Xamarin XAML 语言教程基础语法篇

（内部资料 v1.0）

大学霸
www.daxueba.net

前　言

Xamarin 是一个跨平台开发框架。它可以用来开发 iOS、Android、Windows Phone 和 Mac 的应用程序。使用 Xamarin 框架中的 Forms 子框架，用户可以一次性的开发多个平台的应用，如 iOS、Android、Windows Phone，从而节省大量的开发时间。

在 Xamarin.Forms 中，用户可以直接使用 XAML 语言直接进行界面设计。这样，就可以将界面和逻辑代码分离，使得应用程序的结构更加清晰。为了满足大家的开发需求，本教程着眼于 Xamarin.Forms 开发，详细讲解 XAML 语言在界面设计中的使用。同时为了方便大家理解，我们为每个知识点都配以小实例。

1.学习所需的系统和软件

- ❑ 安装 Windows 10 操作系统
- ❑ Xamarin 4.2.0.719
- ❑ 安装 OS X 10.11
- ❑ 安装 Xcode 8.0

2.学习建议

大家学习之前，可以到百度网盘（下载链接：xxxxxxxx）获取相关的资料和软件。如果大家在学习过程遇到问题，也可以将问题发送到邮箱 xxxxxxxx。我们尽可能给大家解决。

目　录

第 1 章　XAML 语言基础

Xamarin.Forms 允许开发人员通过 XAML 语法对程序的所有用户界面元素进行详细的定制，如文本、按钮、图像和列表框等。同时，开发者还可以借助它对整个界面进行合理化的布局。通过 XAML 来构建 UI 界面具有简洁、可视化等优点，非常适合 MVVM 的应用程序架构。本章将讲解关于 XAML 语言基础内容，其中包括 XAML 语言简介、创建 XAML 文件、XAML 文件构成、元素构成等内容。

1.1　XAML 语言简介

XAML 是 Extensible Application Markup Language 的英文缩写，相应的中文名称为“可扩展应用程序标记语言”。它是微软公司为构建应用程序用户界面而创建的一种新的描述性语言。它基于 Extensive Markup Language（XML）可扩展标记语言。XAML 提供了一种便于扩展和定位的语法来定义和程序逻辑分离的用户界面，而这种实现方式和 ASP.NET 中的“代码后置”模型非常类似。XAML 是一种解释性的语言，尽管它也可以被编译。它的优点是简化编程式上的用户创建过程，应用时要添加代码和配置等等。

1.2　创建 XAML 文件

在 Xamarin.Forms 中，XAML 代码保存在 XAML 文件中。开发者在编写 XAML 代码时，首先需要创建对应的 XAML 文件。本节将讲解使用两种开发工具创建 XAML 文件，第一种是使用 Visual Studio 创建 XAML；第二种是使用 Xamarin Studio 创建 XAML。

1.2.1　使用 Visual Studio 创建 XAML

使用 Visual Studio 创建 XAML 文件有两种方式。第一种是创建项目后再创建 XAML 文件，第二种是创建项目时创建 XAML 文件。下面对这两种方式详细介绍。

1.创建项目后再创建 XAML 文件

以下我们将以创建项目 Hello 为例，为开发者讲解创建 PCL 类型的项目后，再创建 XAML 文件的具体操作步骤：

（1）打开 Visual Studio，如图 1.1 所示。

（2）单击“新建项目…”按钮，弹出“新建项目”界面，如图 1.2 所示。

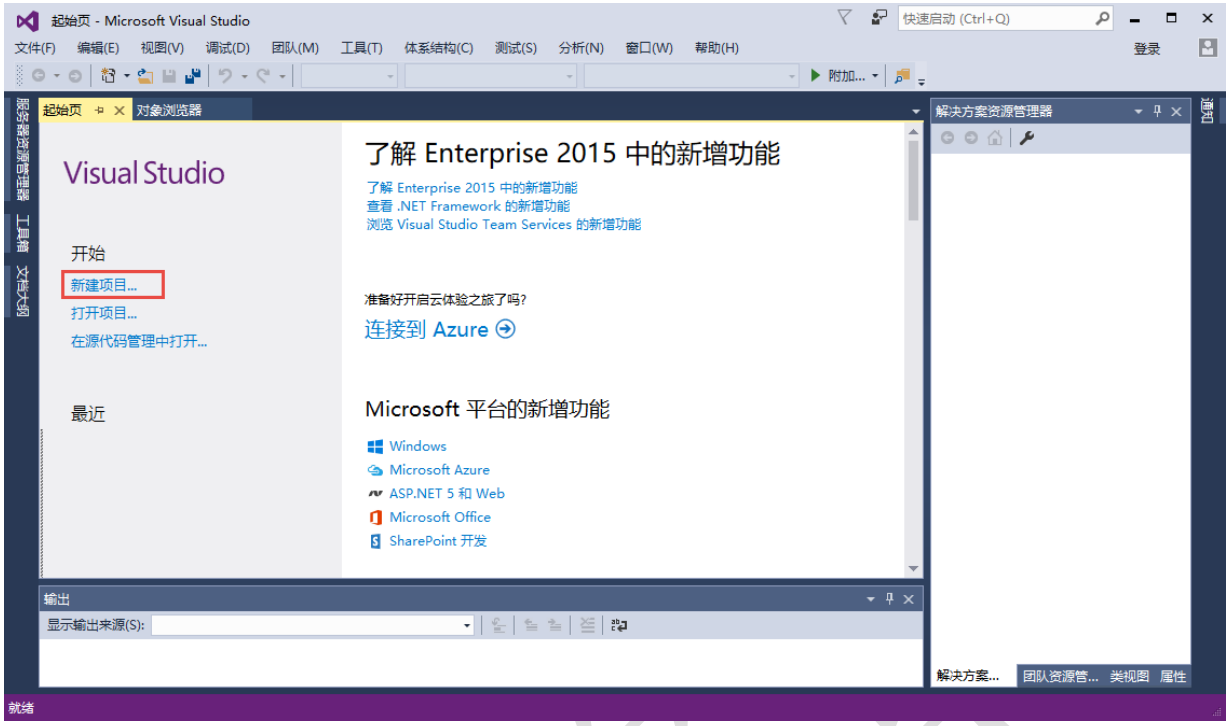

图 1.1　启动 VS

（3）选择 Cross-Plaform 下的 Blank App (Xamarin.Forms Portable) 模板，将名称、位置进行修改，这里我们将名称改为了 Hello，将位置改为了 D:\Code\XAML。

