

Chapter 34: Platform specific visual adjustments

Section 34.1: Idiom adjustments

Idiom specific adjustments can be done from C# code, for example for changing the layout orientation whether the view is shown on a phone or a tablet.

```
if (Device.Idiom == TargetIdiom.Phone)
{
    this.panel.Orientation = StackOrientation.Vertical;
}
else
{
    this.panel.Orientation = StackOrientation.Horizontal;
}
```

Those functionalities are also available directly from XAML code :

```
<StackLayout x:Name="panel">
  <StackLayout.Orientation>
    <OnIdiom x:TypeArguments="StackOrientation">
      <OnIdiom.Phone>Vertical</OnIdiom.Phone>
      <OnIdiom.Tablet>Horizontal</OnIdiom.Tablet>
    </OnIdiom>
  </StackLayout.Orientation>
</StackLayout>
```

Section 34.2: Platform adjustments

Adjustments can be done for specific platforms from C# code, for example for changing padding for all the targeted platforms.

```
if (Device.OS == TargetPlatform.iOS)
{
    panel.Padding = new Thickness (10);
}
else
{
    panel.Padding = new Thickness (20);
}
```

An helper method is also available for shortened C# declarations :

```
panel.Padding = new Thickness (Device.OnPlatform(10,20,0));
```

Those functionalities are also available directly from XAML code :

```
<StackLayout x:Name="panel">
  <StackLayout.Padding>
    <OnPlatform x:TypeArguments="Thickness"
      iOS="10"
      Android="20" />
  </StackLayout.Padding>
</StackLayout>
```

Section 34.3: Using styles

When working with XAML, using a centralized `Style` allows you to update a set of styled views from one place. All the idiom and platform adjustments can also be integrated to your styles.

```
<Style TargetType="StackLayout">
  <Setter Property="Padding">
    <Setter.Value>
      <OnPlatform x:TypeArguments="Thickness"
        iOS="10"
        Android="20" />
    </Setter.Value>
  </Setter>
</Style>
```

Section 34.4: Using custom views

You can create custom views that can be integrated to your page thanks to those adjustment tools.

Select `File > New > File... > Forms > Forms ContentView (Xaml)` and create a view for each specific layout: `TabletHome.xaml` and `PhoneHome.xaml`.

Then select `File > New > File... > Forms > Forms ContentPage` and create a `HomePage.cs` that contains:

```
using Xamarin.Forms;

public class HomePage : ContentPage
{
    public HomePage()
    {
        if (Device.Idiom == TargetIdiom.Phone)
        {
            Content = new PhoneHome();
        }
        else
        {
            Content = new TabletHome();
        }
    }
}
```

You now have a `HomePage` that creates a different view hierarchy for `Phone` and `Tablet` idioms.
