HoloLens Setup – Spring 2020

In this document, you will learn how to setup your Unity Environment for HoloLens development. You will also learn how to Construct A Basic Hello World Application through building a 3D Object and Material as well as a 3D Text Object. You will explore the physics capabilities of the environment and add a Rigid Body and Box Collider to objects. Then you will deploy the product of these efforts to the HoloLens 1 through Microsoft Visual Studio 2017.

Find the IP Address and WiFi network of your HoloLens

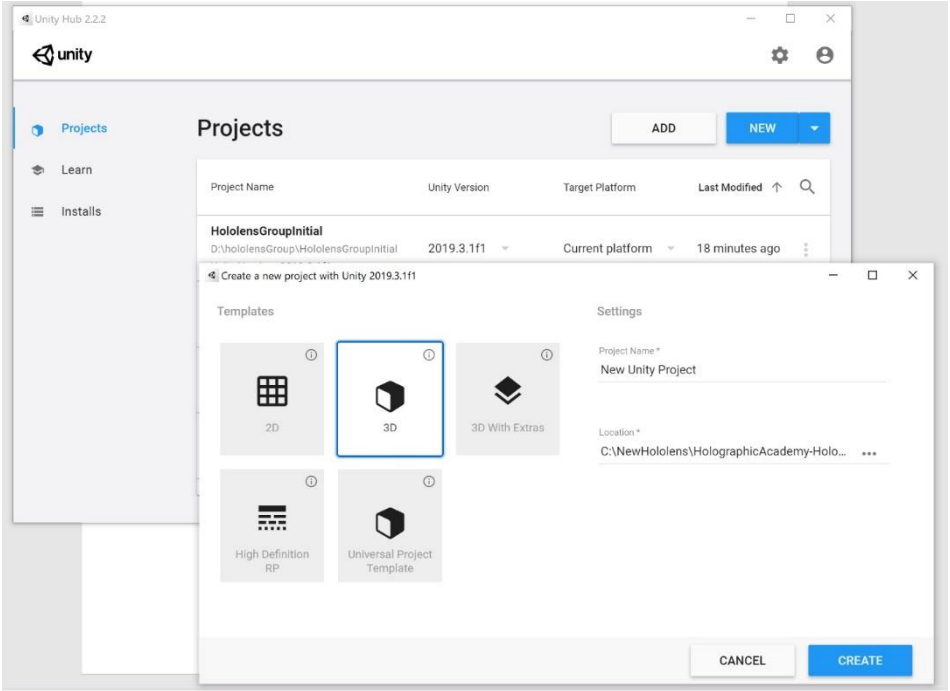
1. Place the HoloLens on your head and turn it on.
2. Wait for the desktop and then locate the Settings icon.
3. Click on Settings>Network & Internet.
4. Select the Wifi network that is connected (make sure it is the same network that your computer is connected to) and select Advanced Options.
5. Look for the IPv4 address. At WPI it should start with a 10.XXX.XX.XX .
6. Write it down. You will need this later.
7. For now, you can shut off the headset.

Setup the Unity Environment for HoloLens Development.

1) Install the latest version of Unity – Unity Hub (your computer needs min. 1080 graphics card)
    a. I am running Unity 2019.3.1f1
2) I am running Visual Studio 2017 (v141)
3) I am using the HoloLens 1 platform

Navigate to projects tab in the Unity Hub

Figure 1 Start a new 3D project in Unity.



Select 3D and give your new project a name and a location. Then click create. (See Figure 1).

Settings that need to be changed:

1) Go to Edit>Project Settings>XR Plugin Management and click on install XR Plugin Management system.
    a. Once that is installed select install Windows XR Plugin
    b. Make sure that the windows store logo is selected.
    c. Click the plus sign above the Plugin Providers list and select Windows MR Loader
    d. Select Input Helpers and click on Install Legacy Input Helpers Package
    e. Go to XR Tracking and Click on Add Tracked Pose Driver To All Cameras in scene.
2) Next select Player from the left column on the same popup
    a. Ignore any warnings about deprecated XR stuff