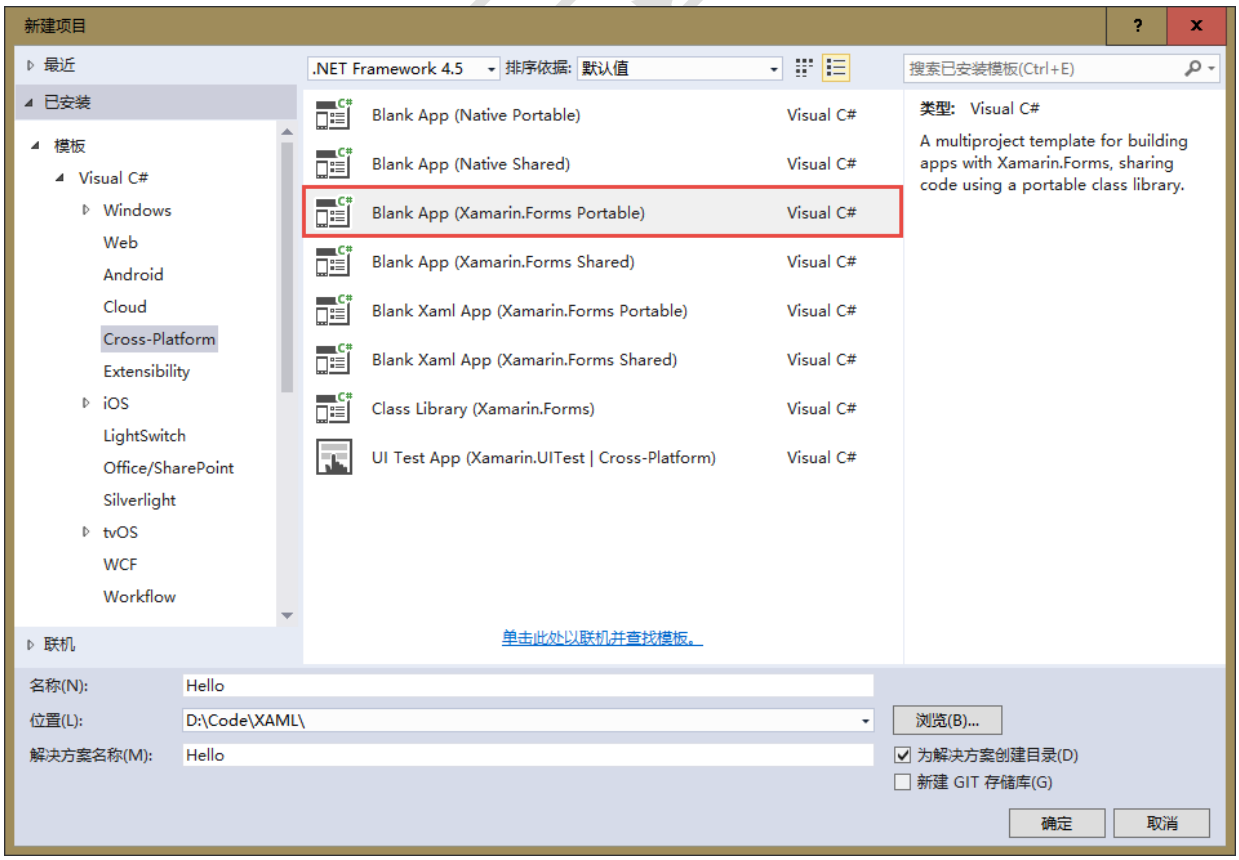

图 1.2 “新建项目”界面

（4）单击“确定”按钮，此时就创建好了一个名称 Hello 的项目。

（5）右击 Hello (可移植的) 项目，弹出快捷菜单，如图 1.3 所示。

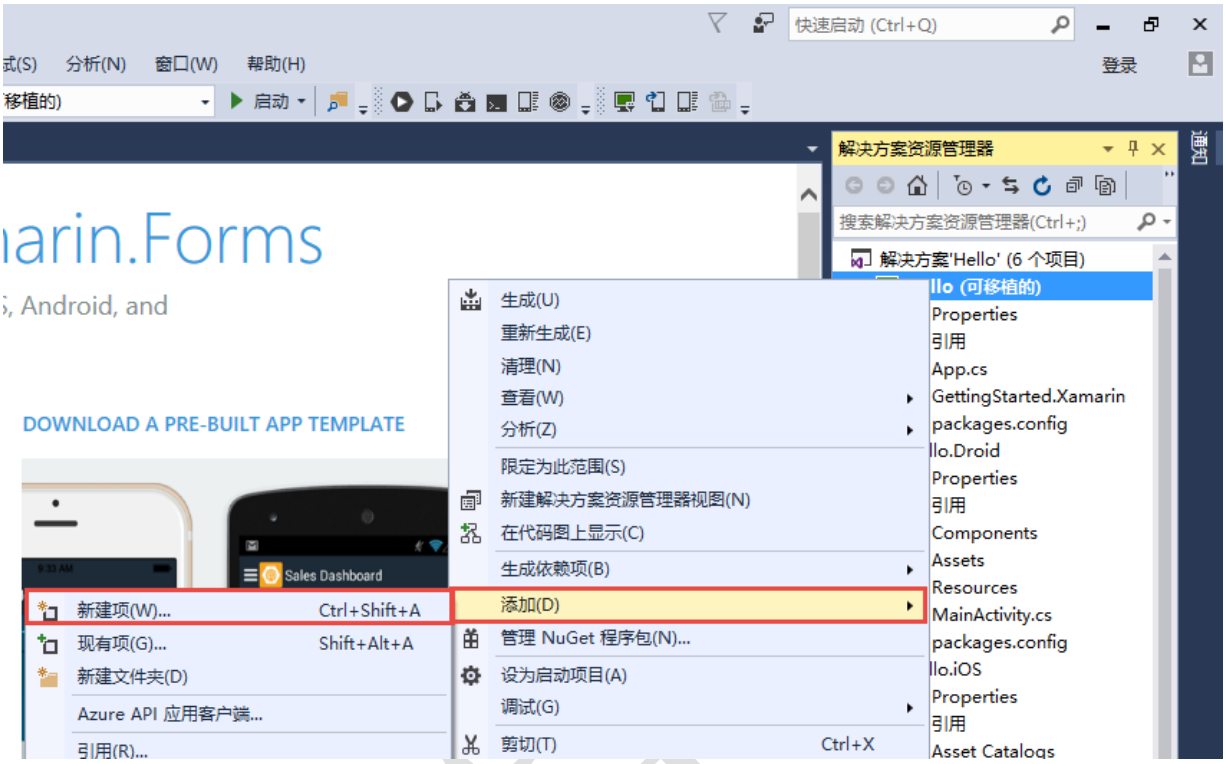

图 1.3 菜单

（6）在弹出的快捷菜单中选择“添加(D)”|“新建项(W)…”命令，弹出“添加新项 - Hello”界面，如图 1.4 所示。

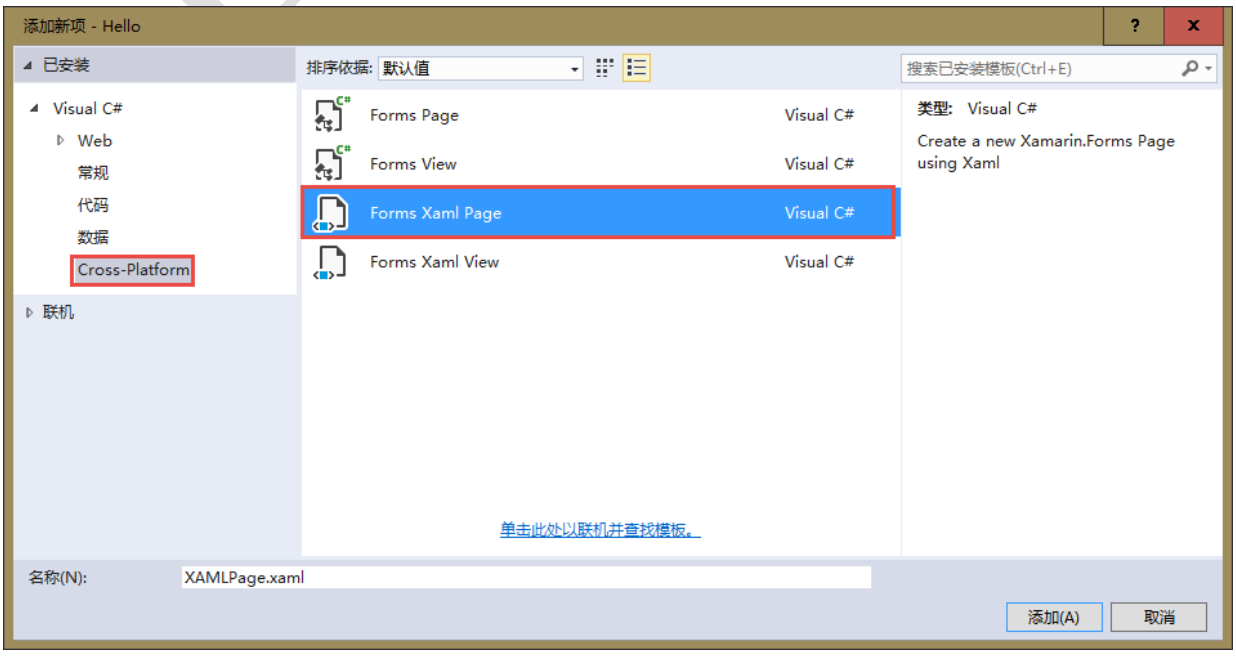

图 1.4　“添加新项 - Hello”界面

（7）选择 Cross-Platform 下的 Forms Xaml Page 项，并将名称改为 XAMLPage.xaml。

注意：如果开发者不修改图 1.4 中的名称，默认为 Page1.xaml。

（8）单击“添加(A)”按钮，此时一个 XAML 文件就创建好了，此文件的名称为 XAMLPage，如图 1.5 所示。

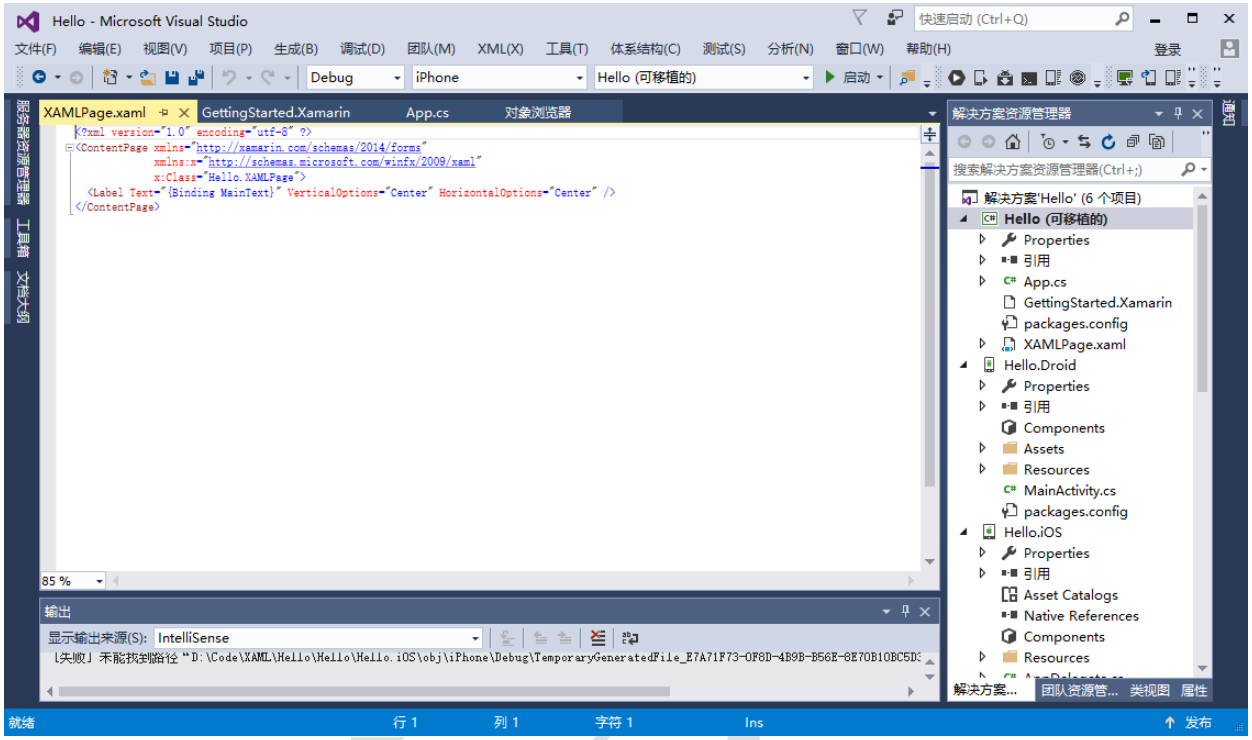

图 1.5　XAMLPage.xaml 文件

注意：在此 Xamarin 版本中，还不支持基于 Share 类型项目创建 XAML 文件。

2.创建项目时创建 XAML 文件

