

Hydroform Ocean System Manual

Quick Start

Add the "HydroMultiCamComp" component to any cameras that need to render the ocean. Even if there is only one camera. Also add the UnderwaterFilter component to the camera(s) if you want to render underwater fx (attach the "underwater" shader to it).

Then find Ocean.prefab located in the root Hydroform directory and drag it into the Scene or Hierarchy window. The ocean should be immediately visible at a height of 0 on the y axis. If you see a black or purple square, see the Troubleshooting section.

Note that you cannot change the position or orientation of the ocean other than raise or lower it. To raise/lower the ocean height, find the 'Water Height' field under 'Wave Settings' in the Hydroform Component attached to the prefab object.

Note - to see the ocean animate in the editor, make sure 'Animated Materials' is checked in the toggles located at the top of the Scene window.

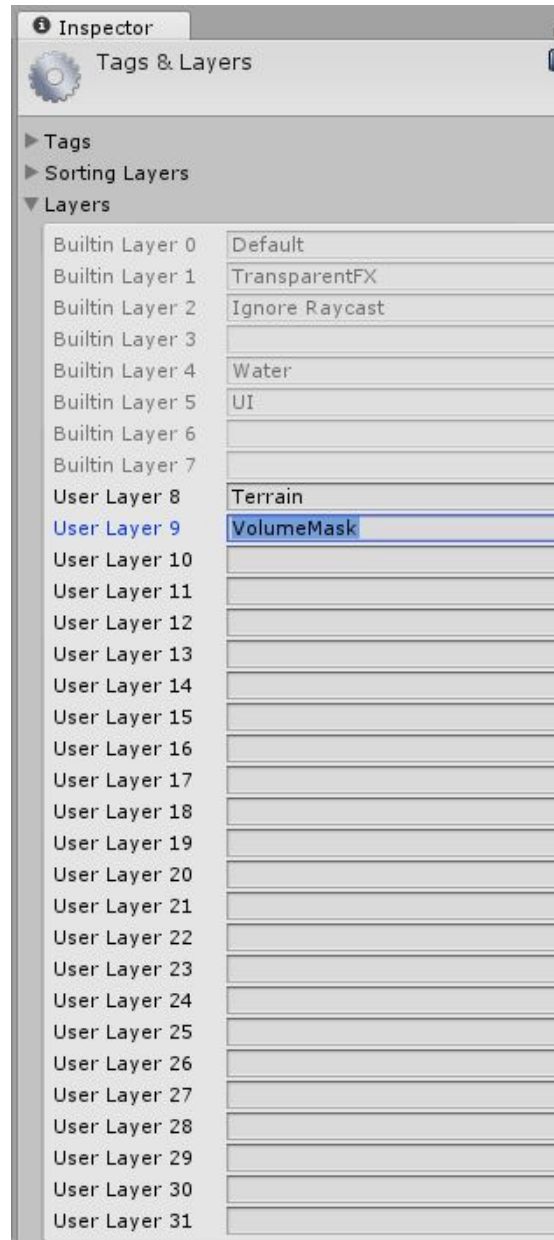
Setting up the tropical island demo

To see the palm trees, they will need to be loaded from Standard Assets. Download the Standard Assets package if this wasn't already done on Unity installation. Go to Assets/ImportPackage/Environment. Click 'None' at the bottom of the window. Scroll down to Environment/SpeedTree/Palm and click the 'Palm' folder so that only Palm and its subdirectories are selected. Click the 'Import' button at the bottom left of the window. Sometimes the trees do not show up until you have run the scene in game mode.

Lightmaps will need to be generated. Load the TropicsIsland demo and then go to Window/Lighting in the top menu. Click the 'Scene' tab and click the 'Build' button at the bottom of the window. It can take a few minutes to generate the lightmaps, be patient.