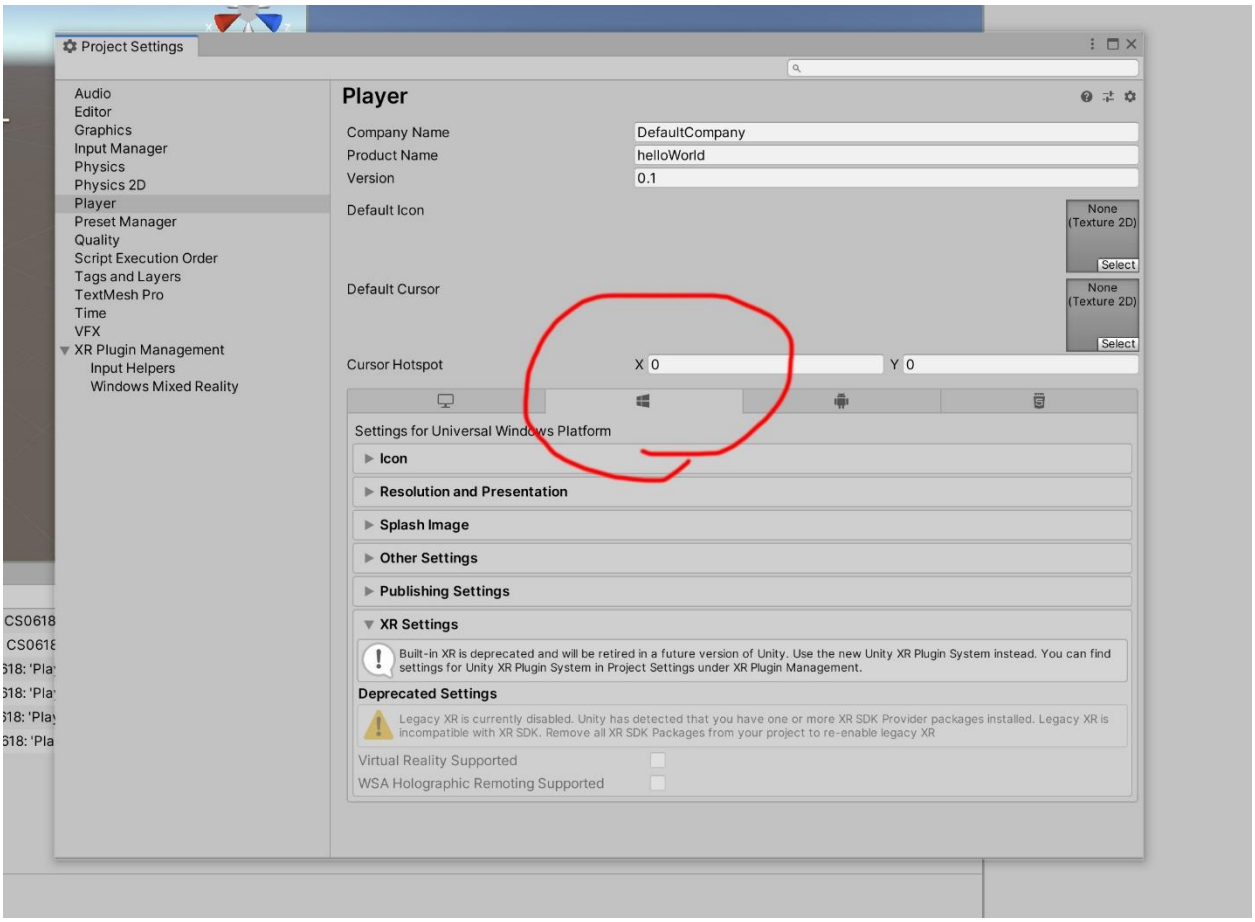


Figure 2 Change the tab to the windows store icon.

3) Change the tab to the windows store icon. (See Figure 2).
    a. Under **Other Settings>Configuration Scripting Backend** should be grayed out and it should say **IL2CPP**. Also the **API Compatibility Level** should say **.NET 4.x**
    b. Leave everything else on that tab alone.
4) On the Quality tab, make sure that the windows store setting is set to Very Low quality.

5) Just to show you what is possible, click on **Player>Publisher Settings**.
    a. This is the tab on which you could select to add Bluetooth, microphone, proximity sensors, and gaze input in a future project. For now, leave this tab alone as well.
    b. Under Supported Device Families, select Holographic
6) For **specifically** this project, we are going to mess with the gravity in our system. For this reason, click on the Physics menu item on the left.
    a. About halfway down the Physics page, you will see that gravity is set to -9.81 in the y-direction. This typical gravity for earth. For this project **set it to -6.0**. The negative sign indicates that things will fall down in the negative y direction.
7) Close the window.
8) At the bottom of the interface, make sure the project tab is selected. (See Figure 3). Check to see if there is a Scene folder inside the Assets folder. If there is not, right click on any free space on the right and select create>folder. Name it Scenes.

