VR TRAINING

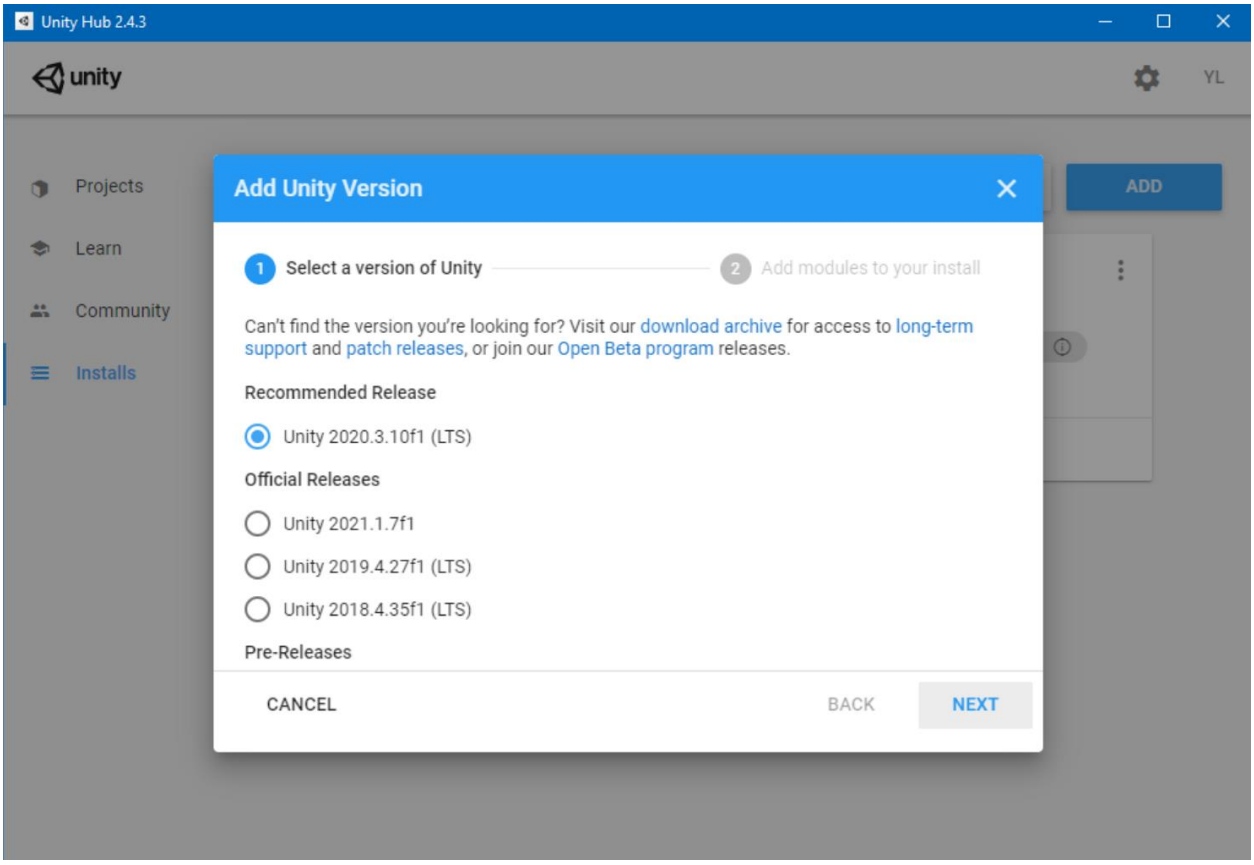
Contents

Setting up your project: Choosing a Unity Version and XR Plugin

While we currently **recommend installing Unity 2020.3 LTS with the latest Mixed Reality OpenXR plugin** for Mixed Reality development, you can build apps with other Unity configurations as well.

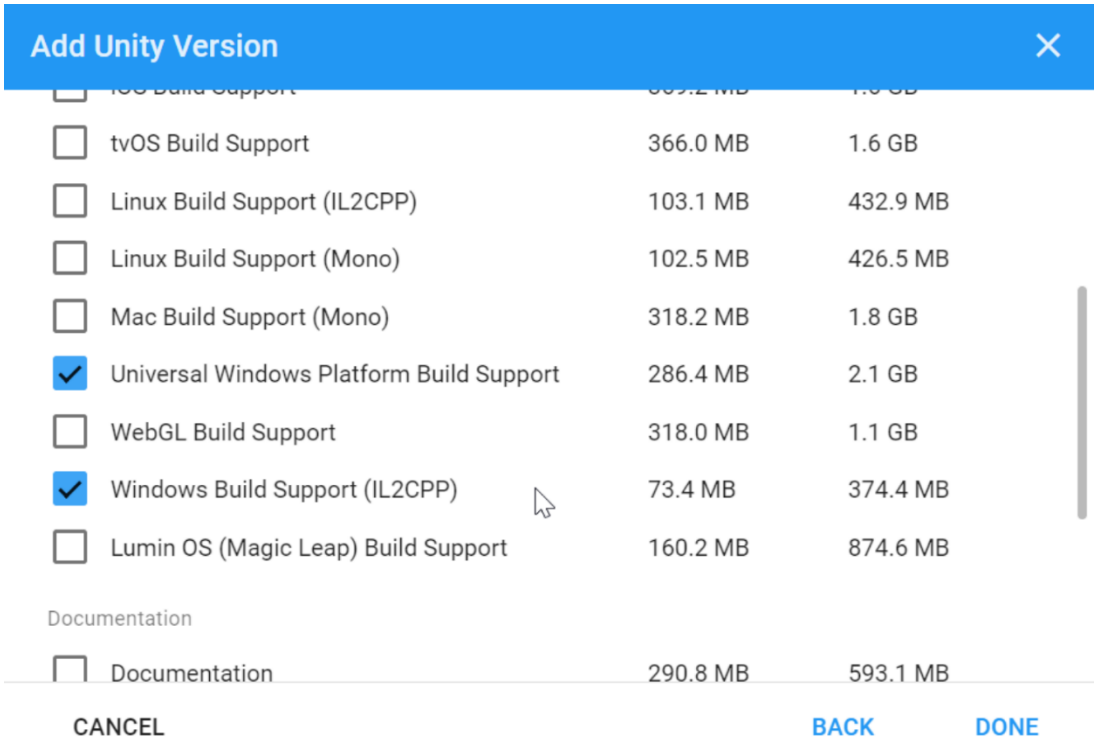
The best way to install and manage Unity is through the **Unity Hub**:

1. Install **Unity Hub**.
2. Select the **Installs** tab and choose **Add**.
3. Select **Unity 2020.3 LTS** and click **Next**.

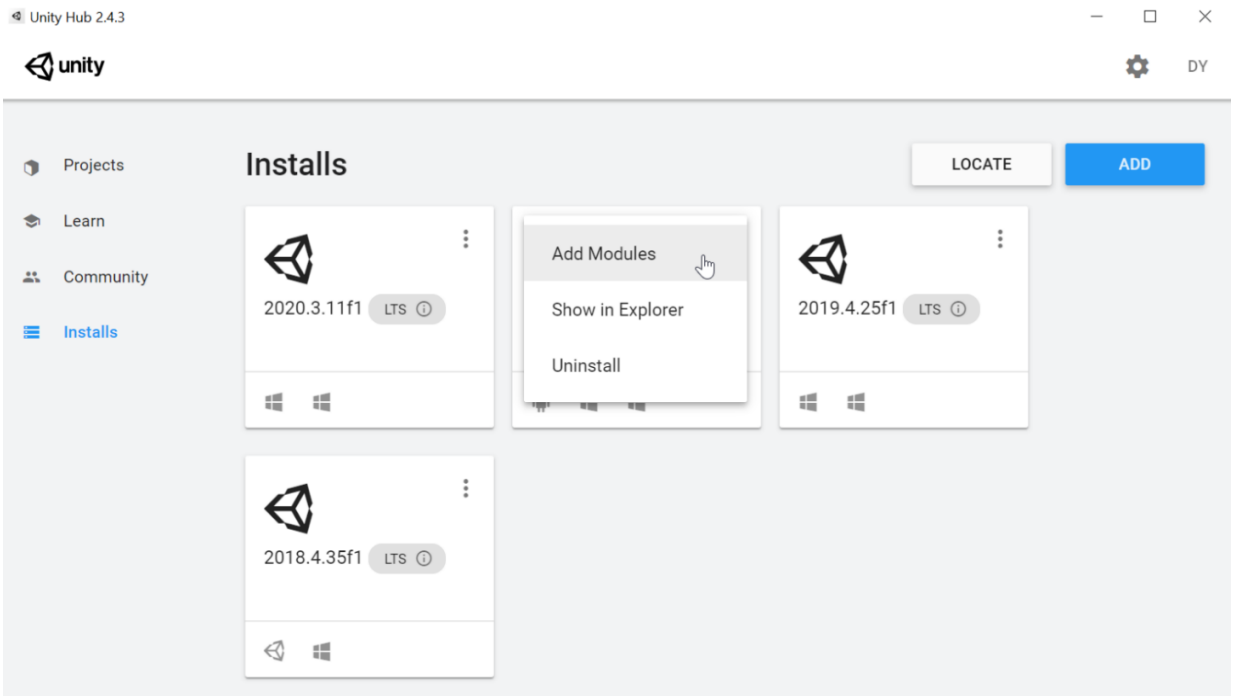


Check the following components under **'Platforms'**:

- **Universal Windows Platform Build Support**
- **Windows Build Support (IL2CPP)**



If you previously installed Unity without these options, you can add them through **'Add Modules'** menu in Unity Hub:



