17,0,28">
        <TextBlock x:Name="ApplicationTitle" Text="MY APPLICATION" Style="{StaticResource PhoneTextNormalStyle}"/>
        <TextBlock x:Name="PageTitle" Text="page name" Margin="9,-7,0,0" Style="{StaticResource PhoneTextTitle1Style}"/>
    </StackPanel>
    <!--ContentPanel - place additional content here-->
    <Grid x:Name="ContentPanel" Grid.Row="1" Margin="12,0,12,0">
        <TextBlock Text="Hello World" />
    </Grid>

</Grid>
```

The code can be reduced to:

```
<Grid x:Name="LayoutRoot" Background="Transparent">

    <Grid.RowDefinitions>
        <RowDefinition Height="Auto" />
        <RowDefinition Height="Auto" />
    </Grid.RowDefinitions>

    <Controls:TitleControl TitleName="SETTINGS" PageName="theme" />

    <TextBlock Grid.Row="1" Margin="24,0" Text="Hello World" />

</Grid>
```

If you have a brand theme, you can override the color and the font directly in the TitleControl or you can add additional dependency properties to the control.

Download the sample project that includes the TitleControl: File:TitleApp.zip

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.