在 Visual Studio 中，创建项目时创建 XAML 文件是一种最常用的方式，并且是最为简单的的方式。以下我们将以 Hello 为例，为开发者讲解创建项目时创建 XAML 文件的具体操作步骤：

（1）在计算机上找到 Visual Studio，将其打开示。

（2）选择“新建项目...”按钮，弹出“新建项目”界面，如图 1.6 所示。

（3）选择 Cross-Plaform 下的 Blank Xaml App (Xamarin.Forms Portable) 模板，将名称、位置进行修改，这里我们将名称改为了 Hello，将位置改为了 D:\Code\XAML。

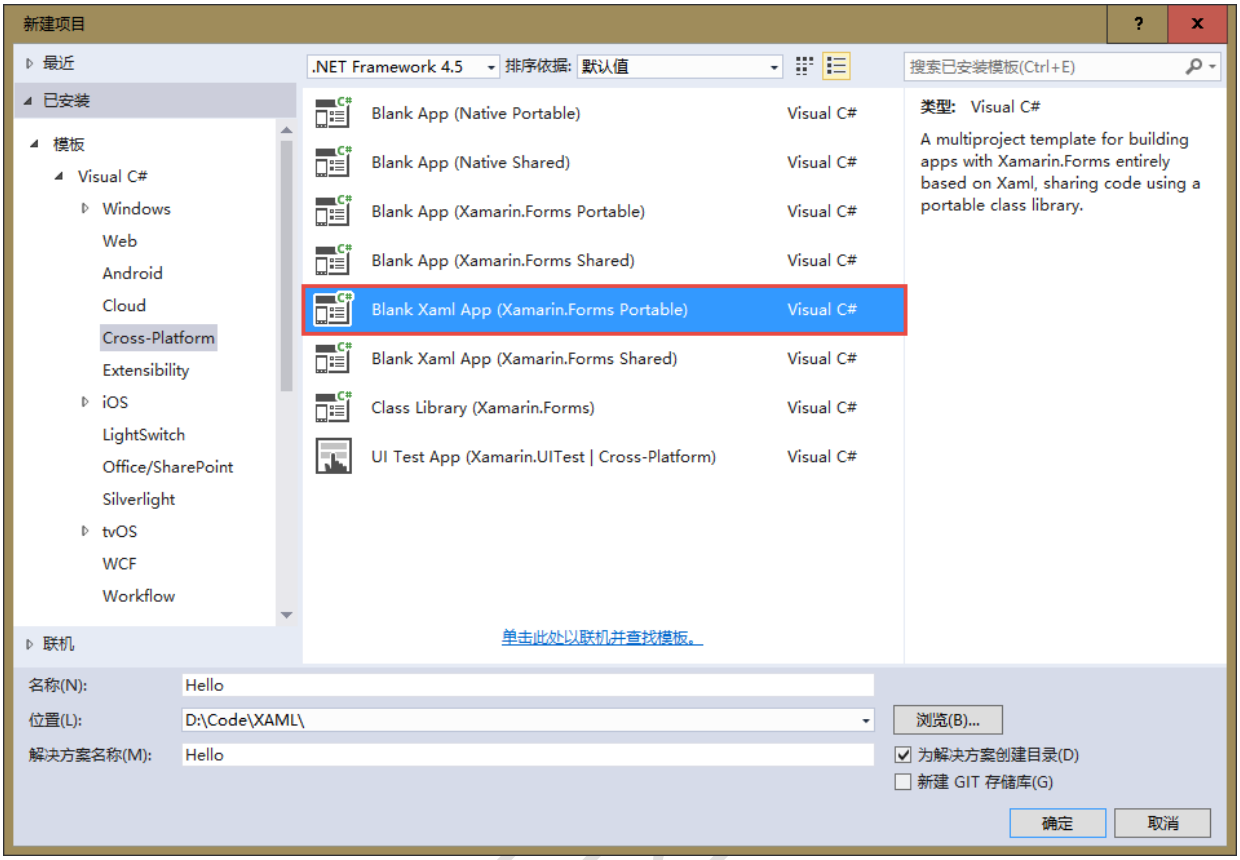

图 1.6 “新建项目”界面

（4）单击“确定”按钮，此时就创建好了一个名称 Hello 的项目。我们可以看到，在创建的项目中存在一个 XAML 文件，此文件的名称 MainPage.xaml，如图 1.7 所示。

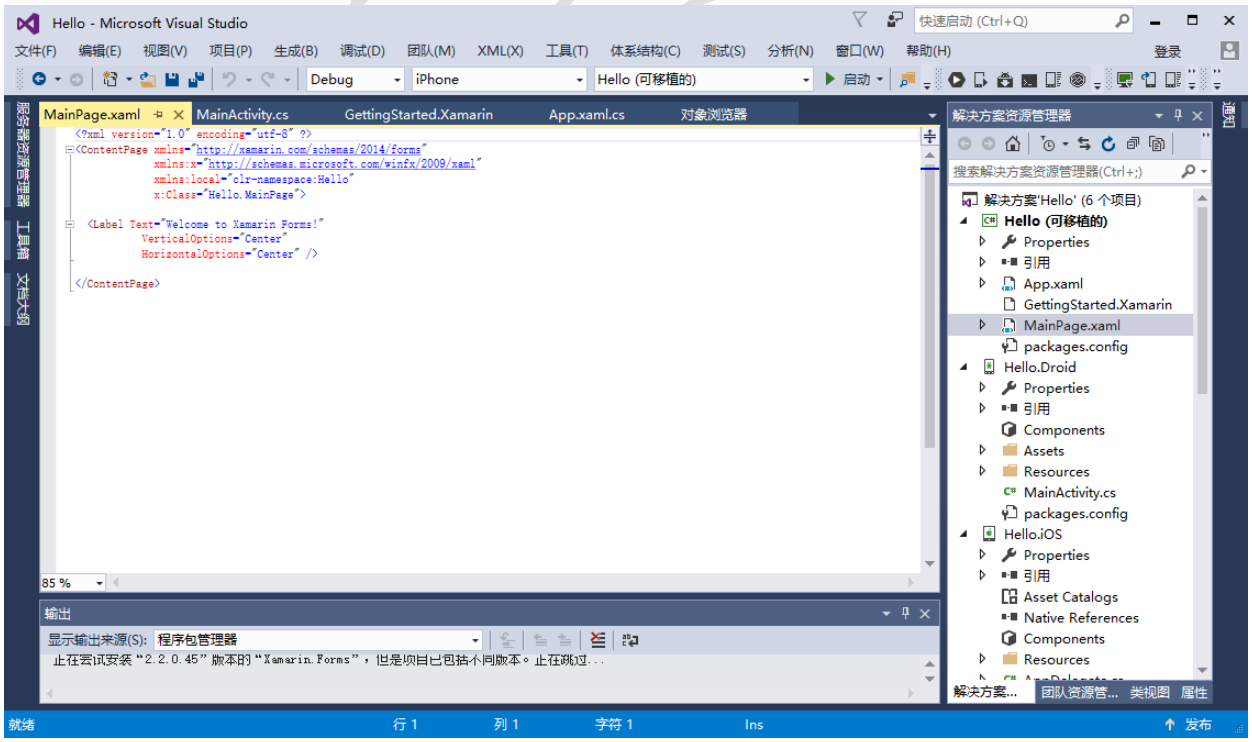

图 1.7　MainPage.xaml 文件

注意：如果开发者想要在此项目中再创建一个 XAML 文件，需要按照 1.2.1 小节中的步骤 5 到步骤 7 的方式操作。

1.2.2　使用 Xamarin Studio 创建 XAML

Xamarin Studio 和 Visual Studio 创建 XAML 文件一样，也分为两种创建方式。第一种是创建项目后再创建 XAML 文件，第二种是创建项目时创建 XAML 文件。以下对这两种方式进行详细介绍。

