

VOLUMETRIC FOG & MIST 2

Universal Rendering Pipeline Edition

Quick Start Guide



Contents

Introduction	3
Requirements	4
Setup Instructions	4
Demo Scene	5
Adding Volumetric Fog & Mist to your scene	6
Special Features.....	9
Orthographic Camera Support	9
Fog Voids	9
Native Lights	10
Point Lights	10
Terrain Fitting	11
Fog of War	12
Fog of War Editor	12
Transparency support	13
Volume fade in/out and Sub-Volume support	14
Scripting Support	16
Accessing fog properties	16
Creating fog volumes at runtime	16
Creating fog voids at runtime	16
Fog of War methods.....	17
Performance Tips.....	18
Known Issues / Questions	19
Support	20

Introduction

Thanks for purchasing!

Volumetric Fog & Mist 2 is an advanced full-screen post-image effect for Unity that adds realistic fog, mist, dust and clouds effects to your scenes.

This version of Volumetric Fog & Mist 2 has been created from scratch for Universal Rendering Pipeline. It won't work with other pipelines. Please use regular Volumetric Fog & Mist package for built-in pipeline.

Requirements

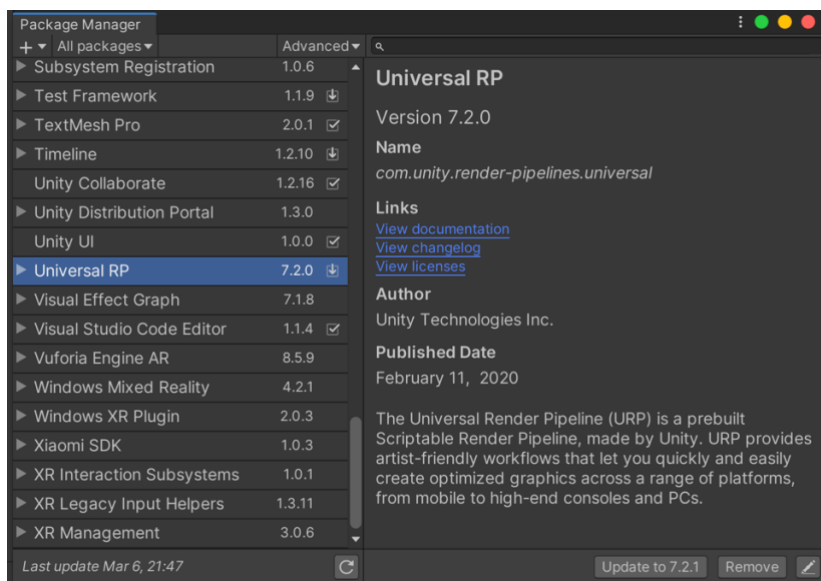
Volumetric Fog & Mist 2 for Universal Rendering Pipeline requires:

- Unity 2019.3 or later
- Universal Rendering Pipeline 7.1.8 or later
- Universal Rendering Pipeline asset must have “Depth Texture” enabled.

Setup Instructions

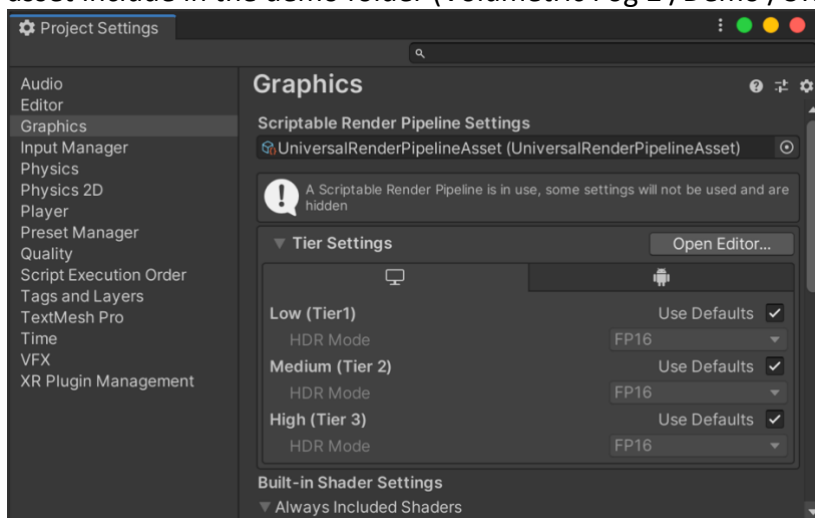
Step 1: Install Universal Rendering Pipeline

Go to Windows -> Package Manager. Select Universal RP (7.2.0 or later preferably) and click “Install”.



