# How to run Hello World using Universal Windows Platform

## Introduction

The sample provided is a Hello World application that targets the Universal Windows Platform (UWP) on Windows 10, written using Extensible Application Markup Language (XAML) with C#. UWP allows you to build applications that run on any Windows 10 device with a single project in Microsoft Visual Studio.

## Building the Sample

This sample is for use with Microsoft Visual Studio 2015 and Windows 10. It won't work correctly with earlier versions. Before you start, make sure you are [set up](https://msdn.microsoft.com/en-us/windows/uwp/get-started/get-set-up) with Windows 10 and Visual Studio 2015.

1. Start Microsoft Visual Studio 2015 and select 2015 **File > Open > Project/Solution.**
2. Go to the directory to which the sample was unzipped. Then go to the subdirectory named for the sample and double-click the Visual Studio 2015 Solution (.sln) file.
3. Build the sample. **Build > Build Solution.**

## Running the Sample

### Starting the app on a Desktop device

In the target device menu () on the **Standard** toolbar, make sure that **Local Machine** is selected. (It's the default selection.)

Click the **Start Debugging** button () on the toolbar. OR

From the **Debug** menu, click **Start Debugging**. OR

Press F5.

The app opens in a window, and a default splash screen appears first. The splash screen is defined by an image (SplashScreen.png) and a background color (specified in your app's manifest file).

The splash screen disappears, and then your app appears. It looks like this.

Press the Windows key to open the **Start** menu, then show all apps. Notice that deploying the app locally adds its tile to the **Start** menu. To run the app again (not in debugging mode), tap or click its tile in the **Start** menu.

**To stop debugging**

Click the **Stop Debugging** button () in the toolbar. OR

From the **Debug** menu, click **Stop debugging**. OR

Close the app window.

### Start the app on a mobile device emulator

In addition to the options to debug on a desktop device, Visual Studio provides options for deploying and debugging your app on a physical mobile device connected to the computer, or on a mobile device emulator. You can choose among emulators for devices with different memory and display configurations.

* **Device**
* **Emulator WVGA 4 inch 512MB**
* **Emulator WVGA 4 inch 1GB**
* etc... (Various emulators in other configurations)

It's a good idea to test your app on a device with a small screen and limited memory, so use the **Emulator 10.0.10240.0 WVGA 4 inch 512MB** option. **To start debugging on a mobile device emulator**

In the target device menu () on the **Standard** toolbar, pick **Emulator 10.0.10240.0 WVGA 4 inch 512MB**.

Click the **Start Debugging** button () in the toolbar. OR

From the **Debug** menu, click **Start Debugging**. OR

Press F5.

Visual Studio starts the selected emulator and then deploys and starts your app. On the mobile device emulator, the app looks like this.

## Using the Code

### MainPage.xaml

In MainPage.xaml you define the UI for your app. You can add elements directly using XAML markup, or you can use the design tools provided by Visual Studio. MainPage.xaml.cs is the code-behind page for MainPage.xaml. It's where you add your app logic and event handlers.

Together these two files define a new class called MainPage, which inherits from **Page**, in the HelloWorld namespace.

MainPage.xaml

|  |
| --- |
| -Code block start-  --C# code snippet start--  <Page  x:Class="CSHelloWorldUniversalApp.MainPage"  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"  xmlns:local="using:CSHelloWorldUniversalApp"  xmlns:d="http://schemas.microsoft.com/expression/blend/2008"  xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"  mc:Ignorable="d">  <Grid Background="{ThemeResource ApplicationPageBackgroundThemeBrush}">    <VisualStateManager.VisualStateGroups>  <VisualStateGroup>  <VisualState x:Name="wideState">  <VisualState.StateTriggers>  <AdaptiveTrigger MinWindowWidth="641" />  </VisualState.StateTriggers>  </VisualState>  <VisualState x:Name="narrowState">  <VisualState.StateTriggers>  <AdaptiveTrigger MinWindowWidth="0" />  </VisualState.StateTriggers>  <VisualState.Setters>  <Setter Target="inputPanel.Orientation" Value="Vertical"/>  <Setter Target="inputButton.Margin" Value="0,4,0,0"/>  </VisualState.Setters>  </VisualState>  </VisualStateGroup>  </VisualStateManager.VisualStateGroups>    <StackPanel x:Name="contentPanel" Margin="8,32,0,0">  <TextBlock Text="Hello, world!" Margin="0,0,0,40"/>  <TextBlock Text="What's your name?"/>  <StackPanel x:Name="inputPanel" Orientation="Horizontal" Margin="0,20,0,20">  <TextBox x:Name="nameInput" Width="280" HorizontalAlignment="Left"/>  <Button x:Name="inputButton" Content="Say 'Hello'" Click="inputButton\_Click"/>  </StackPanel>  <TextBlock x:Name="greetingOutput"/>  </StackPanel>    </Grid>  </Page>  --C# code snippet end--  Insert other Programming Language Code Snippet here  -Code block end- |

MainPage.xaml.cs

|  |
| --- |
| -Code block start-  --C# code snippet start--  public sealed partial class MainPage : Page  {  public MainPage()  {  this.InitializeComponent();  }  private void inputButton\_Click(object sender, RoutedEventArgs e)  {  greetingOutput.Text = "Hello, " + nameInput.Text + "!";  }  }  --C# code snippet end--  Insert other Programming Language Code Snippet here  -Code block end- |

## More Information

What's new for developers in Windows 10: <https://developer.microsoft.com/en-us/windows/getstarted/whats-new-windows-10>

What's a Universal Windows Platform (UWP) app: <https://msdn.microsoft.com/en-us/windows/uwp/get-started/whats-a-uwp>

Create a "Hello, world" app: <https://msdn.microsoft.com/en-us/windows/uwp/get-started/create-a-hello-world-app-xaml-universal>