Работа с изображениями

Изображения являются важной частью навигации по приложениям, их удобства использования и фирменной символики. Программные приложения должны иметь возможность совместно использовать изображения на всех платформах, но на каждой платформе изображения могут отображаться различным образом.

Платформа Xamarin.Forms использует элемент управления Image для отображения изображений на странице. Он имеет несколько важных свойств:

Свойство Source с помощью объекта ImageSource определяет содержит путь к файлу, URI или ресурс, который задает отображаемое изображение;

Свойство Aspect задает изменение размера изображения в пределах границ, в которых оно отображается.

Объект ImageSource использует следующие методы для доступа к файлам изображений:

FromFile задает имя файла или путь к файлу;

FromUri задает URI (например, new Uri("http://server.com/image.jpg");

FromResource задает идентификатор ресурса для файла изображения, внедренного в проект библиотеки приложения или .NET Standard;

FromStream задает поток, из которого получается изображение.

Свойство Aspect определяет, каким образом изображение будет масштабироваться в соответствии с размерами той области, в котором оно будет отображаться:

значение Fill обозначает растяжение изображения для того, чтобы полностью и точно заполнить размеры области отображения;

значение AspectFill означает обрезание краев изображения таким образом, чтобы оно заполнило область отображения (в случае, если размеры изображения больше размеров области отображения);

значение AspectFit означает размещение изображения в пределах области отображения с добавлением пробелов для дополнения размеров изображения до размеров области отображения (в случае, если размеры изображения меньше размеров области отображения).

Для отображения изображений могут быть использованы следующие элементы управления:

элемент управления Button, который имеет свойство ImageSource для работы с изображениями;

элемент управления ImageButton, который имеет свойство Source, для работы с изображением;

элемент управления ToolbarItem, являющийся специальной разновидностью кнопки, которую можно добавить на страницу, и который имеет свойство IconImageSource для работы с изображением;

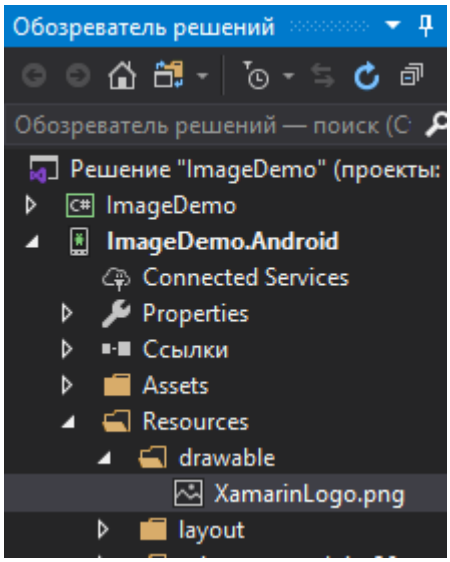
элемент управления ImageCell который имеет свойство ImageSource для работы с изображением;

страницы любого типа, которые могут работать с изображениями с помощью свойства BackgroundImageSource.

Файл изображения можно добавить в проект приложения и ссылаться на него из программного кода. Для работы с одним и тем же изображением на разных платформах необходимо использовать одно и то же имя файла, являющееся допустимым на каждой платформе.

Для работы на платформе Android с локальным изображением необходимо разместить его в папке Drawable, входящей в папку Resources.

Для этого необходимо развернуть папку Resources в проекте Android. Затем необходимо перетащить файл из той папки, где он находится в папку **Drawable**.