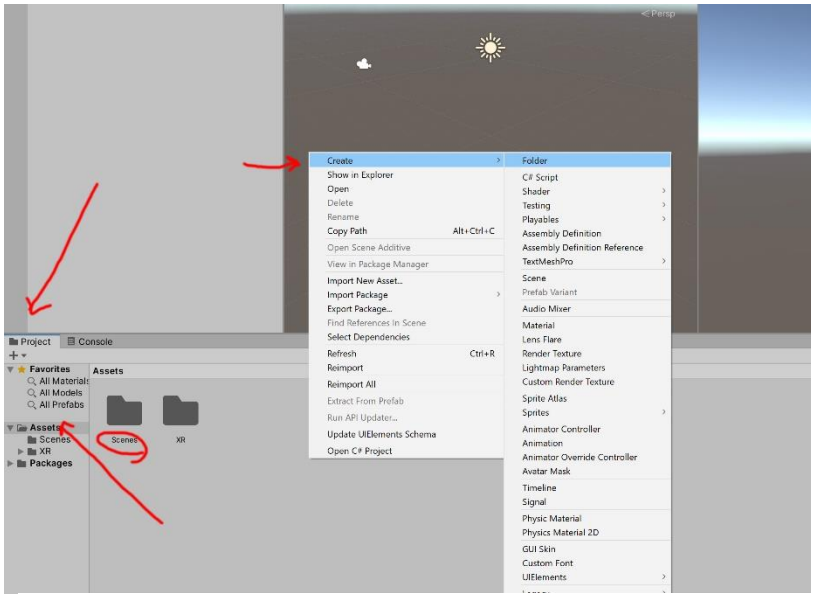


Figure 3 Create a Scene folder.

9) Go to the File tab and select save as and save a new scene inside the Scenes folder. Call it main.

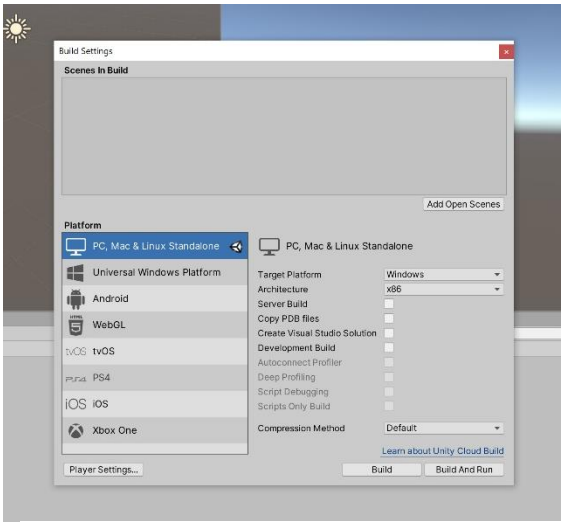


Figure 4 Build Settings

10) Select the File tab again and select Build Settings. (See Figure 4).
11) Make the following changes. (See Figure 5)
    a. Add Open Scenes – they will show up in the Scenes in Build list above.
    b. Change the Platform from PC, Mac to Universal Windows Platform.
    c. Change the Target Device to HoloLens
    d. Change Architecture to x86 (HoloLens 1 is x86, I believe HoloLens 2 is ARM).
    e. Change build type to D3D if it is not already selected.
    f. Keep Build and Run on Local Machine for now. This may change in later documents.
    g. And finally click Switch Platform. Let it think for a bit. Close the popup when it is done.