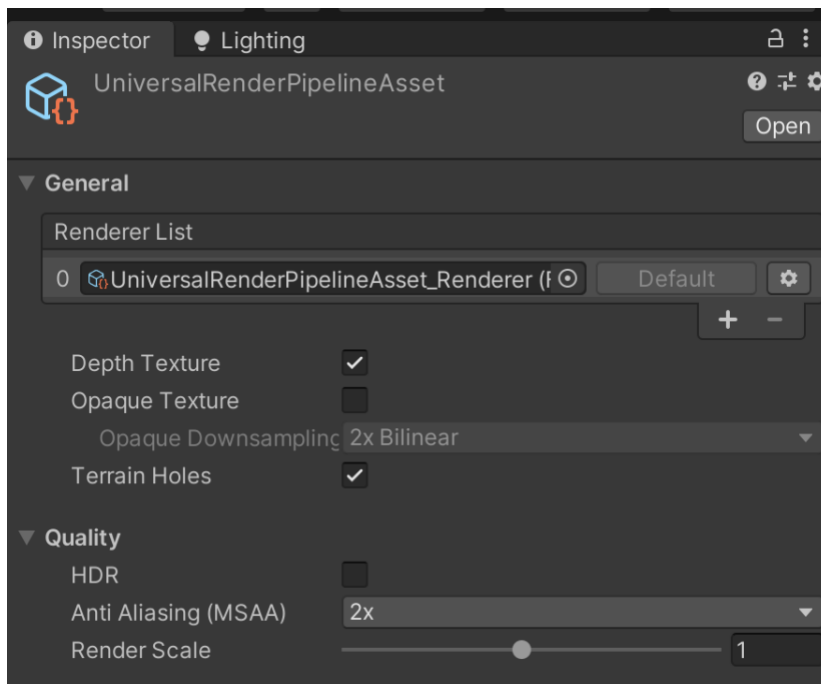
Step 2: Assign the Universal Rendering Pipeline asset

Go to Project Settings -> Graphics and assign an Universal Rendering Pipeline asset. You can use the asset include in the demo folder (Volumetric Fog 2 /Demo /URP Pipeline Settings folder).



Step 3: Configure Universal Rendering Pipeline asset

Double click the Universal Rendering Pipeline asset to show its properties. Then enable “Depth Texture”. Enabling MSAA is also recommended:



Important note!

Check both “Project Settings / Graphics” and “Project Settings / Quality” sections since you can define URP settings overrides in Quality levels. You need to enable Depth Texture option in all URP settings used in any Quality Level.

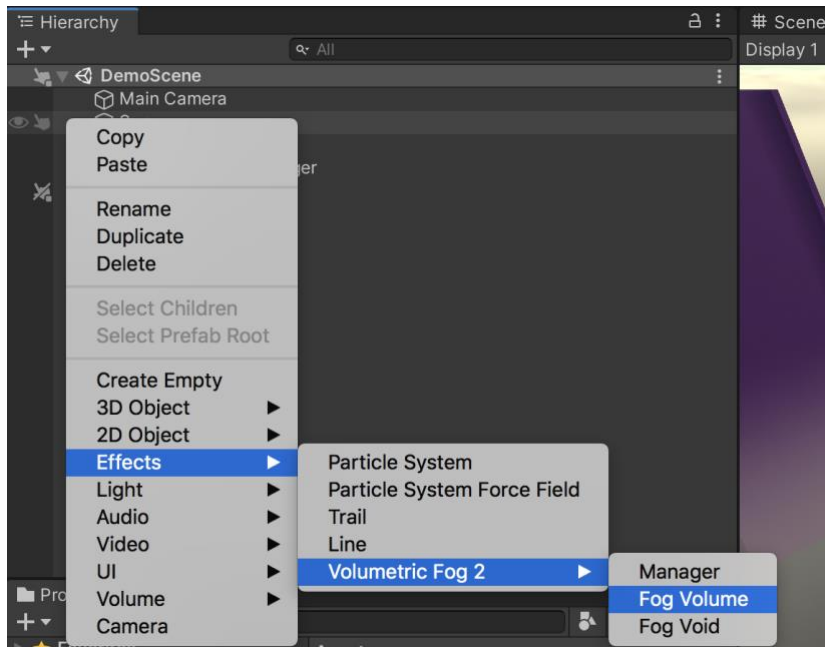
Now you can use Volumetric Fog & Mist 2!

Demo Scene

The asset includes a demo scene to let you quickly familiarize with the different features. Please make sure you have completed the Setup explained in previous section.

Adding Volumetric Fog & Mist to your scene

Step 1) Right click in the Hierarchy panel and select “Effects” -> “Volumetric Fog & Mist 2” -> Fog Volume:

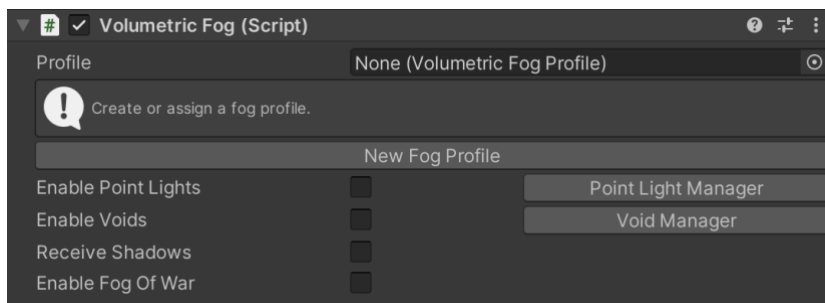


A Fog Volume is a boxed area that contains the fog effect.

