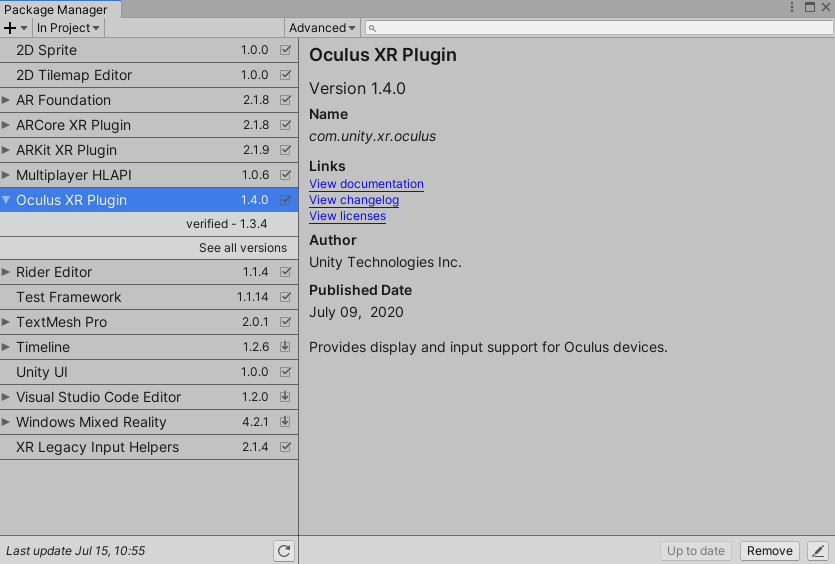
**Periodic Table Deployment to Oculus Quest**

**Using OculusXR Plugin**

1. The [Unity XR SDK Pipeline](https://docs.unity3d.com/Manual/XR.html)(**Open XR**) enables the use of Oculus Touch controllers and head tracking with the Oculus Quest.
2. The [Oculus Integration Unity package](https://assetstore.unity.com/packages/tools/integration/oculus-integration-82022) allows for the use of **hand tracking** and controller models with the Oculus Quest.

**Setting up the XR SDK Pipeline for Oculus Quest**

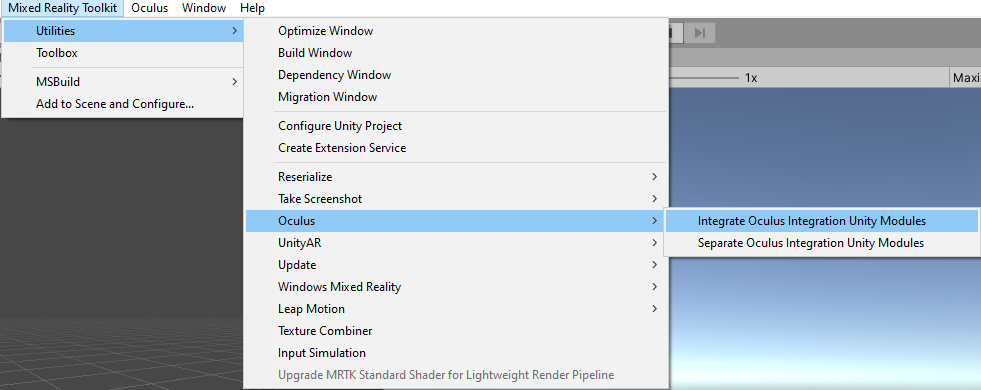
1. Ensure that the **Oculus XR Plugin** is installed under **Window --> Package Manager**

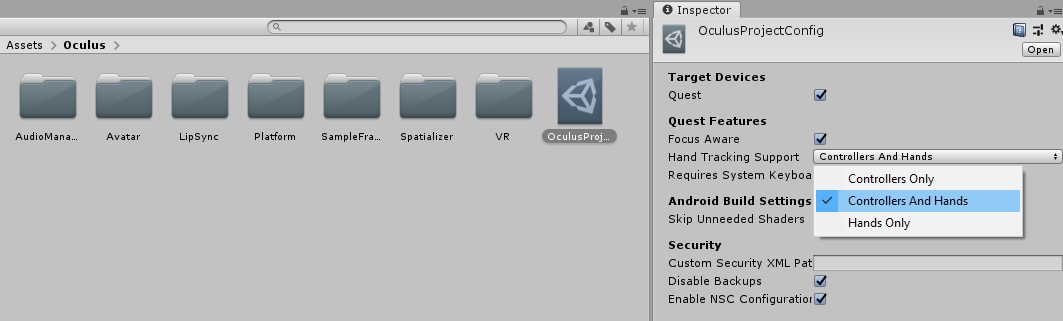


1. Make sure that the Oculus Plug-in Provider is included in your project by going to **Edit --> Project Settings --> XR Plug-in Management --> Plug-in Providers**



**Setting up the Oculus Integration Unity package to enable handtracking**

1. Download and import [Oculus Integration](https://assetstore.unity.com/packages/tools/integration/oculus-integration-82022) from the Unity Asset Store.
2. Navigate to Mixed Reality Toolkit > Utilities > Oculus > Integrate Oculus Integration Unity Modules. 
3. In the imported Oculus folder (It should be found at Assets/Oculus), there is a scriptable object called **OculusProjectConfig**. In that config file, you need to set HandTrackingSupport to "**Controllers and Hands**".

