

# Mixed Reality

This book is the home for all AR, VR, MR knowledge regarding different soft- and hardware

- [Meta Quest 2 tips&tricks](#)
- [Using Vive without the headset for positional tracking](#)

# Meta Quest 2 tips&tricks

## Oculus Passthrough Tutorial Overview

This tutorial guides you through the process of setting up passthrough functionality for Oculus devices in a Unity 3D project. Passthrough allows users to see real-world surroundings through the device's cameras, enhancing the mixed reality experience. Here's a breakdown of the steps:

### Step 1: Begin with a New 3D Project

Start by creating a new 3D project in Unity. This will serve as the foundation for integrating Oculus passthrough features.

### Step 2: Import SDK from Oculus

1. Go to the [Oculus Developer Center](#) and download the necessary SDK.
2. Open Unity and your new project.
3. Navigate to **Assets > Import Package > Custom Package**.
4. Locate and select the downloaded .unitypackage file, then click **Import**.
5. When prompted to update the Oculus Utilities plugin, click **Yes** to ensure compatibility.  
Note: If asked to use OpenXR as the backend, click **Use OpenXR**. This is necessary for the Oculus XR Plugin to work with the Oculus Utilities Plugin.
6. If prompted to clean up old Interaction SDK assets, click **Show Assets**, then **Delete Assets**.
7. If prompted to update Spatializer plugins, click **Upgrade**, and then click **Restart**.

## Follow the Passthrough Tutorial

Continue with the Oculus Passthrough tutorial available [here](#). This tutorial provides detailed steps on implementing passthrough functionality in your Unity project.

## Additional Resources

- For support and community discussions, refer to the [Unity Forum thread](#).

## XR Plugins Management

In Unity's XR Plugin Management, make sure to enable the Oculus checkbox under the PC platform settings. This ensures that your project integrates with Oculus devices correctly. This setting is crucial for XR functionality on Oculus platforms.

By following these steps and consulting the provided resources, you should be able to successfully set up Oculus passthrough in your Unity 3D project.

## Sources:

### Build VR for Quest / PC

- [Tutorial Video](#) (tutorial going through the process of building on Quest and PC)
- [Circuit Stream Blog](#) (very simple has the basics but old!)

### Passthrough Quest

- [Tutorial Video - Standalone](#) (for running standalone on Quest)
- [Tutorial Video](#)
- [Forum Post](#) (Forum post pointing to the API doc)
- [API Reference](#) (api reference)

### PassThrough via link

- [GitHub Repository](#) (does not currently work with OpenXR)

### PassThrough Tutorials

- [Part 1 Tutorial Video](#)
- [Part 2 Tutorial Video](#)
- [Part 3 Tutorial Video](#)

### Quest 2 OSX

- [Blog - Developing for Oculus Quest 2 with MacBook Pro 16](#) (basics, maar niet erg informatief)

### Oculus Rift

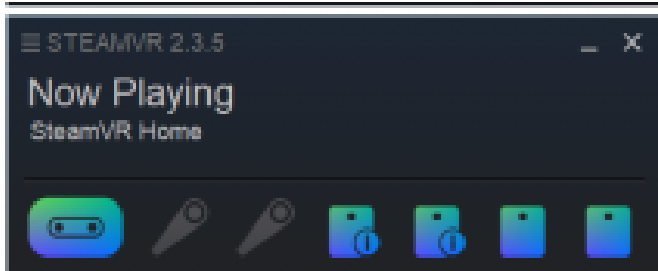
- [Circuit Stream Blog](#)

# Using Vive without the headset for positional tracking

In this setup we are using Vive Trackers in combination with multiple base stations (max 16!) to tracking positional data XYZ

Working without the headset seems less heavy on the GPU & the tracking seems a little more stable.

What you would usually see when connecting the headset & base-stations, in SteamVR:



1. Before adapting the files make sure the headset is disconnected/off.
2. Look up (make a copy as a backup...) & adjust the following files in explorer:

C:\Program Files (x86)\Steam\steamapps\common\SteamVR\drivers\null\resources\settings  
'default.vrsettings'  
(open in texteditor)

```
File Edit View
{
  "driver_null": {
    "enable": false,
    "loadPriority": -999,
    "serialNumber": "Null Serial Number",
    "modelName": "Null Model Number",
    "windowX": 0,
    "windowY": 0,
    "windowWidth": 2160,
    "windowHeight": 1200,
    "renderWidth": 1512,
    "renderHeight": 1680,
    "secondsFromVsyncToPhotons": 0.011111111,
    "displayFrequency": 90.0
  }
}
```

When using the headset "**enable**" : **false**, (standard setting)  
so to not use the headset change this to "**enable**": **true**,

Save the file!