Note: the first time a fog volume is added, a “Volumetric Fog Manager” gameobject is automatically created. This manager is responsible for finding point lights and void areas and send this data globally to all fog volumes.

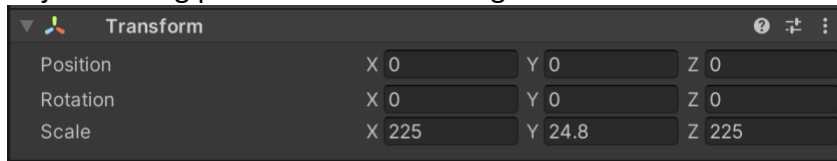
Step 2) Create or assign a fog profile

A Fog Profile contains many appearance options and can be reused among fog volumes. Click “New Fog Profile” or assign an existing profile.

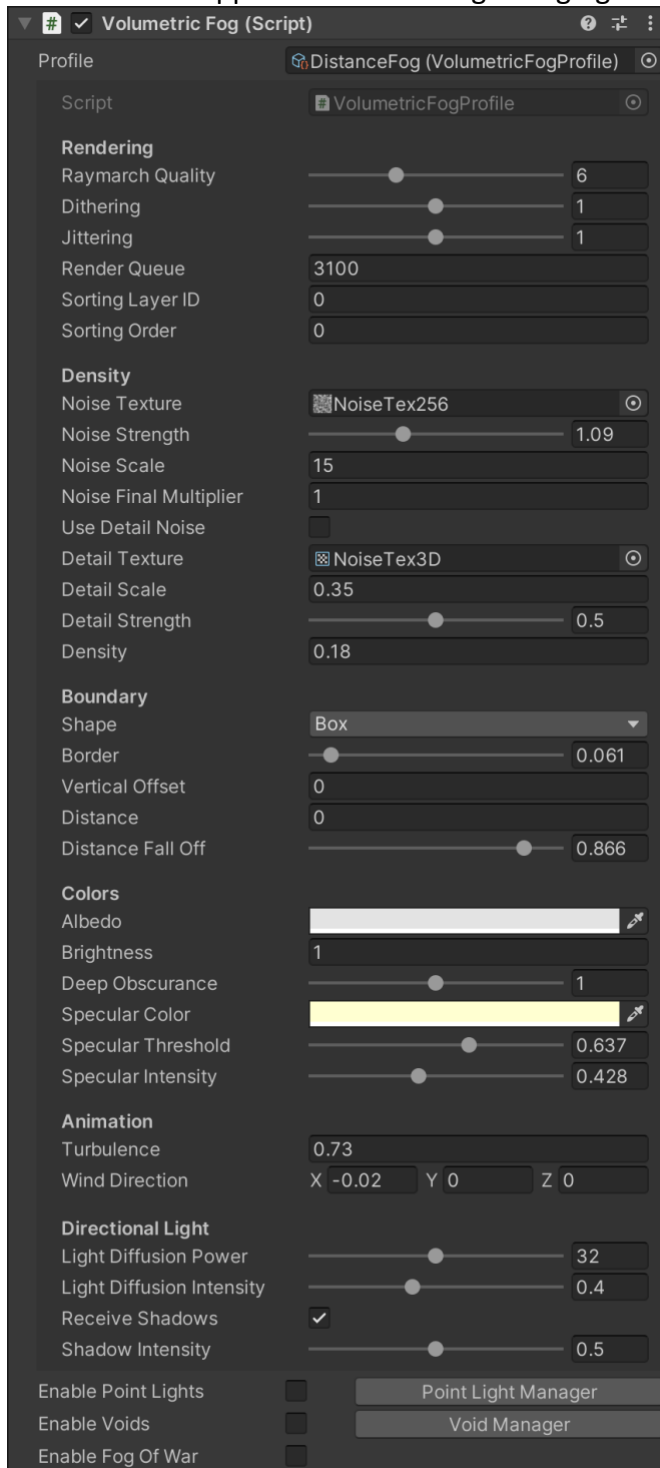


Step 3) Position/scale/customize fog properties

Adjust the fog position and scale using the usual Transform component:



Customize the appearance of the fog changing the profile values:



Most profile settings are self-explanatory. Default values are recommended for most cases:

- **Raymarch Quality:** represents the amount of samples used by the shader. A lower value uses less samples and improves performance.
- **Raymarch Min Step:** increasing the minimum step size can contribute to improved performance.
- **Dithering:** reduces banding artifacts when fog renders in front of solid surfaces.
- **Jittering:** reduces banding artifacts on fog edges.
- **Noise Texture:** you can use any other noise textures from the demo folders or use the Noise Generator tool (from Volumetric Fog Manager) to create a new noise texture.
- **Noise Strength:** defines how noise affects fog shape.
- **Noise Scale / Final Multiplier:** control the size and final amplitude of the noise.
- **Use Detail Noise / Detail Noise options:** combines regular noise with additional 3D noise to create more compelling fog effect.
- **Detail Noise Texture:** custom 3D noise texture (red channel is used only). You can find more noise and detail noise textures inside the demo folder.
- **Density:** defines the sample density. Use a low value to create subtle mist while a high value produces a thicker fog.
- **Border:** smooth fog volume edges.
- **Distance / Distance Fall Off:** the distance at which the fog starts.
- **Albedo:** base color of the fog, including transparency.
- **Brightness:** overall brightness of the fog.
- **Deep Obscurance:** makes the fog darker at the middle of the fog volume.
- **Specular Color/Threshold/Intensity:** simulates light speculars/reflections on the fog by the directional/main light.
- **Turbulence / Wind Direction:** animates the fog along the time.
- **Light Diffusion Power/Intensity:** controls the effect of the directional/main light when crossing through the fog in direction of the camera.
- **Receive Shadows / Shadow Intensity:** controls shadows over the fog.

Special Features

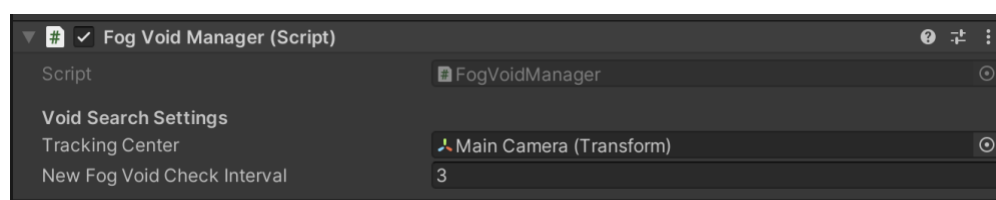
Orthographic Camera Support

Locate and edit **CommonsURP.hlsl** file and uncomment this line:

```
#define ORTHO_SUPPORT
```

Fog Voids

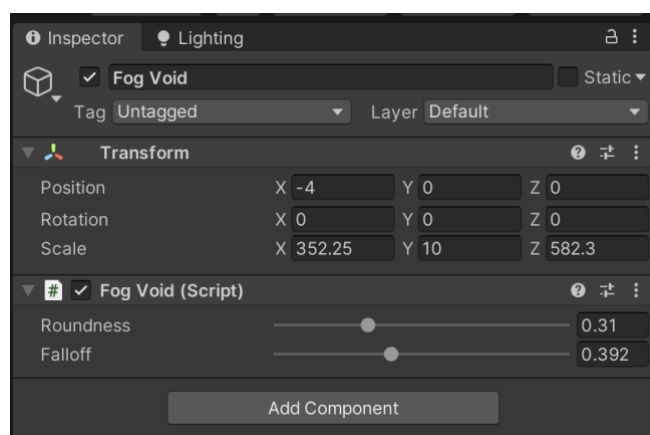
A fog void is a circular area which is not covered by fog. Up to 8 fog voids can be applied to a fog volume. The Fog Void Manager component is responsible for sending the fog void positions and sizes to the shaders:



The Fog Void Manager sorts fog voids by distance to the Tracking Center. You can adjust the time interval for looking new fog voids.

To add a Fog Void to the scene, right click in the Hierarchy and select “Effects” -> “Volumetric Fog & Mist 2” -> Fog Void.

You can **position / scale the fog void** anywhere in the scene. Also, you can parent a fog void around characters so they will not be covered by fog. Fog voids can also be grouped or parented. Use the **Roundness** and **Falloff** slider of the Fog Void component to customize the appearance of each void.



Check this video for an example: <https://youtu.be/3-Y26lqErHg>

Native Lights

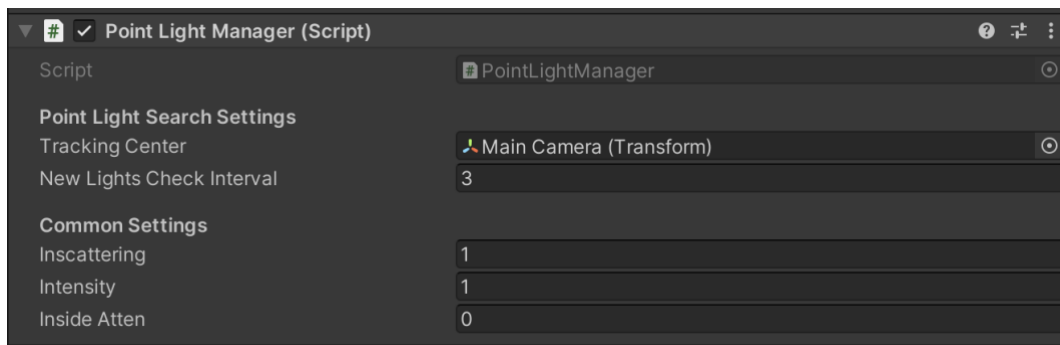
Enabling native lights in the volumetric fog inspector will cast spotlight, point lights and other light types on the fog, including support for shadows. If you enable this option, make sure the URP asset has additional lights/shadows enabled and that the spot/point lights have shadows enabled (if you also wish to render the shadows over the fog).