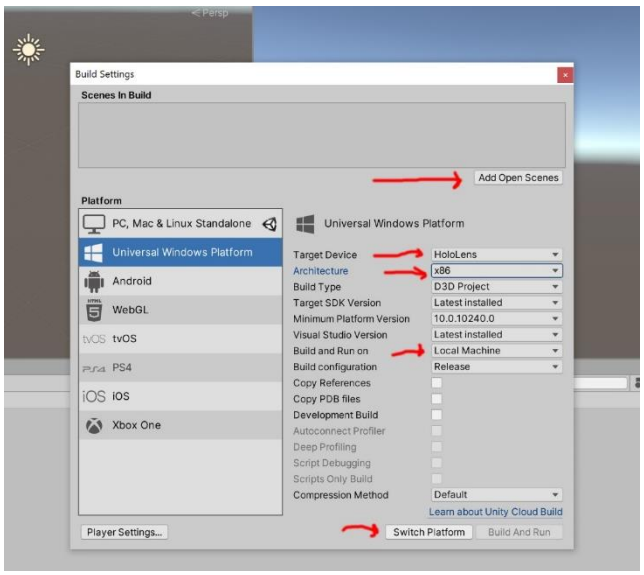


Figure 5 Build Settings for the HoloLens 1

12) Next some changes to the environment.
    a. Select the main camera under the Hierarchy tab. Go to the Inspector tab that displays all of its properties. Change the camera's location to x = 0, y = 0, z = 0.

Because the HoloLens is on your head, you will always have the main camera at the origin.

b. Also, on the Inspector tab, select the Clear Flags drop down and change it to solid color. Change the Background color to black. (In the HoloLens, the black background is not rendered.)

c. The only other thing you definitely need to change here is the Clipping Planes. The default for the near plane is .3 meters. This means, when you get within about a foot of your hologram, it will start clipping. The recommended amount for this in the HoloLens is .85, however, I have found that .5 works quite well. Experiment with this number on your own. For now, set it to .5.

d. You should be able to find a drop down in the Inspector Panel that is called Tracked Pose Driver. If you cannot, then click Add Component, search for Tracked Pose Driver and add it. You should not have to change anything.

e. Save your work.

Construct A Basic Hello World Application

1) Let's add some assets.

   a. On the Project tab at the bottom, **right click on Assets** and select **Create>New Folder. Rename it Materials**. Create another folder and **rename it Holograms**. This is fairly standard protocol for HoloLens apps.

   b. Now it's time to add some assets. There are several ways to create objects in Unity. Select **GameObject>3D Object> Cube** on the top row of tabs or **click the plus sign** on the Hierarchy tab 3D Objects>Cube or **right click anywhere in the Hierarchy tab** and select **3D Objects>Cube** and create a cube. Right click the name Cube in the Hierarchy tab to **change the name** or use the top text box on the Inspector tab and rename it **Platform**. (See Figure 6)

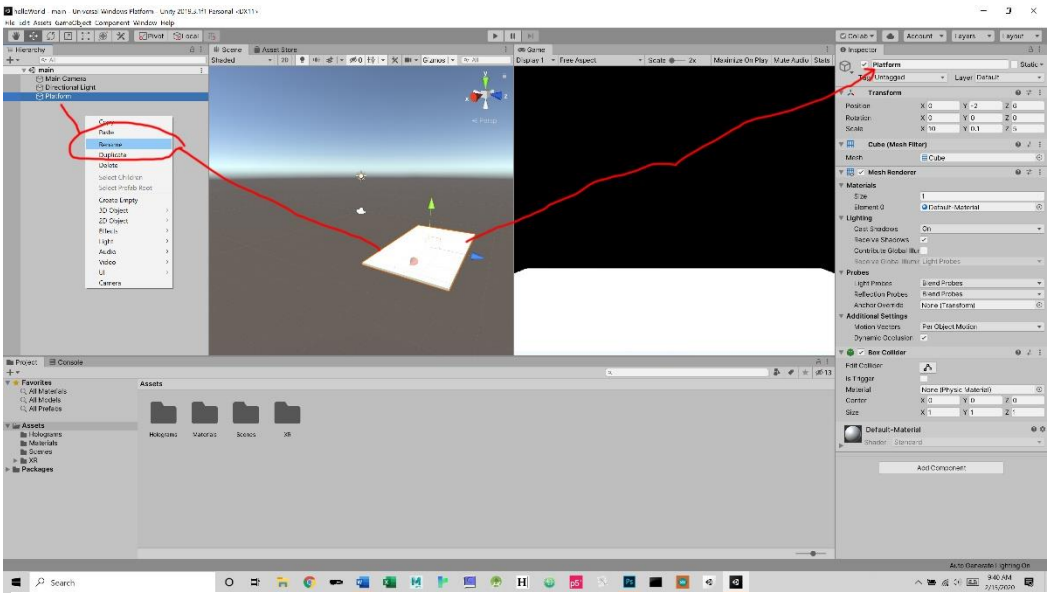


Figure 6

c. Below the text box on the Inspector tab, reset the location of the Platform to x=0, y = -2 and z = 6. Change its scale to x = 10, y = .1 and z = 5.