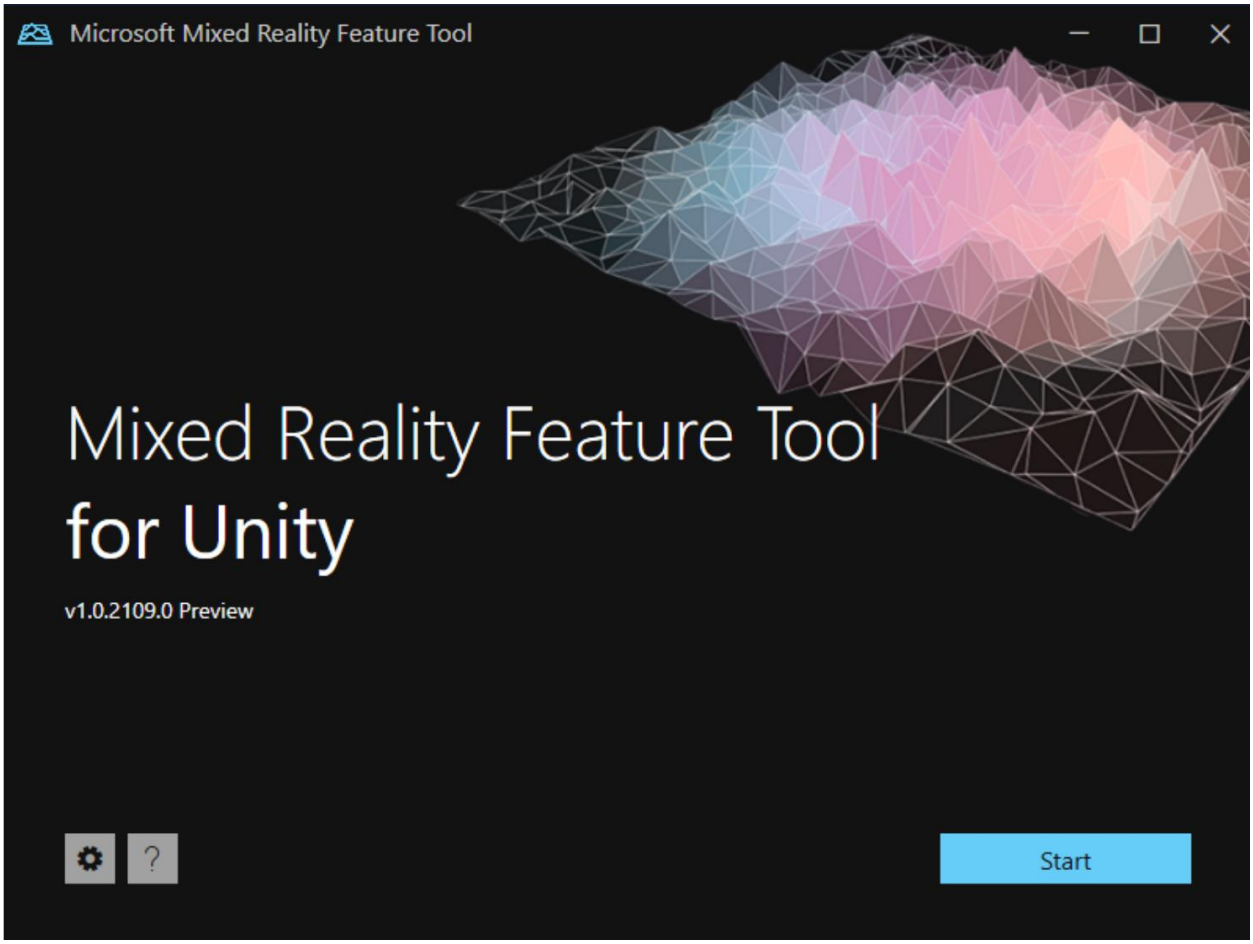
Setting up your XR configuration

Once you have your environment set up:

- Download the latest version of the Mixed Reality Feature Tool from the Microsoft Download Center.
- When the download completes, unzip the file and save it to your desktop.

Follow the below stated steps:

1. Launch the Mixed Reality Feature Tool from the executable file, which displays the start page on first launch:

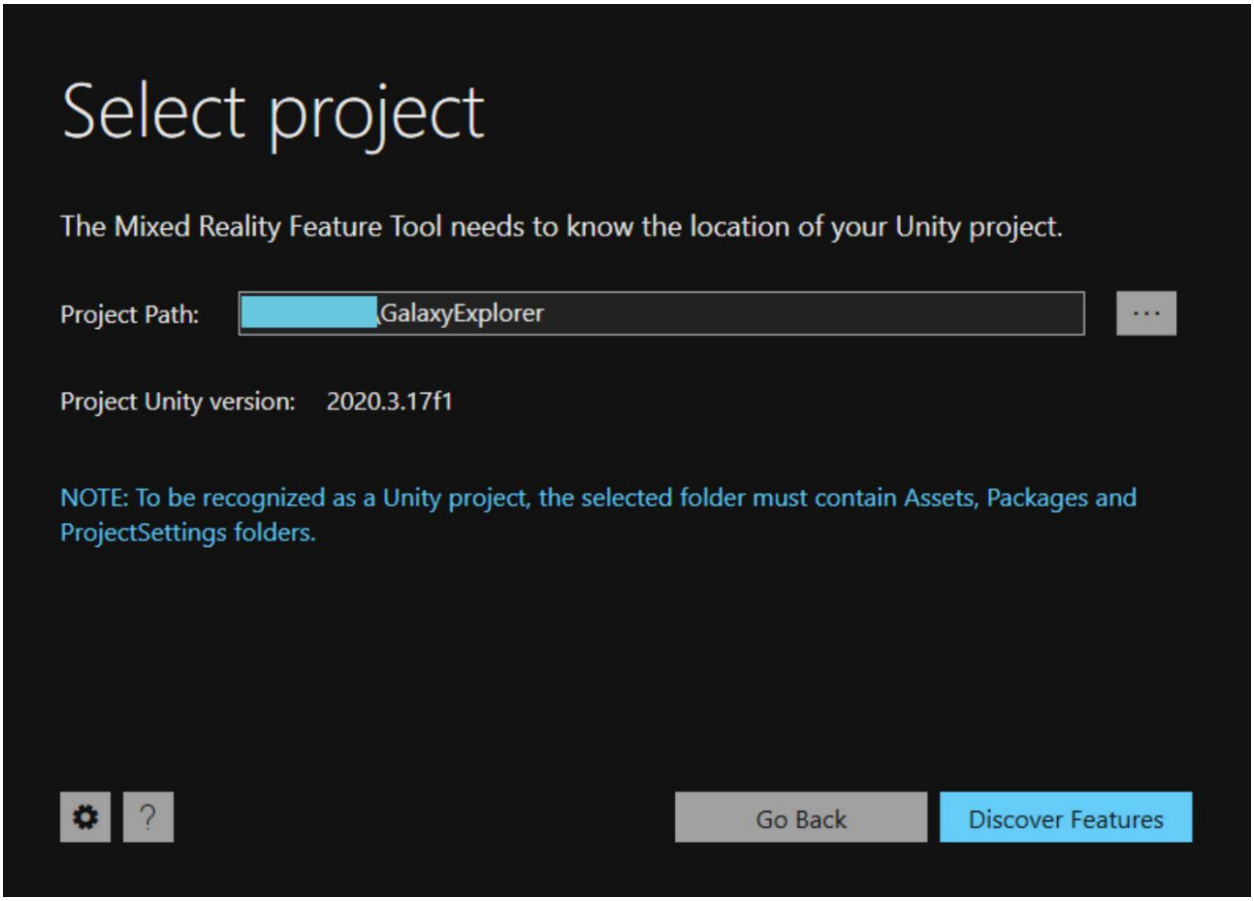


From the start page, you can:

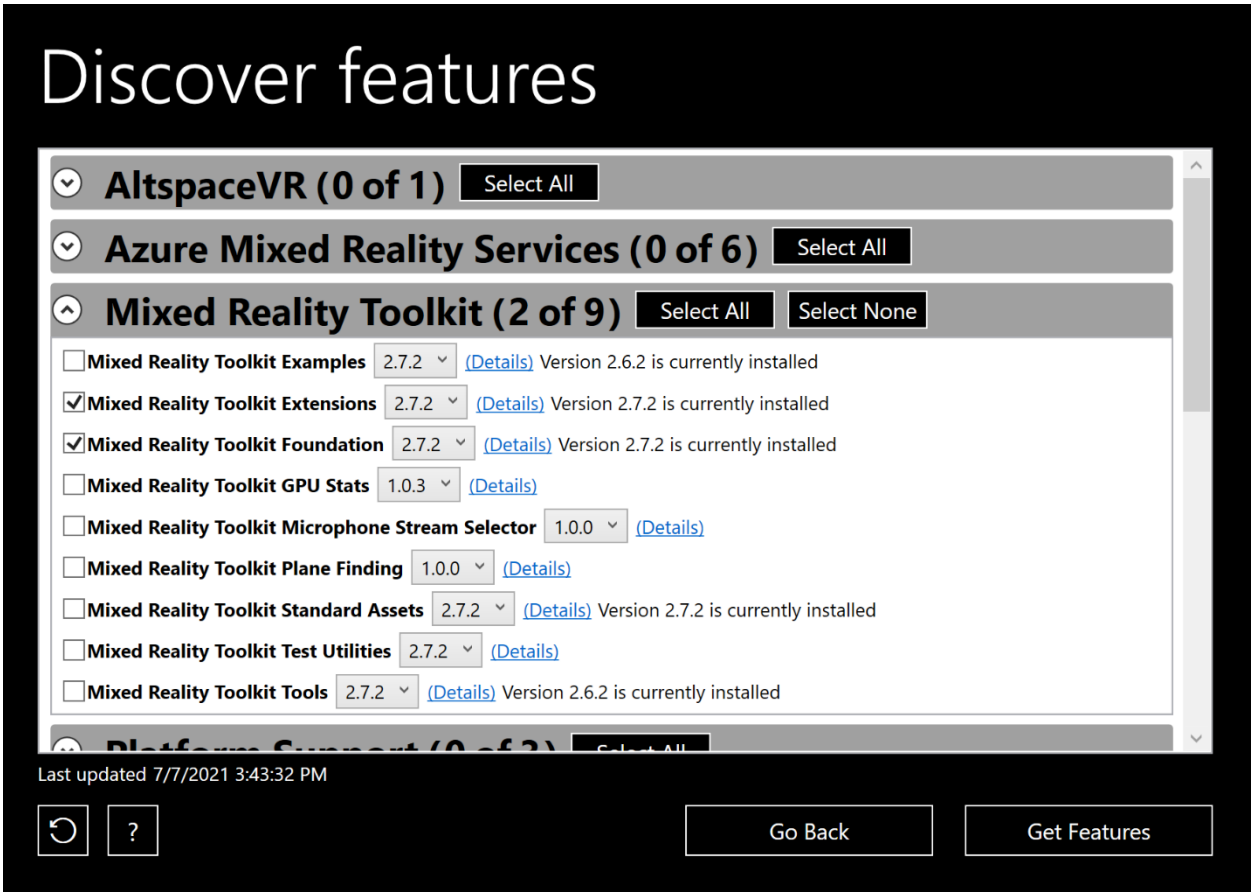
- Configure tool settings using the **gear icon** button
- Use the **question mark** button to launch the default web browser and display our documentation
- Select **Start** to begin discovering feature packages

