

# Uploading to VRChat and Quest Compatibility

As Quest users become more and more prominent on VRChat, compatibility for the Oculus Quest is more and more sought after. Thankfully, making a world Quest compatible is not very difficult, though it can be a bit tedious.

Since switching to the Quest build is slow the first time, it's best to make edits and corrections to the PC version first. That and having a different hierarchy order between PC and Quest can mess with synced objects! So make sure that both worlds have synced objects in the same place in the hierarchy, and with the same names!

There are a few final optimizations that you should ensure you have before uploading:

Ensure that your lighting bake is accurate (you didn't edit any static objects since the last lighting bake and your light probes don't have errors)

Check VRWorld Toolkit's world debugger for errors

Bake occlusion culling

Use the mass texture importer feature of VRWorld Toolkit to compress any textures that you haven't already (click get textures from scene and then apply)]

You can compress your meshes too! To do this, click on the .fbx file and under mesh compression click medium (Click the circle with the ? as always for more info)

If you are using parallax maps, set the steps lower if you can, or disable them entirely if they are not super important to the material (they are relatively computationally expensive, and even Oculus recommends not using them for VR)

Ensure all of your pickupables are the child of one gameobject in the root of the hierarchy (a rigidbody component updates every frame, and any parent gameobjects also must update, so having only one parent gameobject makes our physics more optimized)

Have options to toggle pickupables and shadows on realtime light sources if you have any. The latter requires using two separate buttons with Udon and the script looks like this. If you're confused on how to set either up then watch my UI and Udon videos.

Before uploading you should test your world first, not just in CyanEmu but also in VRChat. To do this go to VRChat SDK>Control Panel and switch to the build tab, and then click build and test; you can increase the number of clients to simulate multiple players. Doing this not only gives you a perfect representation of how your world will function in VRChat but also gives us access to the build report in VRWorldToolkit, which we can use to find any uncompressed assets.

There's another add-on I'd like to tell you about in this series; comparatively it's optional but can come in handy. It's called VRBuildHelper, and its main feature is that it allows you to manage multiple branches of a world like version control, but it also comes with some helpful features (which I mostly use it for) which I will get into in a moment.

Once you're ready to upload, check VRWorldToolkit>World Debugger to see if there's anything you need to fix. Then position the main camera to where you want your world's preview picture to be taken.

Install VRChat ApiTools and then VRBuildHelper. Go to Window>VRBuildHelper, click "Set up Build Helper in this scene", and then click "Create new branch".

Open VRChat SDK>Control Panel and switch to the build tab, and click build and upload.

After the build has completed but before it uploads, Unity will be in game view and you need to enter the details of the world (which can be later edited on Vrchat.com). Go back into the scene view, and on the transform of the main camera click the 3 dots > copy component and then on the new game object VRCCAM's transform click the 3 dots on its transform component and paste component values. Repeat that process copying the post processing layer component from the main camera and then click the 3 dots on VRCCAM and click "paste component as new".

If you have VRBuildHelper installed you will also see two options next to the preview image on the upload screen: you will probably want to tick save VRCCAM position (the other option is for the branch management feature I talked about before).

At the bottom of the upload screen it says you can publish this world to labs; when you publish a new world, you will have to wait 7 days until you can publish another. However, you can push as many updates as you want to existing worlds. When your world is published to labs it can be accessed by other users if they have "show community labs" turned on in their settings. When your world is in labs there will be warnings when loading into it that it has not yet been approved, and may perform poorly or contain offensive content; in addition, other users will not be able to set it as their home world. To get your world approved and out of labs faster, people can visit it, spend time in it, and favorite it. Labs is an obtuse system and if you have any more questions about it read the FAQ or ask in the VRChat Discord. I don't answer questions in the comments section anymore. <https://docs.vrchat.com/docs/vrchat-community-labs>

If you go to Window>VRBuildHelper it will bring up its respective window, and you can see at the bottom it has two buttons next to autonomous builder: current platform (PC or Quest, whichever the editor is set to which you can see in the title of the window) and all platforms. This will allow you to upload your world with a single click, meaning you don't have to go through the upload screen every time you want to push an update! The catch is you have to upload it the normal way first.

Now let's focus on the Quest version of our world.

Switching from PC to Android versions usually takes a long time, but to make it faster, we can set up a local cache server. Go to Edit>Preferences>Cache Server and set the Cache Server Mode to

Local. If the space on your main drive is limited then you can use a custom cache location on a larger drive. Now, the first time we switch from PC to Android and vice versa will be slow, but subsequent switches will be much, much faster.

There is one last tool we should import called EasyQuestSwitch. Once opened and set up, you can add gameobjects and assets to automate changes between the two platforms. Since Quest does not support post processing, you can add a new slot and drag your post processing volume into it, then uncheck Active on Quest. In addition, make slots for all your materials using the Standard shader. Then in the project tab search bar, type standard lite, and drag the shader file into the Quest shader slot.

In the VRChat SDK window, there is a button to switch to Android (for Quest), but this has been reported to have bugs so we will switch by going to File>Build Settings (Ctrl+Shift+B), selecting Android and then clicking Switch Platform. This will take a while on the first switch, but subsequent switches will be faster.

Ensure that you aren't using shaders that are completely incompatible with Quest (like the aforementioned water shaders with a grabpass).

Quest worlds have a size limit of 100MB. If it goes over the world will not upload. The quickest ways to fix this are to compress/lower the resolution of textures/lightmaps/meshes/audio files or to use a larger ATSC block size for textures. All of that can be done with just a few clicks inside of Unity.

Open VRWorldToolkit and check to ensure there aren't any problems.

Finally test the world using CyanEmu to ensure that everything works. If it's all good, you can now upload the world to Quest!

If you have a Quest, test out the world yourself, if not, get a Quest tester to look for any issues. If everything works then congratulations! You did it! You now have the knowledge you need to make your own optimized, Quest compatible VRChat worlds!

---

Revision #3

Created 2022-02-19 23:06:12 UTC by Iconoclass

Updated 2022-05-23 23:25:33 UTC by Iconoclass