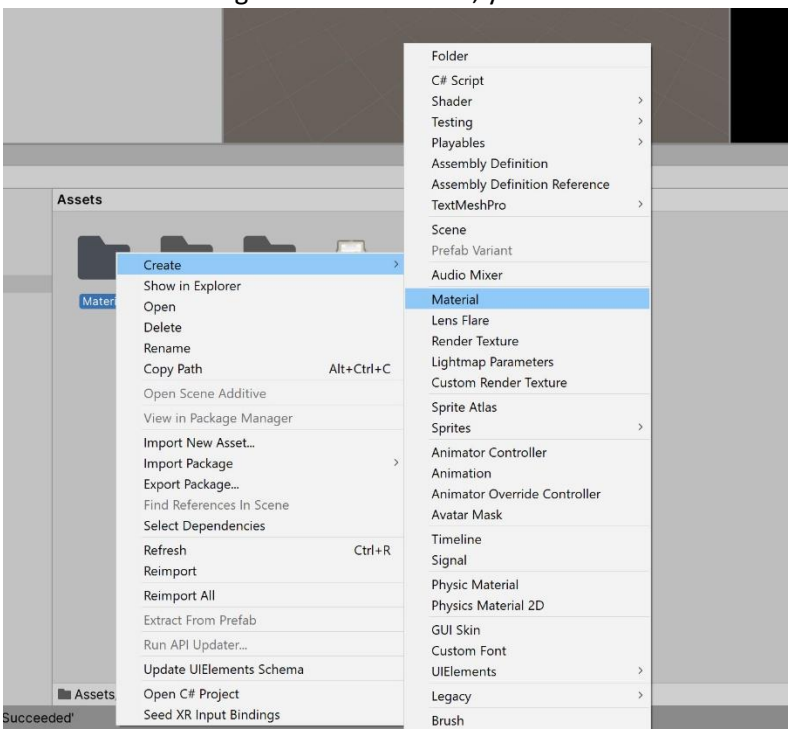


Figure 7 Create a new Material

d. Right click on the Material folder and click on Create>Material. Rename it Platform_m. (See Figure 7).
e. Change the material to a color of your choice.
    i. Click on the material sphere and look at the Inspector tab.
    ii. At the top, Standard shader is the default. For now, that is ok to leave.
    iii. Below that you have Albedo (translates to your diffuse map in Maya). Below that is Metallic, Normal, Height etc.
    iv. For this project, change the Albedo only by left clicking in the white box next to the eye dropper. A color picker opens for you to select your color. Green might be nice.
f. There are several ways to apply your material to the platform. Easiest is to grab the material sphere from the Project tab and drag it onto your platform object.
    i. Alternatively, you can drag the material directly onto the platform object in the Hierarchy list or you can select the platform from the Hierarchy list and drag the material down onto the bottom of the Inspector panel.
g. Your platform should now be whatever color you chose for your material.

2) Now it is time for the text. I prefer right clicking in the Hierarchy panel and selecting 3D Object>3D Text. You should see a New Text object listed in the Hierarchy and the words Hello World at the origin.
3) Ensure your text object is selected in the Scene Panel. It will glow red around the edges if it is.
4) Look at the Inspector panel. In the top text box, change New Text to read Hello.

5) Still on the Inspector panel, check out the Text Mesh drop down. Change the Text to just Hello by deleting the space and World.
6) If you look at the word in the Scene panel you will notice that the text is not clear. This is because Unity renders the font as a mesh and applies a texture map to it. Try changing the Font Size from 0 to 13. Nothing appears to happen. Double that to 26 and change the character size to .5. There is an inverse relationship. As you continue to increase the Font Size and decrease the character size, you approach an appropriately clear font. If you wish just set the Font Size to 78 and the Character Size to .125. Now your word should be sufficiently clear.
7) Set its position to x = -3, y = 2 and z = 6.
8) Ctrl/Cmd +D will duplicate your first word. Transform its position to x = -.4, y = 2 and z = 6.
9) Change the name in the top text box on the Inspector Panel or right click on the new word in the Hierarchy panel and rename it. In the Text Mesh drop down, change the text to World!.