2. Selecting your Unity project

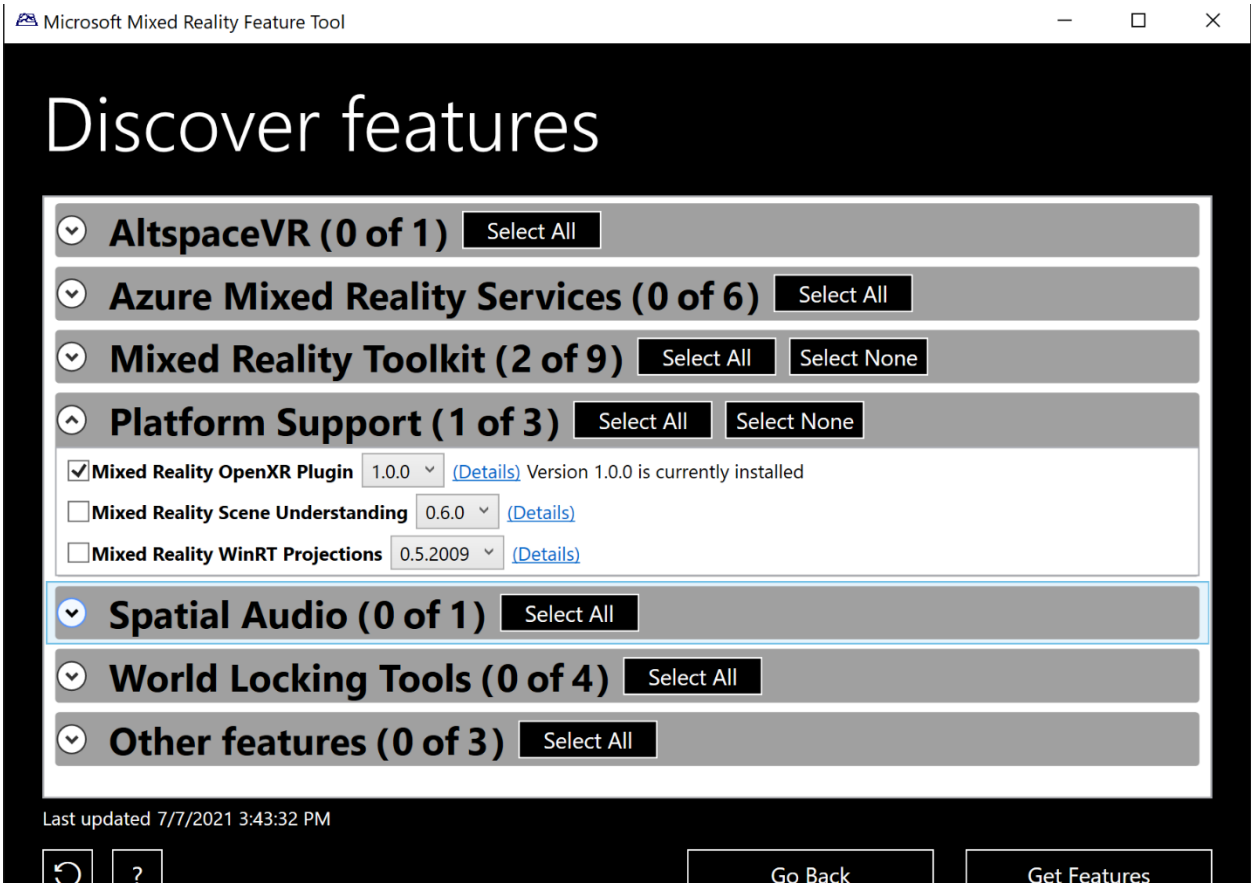
To ensure that all discovered features are supported on your project's version of Unity, the fist step is to point the Mixed Reality Feature Tool to your project using the **ellipsis** button (to the right of the project path field).



3. Discovering and acquiring feature packages



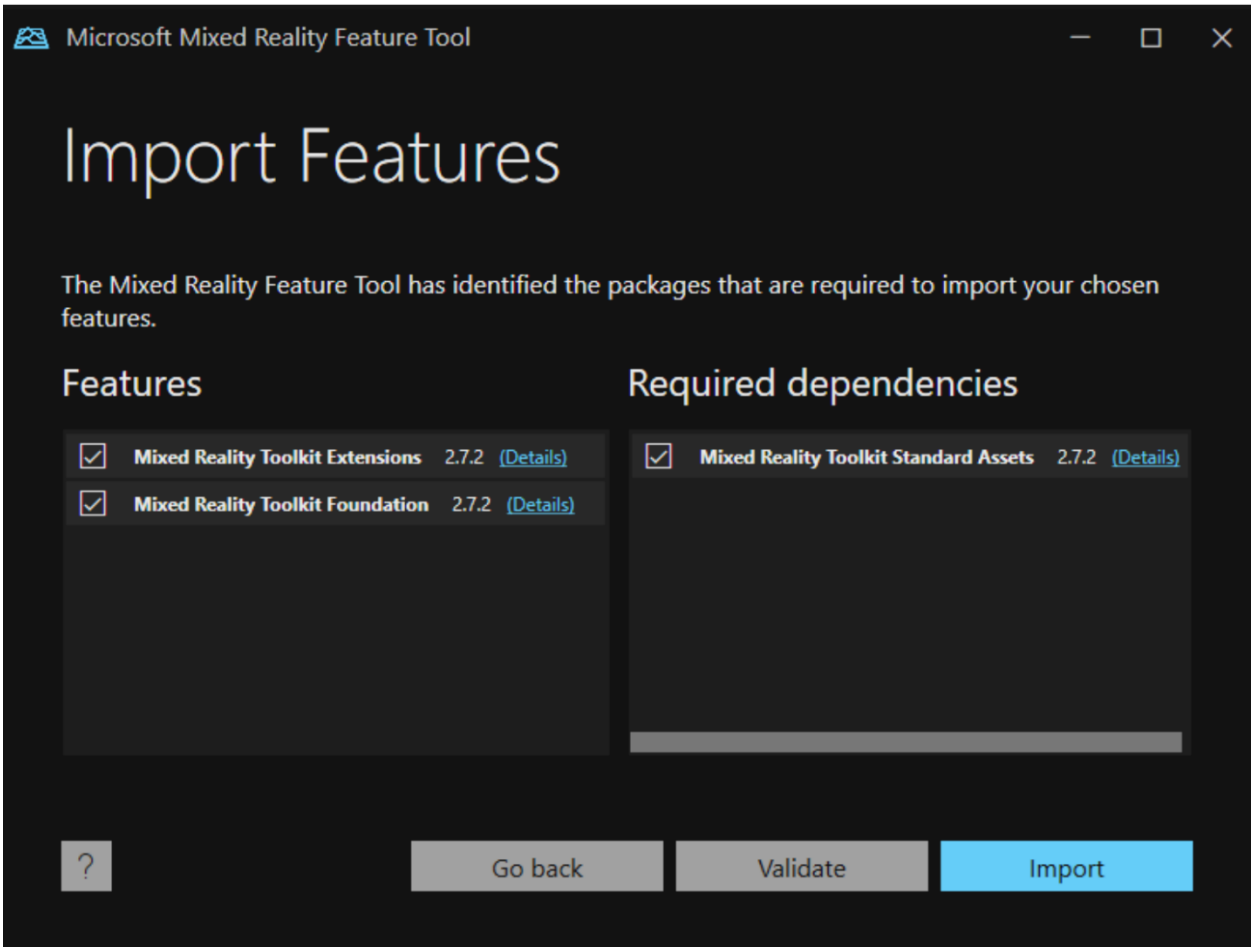
When the Mixed Reality Feature Tool recognizes previously imported feature(s), it displays a notification message by each.



Once you've made your choices, select **Get features** to fetch all the required packages from the catalog.

4. Importing feature packages

Following acquisition, the complete set of packages is presented, along with a list of required dependencies. If you need to change any feature or package selections, this is the time:



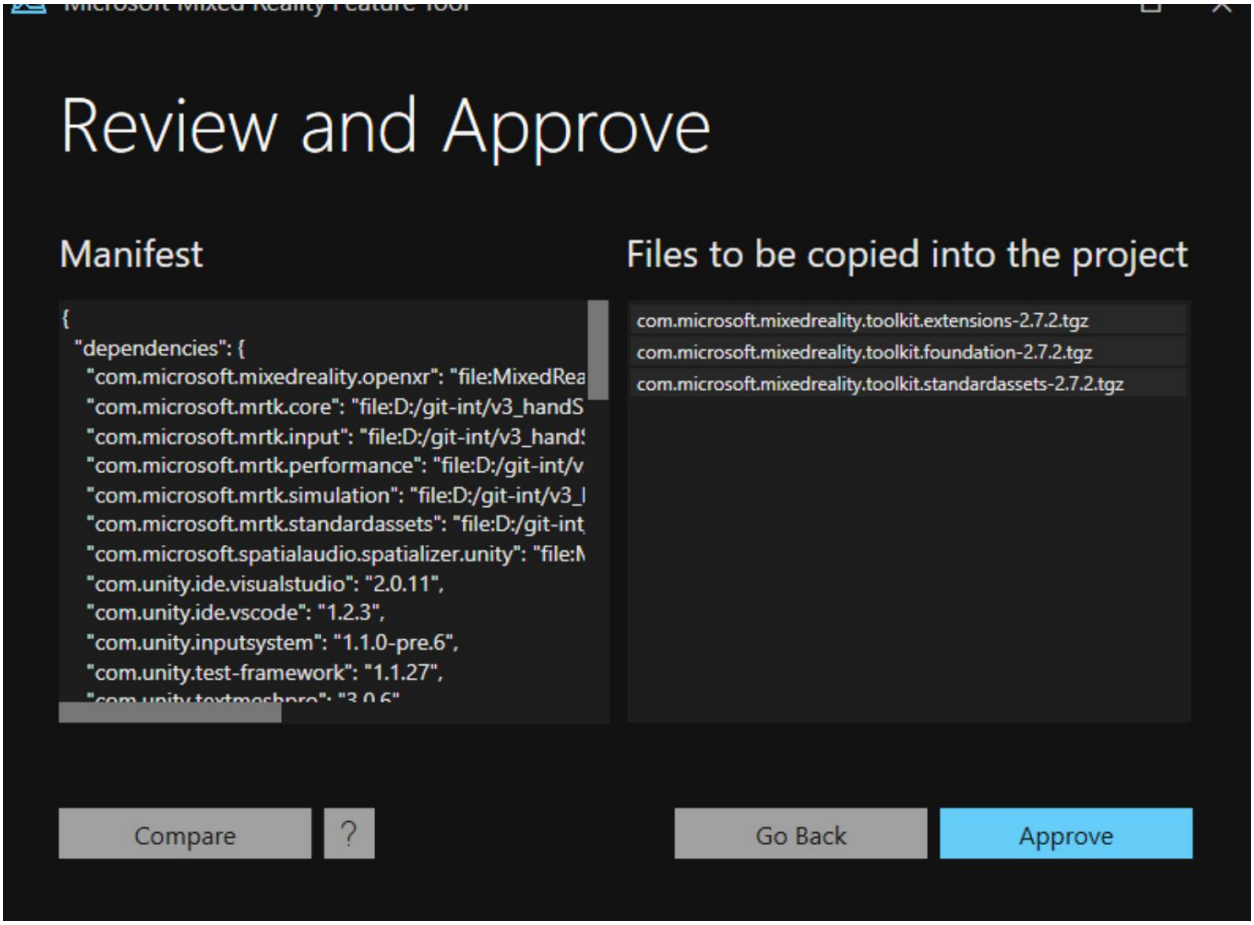
We highly recommend using the **Validate** button to ensure the Unity project can successfully import the selected features. After validation, you'll see a pop-up dialog with a success message or a list of identified issues.