Point lights with shadows require Unity 2021.1 or later.

Point Lights

Up to 16 fast point lights can illuminate the fog volume using this option. This implementation is faster than the option to support native lights, however, this doesn't support shadows. Also, if you enable native lights, this option won't be available.

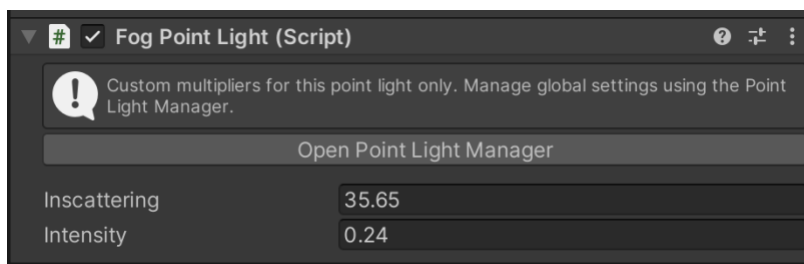
The Point Light Manager is responsible for sending the positions and settings of each point light to the shader.



The Point Light Manager sorts lights by distance to the Tracking Center. You can adjust the time interval for checking new point lights in the scene. You can also call `TrackPointLights(true)` of this component to cause an immediate refresh.

The **Inscattering** and **Intensity** are global multipliers to the point lights range and intensity respectively. The “**Inside Atten**” setting reduces the point light intensity when camera is very near to the point light to avoid “screen burn” effect.

Important: this option requires you to add the `FogPointLight` script to each point light you want to be rendered in the fog:



This script also allows you to specify custom in-scattering & intensity multipliers for each point light.

Note: fast point lights can bleed through geometry. You can enable the option “FAST_POINT_LIGHTS_OCCLUSION” editing the PointLights.cginc to force the shader to perform a depth occlusion test at the light position. This will hide the point light completely when it’s behind solid surfaces (this option affect performance).

Terrain Fitting

By default, the fog renders according to the selected shape (Sphere or Box) and the volume size. When enabling “Terrain Fit” in the profile, the fog will modify the vertical position to fit the underline terrain heightmap.

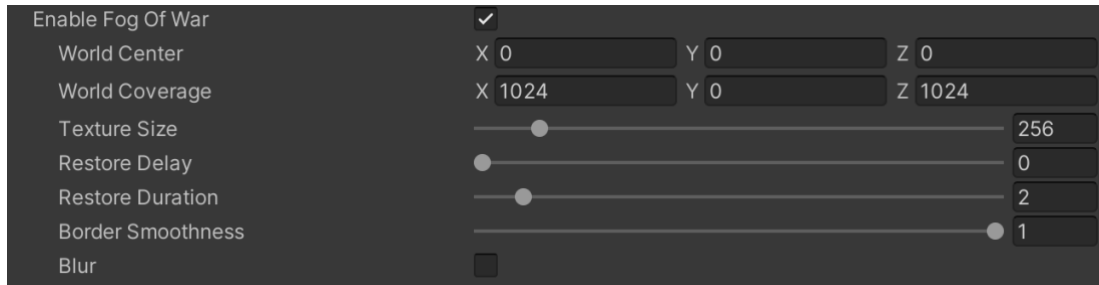
To use this feature please make sure that:

- There’s a terrain (Unity terrain or mesh based terrain) under the fog volume.
- The fog volume has enough size to cover the desired area of the terrain, both in horizontal extension and also from the base altitude to the desired top point of the terrain.
- Use the Terrain Fog Height, Min Altitude and Max Altitude to customize the height range.