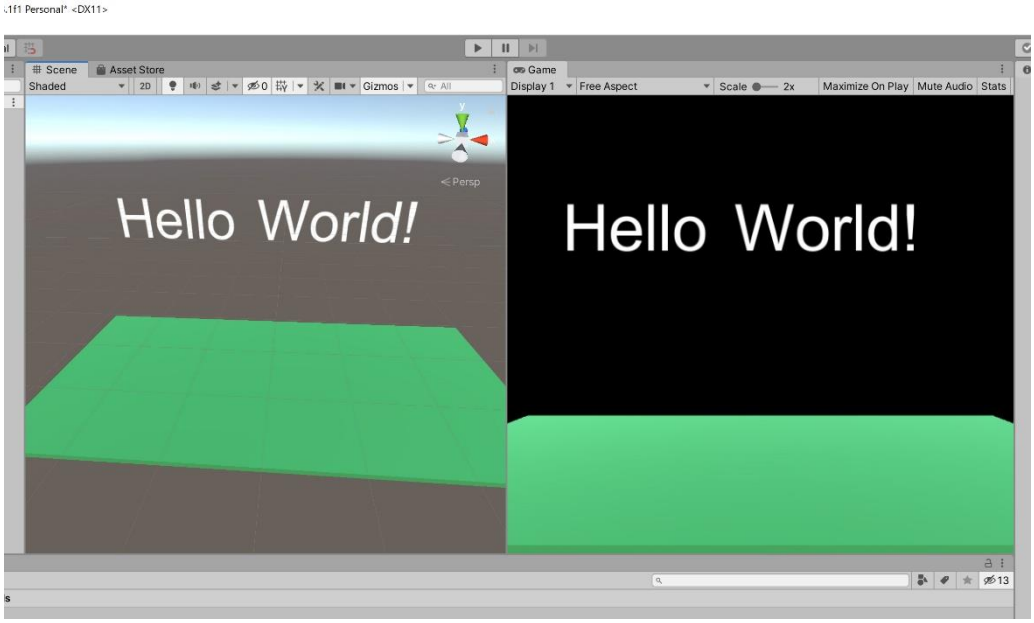


Figure 8 Hello World

10) You could import this into the HoloLens, but it would not be very interesting. Time to ad movement.

Adding Movement to Hello World!

1) In the Hierarchy panel, make sure the 3D Text object Hello is selected. At the bottom of the Inspector panel, click the button that says Add Component. Scroll down to Physics>Rigid Body and select it. This will allow the Unity physics engine to act on your object. Change the settings to Drag = 2 and Angular Drag = 3. Make sure gravity is clicked. Remember, you set up the gravity in this environment to be slightly less than earth gravity.
2) If this application ran now, the word Hello would drop for eternity without stopping. In order for it to react with other objects in the scene, it needs a box collider. Add Component and then select Physics>Box Collider.

3) Next do the same thing to the 3D Text object, World and set the Drag = 4 and Angular Drag = 5. Add a Box Collider.
4) Save your work.

Building your application

1) Now you are ready to build your application. You have already set up your build settings.
2) Go to File>Build Settings again. Make sure the scene you in which you are building shows up in the list at the top.
3) For now, do not build and run only click on Build.
4) A popup will ask which folder you wish to save your build in. Add a folder called App and click build. This should be the only time you have to create this folder. Every other time you build, simply select that folder to store your build in. The first build is always the longest. Give it a while.
5) If the folder does not open on its own when it is finished, you can always navigate to it. Find the App folder in your project and open it. Inside you will find a file with a .sln extension on it. It will be named based on your project name.
6) Double click on the .sln file inside the App folder. It should open up Visual Studio for you.