Такое размещение файлов с изображениями позволяет программному коду загрузить и отобразить изображение:

Программный код на XAML

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

Программный код на Си Шарп:

```
var image = new Image {Source = " XamarinLogo.png"};
```

В случае, если отображаемое изображение зависит от платформы, на которой запущено программное приложение, то для выбора отображаемого изображения необходимо использовать свойство Device.RuntimePlatform, что отображено в следующем фрагменте программного кода на языке Си Шарп:

```
image.Source = Device.RuntimePlatform == Device.Android
        ? ImageSource.FromFile("ImageAndroid.jpg")
        : ImageSource.FromFile("Images/ImageIOS.jpg");
```

Расширение разметки OnPlatform позволяет настраивать внешний вид пользовательского интерфейса в зависимости от платформы, на которой запущено программное приложение.

```
<Image Source="https:// upload.wikimedia.org/wikipedia/Papio.jpg"
       Aspect="Fill"
       HorizontalOptions="Center"
       HeightRequest="{OnPlatform iOS=300, Android=200}"
       WidthRequest="{OnPlatform iOS=300, Android=200}"
       />
```

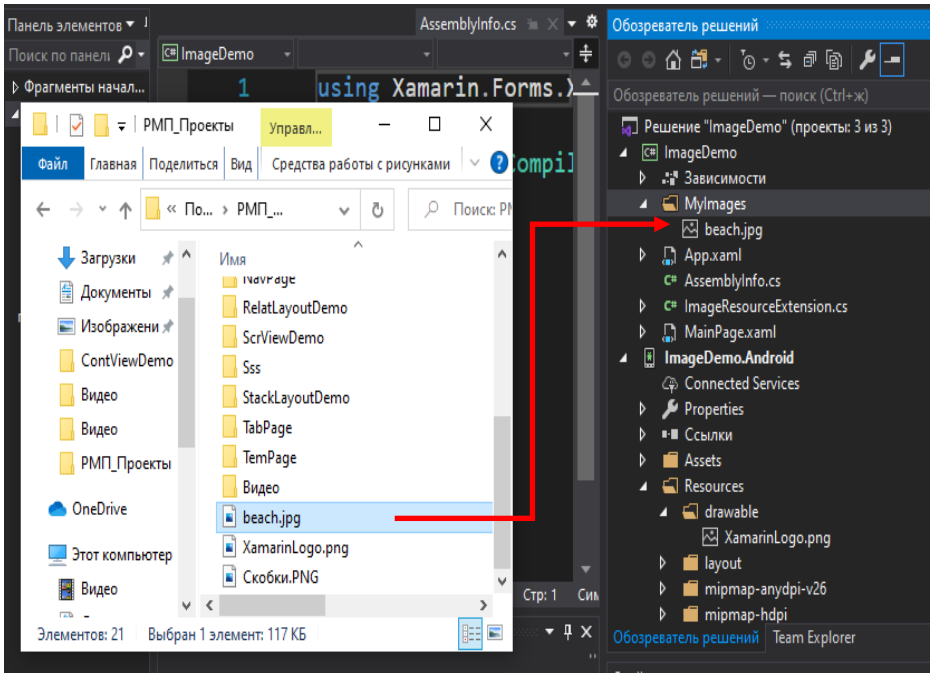
Этот код задает свойство Aspect, которое определяет режим Fill для масштабирования изображения.

Расширение разметки OnPlatform используется для установки свойств HeightRequest и WidthRequest равным 300 единицам для платформы iOS и 200 единиц для Android.

Свойство HorizontalOptions указывает, что изображение будет отцентрировано по горизонтали.

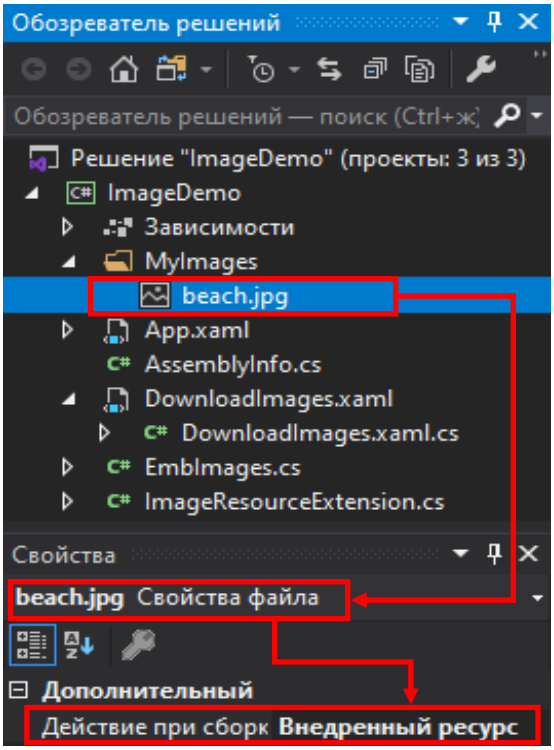
Программные приложения могут поставляться с внедренными изображениями. При этом файл изображения используется в качестве ресурса.