Select **Import** to continue.

5. Reviewing and approving project changes

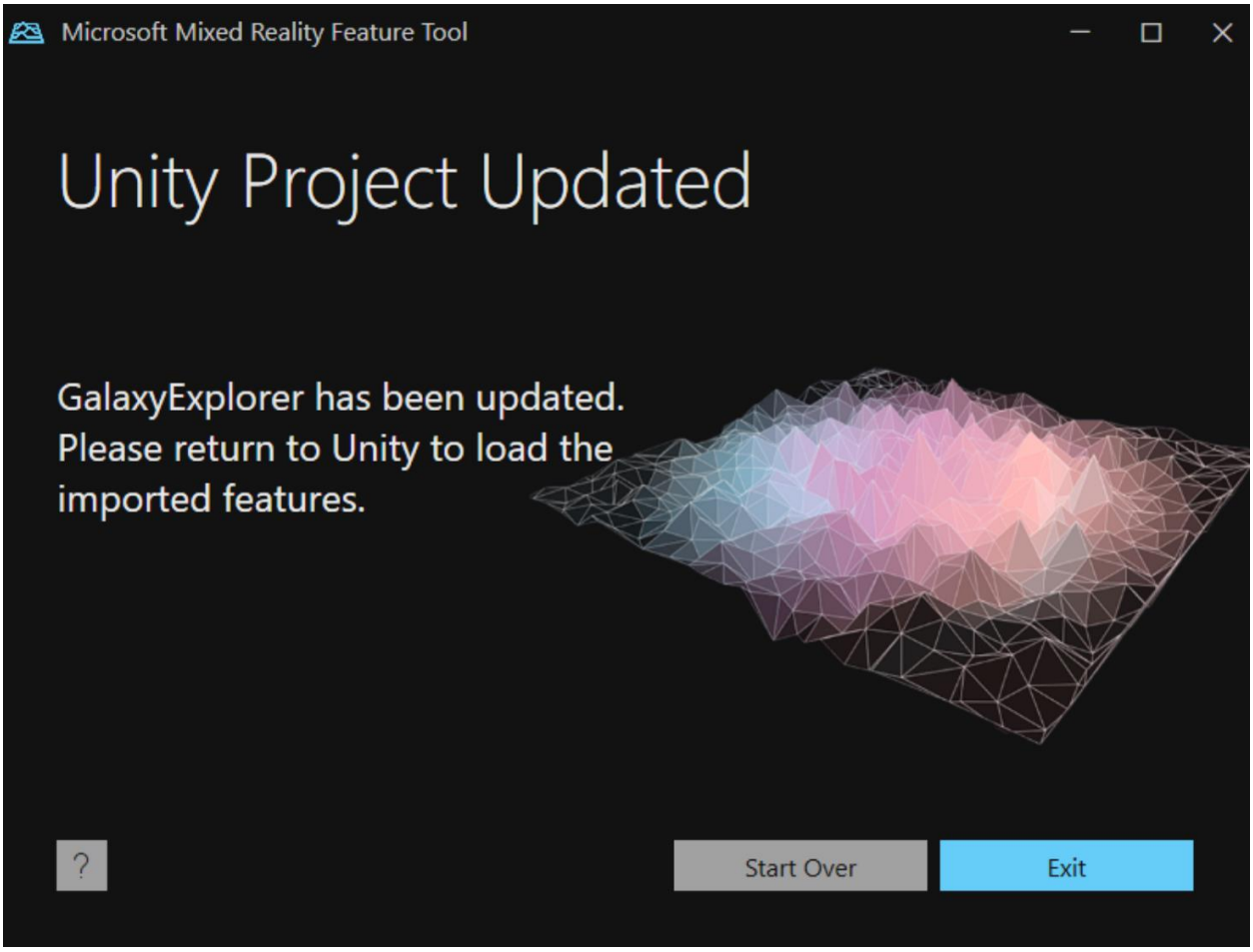
The final step is reviewing and approving the proposed changes to the manifest and project files:

- The proposed changes to the manifest are displayed on the left
- The files to be added to the project are listed to the right
- The **Compare** button allows for side-by-side viewing of the current manifest and the proposed changes



6. Project updated

When the proposed changes are approved, your target Unity project is updated to reference the selected Mixed Reality features.



The Unity project's **Packages** folder now has a **MixedReality** subfolder with the feature package file(s) and the manifest will contain the appropriate reference(s).

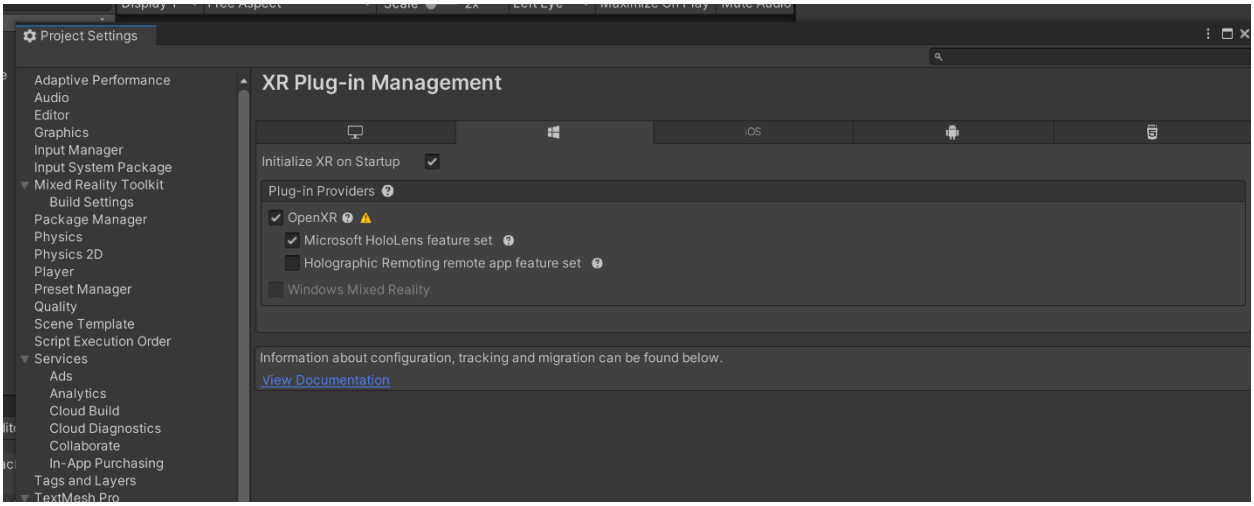
Return to Unity, wait for the new selected features to load, and start building!

Configuring XR Plugin Management for OpenXR

To set OpenXR as the the runtime in Unity:

1. In the Unity Editor, navigate to **Edit > Project Settings**
2. In the list of Settings, select **XR Plugin Management** (should already be installed if you installed the Mixed Reality OpenXR plugin using MRFT)
3. Check the **Initialize XR on Startup** box
4. If targeting Desktop VR, stay on the PC Standalone tab (the monitor) and check the **OpenXR** and **Windows Mixed Reality feature set** boxes

5. If targeting HoloLens 2, switch to the Universal Windows Platform tab (the Windows logo) and select the **OpenXR** and **Microsoft HoloLens feature set** boxes

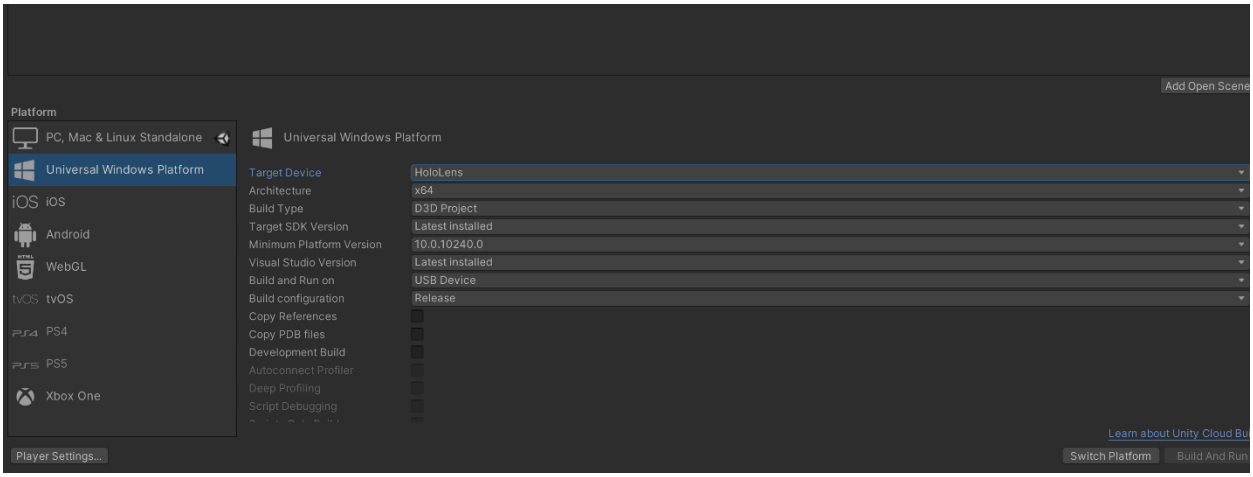


If you see a yellow warning icon next to **OpenXR Plugin**, click the icon and select **Fix All** before continuing. The Unity Editor may need to restart itself for the changes to take effect.

Setting your build target

Since we are targeting HoloLens 2, you need to switch to the Universal Windows Platform:

1. Select **File > Build Settings...**
2. Select **Universal Windows Platform** in the Platform list and select **Switch Platform**
3. Set **Architecture** to **ARM64**
4. Set **Target device** to **HoloLens**
5. Set **Build Type** to **D3D Project**
6. Set **Target SDK Version** to **Latest installed**



7. Click Switch Platform

Importing Maps SDK package

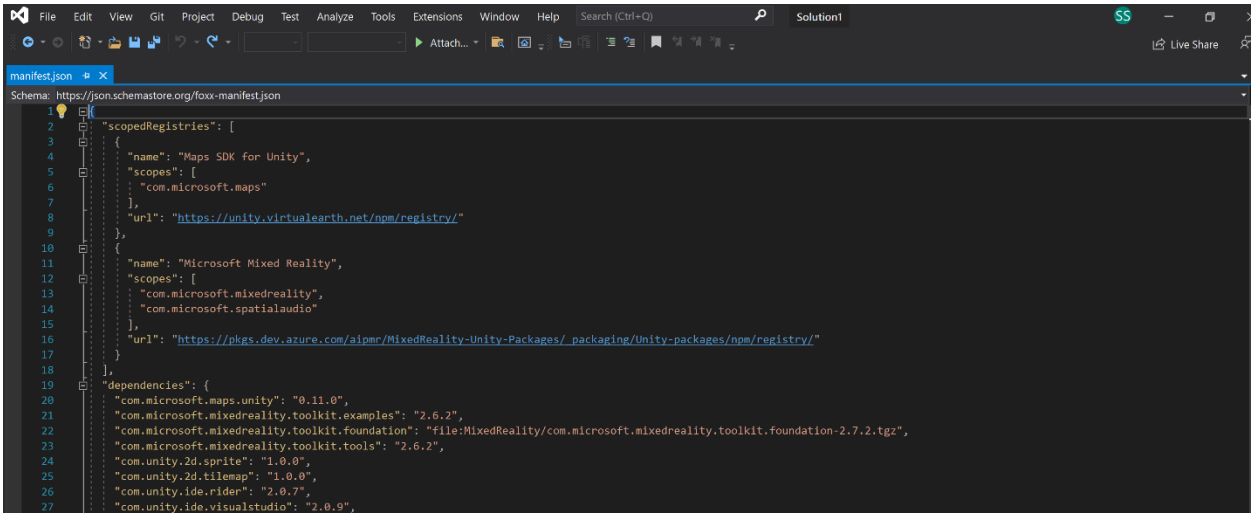
The SDK can also be added to an existing Unity project. This requires modifying Unity's package manifest to include a reference to the SDK package.

