

# Valores específicos de plataforma em XAML

5 minutos

A experiência visual do aplicativo será diferente em cada plataforma. Geralmente, você precisará ajustar a interface do usuário para cada plataforma com base nos elementos visuais que você usa. O .NET MAUI permite que você gerencie o layout do aplicativo com base nessas propriedades do dispositivo.

Nesta unidade, você aprenderá sobre os recursos fornecidos pelo .NET MAUI para ajustar a interface do usuário de seus aplicativos de acordo com a plataforma na qual eles estão em execução.

## Usando a classe Device

A classe `DeviceInfo` é uma classe de utilitário que fornece informações específicas do dispositivo em que o aplicativo está em execução. Ela expõe essas informações por meio de um conjunto de propriedades. A propriedade mais importante é `DeviceInfo.Platform`, que retorna uma cadeia de caracteres indicando o tipo de dispositivo atualmente em uso: `Android`, `iOS`, `WinUI` ou `macOS`.

Considere o cenário a seguir como um exemplo de quando usar esse recurso. O comportamento padrão em um aplicativo .NET MAUI no iOS é que o conteúdo adicionado a uma página invada a barra de status do iOS na parte superior da tela. Você quer alterar esse comportamento. A solução mais simples é deslocar o conteúdo para baixo na página. A solução Notas criada no exercício anterior resolve esse problema definindo a propriedade `Padding` do controle `VerticalStackLayout` para mover o conteúdo para baixo em 60 pontos:

XML

```
<VerticalStackLayout x:Name="MyStackLayout" Padding="30,60,30,30">
    ...
</VerticalStackLayout>
```

A questão é que esse problema só se aplica ao iOS. Deslocar o conteúdo para baixo nessa proporção no Android e no WinUI resulta em desperdício de espaço na tela na parte superior da página.

Você pode consultar a propriedade `DeviceInfo.Platform` para resolver esse problema de exibição. Você pode adicionar o seguinte código ao construtor de página do aplicativo para expandir o preenchimento na parte superior da página, mas somente para iOS:

C#

```
MyStackLayout.Padding =  
    DeviceInfo.Platform == DevicePlatform.iOS  
        ? new Thickness(30, 60, 30, 30) // Shift down by 60 points on iOS only  
        : new Thickness(30); // Set the default margin to be 30 points
```

### ⚠ Observação

`DevicePlatform.iOS` é um struct `DevicePlatform` que retorna o valor de cadeia de caracteres `ios`. Há propriedades equivalentes para outras plataformas com suporte. Você deve usar essas propriedades em vez de comparar com cadeias de caracteres codificadas; é uma boa prática e torna seu código à prova de futuro se alguns desses valores de cadeia de caracteres mudarem no futuro.

Esse código funciona, mas está no arquivo code-behind da página. O preenchimento é um valor específico da interface do usuário. Sem dúvida, é mais apropriado e conveniente fazer isso usando XAML em vez de code-behind.

## Uso da extensão de marcação `OnPlatform`

O XAML do .NET MAUI fornece a extensão de marcação `OnPlatform`, que permite detectar a plataforma de runtime de dentro do código XAML. Você aplica essa extensão de marcação como parte do código XAML que define um valor da propriedade. A extensão requer que você forneça o tipo da propriedade, juntamente com uma série de blocos `on Platform` em que você define o valor da propriedade de acordo com a plataforma.

### ⚠ Observação

A extensão de marcação `onPlatform` é genérica e usa um parâmetro de tipo. O tipo especificado pelo atributo `TypeArguments` garante que o tipo correto de extensão seja usado.

Você pode definir a propriedade `Padding` dessa forma. Observe que o tipo da propriedade `Padding` é `Thickness`:

XML

```
<VerticalStackLayout>
  <VerticalStackLayout.Padding>
    <OnPlatform x:TypeArguments="Thickness">
      <On Platform="iOS" Value="30,60,30,30" />
    </OnPlatform>
  </VerticalStackLayout.Padding>
  <!--XAML for other controls goes here -->
  ...
</VerticalStackLayout>
```

Para plataformas diferentes do iOS, o preenchimento permanecerá definido com o valor padrão de "0,0,0,0". Para WinUI e Android, você pode definir o preenchimento como 30 pontos com blocos `OnPlatform` adicionais:

XML

```
<VerticalStackLayout>
  <VerticalStackLayout.Padding>
    <OnPlatform x:TypeArguments="Thickness">
      <On Platform="iOS" Value="30,60,30,30" />
      <On Platform="Android" Value="30" />
      <On Platform="WinUI" Value="30" />
    </OnPlatform>
  </VerticalStackLayout.Padding>
  ...
</VerticalStackLayout>
```

Você pode aplicar a mesma técnica a outras propriedades. O exemplo a seguir altera a cor da tela de fundo do layout empilhado em uma página para prata no iOS, verde no Android e amarelo no Windows.

XML

```
<VerticalStackLayout>
  ...
  <VerticalStackLayout.BackgroundColor>
    <OnPlatform x:TypeArguments="Color">
      <On Platform="iOS" Value="Silver" />
      <On Platform="Android" Value="Green" />
      <On Platform="WinUI" Value="Yellow" />
    </OnPlatform>
  </VerticalStackLayout.BackgroundColor>
  ...
</VerticalStackLayout>
```

Essa sintaxe é um pouco detalhada, mas uma sintaxe reduzida está disponível para a extensão `OnPlatform`. Você pode simplificar o exemplo que define o preenchimento da seguinte maneira:

XML

```
<VerticalStackLayout Padding="{OnPlatform iOS='30,60,30,30', Default='30'}">  
    <!--XAML for other controls goes here -->  
</VerticalStackLayout>
```

Especifique um valor padrão para uma propriedade, junto com qualquer valor específico da plataforma. Nesse formulário, o parâmetro de tipo é inferido com base na propriedade à qual o atributo `onPlatform` é aplicado.

Para definir a cor da tela de fundo, você pode usar este fragmento XAML no lugar do segundo exemplo anterior:

XML

```
<VerticalStackLayout BackgroundColor="{OnPlatform WinUI=Yellow, iOS=Silver,  
    ...  
    Android=Green}">  
</VerticalStackLayout>
```

## Verificação de conhecimentos

1. Qual extensão de marcação você pode usar no código XAML para detectar a plataforma na qual o aplicativo está em execução? \*

☐ `DevicePlatform`

☐ `DeviceInfo`

☒ `OnPlatform`

✓ Esta é a resposta correta.

2. Qual é a finalidade do atributo `TypeArguments` para a extensão `OnPlatform`? \*

☒ Ele especifica o parâmetro de tipo para a extensão e garante que o tipo correto de extensão seja criado.

✓ Esta é a resposta correta.

- ☐ Ele especifica o tipo da plataforma na qual o aplicativo está em execução
- ☐ Ele permite que você especifique a propriedade a ser definida com um valor específico da plataforma.

---

## Unidade seguinte: Exercício: Adicionar comportamento à sua página XAML

[Continuar >](#)