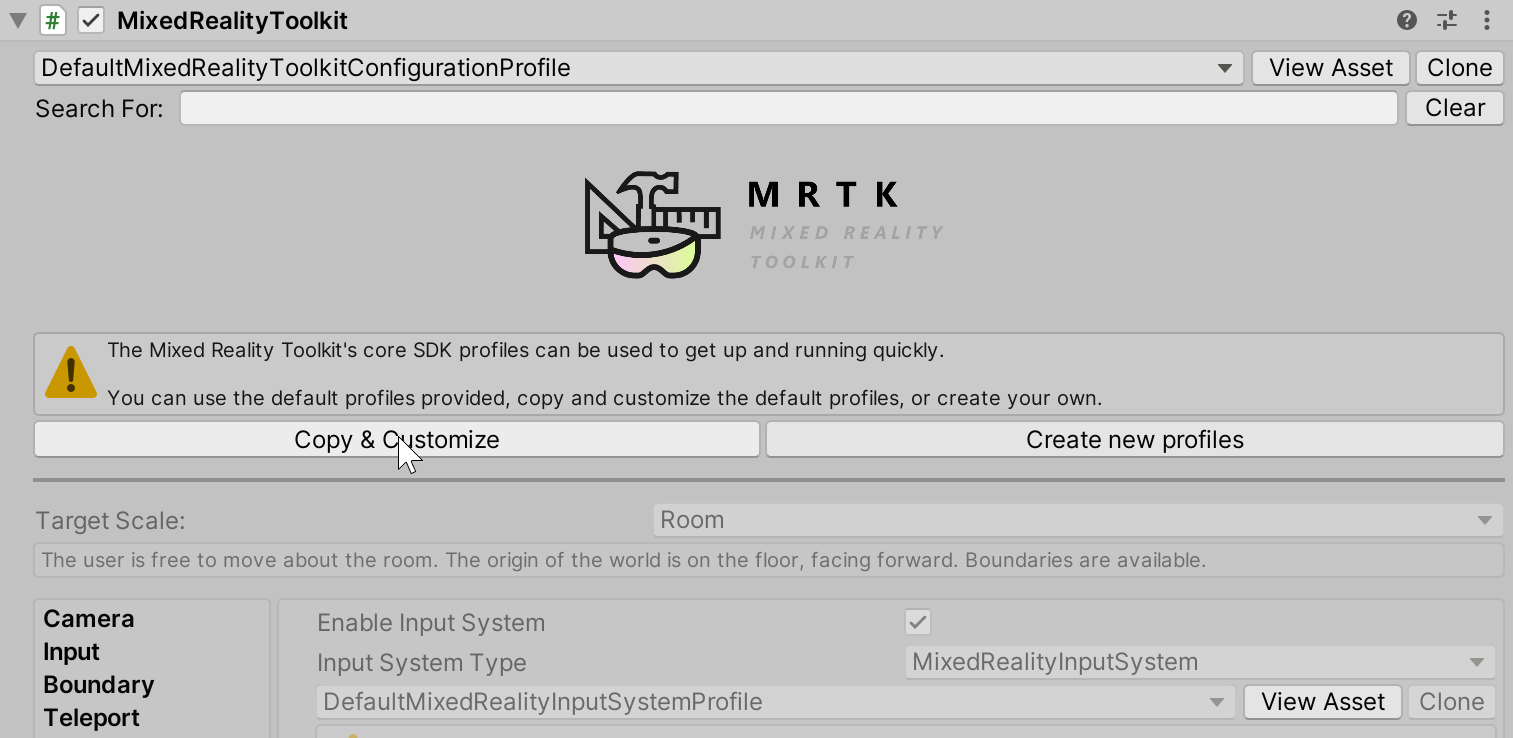


**Setting up the scene**

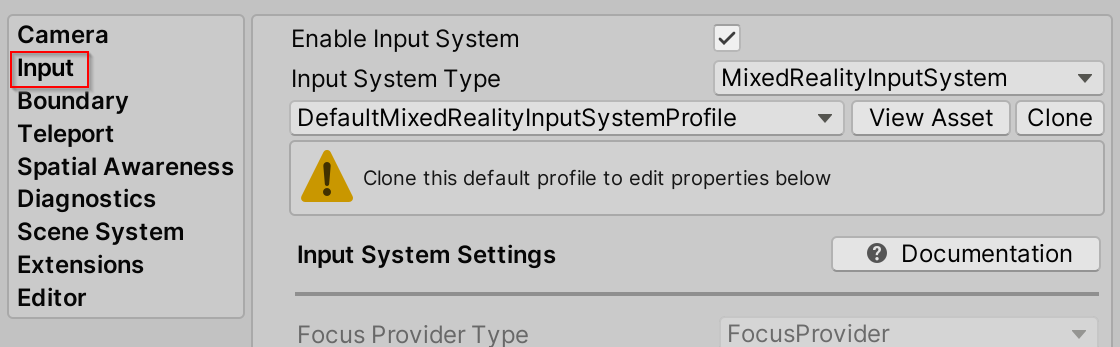
1. Create a new Unity scene or open a pre-existing scene.
2. Add MRTK to the scene by navigating to **Mixed Reality Toolkit** > **Add to Scene and Configure**.

**Using the Oculus XR SDK Data Provider**

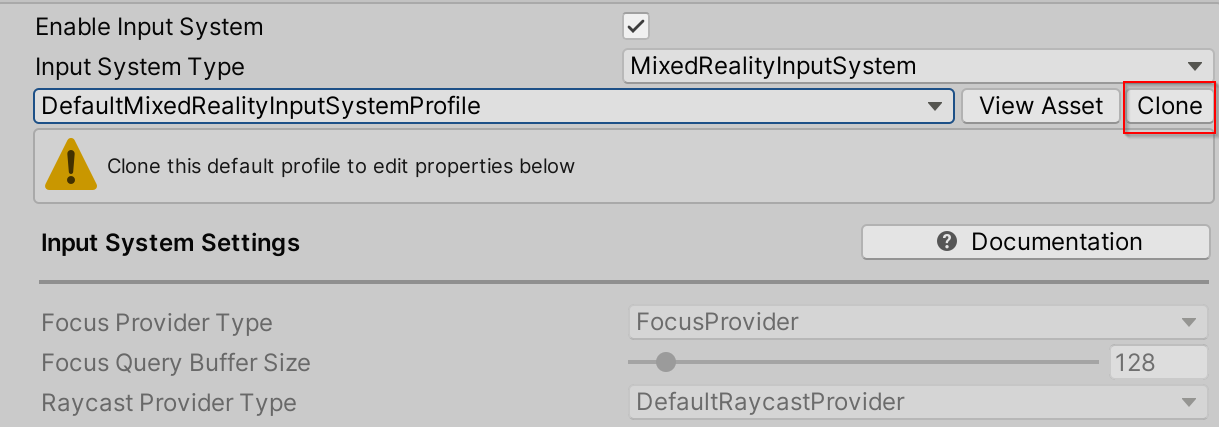
1. Configure your profile to use the **Oculus XR SDK Data Provider**
   * If not intending to modify the configuration profiles
     + Use any of the default MRTK profiles/ ObsoleteXRSDKConfigurationProfile, which are all configured across Unity's XR pipelines.