1.创建项目后再创建 XAML 文件

以下我们将以创建项目 Hello 为例，为开发者讲解创建项目后创建 XAML 文件的具体操作步骤：

（1）在计算机上找到 Xamarin Studio，将其打开，如图 1.8 所示。

（2）选择“New Solution…”按钮，弹出 Choose a template for your new project 界面，如图 1.9 所示。

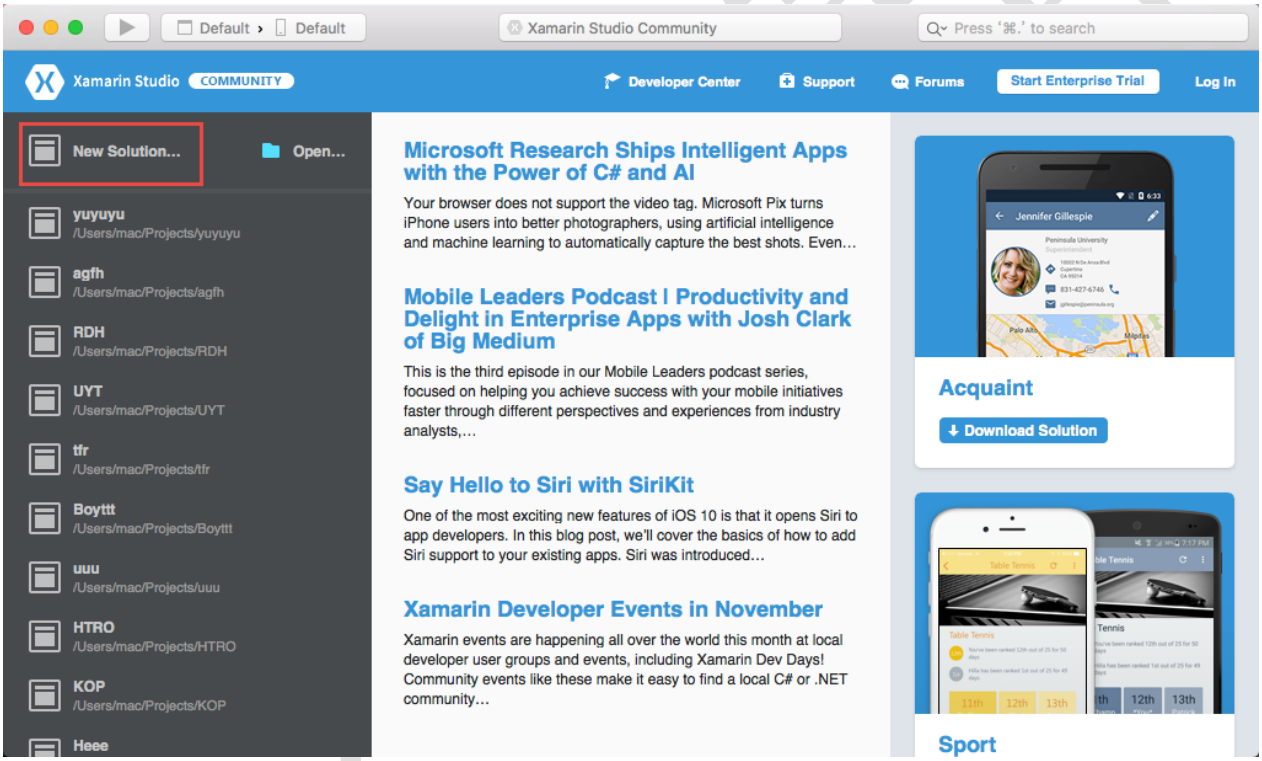

图 1.8　打开 Xamarin Studio

（3）选择 Multiplatform 下的 App 中的 Forms App 模板，单击 Next 按钮，进入到 Configure your Forms App 界面，如图 1.10 所示。

（4）在 App Name 对应的文本框中输入应用程序的名称，这里我们输入的是 Hello，单击 Next 按钮进行 Configure your new project 界面中，如图 1.11 所示。

（5）单击 Create 按钮，一个名为 Hello 的项目就创建好了。选择菜单中的 File|New File…命令，弹出 New File 界面，如图 1.12 所示。