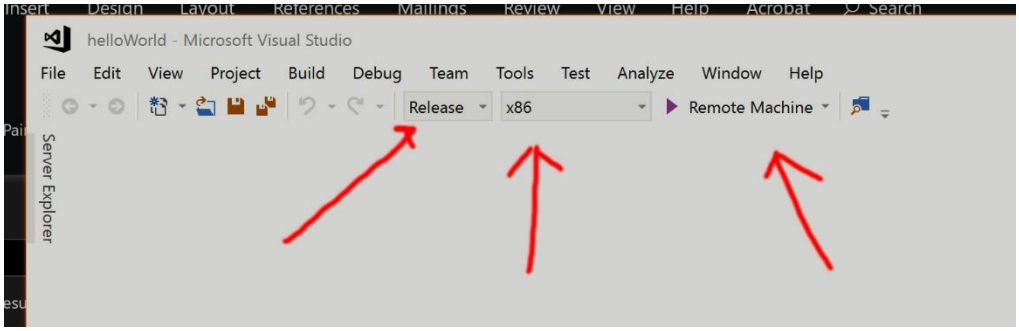


Figure 9 Visual Studio deploy settings.

7) You may need to let Visual Studio know how to locate your remote HoloLens. To check, go to .

8) To deploy your app make sure you are deploying a Release x86 version to your Remote Machine and that your HoloLens is on. If the IP address of your HMD shows here you are good to go. (See Figure 10)

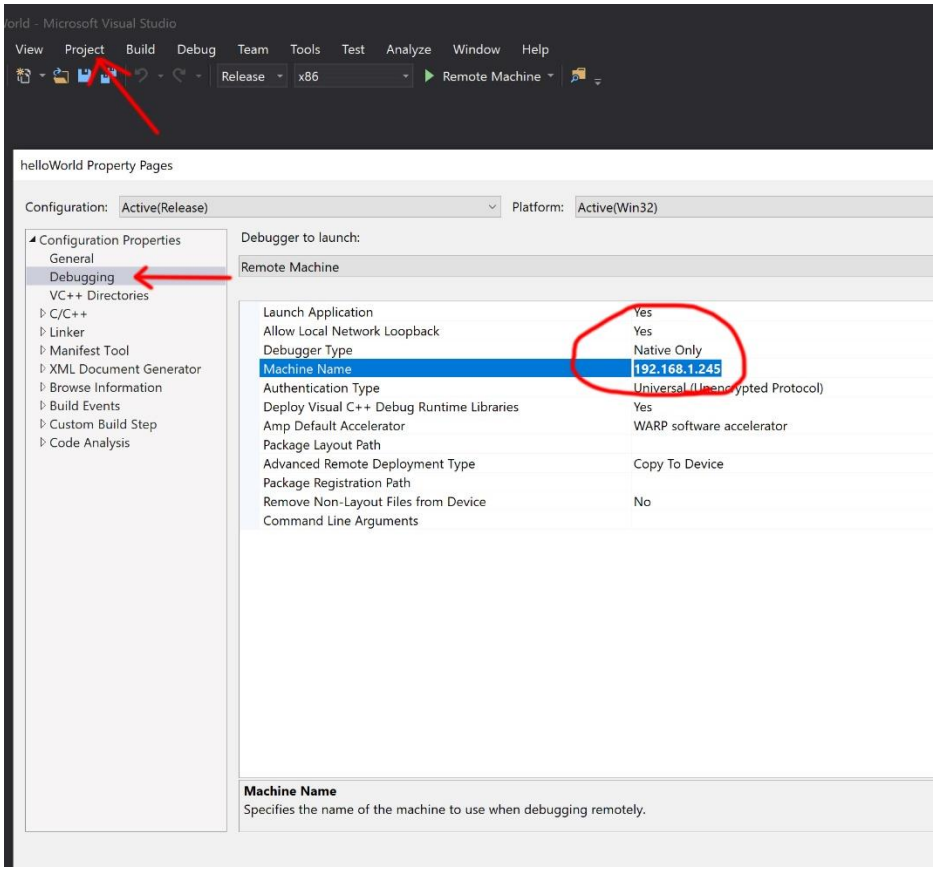


Figure 10 In Visual Studio click Project>Properties>Debugging and enter your IP address.

9) Place the HoloLens on your head and make sure it is on.
10) Now you are ready to Deploy your build. Click on the arrow to the left of Remote Machine and wait….wait….wait for it….
11) You will eventually see the Unity Splash image in the HoloLens and then very quickly, the words will drop and land on the platform.
12) To see it happen again, use the bloom gesture and find your app in the list of installed app. Click on it and wait for a billboard. Click on the billboard and your app will run.

Yeah!!! First app built.