C:\Program Files (x86)\Steam\steamapps\common\SteamVR\resources\settings  
'default.vrsettings'

(open in texteditor)

```
File Edit View

{
  "!!_WARNING_": {
    "000": "DO NOT EDIT THIS FILE TO CHANGE YOUR PERSONAL SETTINGS.",
    "001": "THIS FILE WILL BE REPLACED WHEN STEAMVR UPDATES.",
    "002": "These are the default values for settings which are not mentioned",
    "003": "in the user's personal steamvr.vrsettings file (use vrpsthreg.exe",
    "004": "to find the location of this file). User settings which match the",
    "005": "default in this file are not written to the user settings file, but",
    "006": "any setting below may be placed in steamvr.vrsettings to override",
    "007": "these defaults."
  },
  "steamvr": {
    "requireHmd": false,
    "forcedDriver": "",
    "forcedHmd": "",
    "displayDebug": false,
    "debugProcessPipe": "",
    "enableDistortion": true,
    "displayDebugX": 0,
    "displayDebugY": 0,
    "allowDisplayLockedMode": false,
    "sendSystemButtonToAllApps": false,
    "loglevel": 3,
    "ipd": 0.063,
    "ipdOffset": 0.0,
    "background": "",
    "backgroundUseDomeProjection": false,
    "backgroundCameraHeight": 1.6,
    "backgroundDomeRadius": 0.0,
    "environment": "",
    "hdcpl4legacyCompatibility": false,
    "gridColor": "",
    "playAreaColor": "",
    "showStage": false,
    "showSkydome": true,
    "drawTrackingReferences": true,
    "showGridCircles": true,
    "saveRecenterStandingZeroPose": true,
    "saveRecenterSeatedZeroPose": true,
    "activateMultipleDrivers": true,
    "usingSpeakers": false,
    "speakersForwardYawOffsetDegrees": 0.0,
    "basestationPowerManagement": 0,
    "ShowBaseStationPowerManagementTip": 1,
    "neverKillProcesses": false,
    "renderTargetMultiplier": 1.0,
    "maxRecommendedResolution": 8192,
    "supersampleScale": 1.0,
    "allowSupersampleFiltering": true,
    "supersampleManualOverride": false,
    "motionSmoothing": true,
    "forceFadeOnBadTracking": true,
    "mirrorView": 0,
    "showLegacyMirrorView": false,
    "mirrorViewDisplayMode": 1,
    "mirrorViewEye": 1,
    "mirrorViewGeometryMaximized": false,
    "showPerfGraph": false,
    "startMonitorFromAppLaunch": true,
    "startCompositorFromAppLaunch": true,
    "startDashboardFromAppLaunch": true,
    "startOverlayAppsFromDashboard": true,
    "enableHomeApp": true,
    "setInitialDefaultHomeApp": false,
    "CycleBackgroundImageTimeSec": -1,
    "retailDemo": false,
    "panelMask": true,
    "panelMaskVignette": true,
    "panelMaskVignetteWidth": 2.0,
    "inputBindingUI": true,
    "legacyInputRebinding": true,
    "debugInputBinding": false,
    "doNotFadeToGrid": false,
    "useNewChaperone": true,
    "enableLinuxVulkanAsync": false,
    "enableSafeMode": false,
  }
}
```

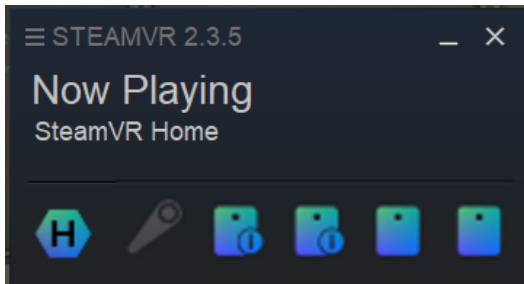
When using the headset **"requireHmd": true**, (standard setting) & **"activateMultipleDrivers": false**, (standard settings)

so to not use the headset change this to **"requireHmd": false,** & **"activateMultipleDrivers": true,**

Save the file!

3. Connect and turn on your headset.

Tracking is enabled, but your headset has no visual output. You should see this:



Remember! If you are using a shared headset in a space like the Blackbox or borrowed one from the loan, to ALWAYS reset these settings, so the headset can be used as expected.