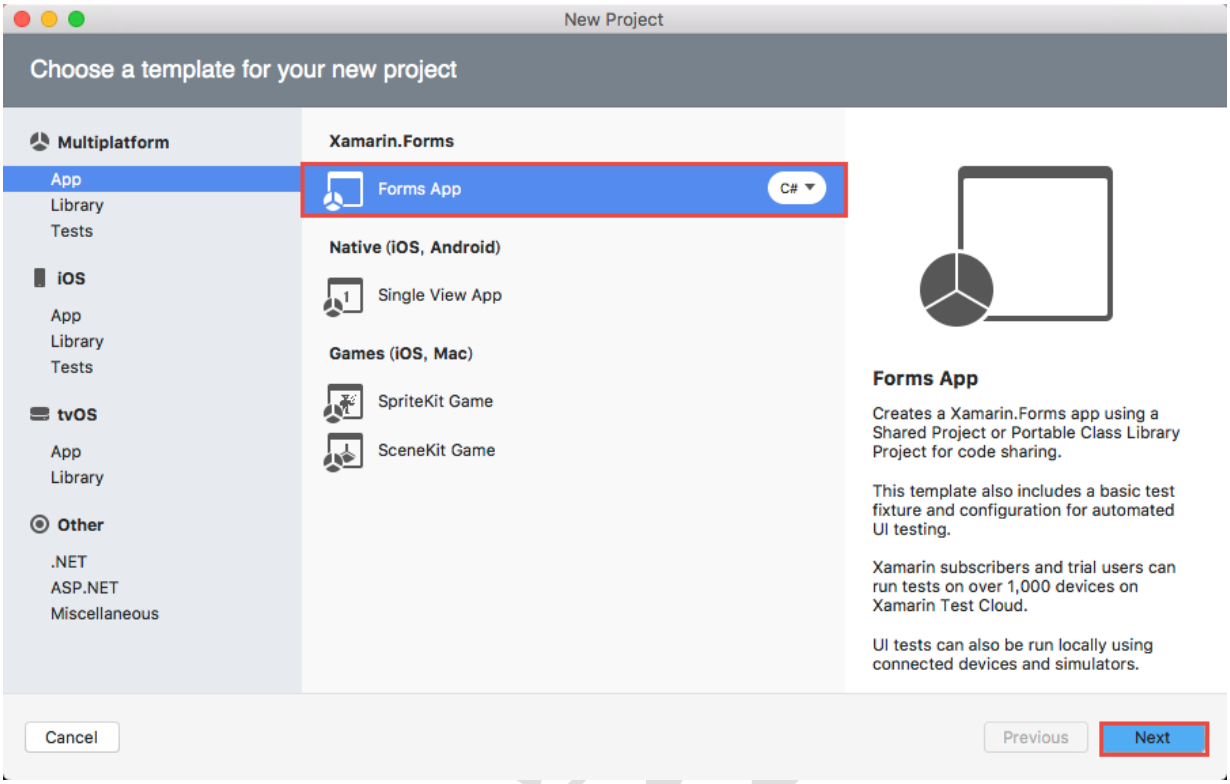

图 1.9　Choose a template for your new project 界面

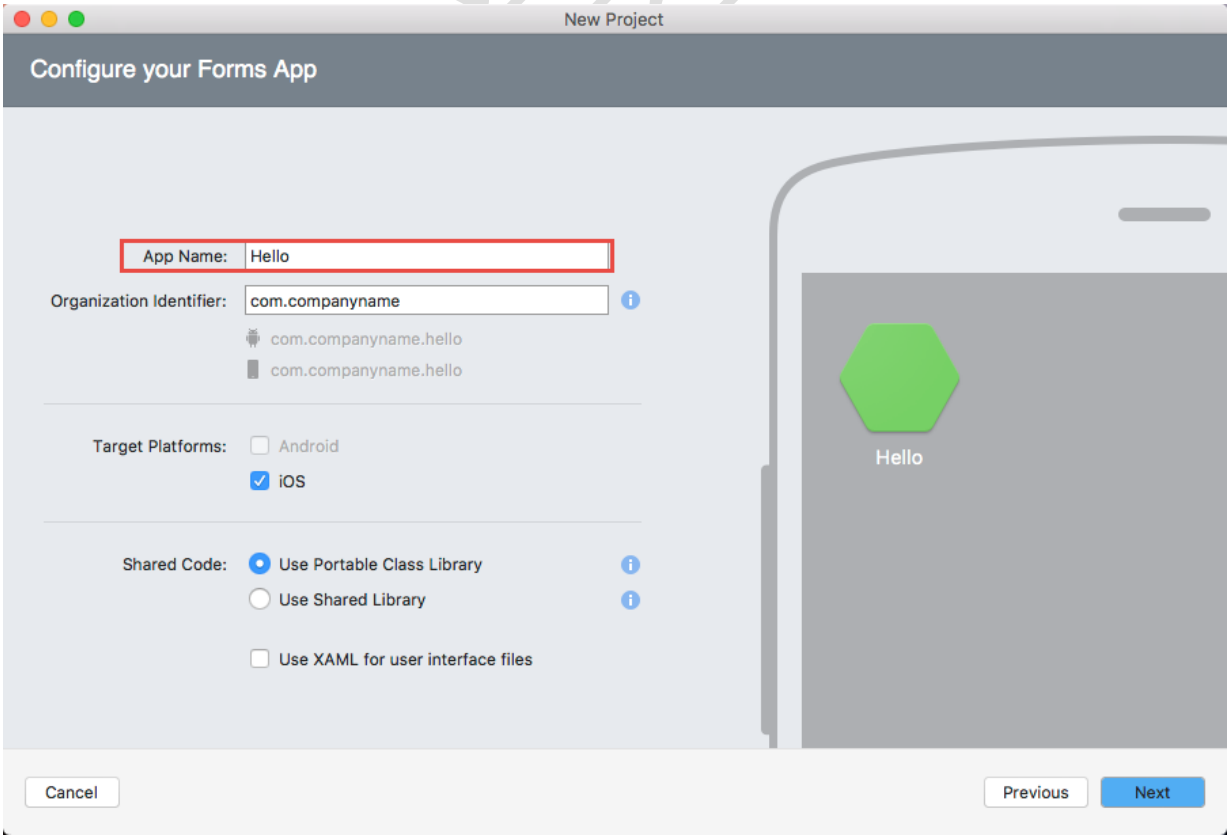

图 1.10　Configure your Forms App 界面

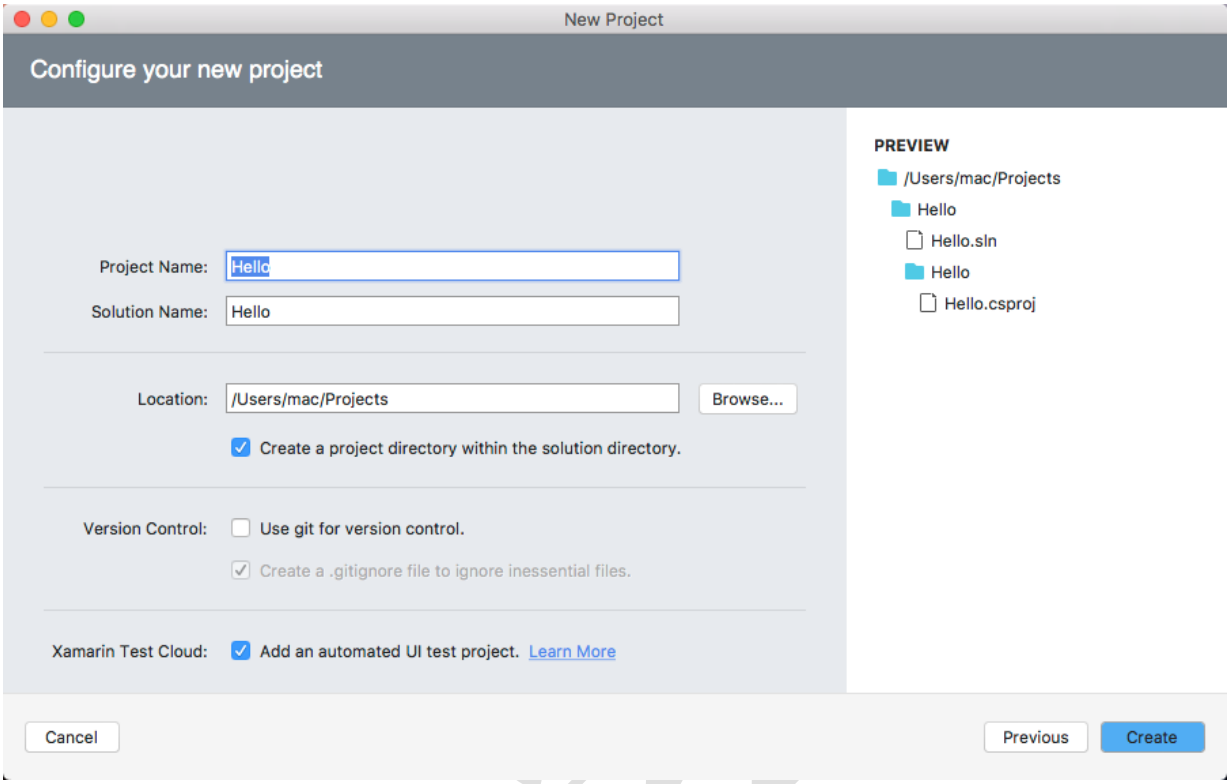

图 1.11　Configure your new project 界面

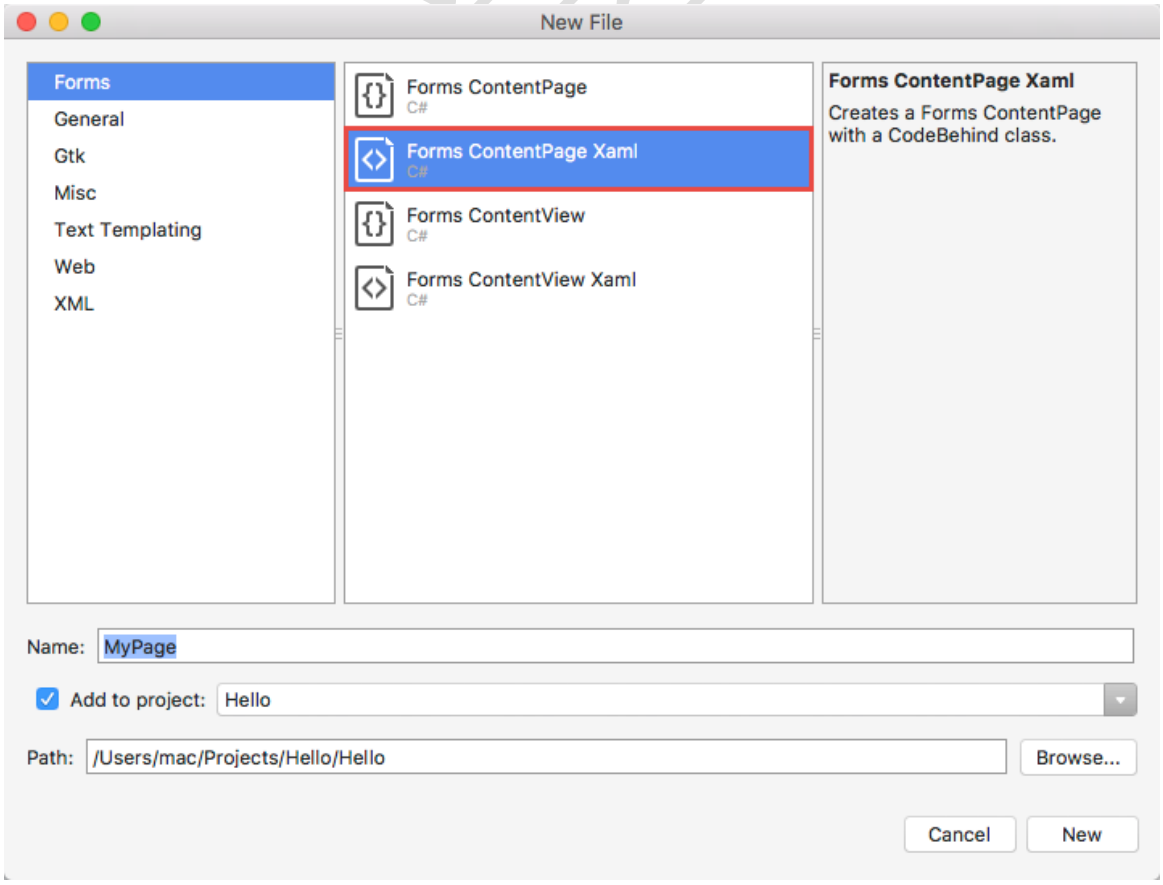

图 1.12　New File 界面

（6）选择 Forms 中的 Forms ContentPage Xaml 选项，在 Name 文本框中输入名称，这里我们使用的是默认的名称。单击 New 按钮，此时一个名为 MyPage 的 Xaml 文件就创建好了，如图 1.13 所示。