1. Locate the Unity project's folder.
   In the Packages directory, open manifest.json.
   Append the scopedRegistries section to the beginning of the manifest and modify dependencies as follows:

```
{
 "scopedRegistries": [
  {
   "name": "Maps SDK for Unity",
   "url": "https://unity.virtualearth.net/npm/registry/",
   "scopes": [
    "com.microsoft.maps"
   ]
  }
 ],
 "dependencies": {
  "com.microsoft.maps.unity": "0.11.1",

  <Existing dependencies: Do not remove>
```

```
    }
}
```

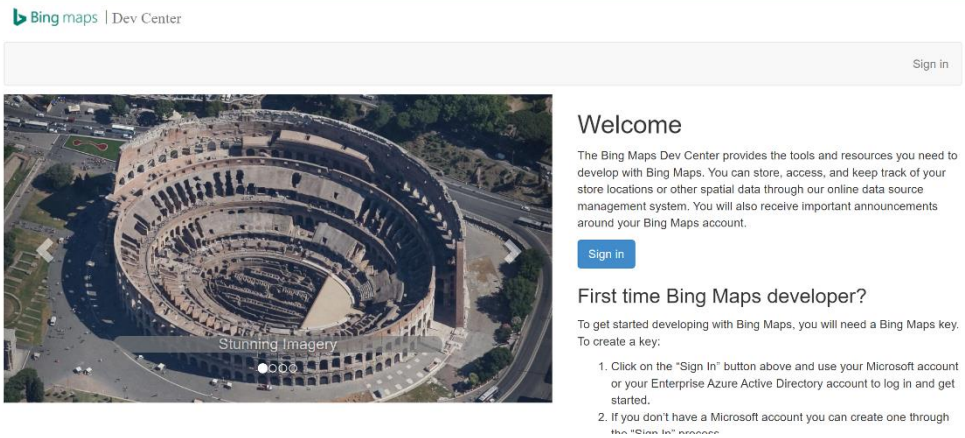


2. Open the Unity editor for the corresponding project. A dialog should appear showing the progress of importing the new package.

Create a Bing Maps key

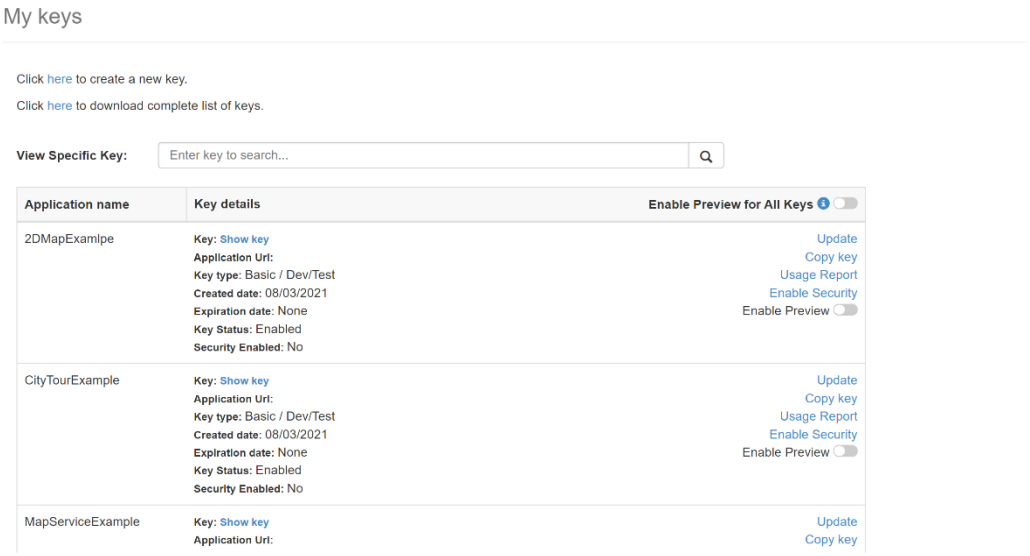
A Bing Maps developer key is required to enable the mapping functionality of the SDK.

1. Sign-in to the Bing Maps Dev Center.

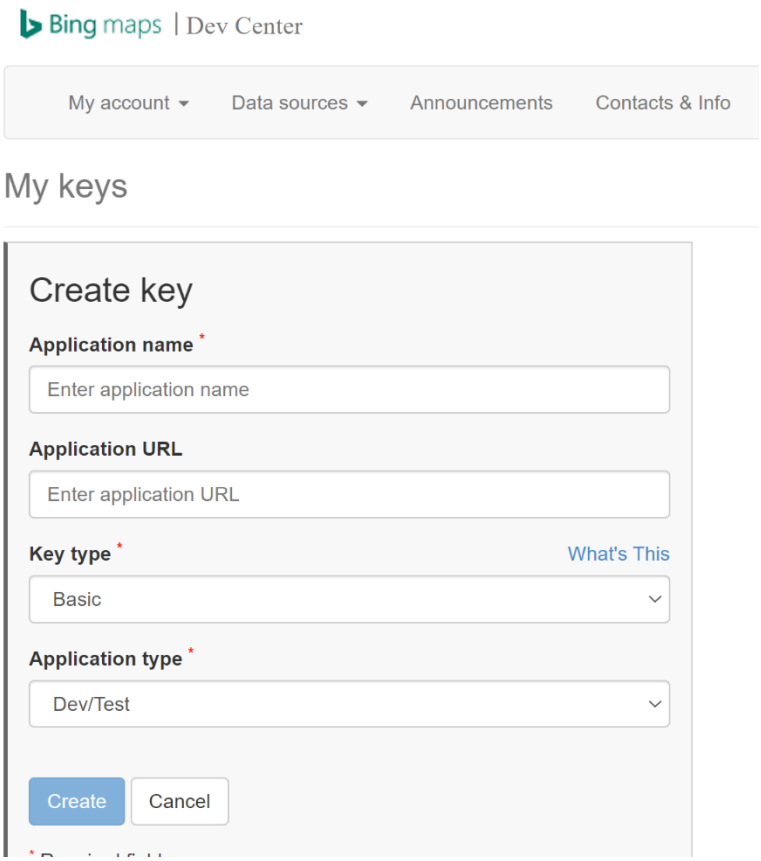


   - For new accounts, follow the instructions at Creating a Bing Maps Account.

2. Select **My keys** under **My Account** and select the option to create a new key.



3. Provide the following required information to create a key:
   - **Application name**: The name of the application.
   - **Key type**: *Basic* or *Enterprise*.
   - **Application type**: Select *Other Public Mobile App*.

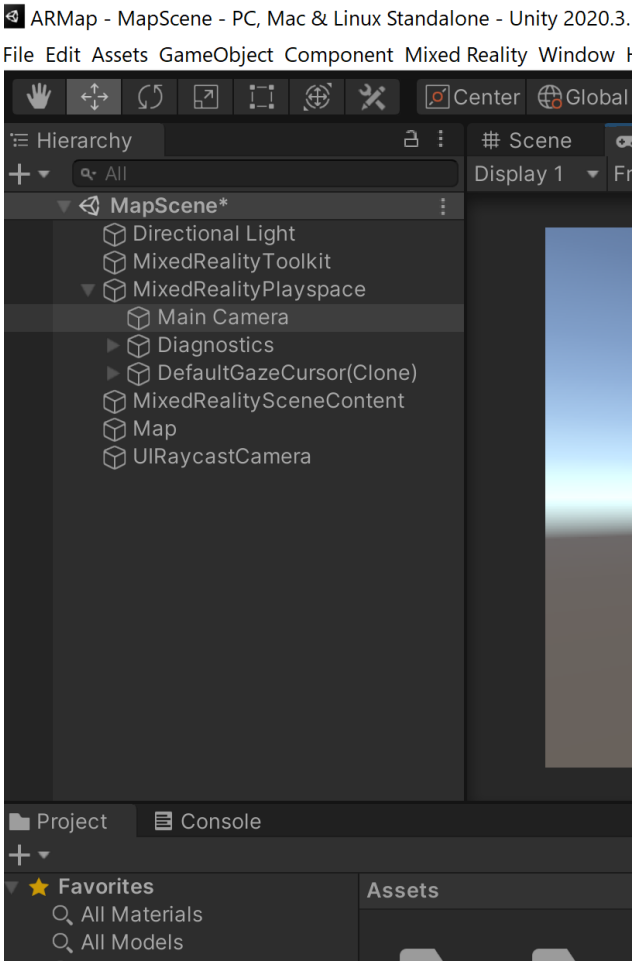


4. Click the **Create** button. The new key displays in the list of available keys. This key will be used later when setting up the Unity project.

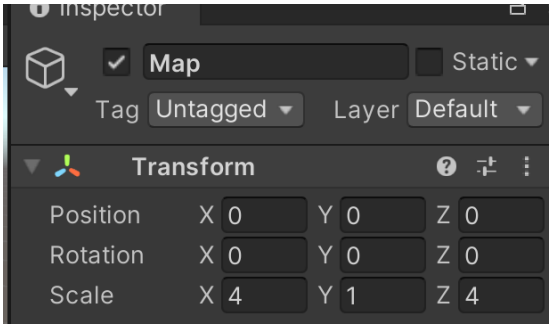
Building our first application with Unity

Open your configured project and follow the steps:

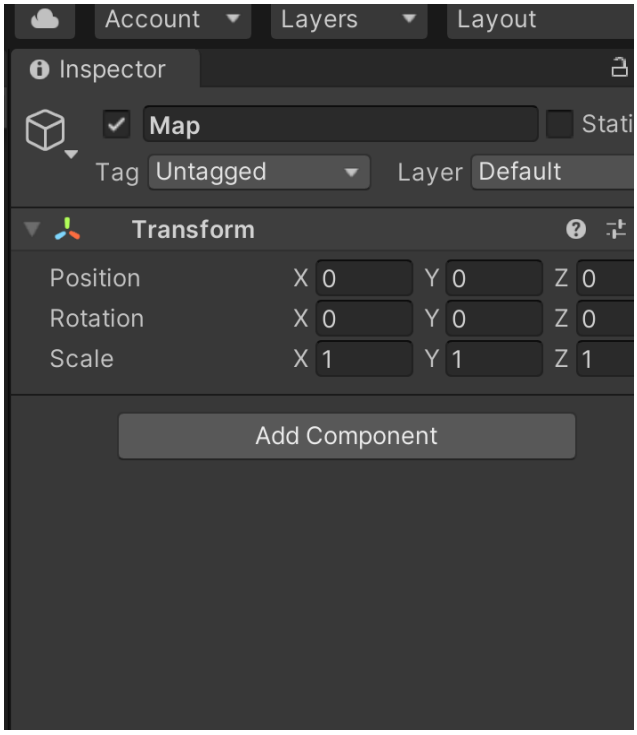
1. Select Mixed Reality > Toolkit > Add to Scene and Configure.
2. Create a new Empty Object in your scene and rename it to Map.



