

# Volumetric Fog & Mist



## **VOLUMETRIC FOG & MIST**

Reference Guide

## Contents

Introduction .....	3
Demo Scenes .....	3
Quick Start.....	3
Special Features.....	4
Fog Profiles .....	4
Fog Volumes .....	5
Transparency Support .....	7
Automatic fog light alignment with Sun .....	8
Elevated Fog & Clouds.....	8
Fog Colors and Lighting Model.....	8
Fog Void.....	9
Fog Areas .....	11
<i>Method 1: Limit the fog of an existing Volumetric Fog component attached to the camera.</i> .....	11
<i>Method 2: Drag &amp; drop the fog area prefabs.</i> .....	11
<i>Method 3: Use the API to create fog areas.</i> .....	12
<i>Setting rendering order of fog areas.</i> .....	12
Terrain Fit option for Fog Areas .....	14
Fog of War .....	15
Point Light support.....	16
Light Scattering.....	17
Depth Blur .....	17
Sun Shadows (directional light + cookie support).....	18
Geometry Mask.....	19
Orthographic Camera support .....	20
VR and Single Pass Stereo Rendering.....	20
Scripting Support .....	21
Accessing global fog properties.....	21
List of useful runtime Volumetric Fog properties .....	21
Fog of War methods.....	24
Enhanced compatibility .....	24
Enhanced Sprite Compatibility .....	24
Compatibility with Time of Day .....	25
Compatibility with Enviro .....	25
Compatibility with other Sky Managers.....	26
Compatibility with Gaia .....	26
Compatibility with Horizon(ON).....	27
Compatibility with other shaders (surface shaders) .....	27
Compatibility with other shaders (vertex/fragment shaders) .....	28
Performance Tips.....	29
Common Issues (FAQ) .....	31
Additional Support .....	33
Dynamic Fog & Mist .....	33

## Introduction

### Thanks for purchasing!

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

This asset is highly optimized but due to the nature of its algorithm it's not suitable for mobile devices. For mobile devices, use Dynamic Fog & Mist, also included in the package. You can add both assets to your project and enable the most appropriate for the running platform (see Dynamic Fog & Mist section at the end of this document).

## Demo Scenes

Just load any demo scene included and click "Play". You will be able to move around using WASD or cursor keys. Press spacebar to jump, F to change fog style and T to toggle on/off the fog.

The Fog Of War demo scene allows you to "Cut" the fog as you pass through it (press "C" to enable fog cutting mode).

You can delete the Demo folder entirely or ignore it when importing the asset into your project.

## Quick Start

1. Add the VolumetricFog script to your main camera in your scene.
2. Choose one of the preset and that's all!

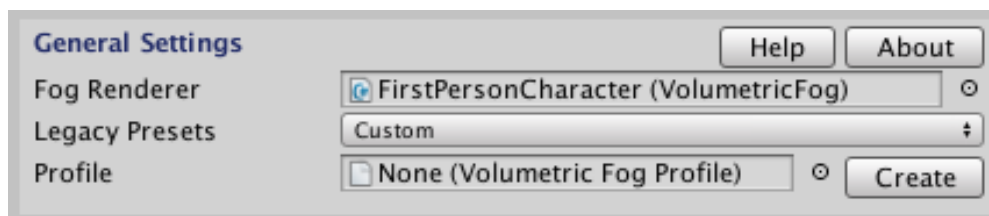
You can of course customize any of its parameters to match your game mood and requirements.

Please read this document to learn more about all supported features.

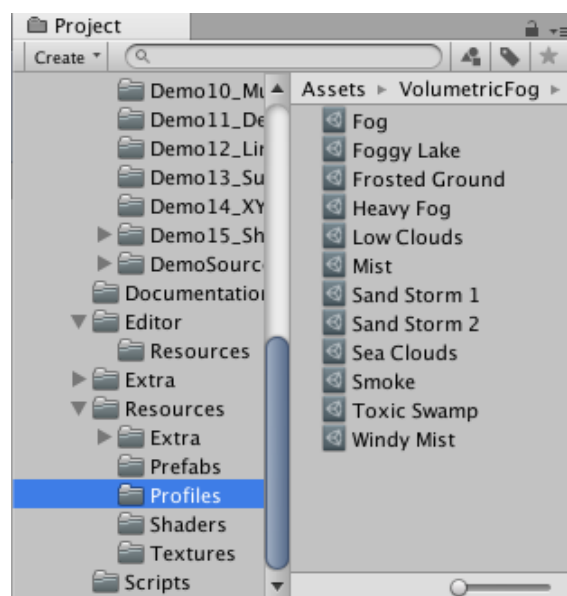
## Special Features

### Fog Profiles

Starting with V8, Volumetric Fog & Mist accepts profiles. A profile is a special file that contains all the settings used to customize look & feel of the Volumetric Fog.