Setting up the VolumeTest demo

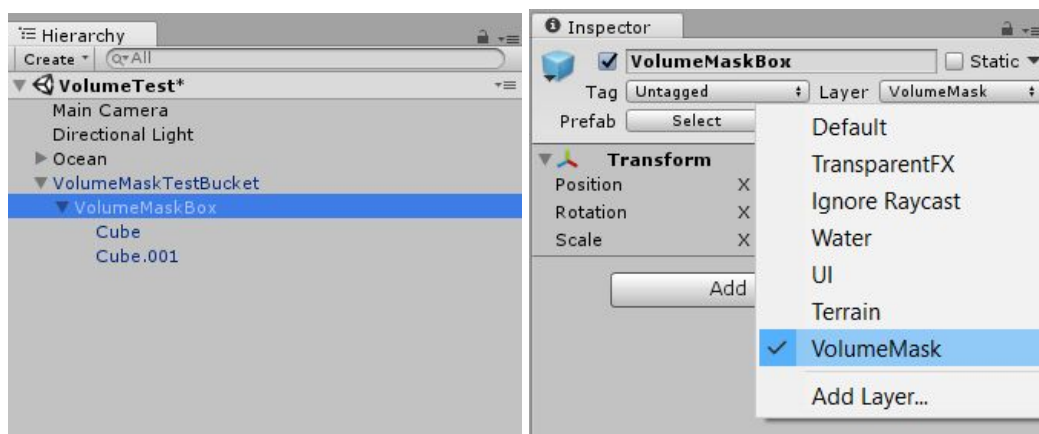
In order for the VolumeTest demo to work, a VolumeMask layer needs to be added. To add the layer, click the “Layers” dropdown in the upper right corner of the editor and click “Edit Layers” at the bottom of the dropdown.



Type in “VolumeMask” into one of the empty User Layer slots. If it is assigned to User Layer 9, that should be all that is necessary to get the demo running. If it doesn’t work, or User Layer 9 is already in use, you will need to do the following step of assigning the VolumeMask layer to the volume test box.

Find the VolumeMaskTestBucket object in the scene and select the next object down in the hierarchy, “VolumeMaskBox”. Then apply the VolumeMask layer to it via the inspector. It will ask if you want to apply the layer to all the children as well, which you do, so select “yes”.

The last thing is to set the culling mask on the Main Camera such that the VolumeMask layer is NOT checked.



That should be all that is necessary, the Volume Masking only works in Game mode, so start that mode to see it in action.

VR

Oculus:

If you are using the standard oculus camera rig, make sure to install the HydroMultiCamComp and UnderwaterFilter scripts onto the CenterEyeAnchor and NOT the LeftEyeAnchor or RightEyeAnchor.

SteamVR:

If you are using the SteamVR camera rig [CameraRig], make sure to install HydroMultiCamComp and UnderwaterFilter scripts onto the 'Camera(eye)' and NOT 'Camera(head)' or the ocean will not appear.

Known issues in VR:

Volume Masking does not work properly

Dynamic reflections do not work properly

Performance Tuning

The Hydroform Ocean System gives a high degree of control over how detailed the waves are vs. how fast the system can render them.

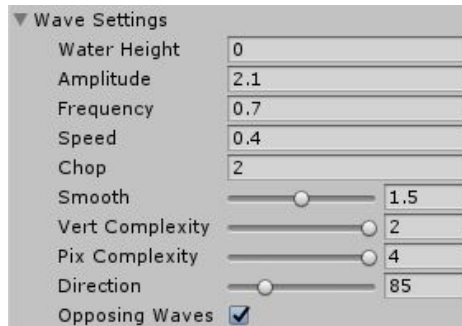
The most important controls are 'Vert Density' and 'Patch Size' located under 'Wave Construction'. Vert Density controls how dense the vertices are in the grid meshes. Higher densities result in more detailed wave geometry, but also more meshes and lower performance.



'Patch Size' refers to the patch of grid meshes that are rendered in front of the camera. A larger patch size will create more grid meshes over a larger distance in front of the camera before they fade into the flat brim mesh. A larger patch size improves the impression that the waves extend all the way to the horizon, but also creates more meshes and lowers performance.

Lowering the frequency and amplitude of the waves (see 'Wave Settings'), will smooth the wave geometry out and require a lower vertex density for high quality looking waves.

For low-end and mobile hardware, the largest performance bottleneck is the pixel shader complexity (see 'Pix Complexity' under 'Wave Settings'). Setting it to lower values will increase performance, but lose detail.