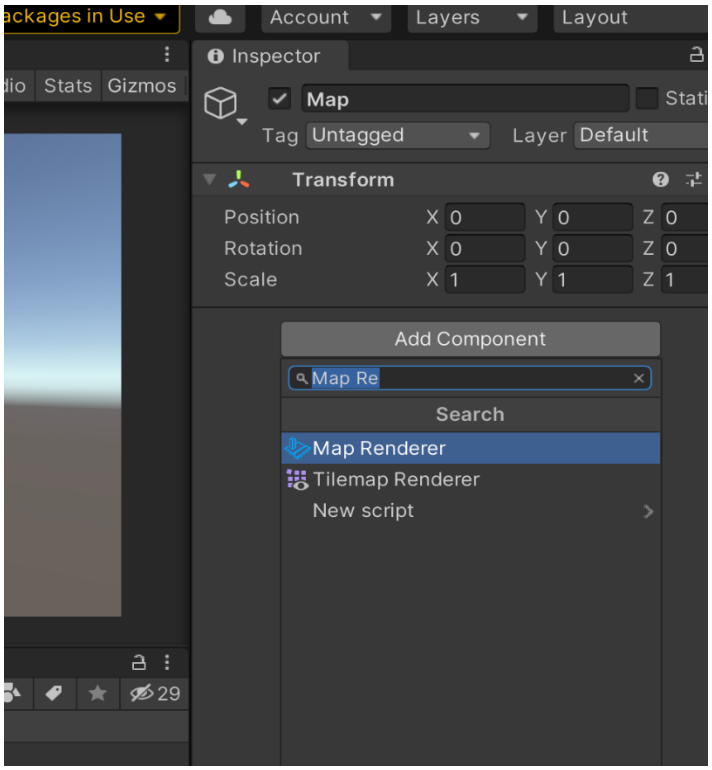
3. Scale the Map in X to 4 and Z to 4.



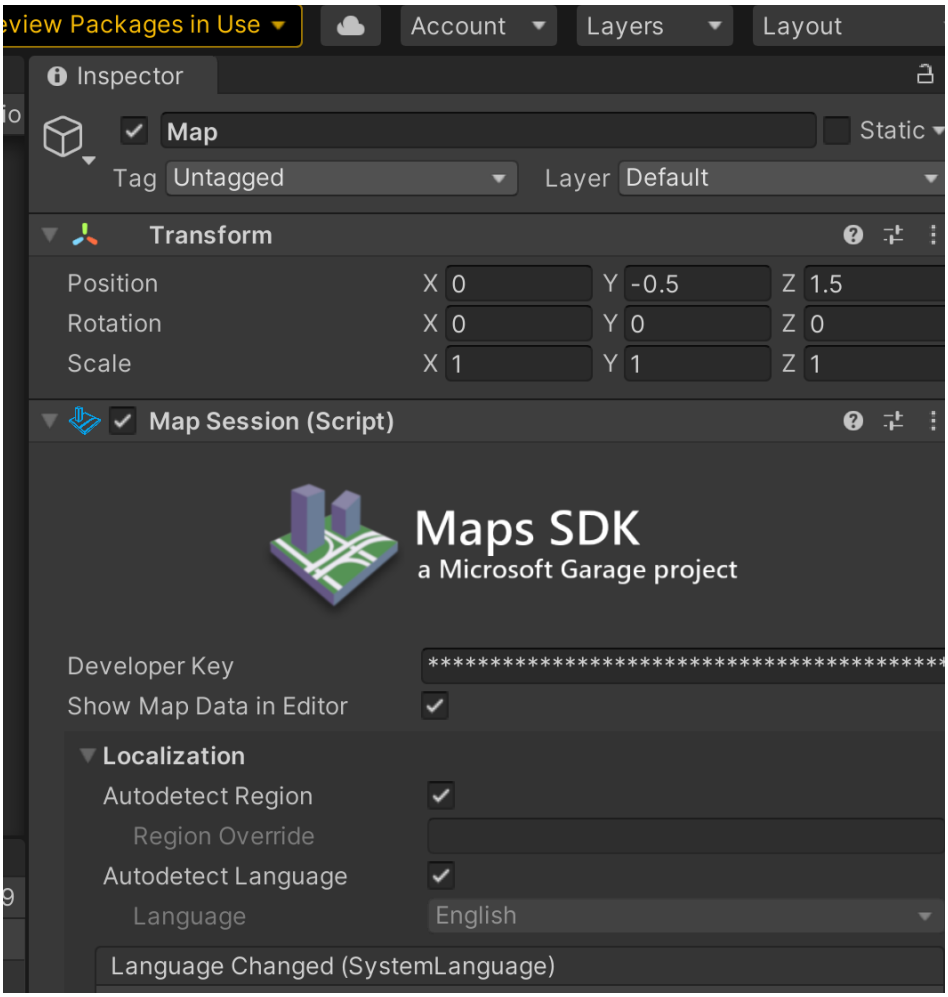
4. Click the Map and on the right side in the Inspector you will have the option: **Add a Component**
5. Click on **Add a Component**



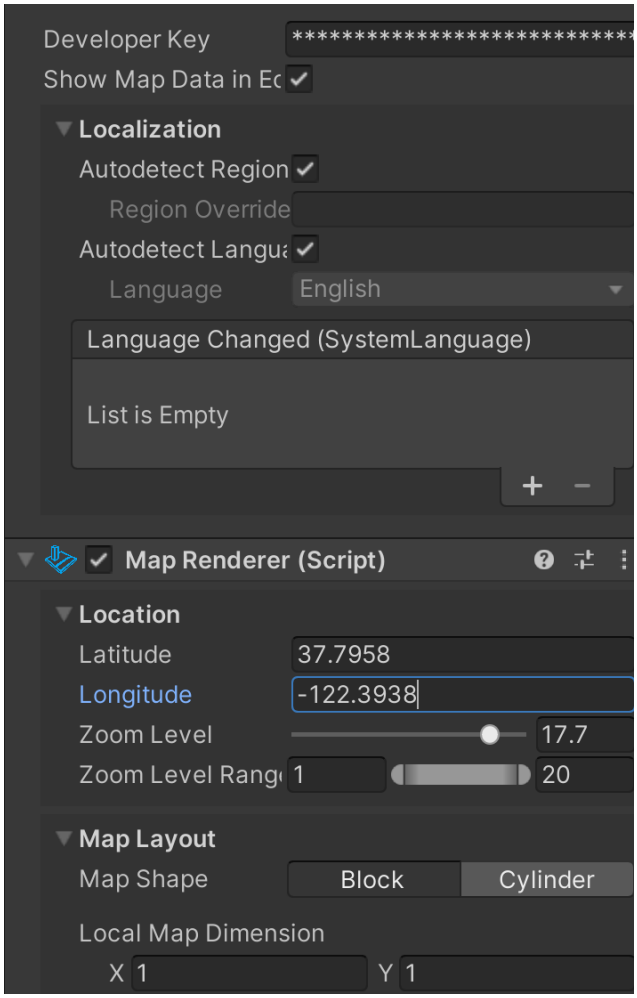
6. Search for **Map Renderer** and add it.



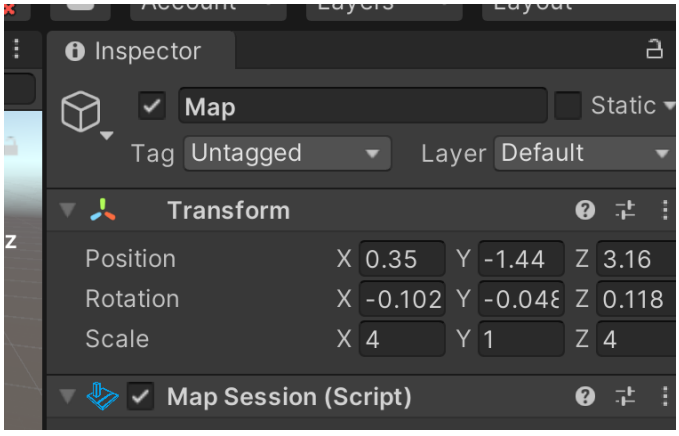
5. Copy the key from Bing Maps Portal and paste it to the **Developer key** placeholder.