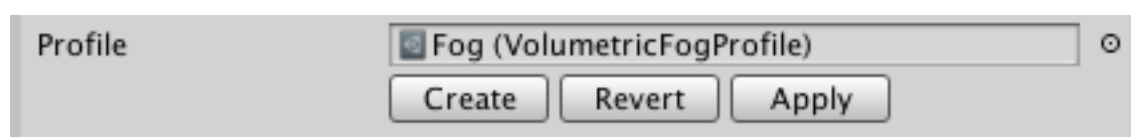


Assign a profile to the “Profile” slot to load its settings. You can find some sample profiles inside the VolumetricFog/Resources/Profiles folder:



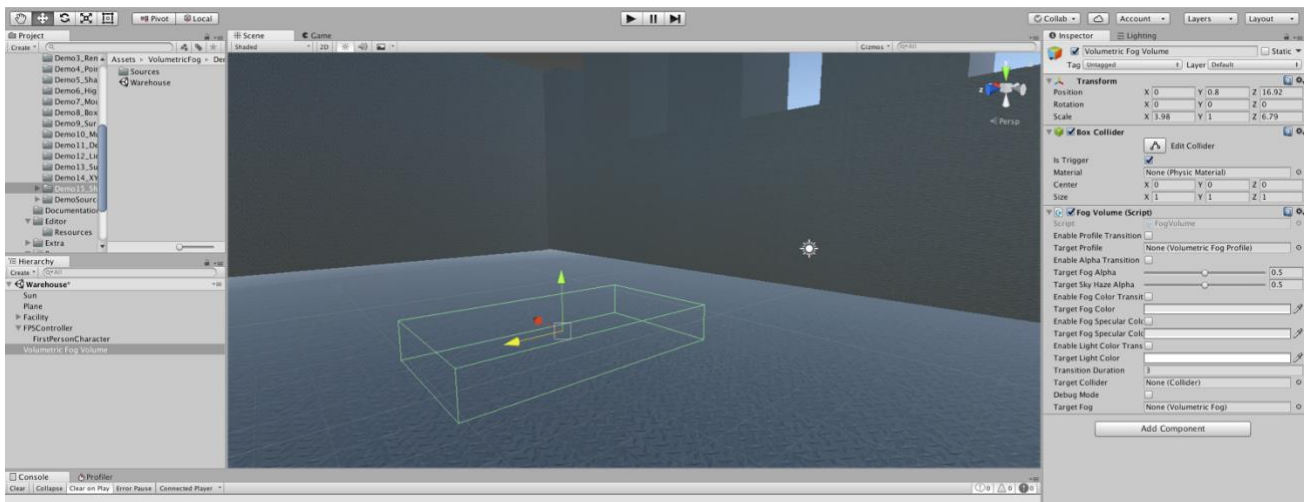
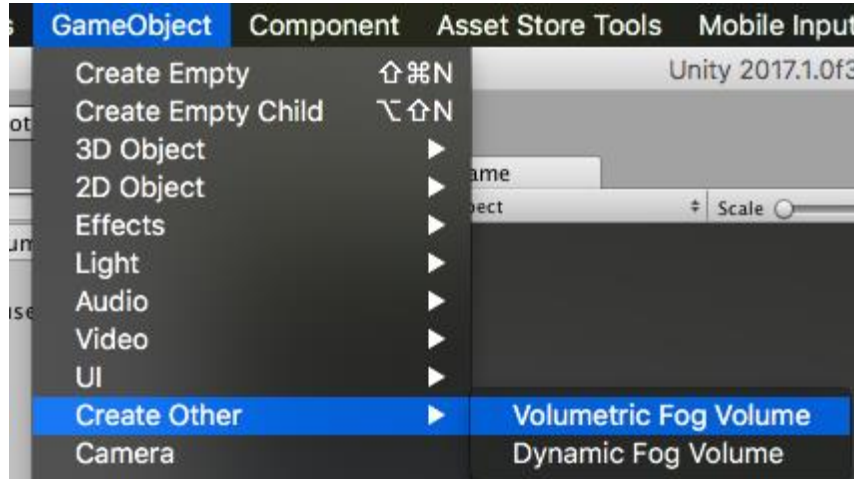
Click “Create” to create a new profile with current settings. A new file will be created and you can rename it and move it into any folder.