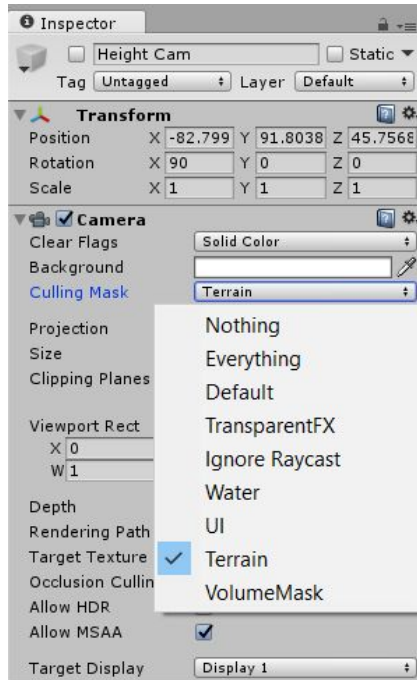


Reflections are also important when looking at performance. Planar reflections look the best and are the most accurate, but they require that the scene be redrawn, so complex scenes can have a large performance hit. To improve performance on planar reflections, set a culling mask on the Reflect Cam (located under the Ocean prefab) so that it only renders major objects that will be seen in a reflection.

Screenspace reflections (SSR) use the existing rendered scene in the frame buffer to provide an approximate reflection of the scene. They are much faster than planar reflections on most hardware, but low-end devices may bog down under the more complex shader.



Another optimization is to set up a Culling Mask for the Height Cam (child of Ocean). For instance, if you set up the terrain as a Layer, and then set the Height Cam to only render the terrain Layer via a Culling Mask, it can avoid rendering much of a complex scene that is not affected by the ocean.



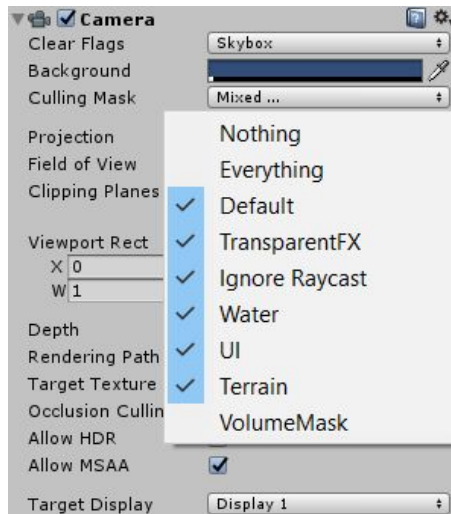
Volume Culling

Volume culling is using mesh geometry to define areas where the ocean should not be drawn. Examples would be the inside of a ship or an underground tunnel. To set up volume culling, first create a layer called “VolumeMask”:

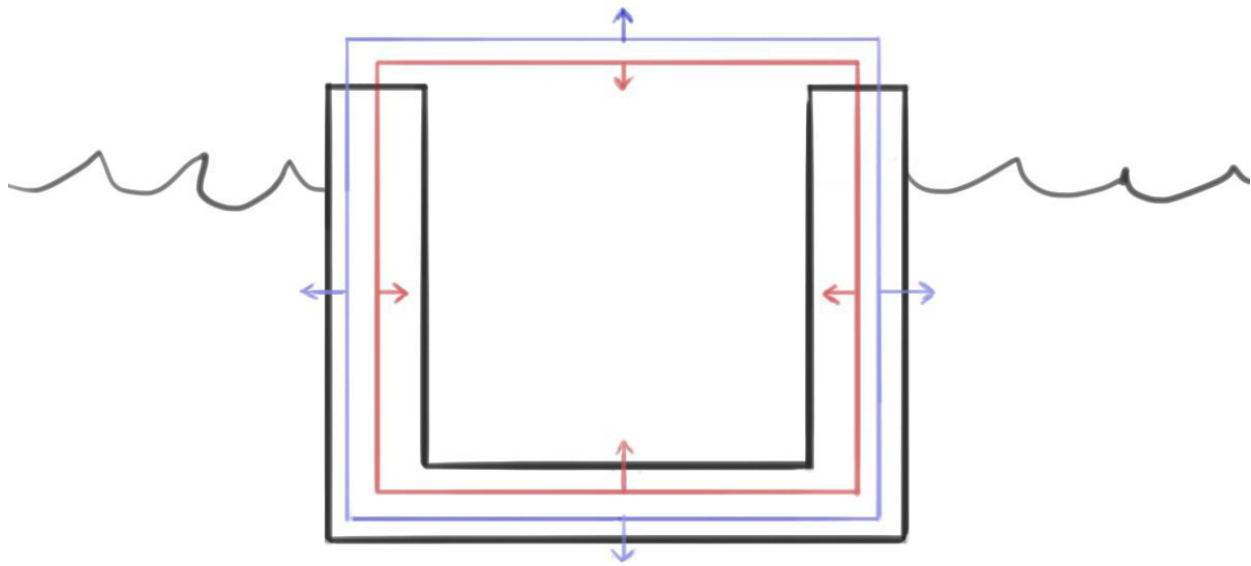


That layer will need to be applied to any geometry that will be used to cull water pixels. In addition to applying the VolumeMask layer, either the VolumeMaskMat or VolumeMaskInteriorMat material will need to be applied to volume culling geometry. The Culling

Mask for camera(s) with the ocean visible should uncheck the VolumeMask layer so that they don't render objects with that layer applied.