Чтобы внедрить изображение в проект, необходимо перенести изображение рисунка из внешнего источника в проект. Далее приведен пример, когда файл beach.jpg перенесен из внешнего источника в заранее созданную папку MyImages.



Перемещение изображения beach.jpg в папку MyImages в проекте ImageDemo приведет к получению идентификатора ресурса ImageDemo.MyImages.beach.jpg.

По умолчанию в окне свойств для изображения будет задано значение свойства Build Action (Действие при сборке), равное None (Нет). Для этого свойства необходимо задать значение EmbeddedResource (Внедренный ресурс).



Для загрузки внедренного изображения идентификатор ресурса передается в метод ImageSource.FromResource:

```
Image embeddedImage = new Image
{
   Source = ImageSource.FromResource("ImageDemo.MyImages.beach.jpg",
                                     typeof(MyClass).GetTypeInfo().Assembly)
};
```

Также отображаемые в пользовательском интерфейсе изображения могут быть скачаны из внешних источников. Программный код для такого случая имеет следующий вид:

Для XAML

```
 <Image Source="https://ak.ms/camp.jpg" />
```

Для Си Шарп:

```
var webImage = new Image {Source =
      ImageSource.FromUri(new Uri("https://ak.ms/camp.jpg"))};
```

Платформа Xamarin.Forms также поддерживает отображения анимированных изображений из файлов с расширением GIF. Поддержка работы с такими изображениями производится с помощью присвоения имени файла свойству Source

```
<Image Source="GIF_File.gif" />
```

Анимированный GIF-файл не будет воспроизводиться до тех пор, пока для свойства IsAnimationPlaying не будет установлено значение true.

Воспроизведение можно остановить, задав значение false для свойства IsAnimationPlaying. Если расширение файла изображения отличается от GIF, то свойство не действует при отображении изображения.

Для демонстрации работы с изображениями создадим проект ImageDemo для мультиплатформенного программного приложения Xamarin.Forms с использованием шаблона «Пустой».

Перед началом работы с программным кодом в папку Drawable проекта ImageDemo.Android необходимо скопировать файлы изображений с расширениями png и gif (файлы XamarinLogo.png и GIF_File.gif).

Далее необходимо в проекте ImageDemo создать папку MyImages. Для этого курсор мыши необходимо подвести к этому проекту, нажать правую кнопку мыши, выбрать пункт меню «Добавить» и создать папку. В созданную папку необходимо перенести файл с изображением с расширением jpg. В примере, приведенном далее, в папке MyImages находится файл beach.jpg. Необходимо для этого файла установить значение «Внедренный ресурс» для свойства «Действие при сборке».

Откроем файл App.xaml.cs и наберем программный код Си Шарп, приведенный далее.

В программном коде сначала задается переменная xamlTab, с помощью которой задается страница типа TabbedPage с четырьмя вкладками. С помощью переменной xamlTab задается страница MainPage типа TabbedPage, которая будет отображаться при запуске программного приложения.

Первая вкладка отображает страницу, которая реализуется с помощью программного кода, приведенного в файле LocalImages.xaml.

Вторая вкладка отображает страницу, которая реализуется с помощью программного кода, приведенного в файле DownloadImages.xaml.

Третья вкладка отображает страницу, которая реализуется с помощью программного кода, приведенного в файле EmbImages.xaml.