When a profile is assigned, you will be able to Create a copy, Revert current settings (reload the profile) or update the profile with current settings:

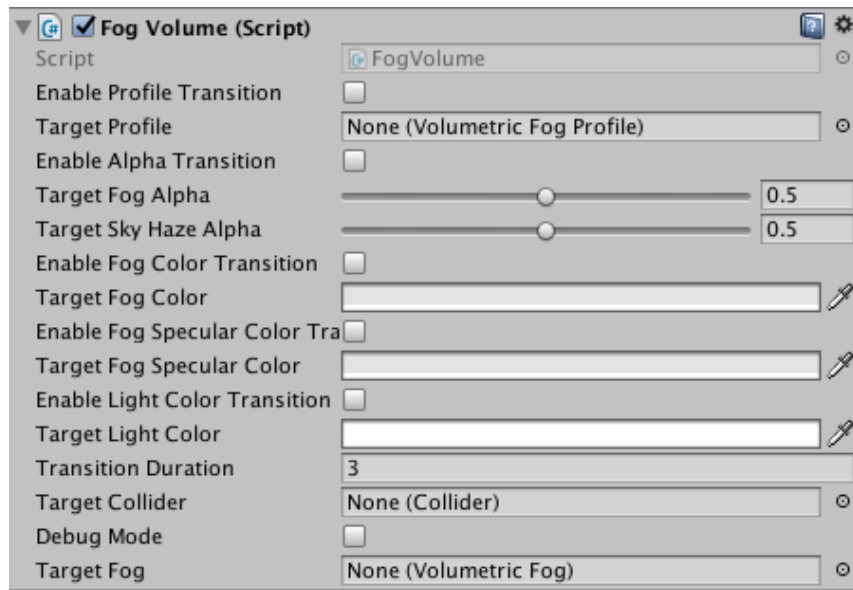


## Fog Volumes

You can define special zones (fog volumes) where fog alpha, color and other properties can change automatically. Create a fog volume from the menu **GameObject / Create Other / Fog Volume**. Position the fog volume over the desired area and edit the transform scale, then customize what attributes of the fog should be changed in the **FogVolume** script (see below).



Fog Volumes can change fog alpha, fog color, fog specular color, fog light color and sky haze alpha. But also you can assign a **fog profile** which holds many options:



**Enable Profile Transition:** when this option is enabled and a profile is assigned to the Target Profile slot, all the settings of that profile will be used in the transition. This is the most complete option to create great transitions.

**Enable Alpha Transition:** when enabled, will make a transition with fog alpha value (transparency of the fog) and also the sky haze alpha (transparency of the sky haze effect if used).

**Enable Fog Color Transition:** when enabled, the albedo color of the fog will transition to this value.

**Enable Fog Specular Transition:** same but affecting the fog specular color.

**Enable Light Color Transition:** same but affects the light color value.

**Transition Duration:** duration in seconds for the transition.

**Target Collider:** the collider that will be detected to trigger the transition. By default it uses the collider attached to the camera (if any).