The VolumeMaskMat material should be applied to geometry that is used to cull the water surface. The VolumeMaskInteriorMat should be applied to geometry that is a) underwater and b) in a volume that the camera will be within. It essentially masks out underwater screen fx. Make sure that geometry that is used with the VolumeMaskInteriorMat has normals facing inwards, towards where the camera will be.



Bucket Mesh

Volume Cull Mesh

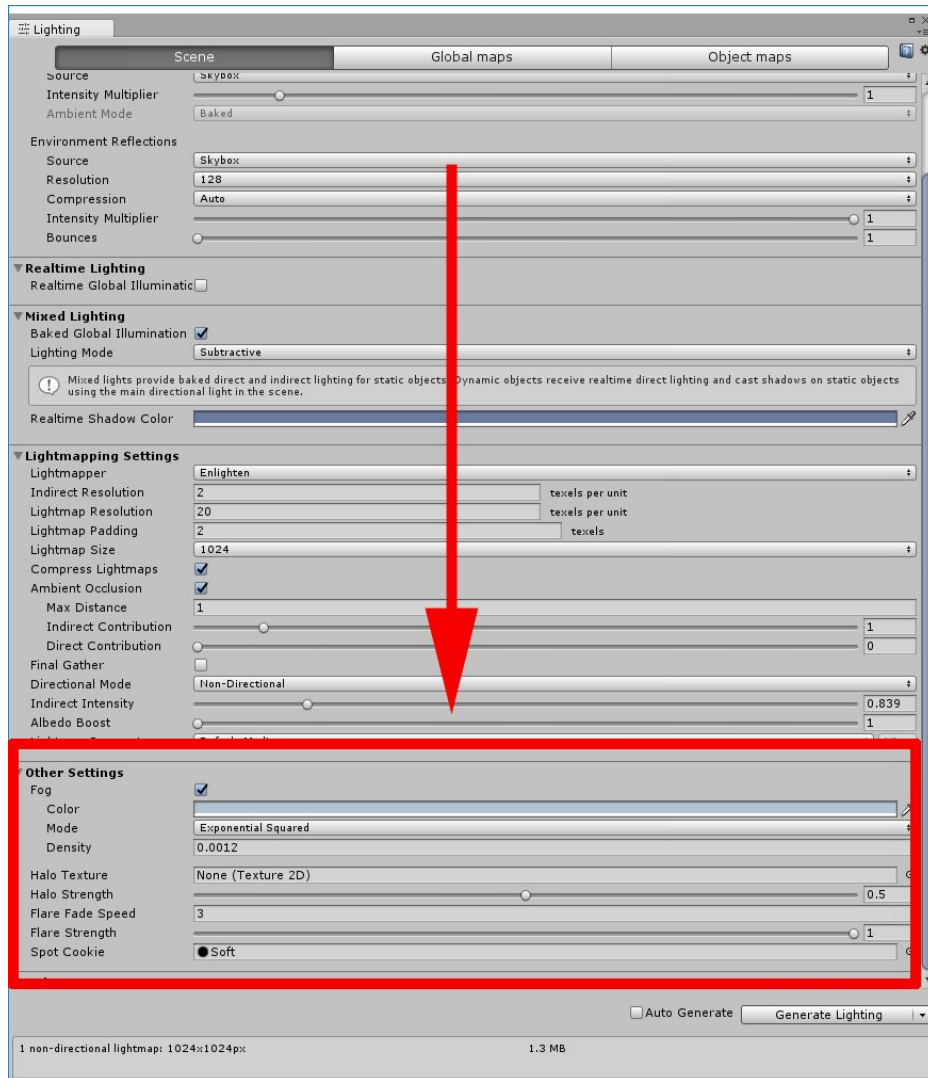
Volume Cull Interior Mesh

In the example above, the “bucket” mesh is just a typical mesh with a standard material applied. The Volume Cull Mesh is a box with the normals facing outwards and the VolumeMaskMat material applied. It is not visible to the game camera and removes the ocean from view when seen from above. Note that its height covers the complete amplitude range of the waves. Also note that it completely covers the Volume Cull Interior Mesh. The Volume Cull Interior Mesh is there to block out the underwater fx if the camera goes into the “bucket” and below water level. It has the VolumeMaskInteriorMat material applied and normals pointed inwards. Both the Volume Cull Mesh and the Volume Cull Interior Mesh have the VolumeMask layer applied to them. With all of these meshes set up, no water will appear within the bucket, even if the camera goes below the water level.

Another way to remove underwater fx when the camera goes below the water level is to modify the ‘enableUnderwater’ variable in HydroformComponent via script. This may be more convenient than creating a lot of geometry to mask out an underwater tunnel for example.

Fog Settings

Hydroform Ocean System uses fog settings from the Unity project. To adjust the fog go to Window/Lighting and click the 'Scene' tab. The 'Fog' drawer is near the bottom of the Lighting window. Check the Fog box to turn on fog, and then changes made to the color and density will affect the ocean. The ocean uses Exponential Squared fog calculations, so make sure that is selected as the fog mode so that the ocean fog matches other geometry in the scene.



Deferred Rendering Path

In order to get fog working properly with Hydroform when using the deferred rendering path, you must do the following:

1. Install the GlobalFog Image Effect from Standard Assets (version 5.4)

2. Replace the "GlobalFog.shader" link in the GlobalFog component with the "HydroformFog.shader" located in Hydroform/Resources directory.

Note that you will need at least Unity 5.4 for the shader to compile

Known Issues

These are on my list of things to fix, if they can be fixed:

- Hydroform dynamic reflections (SSR and planar) will not work properly with VR. This cannot be fixed until Unity opens up API (might be able to fix with 2018)