Четвертая вкладка отображает страницу, которая реализуется с помощью программного кода, приведенного в файле GIFImages.xaml.

```
using System;
using Xamarin.Forms;
using Xamarin.Forms.Xaml;

namespace ImageDemo
{
  public partial class App : Application
  {
    public App()
    {
```

```
        var xamlTab = new TabbedPage();
        xamlTab.Children.Add(new LocalImages { Title = "Local" });
        xamlTab.Children.Add(new DownloadImages
                             { Title = "Downloaded" });
        xamlTab.Children.Add(new EmbImages
                             { Title = "Embedded" });
         xamlTab.Children.Add(new GIFImages
                              { Title = "GIF Files" });

        MainPage = xamlTab;
      }
    }
}
```

Добавим в проект ImageDemo страницу содержимого с именем LocalImages.xaml.

Откроем этот файл и наберем программный код, приведенный далее.

Программный код предусматривает работу со страницей типа ContentPage, которая предназначена для отображения изображения, получаемого из локального файла. Страница содержит макет StackLayout, внутри которого располагается изображение и элемент управления Label для отображения поясняющего текста. Производится работа с файлом XamarinLogo.png, который содержится в папке Drawable. В программном коде задана область, которую занимает изображение (то есть, его ширина и высота), а также задано значение свойства Aspect, которое равно Fill. То есть изображение растягивается в пределах выделенной под него области. Также с помощью свойства Margin задано расположение элементов управления внутри макета.

```
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
         xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
         x:Class="ImageDemo.LocalImages"
         BackgroundColor="YellowGreen">
  <ContentPage.Content>
    <StackLayout Margin="10,150,10,10">
      <Label
      Text="Изображение из локального файла"
        FontAttributes="Bold"
        HorizontalOptions="Center"
        FontSize="50" TextColor="Black"
        HorizontalTextAlignment="Center"/>
      <Image
      HeightRequest="200"
      WidthRequest="350"
      HorizontalOptions="Center"
      Source="XamarinLogo.png"
      Aspect="Fill"/>
    </StackLayout>
```

```
  </ContentPage.Content>
</ContentPage>
```

Добавим в проект ImageDemo страницу содержимого с именем DownloadImages.xaml. Откроем этот файл и наберем программный код, приведенный далее.

Программный код предусматривает работу со страницей типа ContentPage, которая предназначена для отображения изображения, загружаемого с внешнего Интернет-ресурса.

Страница содержит макет StackLayout, внутри которого располагается изображение и элемент управления Label для отображения поясняющего текста.

С помощью свойства Source производится подключение к Интернет ресурсу.

Следует учесть, что при присвоении значения свойству Source текст с адресом ресурса следует расположит не в несколько строк, как это приведено далее для наглядности, а в одну строку.

В случае многострочного представления адреса ресурса изображение из внешнего источника может не отобразиться на странице.

В программном коде с помощью расширения разметки OnPlatform производится настройка области, которую занимает изображение, в зависимости от платформы, на которой запущено программное приложение. У изображения задано значение свойства Aspect, равное Fill. Также с помощью свойства Margin задано расположение элементов управления внутри макета.

```
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
        xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
        x:Class="ImageDemo.DownloadImages"
        BackgroundColor="Coral">
  <ContentPage.Content>
    <StackLayout Margin="10,100,10,10">
      <Label
        Text="Изображение
            загружается"
        FontAttributes="Bold"
        HorizontalOptions="Center"
        FontSize="50" TextColor="Yellow"
        HorizontalTextAlignment="Center"/>

      <Image
        Source="https://upload.wikimedia.org/wikipedia/commons/
                thumb/f/fc/Papio_anubis_%28Serengeti%2C_2009%29.jpg/200px-
                Papio_anubis_%28Serengeti%2C_2009%29.jpg"
        Aspect="Fill"

        HeightRequest="{OnPlatform iOS=300, Android=400}"
        WidthRequest="{OnPlatform iOS=300, Android=400}"
```

```
        HorizontalOptions="Center" />
    </StackLayout>
  </ContentPage.Content>
</ContentPage>
```

Добавим в проект ImageDemo класс с именем EmbImages. Откроем файл EmbImages.cs и наберем программный код Си Шарп, приведенный далее.