**Debug Mode:** if enabled, it will print a debug message into the console when the collider enters or exits the volume.

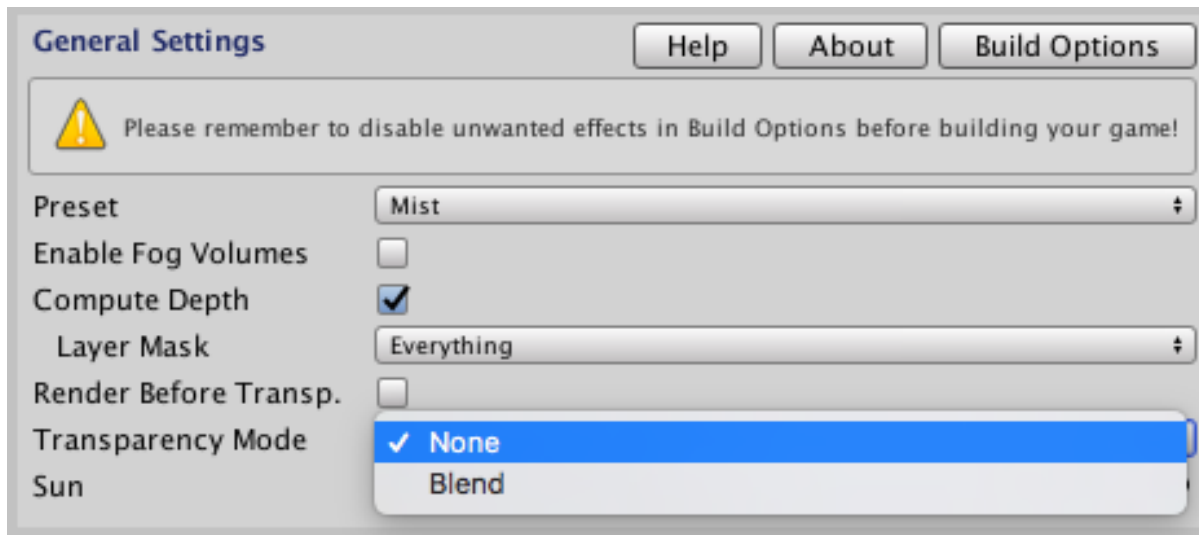
**Target Fog:** the target Volumetric Fog & Mist that will receive the changes. By default it's the script attached to the camera.

By default, Fog Volumes will react to any collider that contains the camera to which the fog image effect is attached (for first person controllers this is the default behaviour). But you can also assign a different collider so they will also work with third person controllers.

Check out Demo Scene #7 "MountainClimb" for an example of Fog Volumes.

## Transparency Support

Under General Settings, you can find a few options to control the order of execution of Volumetric Fog & Mist as well as some specific options for dealing with transparent objects like particles and tree billboards:



**Compute Depth:** this option will calculate the depth for standard Unity tree billboards and/or transparent objects included in the specific layer mask (not necessary for SpeedTree). This custom render pass will prevent these trees to be clipped when you look to the horizon or sky and there's a billboard tree in the middle.

**Render Before Transparency:** this toggle controls the execution order of the image effect. If enabled fog will be rendered before any transparent object. If not (default value), it will render after everything has been rendered.

**Transparency Mode:** this option is only enabled if the image effect renders after transparent objects (see previous option). In this case, if the fog is very thick, it can occlude any particle or transparent object. Using this special Blend feature they will be combined with the fog color. Experiment with different Blend Power values (shown when Transparency Mode is set to Blend).

It's important to note that proper transparency and fog support is not possible in Unity due to how the render pipeline works. Depending on the type of transparent objects you use and your scene setup one of the following 3 options may work for you:

1. **Render Before Transparency:** this option will make Volumetric Fog & Mist render before any transparent object is drawn so they are not affected nor clipped by the fog.  
Pros: best performance.  
Cons: a side effect of this option is that transparent objects may look weird rendered on top of the fog.
2. **Enable Compute Depth option with Scope: Tree Billboards And Transparent Objects.** This option will enable a custom depth pass and will treat transparent objects as cutout shaders. You can adjust the Cut Off value to match the best result.

Pros: transparent objects receive fog effect based on their exact depth.

Cons: requires an additional depth pass which is slower. Also some transparent objects with smooth gradients or edges may look bad since this option treats them as CutOut shaders (hard edges).

3. Transparency Mode set to Blend: this option can be used instead of the previous one (they are exclusive options). It renders after transparent objects have been drawn but afterwards blends the transparent objects on top of the fog with a customizable blending intensity.

Pros: looks similar to “Render Before Transparency” but with customizable blending intensity and running after all transparent objects have been drawn. This setup is required when also using Unity standard tree.

Cons: although it provides a bit better result than “Render Before Transparent”, the result is non-optimal as no depth-checking is performed.

## Automatic fog light alignment with Sun

A light game object existing in the scene can act as the Sun and be assigned to Volumetric. You’ll find a property in the inspector, called Sun, where you can drag and drop the desired light in your scene.

After assigning a light as the Sun, the fog will react to the sun light direction, intensity and color automatically. Click “Unassign” to break the link and allow to freely customize light direction, intensity and color.

## Elevated Fog & Clouds

You can make the fog start above Camera position to simulate floating smoke or even clouds! Try it assigning a value above the Camera Y position to Base Height property in the inspector.