- On some video cards, there is a pixel or two of noise along the horizon line. As a temporary fix, I recommend increasing fog density so that it covers the noise.
- The height query does not work properly close to shore (doesn't take into account wave damping)
- The height query is not always accurate when the water height is far away from 0
- Volume culling does not work in VR
 - This is because it is 2 pass and involves a screenspace filter. Not easily solved.

Troubleshooting

Problems with the Hydroform Ocean System most typically arise from changing platform settings in Unity. Usually the problem can be solved simply by clicking the 'Apply' button under 'Wave Settings', Going into Game mode and back out, or by restarting Unity.

Ocean doesn't render

Add the "HydroMultiCamComp" component to any cameras that need to render the ocean. Even if there is only one camera. Also add the UnderwaterFilter component to the camera(s) if you want to render underwater fx (attach the "underwater" shader to it).

No water is appearing in the editor, but the water is showing fine in Game mode.

Sometimes the scale can get screwed up if you focus on a large object like terrain in the editor. To fix, double click on a small object in the Hierarchy panel of the editor.

Shader warning: "Gradient instruction used in a loop with varying iteration..."

Solution: Find the line that it's complaining about and replace the texture fetch with a constant color. The warning should disappear. Then revert the change back to the texture fetch and it should proceed with no warning/error.

Example:

the offending line:

```
return texCUBE( _Cube, reflectEye );
```

replace with:

```
return float4( 1, 1, 1, 1 );
```

and then put the original texCube call back.

Shader error: "unable to unroll loop, loop does not appear to terminate in a timely manner.."

Solution: This occurs on some platforms or if D3D9 is targeted. This is always located in getHeightFromTex() in WaterFunc.cginc. Replace the line:

```
for(int i = 0; i < numOctaves; i++)
```

with (or whatever the number is you want for pixel complexity):

```
for(int i = 0; i < 4; i++)
```

ShoreFX not working properly

Look at 'Height Cam' camera located under the Ocean prefab. Try setting 'Culling Mask' to 'Everything'. The script will then set it to everything but water automatically.

Water patch not clipping correctly in editor (works fine in game mode)

Exit Unity, remove the Library directory in your project root, and restart Unity.

Square shapes appear in the water when ShoreFX are on

Set up the Height Cam (child of Ocean) such that the Culling Mask only allows rendering of the terrain. You may have to create a new terrain Layer to use in the Culling Mask. This will also optimize the Height Cam rendering as it will only render the terrain.

Contact

support@xixinteractive.com