Check this video for an example: <https://youtu.be/yBT4no45g2Q>

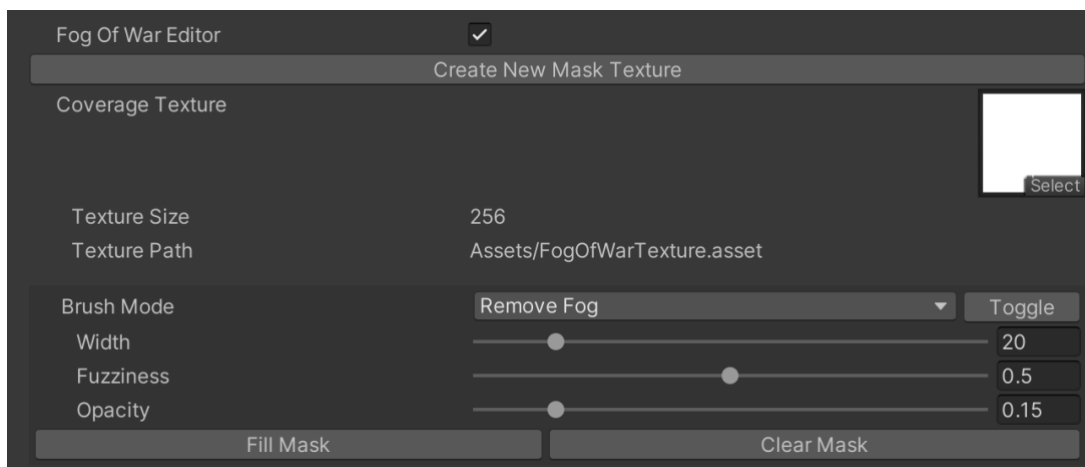
Fog of War

The fog of war feature uses a color texture to control the opacity of the fog on the world. It can be used to clear / add fog at any custom position. This feature is commonly used to show which areas of the world have been visited or are visible to the player.



- **World Center / Coverage:** maps the fog of war texture to the world using this center and size.
- **Texture Size:** the size of the fog of war texture. A big texture allows finer details but it can result in a slower operation.
- **Restore Delay / Duration:** controls when the fog restores its original appearance. It can be used to allow the fog to disappear while the character is at a certain position and gracefully restore the fog when the character abandons the place. A delay of 0 means the fog won't ever be restored. A duration of 0 means immediate change (no transition).
- **Border Smoothness:** controls the smoothness of the border of clear areas.
- **Blur:** produces a softer transition between clear and opaque fog areas.

Fog of War Editor



Enable the Fog of War Editor to interactively add, paint with color or remove fog in Scene View.

Click "Create new Mask Texture" to create the texture asset which will hold the opacity states of the fog of war.

The Brush Mode selects the paint/erase mode as well as Width, Fuzziness and Opacity allow you to customize the artistic result.

Transparency support

Customizing the rendering order

Each Volumetric Fog volume can have a different render queue setting, located in “Rendering Section”:

Render Queue	3100
Sorting Layer ID	0
Sorting Order	0

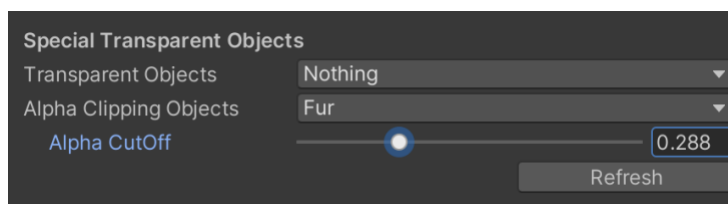
These settings allows you to control the rendering order.

Respecting other transparent objects

The fog effect relays on the depth buffer to compute ray marching distance (that’s why you need to enable “Depth Texture” checkbox in the URP asset). Since transparent objects do not render to depth buffer, it means the fog effect will always render on top (or behind) of previously rendered objects depending on the render queue (for example water, rivers and any other transparent objects located between the fog and the camera may look incorrect either fully visible on top of the fog or completely wiped by the fog).

To avoid this issue, you should first try solving the issue by customizing the rendering order (see previous section). You can change the rendering order of Volumetric Fog volume or change the rendering order of the transparent material so one renders before the other.

Alternatively, you can force a custom depth pre-pass for certain transparent objects from the Volumetric Fog Manager inspector. There’re two ways that can be combined: transparent and alpha clipping.

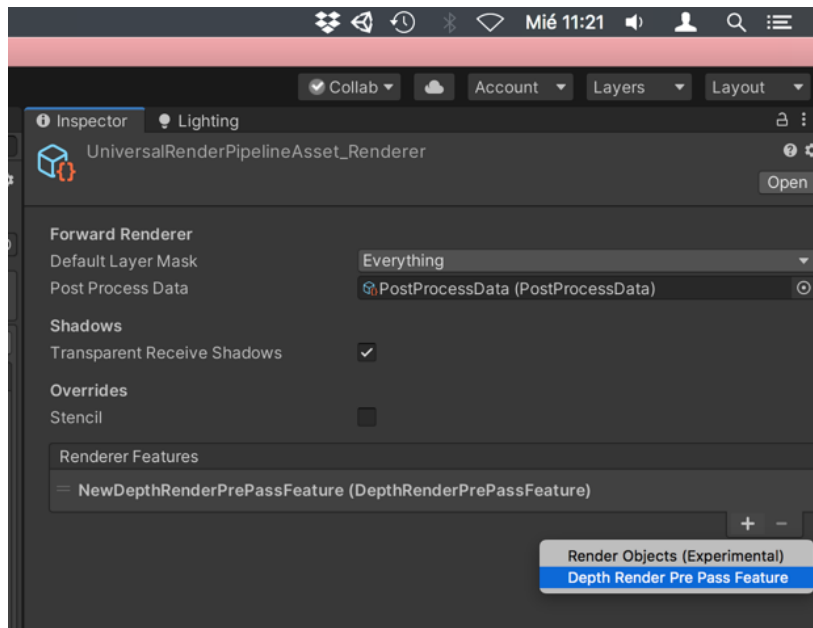


Special Transparent Objects	
Transparent Objects	Nothing
Alpha Clipping Objects	Fur
Alpha CutOff	0.288
Refresh	

- 1) “Transparent Objects”: specify which transparent layers should be included in the depth pre-pass and be considered completely opaque. This option is designed for ocean, lakes and rivers where the surface should behave like ground with respect to the fog.
- 2) “Alpha Clipping Objects”: for other kind of semi-transparent objects, like fur and hair. Increase the Alpha CutOff value.