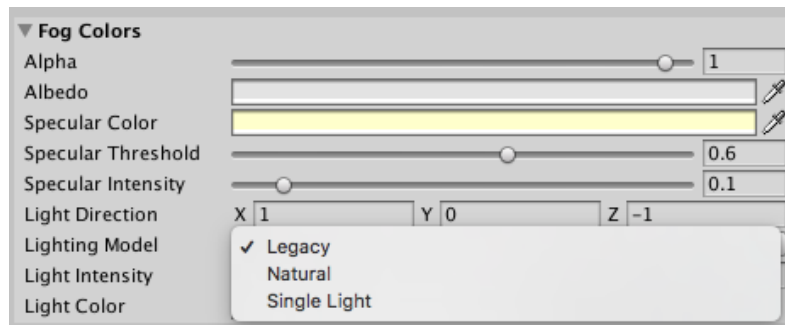
You can increase the Max Distance beyond your Camera Far Clip plane to force the cloud layer to render beyond it and even reach the horizon. Volumetric Fog & Mist will apply a special matrix transform to achieve this effect automatically.

Check out Demo Scene #7 “MountainClimb” for an example of how to enable a dynamic cloud layer that reacts when player climbs a mountain.

## Fog Colors and Lighting Model

To provide more artistic choices, Volumetric Fog provides different “colorization formulae”. This option is located under Fog Colors section:





**Legacy:** this is the traditional technique used in Volumetric Fog & Mist. It lerps between the ambient light and the sun or light color to provide the final fog color.

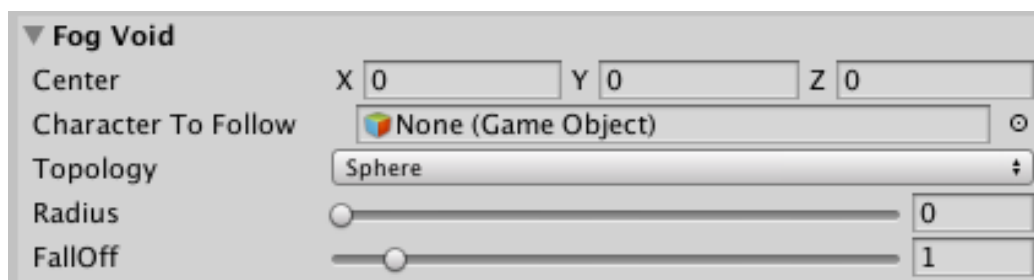
**Natural:** in this mode, the color lerps between the ambient and ambient + light color and the result is multiplied by the light intensity.

**Single Light:** in this mode, the ambient color is not used. Just the light color multiplied by the light intensity.

Note that if a Sun object is assigned, the Sun's light intensity is added to the Light intensity parameter.

## Fog Void

Another great feature included in the asset is the **void area**. This option is useful if you want a clear area around a world space position. For example, in 3rd Person View games, you may want a clear area around the character.



This option is similar to the Distance Fog feature (using Distance > 0) but Distance Fog Works having into account the distance from the Camera. So Distance parameter is useful for 1st Person View and Void Areas for 3rd Person View.

Void areas can be **spherical or boxed**. To make a spherical void area just move the radius slider to the left. To make it boxed, set the width and depth sliders.

Also, you can assign your character game object to the property field in this section, so the center of the void will follow it automatically.



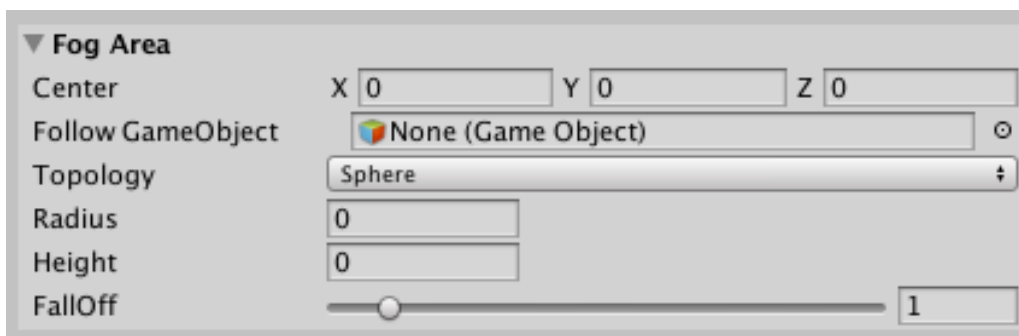
## Fog Areas

Fog Areas can be defined to limit the rendering of the fog inside a box or sphere container. The demo scene #6 “HighClouds” shows an example

There’re 3 ways to add a Fog Area into your scene:

Method 1: Limit the fog of an existing Volumetric Fog component attached to the camera.