     + Go to [Build and deploy your project to Oculus Quest](https://docs.microsoft.com/en-us/windows/mixed-reality/mrtk-unity/supported-devices/oculus-quest-mrtk?view=mrtkunity-2021-05#build-and-deploy-your-project-to-oculus-quest).
   * Otherwise follow the following:
     + Select the MixedRealityToolkit game object in the hierarchy and select **Copy and Customize** to clone the default mixed reality profile.



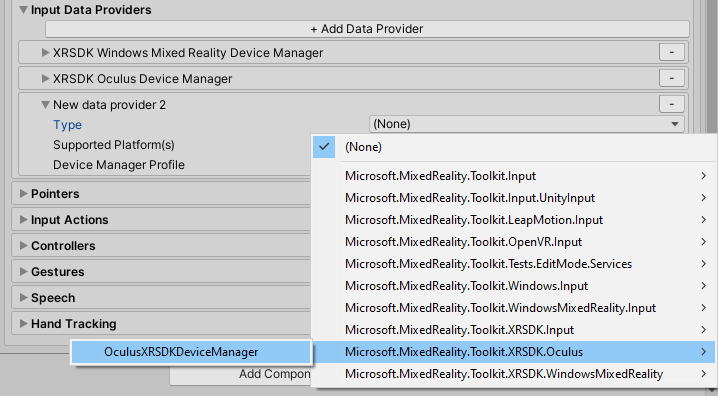
* + - Select the **Input** Configuration Profile.



* + - Select **Clone** in the input system profile to enable modification.