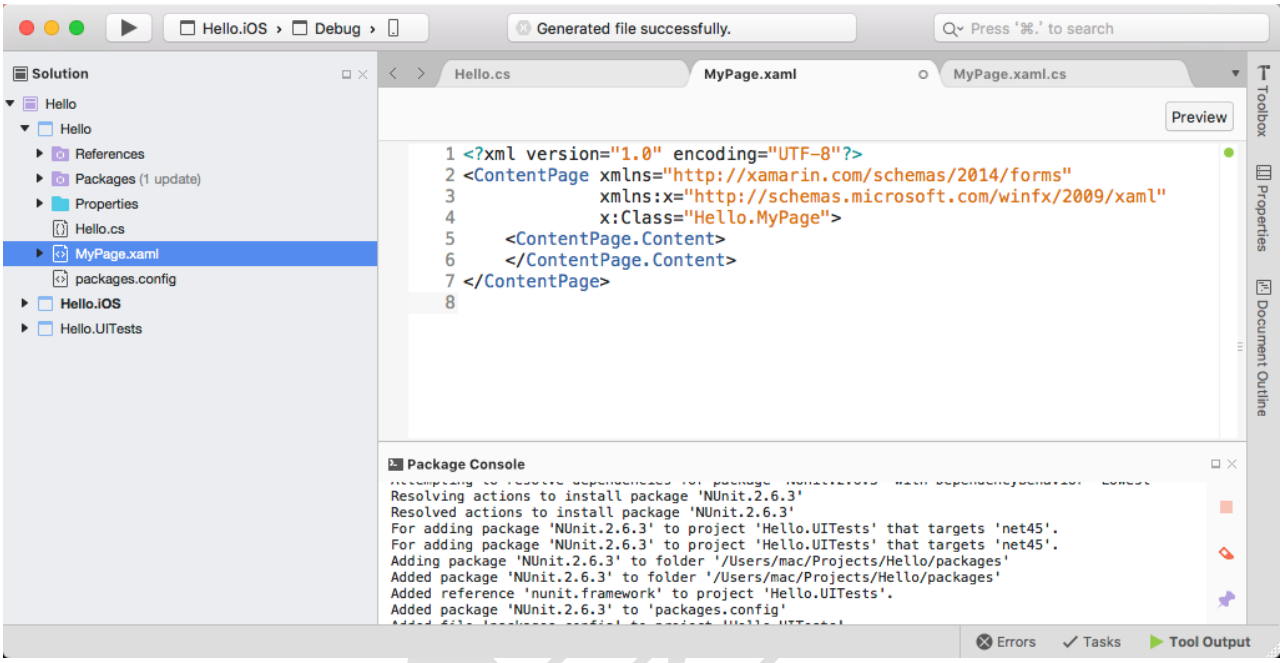

图 1.13　MyPage 文件

2.创建项目时创建 XAML 文件

在 Xamarin Studio 中在创建项目的同时创建 XAML 也是最常用的方式，并且是最为简单的方式。以下我们将以 Hello 为例，为开发者讲解在创建项目的同时创建 XAML 的具体操作步骤：

（1）在计算机上找到 Xamarin Studio，将其打开。

（2）选择“New Solution…”按钮，弹出 Choose a template for your new project 界面。

（3)选择 Multiplatform 下的 App 中的 Forms App 模板，单击 Next 按钮，进入到 Configure your Forms App 界面。

（4）在 App Name 对应的文本框中输入应用程序的名称，这里我们输入的是 Hello，将最下方的 Use XAML for user interface files 复选框选中，如图 1.14 所示。

（5）单击 Next 按钮,进入 Configure your new project 界面中。

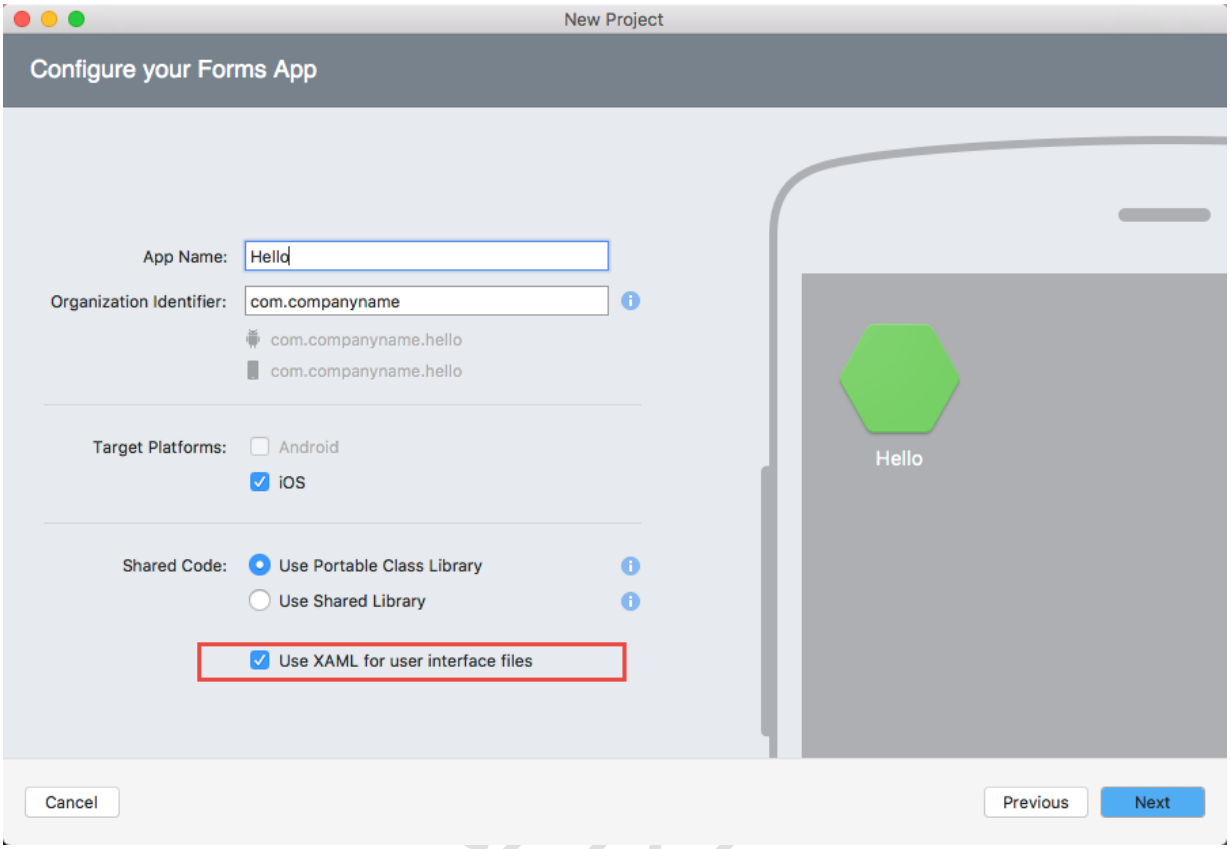

图 1.14　Configure your Forms App 界面

（6）单击 Create 按钮，一个名为 Hello 的项目就创建好了。我们可以看到在创建的项目中存在一个 XAML 文件，此文件的名称 HelloPage.xaml，如图 1.15 所示。

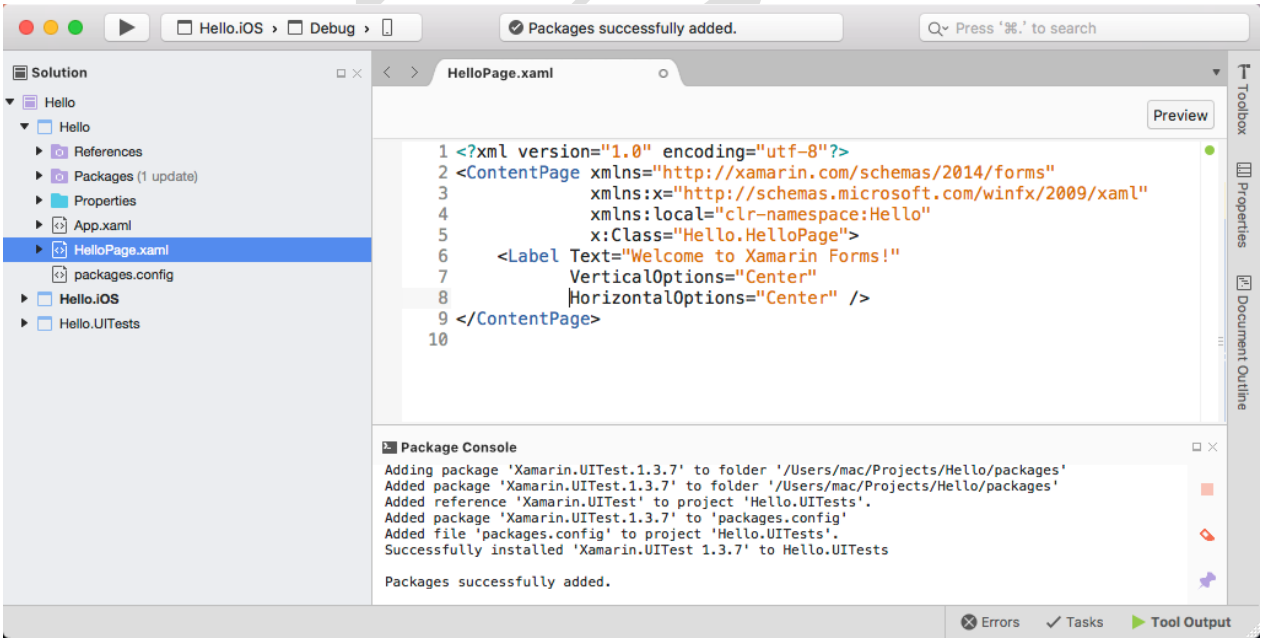

图 1.15　HelloPage.xaml

注意：在后面内容中，我们主要使用 Visual Studio 为环境讲解 XAML 代码的编写。

1.3 XAML 文件结构

在上文中，我们创建 XAML 文件后，会看到类似图 1.16 所示的结构

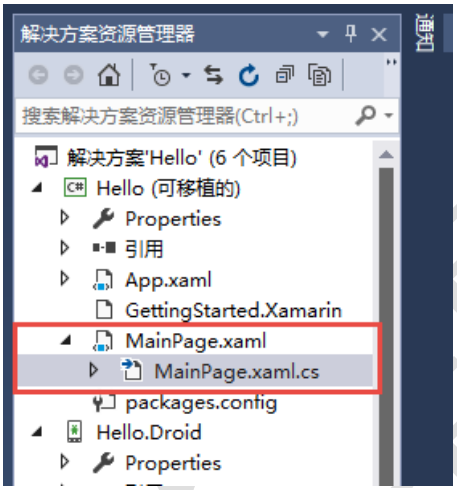

图 1.16 结构

其中，.xaml 文件和.xaml.cs 文件就是 XAML 文件的结构。以下就是对这两个文件的介绍。

- .xaml 文件中包含的就是 XAML 代码，实际上就是 XML 语法。官方的说法：它是一个声明对象的语言，为我们创建对象提供便捷的一种方式。与 HTML 类似，特点是用来描述用户接口（UI）内容。
- 通常我们把与.xaml 文件关联的.xaml.cs 文件叫作代码隐藏文件。如果开发者引用.xaml 中的任何一个事件（如 Button 的 Click 事件），，将在这个文件中编写对应的事件处理代码。