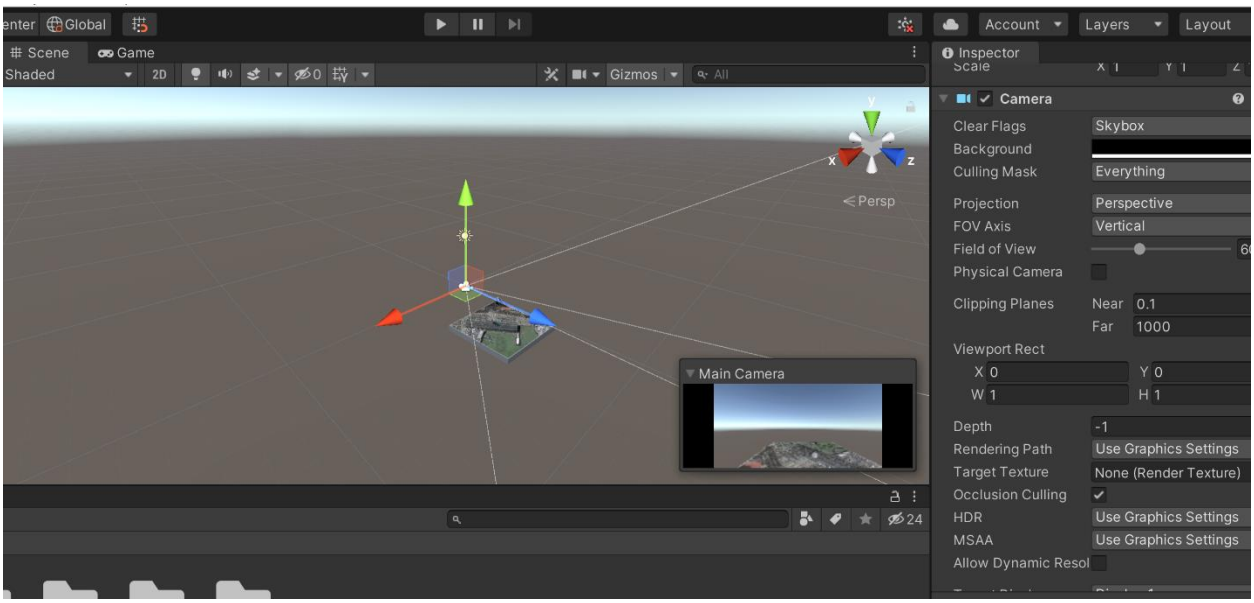


6. Give the **Latitude** and **Longitude** coordinates of a city available in 3D.

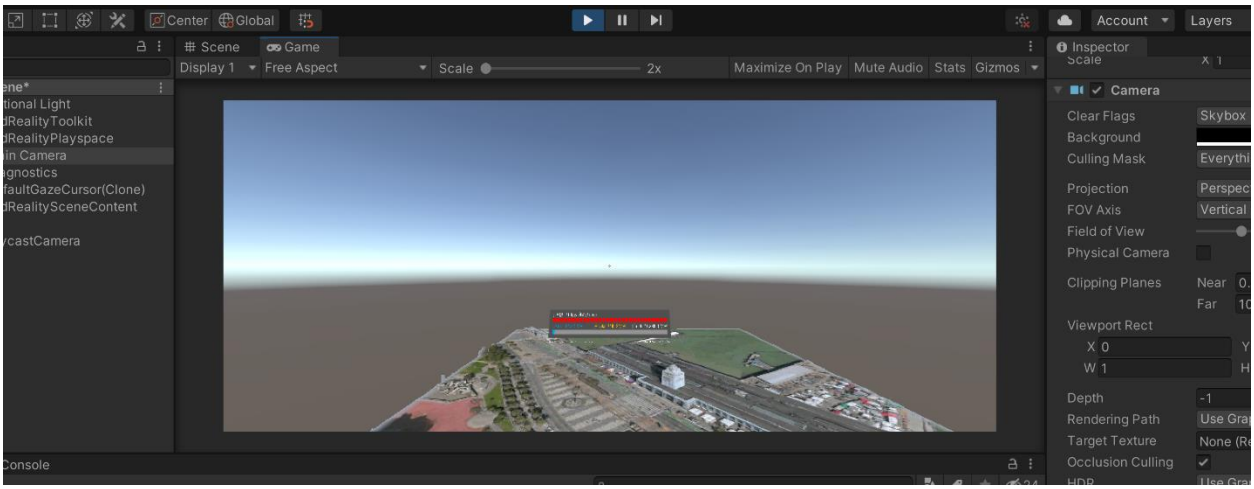


7. Specify **Zoom Level** to 17.5
8. Position the Map so it can be seen by the camera.

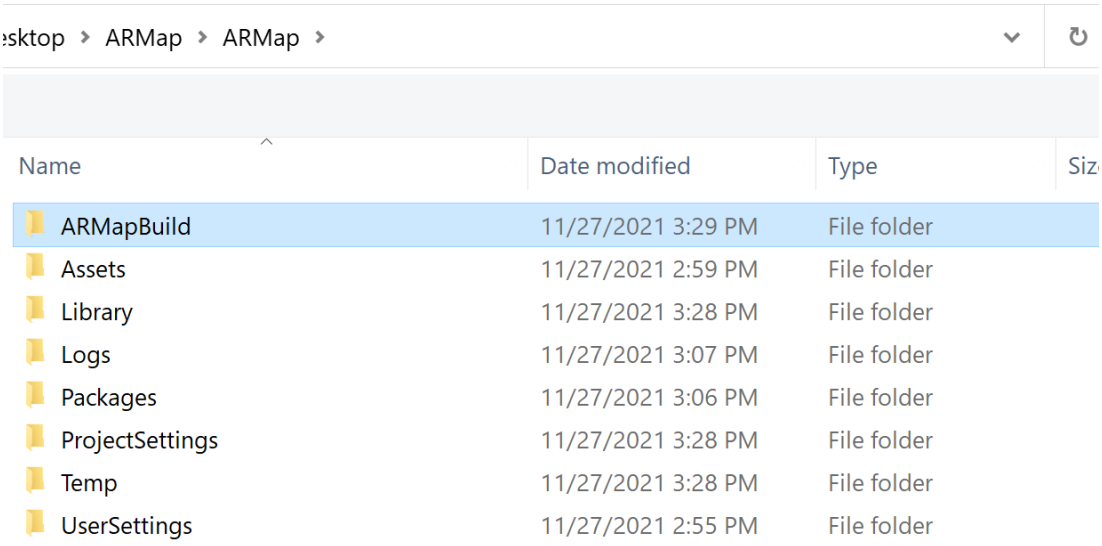




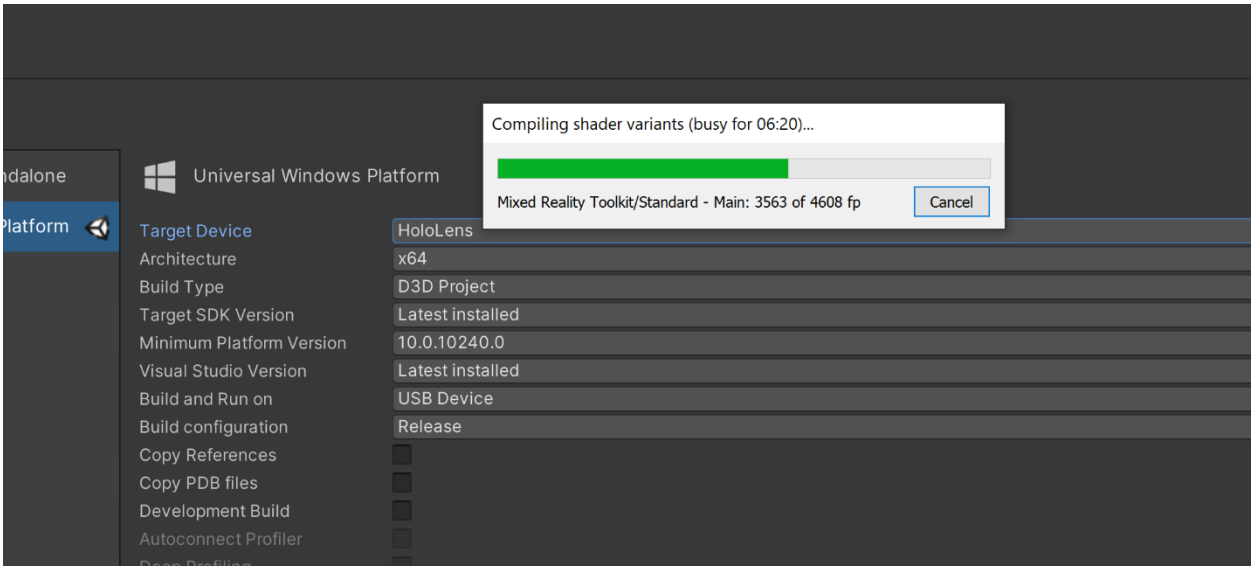
9. Press **Play**



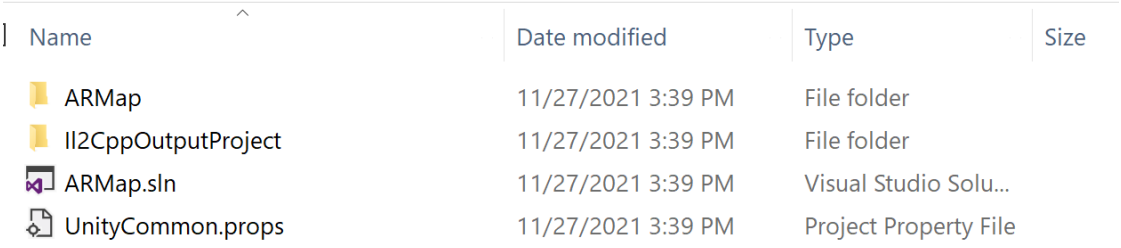
10. Next Build your project by going to File > Build Settings and click Build
11. Create a folder **ARMapBuild** and **Select Folder**



We wait for build to finish



Once the build finishes, we have the following build folder:



Name	Date modified	Type	Size
ARMap	11/27/2021 3:39 PM	File folder	
Il2CppOutputProject	11/27/2021 3:39 PM	File folder	
ARMap.sln	11/27/2021 3:39 PM	Visual Studio Solu...	
UnityCommon.props	11/27/2021 3:39 PM	Project Property File	

Using Visual Studio to debug and deploy

Visual Studio is your go-to tool for debugging and deployment.

Enabling Developer Mode

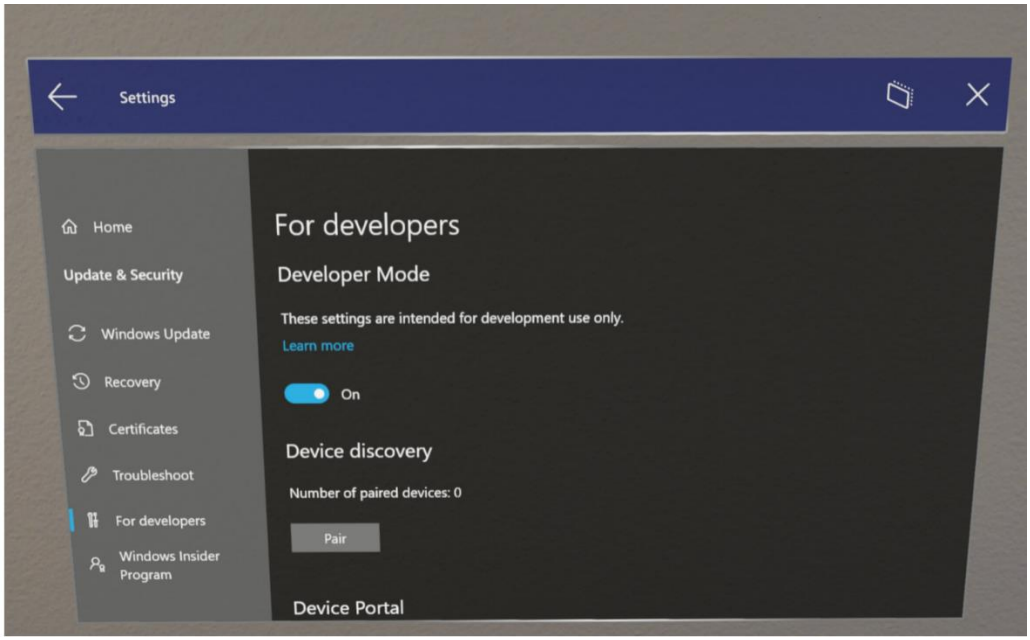
Start by enabling **Developer Mode** on your device, so Visual Studio can connect to it.