* + - Open the **Input Data Providers** section, select **Add Data Provider** at the top, and new data provider will be added at the end of the list. Open the new data provider and set the **Type** to **Microsoft.MixedReality.Toolkit.XRSDK.Oculus > OculusXRSDKDeviceManager**.

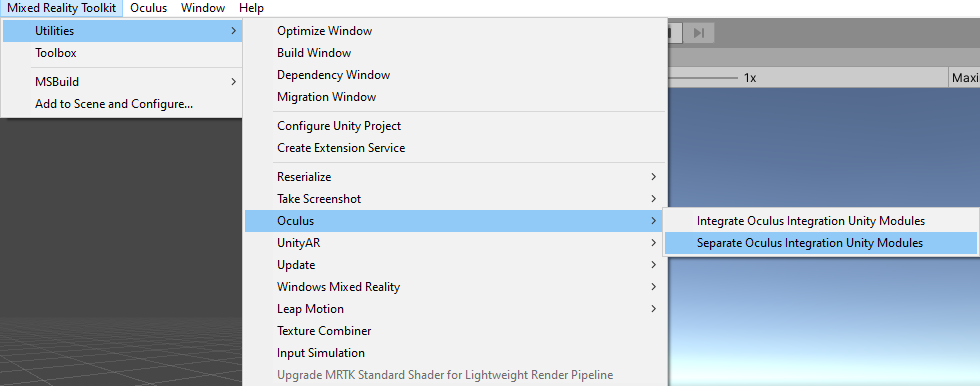


**Build and deploy your project to Oculus Quest**

1. Plug in your Oculus Quest via a USB 3.0 -> USB C cable
2. Navigate to **File > Build Settings**
3. Change the deployment to **Android**
4. Ensure that the Default device is selected as the applicable run device.
5. Select Build and Run

**Removing Oculus Integration from the Project**

1. Navigate to the Mixed Reality Toolkit > Oculus > Separate Oculus Integration Unity Modules



1. Let Unity refresh as references in the Microsoft.MixedReality.Toolkit.Providers.Oculus.asmdef and other files are modified in this step
2. Close Unity
3. Close Visual Studio, if it's open
4. Open File Explorer and navigate to the root of the MRTK Unity project
5. Delete the UnityProjectName/Library directory
6. Delete the UnityProjectName/Assets/Oculus directory
7. Delete the UnityProjectName/Assets/Oculus.meta file
8. Reopen Unity

**Common errors**

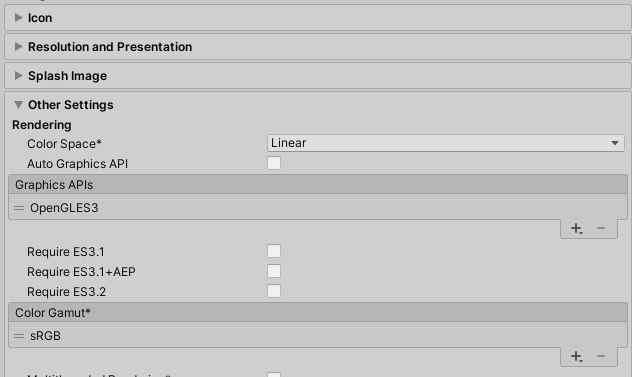
**Switch Between OpenXR and Legacy VRAPI**

With the Oculus Integration SDK v31 and higher, OpenXR is the default backend for the Unity version 2020 or higher, and optional in the Unity version 2019.4. You can easily switch between OpenXR and legacy VRAPI from the Unity Editor anytime.

1. On the **Oculus** menu, go to **Tools** > **OpenXR** > **Switch to Legacy OVRPlugin**.
2. Click **OK** to acknowledge the confirmation and click **Restart** to activate the legacy VRAPI backend.

# Linear color space is not supported

Switch the color space used by the player to "Linear", uncheck “Auto graphics API” and remove **openGLES2.** This can be done by navigating to Edit > Project Settings > Player in main Unity menu:



**Reference**: <https://docs.microsoft.com/en-us/windows/mixed-reality/mrtk-unity/supported-devices/oculus-quest-mrtk?view=mrtkunity-2021-05>

**Using OpenXR Plugin**

1. OpenXR Plugin
2. Oculus Integration

**Setting up the XR SDK Pipeline for Oculus Quest**

1. Ensure that the **Open XR Plugin** is installed under **Window --> Package Manager**
2. Make sure that the Oculus Plug-in Provider is included in your project by going to **Edit --> Project Settings --> XR Plug-in Management --> OpenXR**
3. In **OpenXR >** add **Oculus Touch Controller Profile >** select **Oculus Quest Support** and **OculusXR Feature**

**Build and deploy your project to Oculus Quest**