1.4 解析 XAML

在上文中我们创建 XAML 文件后，会看到类似以下 3 种代码。

（1）第一种是使用 Visual Studio 创建项目后创建 XAML 文件显示的 XAML 代码：

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
             x:Class="Hello.XAMLPage">
  <Label Text="{Binding MainText}" VerticalOptions="Center" HorizontalOptions="Center" />
</ContentPage>
```

（2）第二种是使用 Xamarin Studio 创建项目后创建 XAML 文件显示的 XAML 代码：

```
<?xml version="1.0" encoding="UTF-8"?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
              xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
              x:Class="Hello.MyPage">
    <ContentPage.Content>
    </ContentPage.Content>
</ContentPage>
```

（3）第三种是在创建项目的同时创建 XAML 文件显示的代码：

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
             xmlns:local="clr-namespace:Hello"
             x:Class="Hello.MainPage">
  <Label Text="Welcome to Xamarin Forms!"
         VerticalOptions="Center"
         HorizontalOptions="Center" />
</ContentPage>
```

这 3 种文件代码的公共部分为如下：

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
             x:Class="Hello.MainPage">
</ContentPage>
```

以下就是对 XAML 公共默认的部分进行说明。

- 第一行是对 XML 简单的说明，其中包含了 XML 版本号以及编码格式。
- 第二行代码至最后一行代码的功能是构建界面所需要的内容。其中，第二行和第三行使用两个 XML 命名空间（xmlns）声明引用了 URL。开发者不需要知道这些 URL 指向的具体内容。他们只是 Xamarin 和 Microsoft 拥有的 URL，它们基本上充当版本标识符。第一个 XML 命名空间声明意味着在 XAML 文件中定义了没有前缀的标签，它引用 Xamarin.Forms 中的类，如 ContentPage。第二个命名空间声明定义了 x 的前缀，它用于 XAML 本身固有的几个元素和属性，（理论上）由 XAML 的所有实现支持。

注意：这些元素和属性根据嵌入在 URL 中的年份略有不同。Xamarin.Forms 支持 2009 XAML 规范，但不是所有的。

- 第四行代码。在声明 x 前缀之后，该前缀立即用于名为 Class 的属性，这是因为使用这个 x 前缀在 XAML 文件中非常普遍。例如，Class 简称为 x:Class。x:Class 指定.NET 类名称。

注意：x:Class 属性只能出现在 XAML 文件的根元素中，以定义派生的 C#类。

对于 x:Class 指定类的定义，开发者可以在.xaml.cs 文件中看到，代码类似于以下代码：

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using Xamarin.Forms;
namespace Hello
{
    public partial class MainPage : ContentPage
    {
        public MainPage()
        {
            InitializeComponent();
        }
    }
}
```

注意：x:Class 的值所指示的类型在声明的时候必须使用 partial 关键字。这样由 XAML 解析成的类和.xaml.cs 文件里定义的部分就合二为一了。正是由于这种 partial 机制，我们可以把类的逻辑代码留

在.xaml.cs 文件中，并用 C#语言来实现，而把那些于声明及布局 UI 元素相关的代码分离出来，实现 UI 以逻辑分离。

1.5 对象元素的声明方式

XAML 的对象元素的声明有两种形式，分别为包含属性的特性语法形式以及对象元素语法形式。在 1.4 小节中，我们看到了其中一种对 XAML 对象元素的声明方式，以下将讲解两种对对象元素的声明方式。

1.5.1 包含属性的特性语法形式

在 XAML 中如果是单个元素对象，可以使用包含属性的特性语法形式对这个元素对象进行声明，其语法形式如下：

```
<object    ……      />
```

其中，object 是 Xamarin.Forms 中提到的对象，对象元素的声明是以“<”开始，并以“/>”结束。在 XAML 中使用这种方式可以将 Xamarin.Forms 中的对象实例化化为 XML 格式的对象元素。

注意：在 XAML 中以“<”开始，并以“/>”结束的内容称为标签

【示例 1-1：ObjectElementDeclarationOne】以下将使用该语法形式声明一个 Label 元素对象。代码如下：

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
             xmlns:local="clr-namespace:ObjectElementDeclarationOne"
             x:Class="ObjectElementDeclarationOne.MainPage">
  <Label Text="Welcome to Xamarin Forms!"
         FontAttributes="Bold"
         FontSize="Large"
         VerticalOptions="Center"
         HorizontalOptions="Center" />
</ContentPage>
```

注意：在对象元素中的这些属性可以放在一行进行写，也可以分行进行写。

1.5.2 对象元素语法形式

在 XAML 中第二种声明对象元素的方式是使用对象元素语法形式，其语法形式如下：

```
<object>
  ……
</object>
```

其中，object 是 Xamarin.Forms 中提到的对象，对象元素的声明使用开始标签和结束标签将对象实例化化为 XML 格式的元素。在 XAML 中，如果在对象元素中还包含其它对象元素时，最好使用这种方式。

注意：在 XAML 中以“<”开始以“>”结束的代码称为标签。

【示例 1-2：ObjectElementDeclaration】以下将使用该语法形式声明一个 StackLayout 对象元素，在

StackLayout 对象元素中包含了 3 个标签对象元素，对于标签对象元素的声明，我们使用包含属性的特性语法形式进行声明。代码如下：

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
             xmlns:local="clr-namespace:ObjectElementDeclaration"
             x:Class="ObjectElementDeclaration.MainPage">
  <StackLayout>
    <Label Text="Welcome to Xamarin Forms!"
           FontAttributes="Bold"
           FontSize="Large"
           VerticalOptions="CenterAndExpand"
           HorizontalOptions="Center" />
    <Label Text="Welcome to Xamarin Forms!"
           FontAttributes="Bold"
           FontSize="Large"
           VerticalOptions="CenterAndExpand"
           HorizontalOptions="Center" />
    <Label Text="Welcome to Xamarin Forms!"
           FontAttributes="Bold"
           FontSize="Large"
           VerticalOptions="CenterAndExpand"
           HorizontalOptions="Center" />
  </StackLayout>
</ContentPage>
```

1.6　显示到界面

如果通过 XAML 将 UI 设计好以后，就可以将 XAML 中的内容显示给用户了，也就是显示到界面上。由于创建 XAML 文件方式的不同，所以将 XAML 中的内容显示到界面上的方式也就不一样了。以下针对 XAML 文件创建方式的不同，为开发者讲解如果将 XAML 中的内容显示到界面上。

注意：在 1.2 节中我们使用 Visual Studio 创建 XAML 文件和使用 Xamarin Studio 中创建 XAML 的方式都分为两种，分别为创建项目后创建 XAML 文件和在创建项目的同时创建 XAML 文件。创建项目后创建 XAML 文件和在创建项目的同时创建 XAML 文件这两种创建 XAML 的方式对应就会存在两种显示到界面的方法。

1.6.1　创建项目后再创建 XAML 文件

如果开发者是在创建项目后创建 XAML 文件，此时需要将 XAML 文件中的内容显示到界面上，需要打开 App.cs 文件（App.cs 文件是 Visual Studio 中的文件，如果是 Xamarin Studio 需要打开**.cs，这里的**指定的项目名称），将此文件中的代码进行修改，修改后的代码如下：

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

```
namespace Hello
{
    public class App : Application
    {
        public App()
        {
            // The root page of your application
            MainPage = new XAMLPage();                    //XAMLPage 是创建的 XAML 文件
        }
        ……
    }
}
```

此时运行程序，XAML 文件中的内容就显示到界面上了，从而实现了对用户的显示。

1.6.2 创建项目时创建 XAML 文件

如果开发者是在创建项目的同时创建的 XAML 文件，要将内容显示到界面上我们不需要做任何事情。此时运行程序，会看到 XAML 文件中的内容显示到界面中了。

1.7 XAML 预览

每次通过编译运行的方式查看 XAML 文件效果，需要花费大量的时间。如果开发者使用 XAML 对 UI 进行布局和设计，可以通过预览的方式对 XAML 进行查看。本节将讲解在 Visual Studio 和 Xamarin Studio 中如何实现预览。