Программный код предусматривает работу со страницей типа ContentPage, которая предназначена для отображения изображения, которое внедрено в программное приложение в качестве ресурса.

В примере, приведенном далее, производится работа с файлом beach.jpg, который ранее был загружен в проект в папку и настроен как ресурс.

Страница содержит макет StackLayout, внутри которого располагается изображение и элемент управления Label для отображения поясняющего текста.

С помощью свойства Source и метода ImageSource.FromResource производится подключение к файлу-ресурсу.

Производится настройка высоты отображаемого изображения, а также задано значение свойства Aspect, равное Fill.

Также с помощью свойства Margin задано расположение элементов управления внутри макета

```
using System;
using System.Collections.Generic;
using System.Text;
using Xamarin.Forms;
using System.Reflection;


namespace ImageDemo
{
        public class EmbImages : ContentPage
        {
                public EmbImages()
                {
                        var EmbedImage = new Image {Aspect = Aspect.Fill};

                        EmbedImage.Source = ImageSource.FromResource(
                                "ImageDemo.MyImages.beach.jpg",
                                typeof(EmbImages).GetTypeInfo().Assembly);

                        EmbedImage.HeightRequest = 350;

                        BackgroundColor = Color.Gold;
                        Content = new StackLayout
                        {
                                Children = {
```

```
                                    new Label {
                                Text = "Изображение как присоединенный ресурс",
                                 FontSize = 40,
                                 TextColor=Color.Black,
                                 FontAttributes = FontAttributes.Bold,
                                 HorizontalOptions = LayoutOptions.Center,
                                 HorizontalTextAlignment=TextAlignment.Center
                                    },
                                    EmbedImage
                            },
                            Margin = new Thickness(20, 50, 20, 20),
                            VerticalOptions = LayoutOptions.StartAndExpand,
                            HorizontalOptions = LayoutOptions.CenterAndExpand
                        };

                }
            }
}
```

Добавим в проект ImageDemo страницу содержимого с именем GIFImages.xaml.

Откроем этот файл и наберем программный код, приведенный далее.

Программный код предусматривает работу со страницей типа ContentPage, которая предназначена для отображения анимированного изображения с использованием файла, с расширением GIF.

Страница содержит макет StackLayout, внутри которого располагается изображение и элемент управления Label для отображения поясняющего текста.

Производится работа с файлом GIF_File.gif, который содержится в папке Drawable. В программном коде заданы размеры области, которую занимает изображение. Задано значение свойства Aspect, а также с помощью свойства Margin задано расположение элементов управления внутри макета. Для воспроизведения анимированного GIF-файла установлено значение true для свойства IsAnimationPlaying.

```
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
        xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
        x:Class="ImageDemo.GIFImages"
        BackgroundColor="White">
  <ContentPage.Content>
    <StackLayout Margin="10,150,10,10">
      <Label
      Text="Загрузка GIF-файла"
        FontAttributes="Bold"
        HorizontalOptions="Center"
        FontSize="50" TextColor="Black"
        HorizontalTextAlignment="Center"/>
      <Image
```

```
        IsAnimationPlaying="True"
        HeightRequest="200"
        WidthRequest="350"
        HorizontalOptions="Center"
        Source="GIF_File.gif"
        Aspect="Fill"/>
    </StackLayout>
  </ContentPage.Content>
</ContentPage>
```

Запустим программное приложение для демонстрации работы с изображениями.

Отображается главная страница с четырьмя вкладками.

Первая вкладка отображает страницу, которая реализует работу с локальными изображениями.

Вторая вкладка отображает страницу, которая реализует работу с изображением, скачиваемым из Интернета.

Третья вкладка отображает страницу, которая реализует работу с изображением, которое внедрено в проект как ресурс.

Четвертая вкладка отображает страницу, которая реализует работу с анимированным файлом.