Images

By: Mosh Hamedani

Image Sources

- Images can be platform-independent (e.g. backgrounds) and can be downloaded using a URI or embedded in the PCL (to ship with the app).
- Or, they can be platform-specific (e.g. icons). They should be added to each application project.
- Image.Source is of type ImageSource, which is an abstract class. Its derivates are **UriImageSource**, **ResourceImageSource** and **FileImageSource**.

Downloading Images

In XAML

Caching is enabled by default.

```
<Image Source="http://…" />
```

In Code

Useful for disabling caching or overriding validity period.

```
image.Source = new UriImageSource
{
    Uri = new Uri("http://…"),
    CachingEnabled = false,
    CacheValidity = TimeSpan.FromHours(1)
};
```

Images

By: Mosh Hamedani

Embedding Images

Add to PCL and set the Build Action to **EmbeddedResource**.

In Code

```
image.Source = ImageSource.FromResource("…resourceId…");
```

In XAML

Create a custom markup extension:

```
public class EmbeddedImage : IMarkupExtension { … }
```

```
<ContentPage
   xmlns:local="clr-namespace:HelloWorld;assembly=HelloWorld">

    <Image Source="{local:EmbeddedImage ResourceId}" />
```

Platform-specific Images

Add to each application project.

iOS

- Resources/clock.png
- Resources/clock@2x.png
- Resources/clock@3x.png

Images

By: Mosh Hamedani

Android

- Resources/drawable/clock.png
- Resources/drawable-hdpi/clock.png
- Resources/drawable-xhdpi/clock.png
- Resources/drawable-xxhdpi/clock.png

In Code

```
image.Source = ImageSource.FromFile("clock.png");
```

In XAML

```
<Image Source="clock.png" />
```

Image Aspects

- **AspectFit** (default): image is scaled to fit the container
- **AspectFill**: image is cropped to fill the container
- **Fill**: image fills the container but it can get distorted

Images

By: Mosh Hamedani

Activity Indicator

```
<ActivityIndicator
    Color="White"
    IsRunning="{Binding
         Source={x:Reference image},
         Path=IsLoading}" />
```