Important note 1: These two options require installing the Depth Render Pre-Pass Feature in the Forward Renderer of the Universal Rendering Pipeline asset. Simply go to Project Settings -> Graphics

-> Double click on the Universal Pipeline asset -> Double click on the Forward Renderer asset -> Add the “DepthRenderPrePassFeature” to the list:

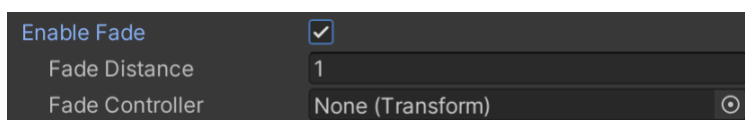


Important note 2: when enabling “Alpha Clipping Objects” option, the asset will find which objects are contained in this layer during start. If you create new objects in this layer at runtime, call **VolumetricFog.FindAlphaClippingObjects()** from a script to update the internal list of objects.

Volume fade in/out and Sub-Volume support

Fading in/out the fog volume

Enable the “Fade” option in the inspector to make the fog volume invisible when the controller is outside the fog volume and make the fog appear smoothly as it enters the volume:

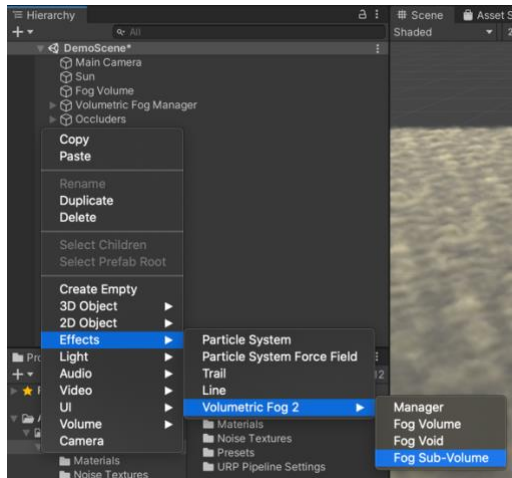


The Fade Controller is usually the player camera or the character controller.

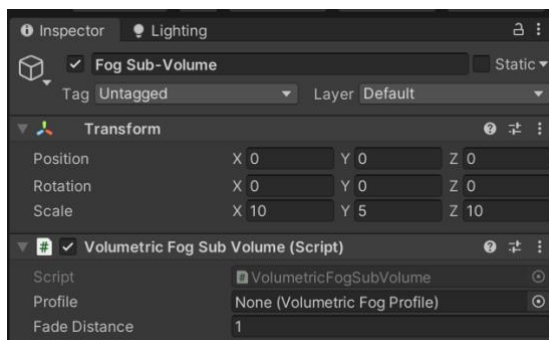
Sub-volumes

Sub-volumes are zones that overlap a fog volume that modify the properties of the fog as the controller enters that sub-volume. For example, you may want to use a different fog density or color on certain areas of a fog volume.

To create a new sub-volume, right click in the hierarchy and select Effects -> Volumetric Fog 2 -> Fog Sub-Volume as shown in this screenshot:



Then, position and scale the new sub-volume using the transform position/scale anywhere in the scene, and assign the desired new profile for that sub-volume and a blending or fade distance:



Scripting Support

Accessing fog properties

All properties shown in the inspector can be set or modified at runtime using scripting. First you need to get a reference to the main script which can be done easily using:

```
using VolumetricFogAndMist2;  
...  
VolumetricFog fog = volumeObject.GetComponent<VolumetricFog>;  
(where volumeObject is the gameobject of the fog volume)
```

Then, you can change any profile property like the fog color or wind speed/direction:

```
fog.profile.albedo = new Color(1,0,0);  
fog.profile.windDirection = Vector3.right;  
etc.
```

You can also access the properties using the “settings” accessor which will create an internal copy of the profile the first time you use it so any change to the properties will only affect that fog instance. Example:

```
fog.settings.albedo = new Color(1,0,0);  
etc.
```

Creating fog volumes at runtime

Use this code to create a new fog volume using code:

```
GameObject fogVolume = VolumetricFogManager.CreateFogVolume(“New Fog Volume”)
```

Creating fog voids at runtime

Use this code to create a new fog void using code:

```
GameObject fogVoid = VolumetricFogManager.CreateFogVoid(“New Fog Void”)
```


Fog of War methods

The following fog of war methods are available:

fogOfWarTexture

Property that returns or set the fog of war texture.