1.7.1 Visual Studio 中实现预览

以下我们将以 Hello 项目为例，在 Visual Studio 中实现预览。在实现预览时，需要选择“视图(V)”|“其它窗口(E)”|Xamarin.Forms Previewer 命令，打开 Forms Previewer 窗口，如图 1.17 所示。

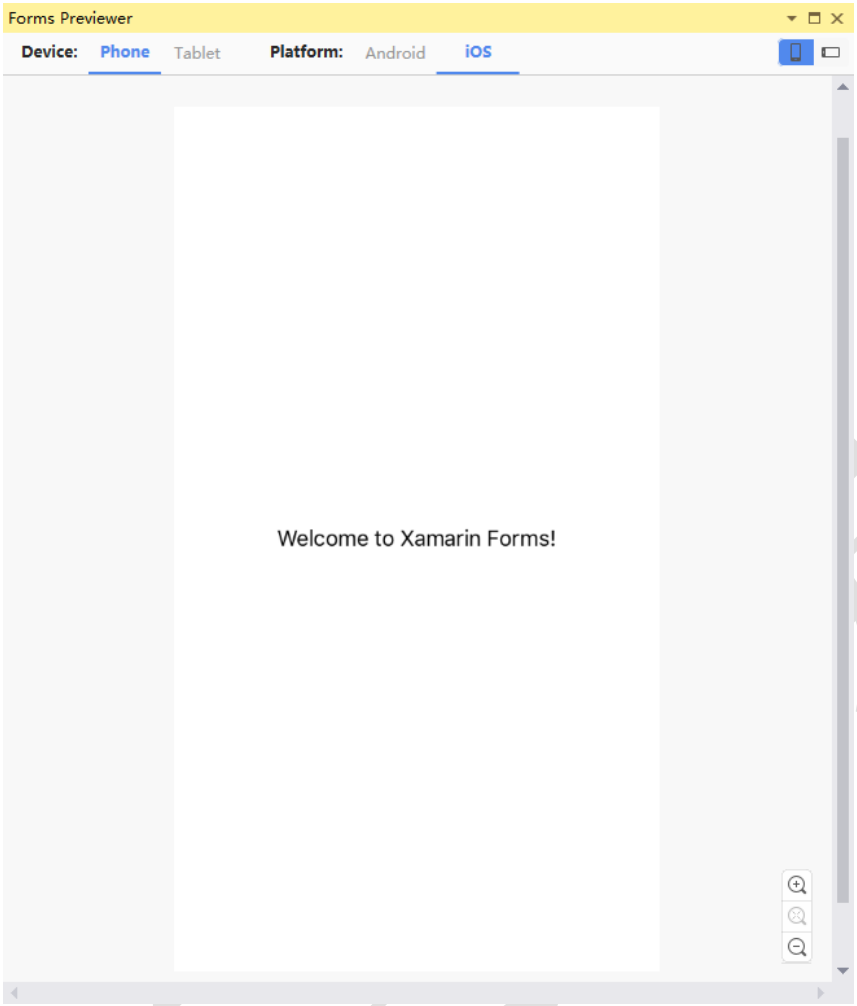

图 1.17　Forms Previewer 窗口

注意：在第一次加载 XAML 文件时，开发者需要耐心等待一会，此时 Forms Previewer 窗口会对 iOS 或者 Android 的 SDKs 进行读取，如图 1.18 所示。

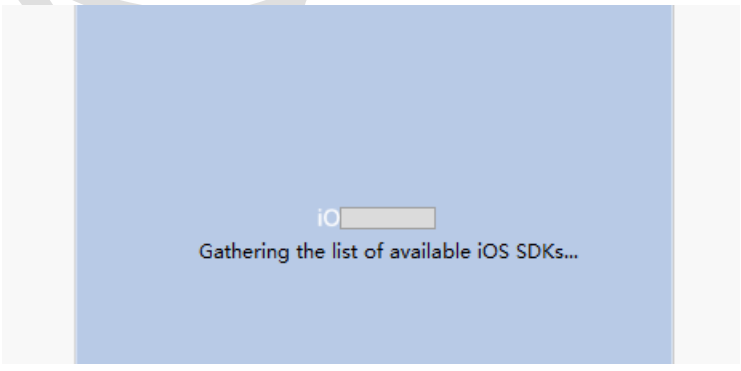

图 1.18　Forms Previewer 窗口

此时我们会看到在 Forms Previewer 窗口中有 6 个选项，这 6 个选项的功能介绍如下：

- Phone：在手机大小的屏幕中呈现。
- Tablet：在平板电脑屏幕中呈现（请注意，窗格右下角有缩放控件）。
- Android：显示 Android 版本的屏幕。

- iOS：显示 iOS 版本的屏幕。
- Portait (icon)：使用纵向方向进行预览。
- Landscape (icon)：使用横向方向进行预览。

开发者可以根据自己代码的需求对这些选项进行选择。图 1.18 选择的是 Phone、iOS 以及 Portait (icon) 模式。

注意：在打开的 Forms Previewer 窗口中很有可能不会对 XAML 中的内容进行显示，如果遇到不显示的问题，开发者可以有两种解决办法：

- 在尝试预览 XAML 文件之前，应该构建（编译）项目。
- 尝试关闭并重新打开 XAML 文件。

1.7.2 Xamarin Studio 中实现预览

如果开发者要在 Xamarin Studio 中实现预览，可以单击.xaml 文件中的 Preview 按钮，此时就会在 XAML 代码的右侧出现预览窗口，如图 1.19 所示。

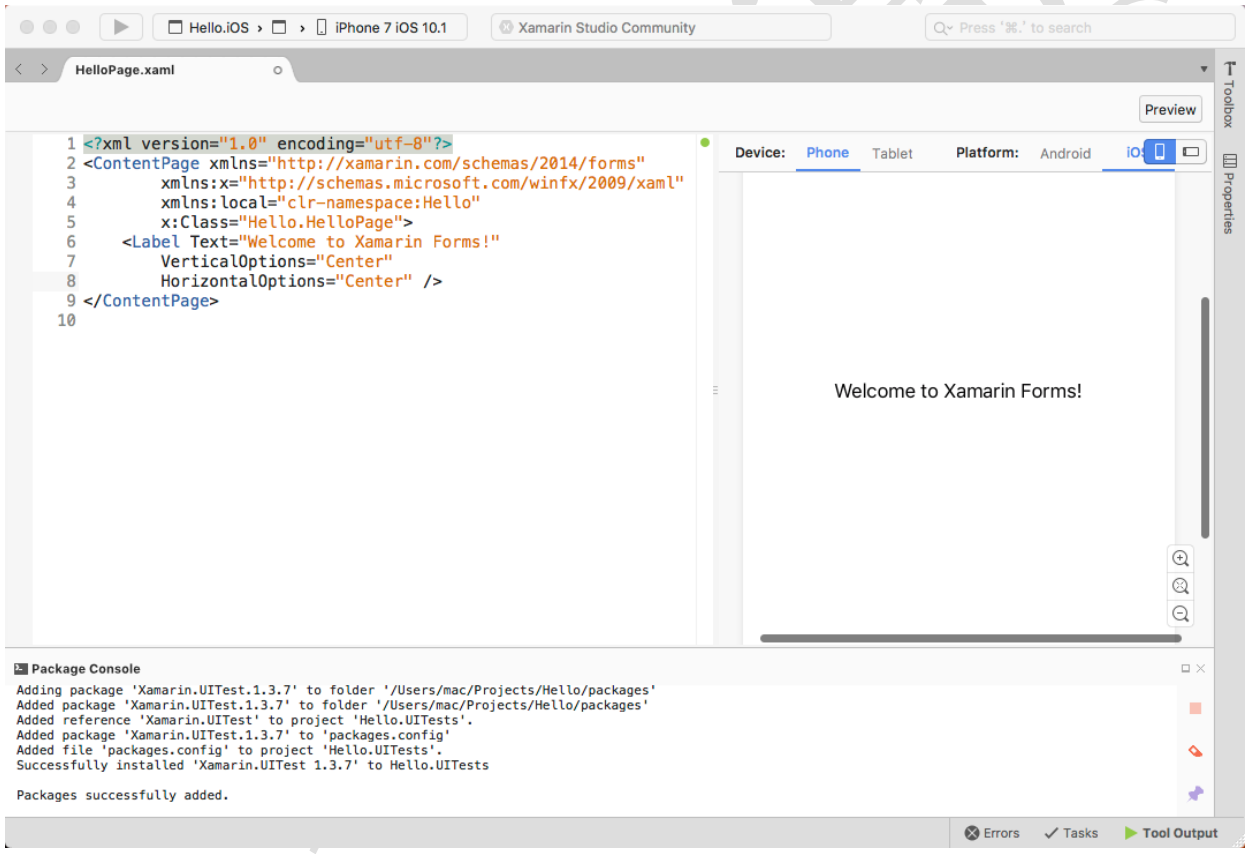

图 1.19　预览窗口