HoloLens. Using the Windows Device Portal

The Windows Device Portal for HoloLens lets you configure and manage your device remotely over Wi-Fi or USB. The Device Portal is a web server on your HoloLens that you can connect to from a web browser on your PC. The Device Portal includes many tools that will help you manage your HoloLens and debug and optimize your apps.

This documentation is specifically about the Windows Device Portal for HoloLens.

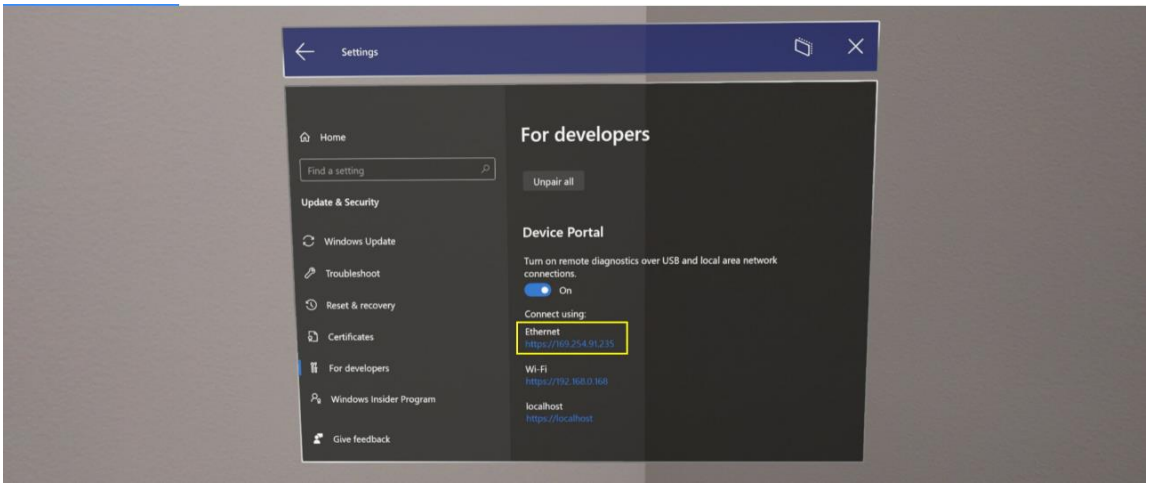
1. Power on your HoloLens and put on the device.
2. Use the Start gesture for HoloLens2 or Bloom on HoloLens (1st Gen) to launch the main menu.
3. Gaze at the **Settings** tile and do an air-tap gesture on HoloLens (1st Gen). You can also select it on HoloLens 2 by touching it or using a Hand ray.
4. Select the **Update** menu item.
5. Select the **For developers** menu item.
6. Enable **Developer Mode**.
7. Scroll down and enable **Device Portal**.
8. If you're setting up Windows Device Portal so you can deploy apps to this HoloLens over USB or Wi-Fi, select **Pair** to generate a pairing PIN. Leave the Settings app at the PIN popup until you enter the PIN into Visual Studio during your first deployment.



Connecting over USB

1. If your HoloLens 2 is running Windows Holographic version 21H1 or higher, go to 'For developers' in the Settings app and make sure that 'Device discovery' is toggled ON.
2. Connect your HoloLens 2 to your PC with a USB-C cable.
3. Find your UsbNcm IP. There are a few ways to do this:

- In the Settings app on the device (This method only works for HoloLenses running Windows Holographic version 21H1 or higher, with 'Device discovery' toggled ON.)
    1. Go into the Settings app on the device.
    2. Go to "Update & Security" > "For developers." This is the same place you enabled Device Portal.
    3. At the bottom of the page, copy your **Ethernet** IP address. This is your UsbNcm IP.



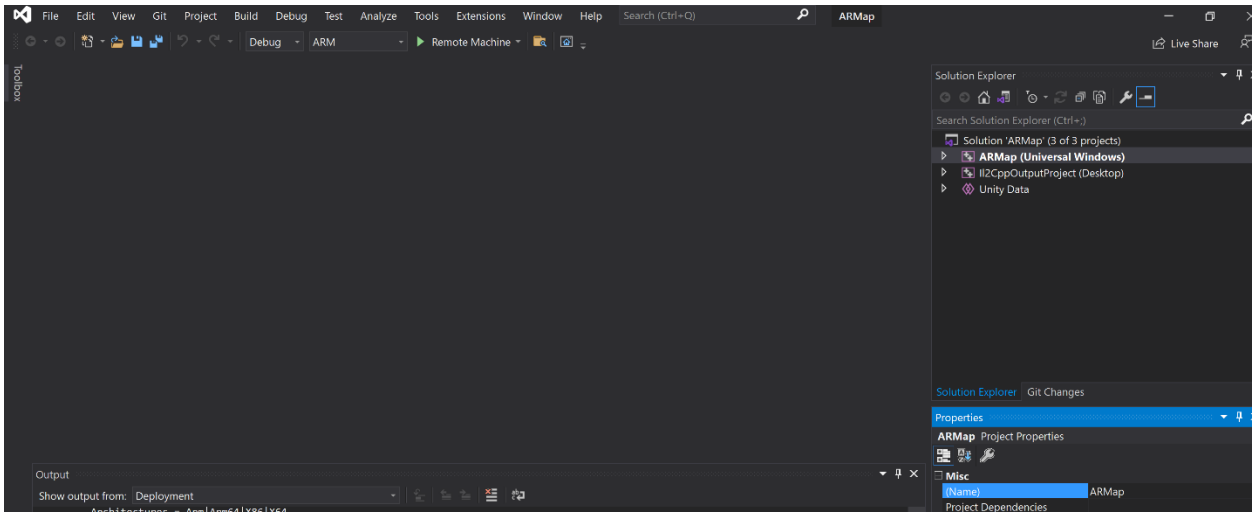
- In Device Portal
    1. On your device, open Device Portal using your HoloLens' WiFi address. If you don't know your HoloLens' WiFi address, you can use the voice command "What's my IP address?"
    2. Go to System > Networking
    3. On the far-right side of the page in the "IP Configuration" panel, locate the section that starts with "Description: UsbNcm Function."
    4. Your UsbNcm IP is the "IPv4 address" line. You can copy the address or just click on the address - it is a hyperlink which will reopen Device Portal using the UsbNcm IP.

- In a command prompt
    1. In any command prompt, navigate to the bin<SDK version>\x86 folder where your Windows 10 SDK is installed, such as C:\Program Files (x86)\Windows Kits\10\bin\10.0.19041.0\x86.
    2. Type "winappdeploycmd devices" and press Enter.
    3. In the output, look for the entry where the Model/Name column is your HoloLens device name, such as HOLOLENS-xxxxxx. The UsbNcm IP is at the start of this line and will be an Automatic Private IP address in the form of 169.254.x.x. Copy this address.

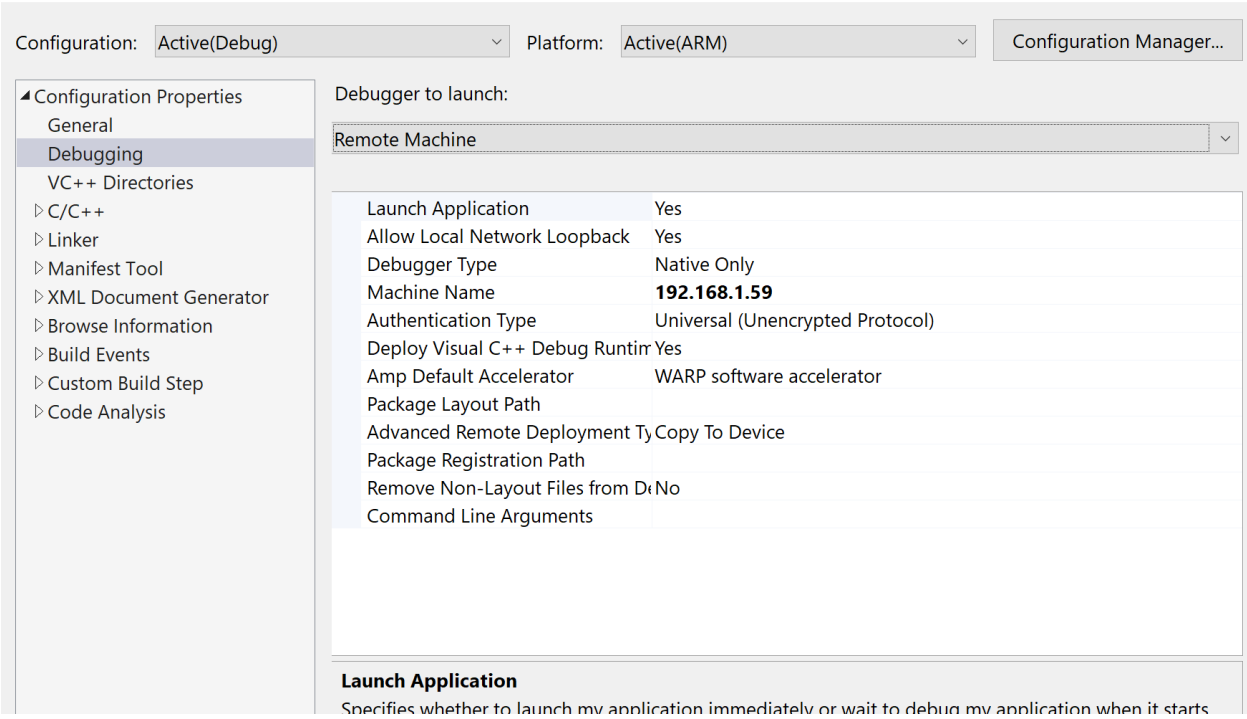
4. If you copied your UsbNcm IP, from a web browser on your PC go to https:// followed by your UsbNcm IP.

Deploying a HoloLens app over USB

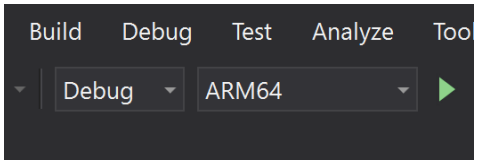
1. Open the build in Visual Studio.



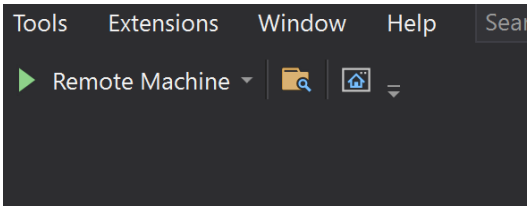
2. Go to Project > Properties > Debugging and set the IPv4 address that you have in HoloLens



3. Next Select your apps compilation options
    - Choose either **Release** or **Master**
4. Select your build configuration based on your device



5. Select **Remote Machine** in the deployment target drop-down menu



6. Build, deploy, and debug your app based on your needs

    - Select **Debug > Start debugging** to deploy your app and start debugging
    - Select **Build > Deploy** to build and deploy without debugging