ResetFogOfWar()

General reset function which removes any previous change to fog of war transparency.

ResetFogOfWarAlpha(...)

Restores fog immediately at a given position with radius or bounds.

GetFogOfWarAlpha(...)

Returns the opacity of the fog at a given world position.

SetFogOfWarAlpha(...)

Changes the transparency of the fog in an area. This method is overloaded and can accept different paremeters and options:

- A world position and a radius, or a bounds object or a world position and size.
- Desired alpha and specify if this value replaces the existing transparency or if it blends with it (blendAlpha parameter).
- The duration of the transition (pass 0 to set the alpha immediately).
- The smoothness/harshness of the border (by default it uses the global setting in Fog Of War section which is 1f).
- The restore delay and duration (by default it uses the global settings in Fog Of War section). Pass 0 to restore delay to make the change permanent.

Performance Tips

Volumetric Fog uses an extremely optimized ray-marching algorithm to provide “volumetric sense” fog in front of your player. If you need to improve performance, you can try the following options:

- Increase **RayMarch Min Step**: increasing this value will improve performance but will make the fog effect less accurate and show more artifacts.
- Reduce **Raymarch Quality**: reducing this value will indeed reduce the number of texture fetches per pixel.
- Use **Fog of War** instead of Fog Voids: consider switching to fog of war instead of fog voids if you use many voids.
- **Detail Noise** adds an additional texture3D sample during the ray-marching loop: consider disabling this option to increase performance if needed.
- Reduce **Render Scale** in the Universal Rendering Pipeline asset. This will reduce the resolution of the framebuffer, improving the overall performance. This setting is especially useful on mobile devices since they use very high DPI screens and can afford to use a lower resolution.
- Edit Commons.hlsl file and reduce the **MAX_ITERATIONS** parameter. By default, a value of 500 as maximum should be good for most open/large areas. If you’re fog areas cover a small area, you may want to reduce this value to ensure no more samples than necessary are used. Note that this value is just a maximum. It doesn’t mean that 500 iterations as used. The maximum is provided to avoid loops excessively large.

Known Issues / Questions

Shadow Cascade Bug

Universal Rendering Pipeline 7.1.8 has a bug with shadow cascade.
Solution: please make sure you use URP 7.2.0 or later.

Shader can take some time to compile

The first time the shader is compiled, it can take some time due to the amount of shader features. To speed up this process, you can edit the Volumetric Fog2DURP shader file and remove some “multi_compile” pragma lines or use Shader Control tool from the Asset Store to list and disable keywords related to features that you are not going to use:

```
#pragma multi_compile _ MAIN_LIGHT_SHADOWS // directional light shadow
#pragma multi_compile _ MAIN_LIGHT_SHADOWS_CASCADE // shadow cascades
#pragma multi_compile _ ADDITIONAL_LIGHT_SHADOWS // additional light shadows
#pragma multi_compile _ VF2_DEPTH_PREPASS // depth prepass
#pragma multi_compile_local _ VF2_POINT_LIGHTS VF2_NATIVE_LIGHTS // fast point lights or native lights
#pragma multi_compile_local _ VF2_VOIDS // voids
#pragma multi_compile_local _ VF2_FOW // fog of war
#pragma multi_compile_local _ VF2_RECEIVE_SHADOWS // receives directional light shadow
#pragma multi_compile_local _ VF2_DISTANCE // short distance fade
#pragma multi_compile_local V2F_SHAPE_BOX V2F_SHAPE_SPHERE // fog shape (box or sphere)
#pragma multi_compile_local _ V2F_DETAIL_NOISE // use 3d noise
#pragma multi_compile_local _ V2F_SURFACE // terrain fit
```

Inverted Depth Bug

Universal Rendering Pipeline 7.2.0 has a bug which produces an incorrect (flipped) depth texture. This bug occurs when HDR is OFF, MSAA is x1 and Render Scale =1. More details here: <https://forum.unity.com/threads/inverted-depth-texture-and-unresolved-shadowmap.824703/#post-5466879>

Solution:

Option 1) Enabling HDR, MSAA or changing Render Scale (any of the three) produces the correct depth texture.

Option 2) Enable “Flip Depth Texture” option in Volumetric Fog Manager.

Shadows not showing in WebGL 2.0

Edit RayMarch2D.cginc file in Resources/Shaders folder and uncomment this line:
`// UNITY_UNROLLX(50)`
(Increase the 50 value if needed)

Fog not rendering correctly when using VR in URP

URP 7.4 has a known bug related to inverted matrices not being set correctly in VR. While this bug is not fixed by Unity, the workaround requires editing the file CommonsURP.hlsl and uncomment the line below:

```
// #define USE_ALTERNATE_RECONSTRUCT_API
```

Additional noise textures

You can find additional noise and detail noise textures inside the Demo / Noise Textures folder.

Support

Please visit kronnect.com/support for questions, support and more info.
If you have any suggestion to improve the asset, please contact us. We'll gladly consider it.