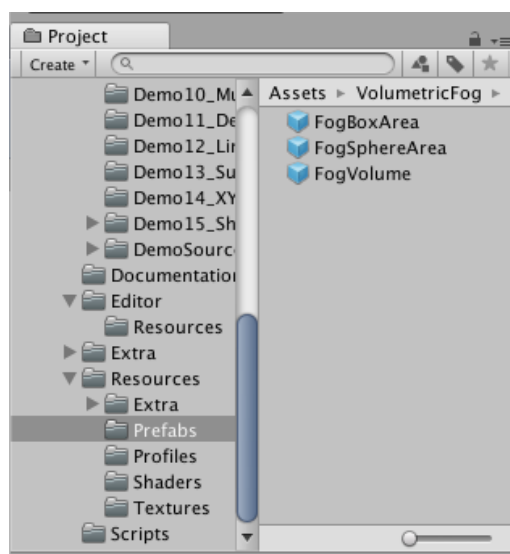
Use the Fog Area section:



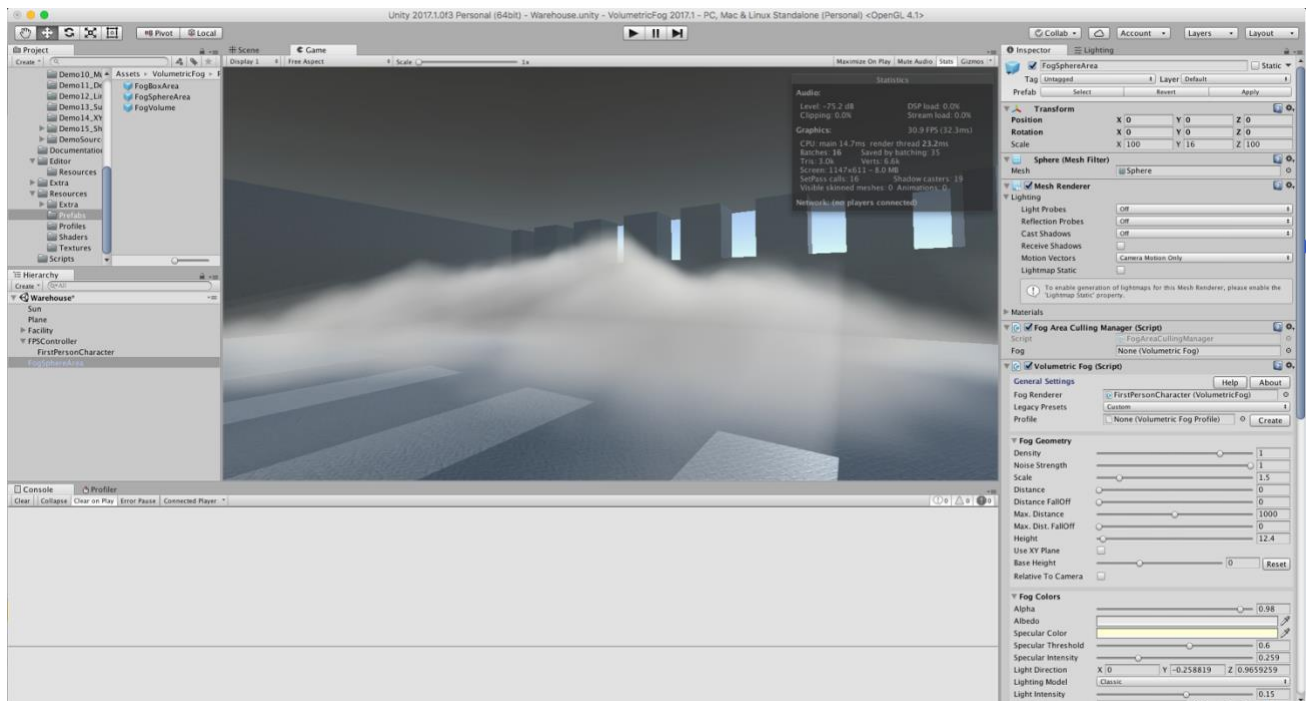
- Set the position of the fog area in Fog Area position property.
- Choose a fog area topology.
- Choose a radius and a falloff for spherical fog area or radius/depth/height for a box-type fog area.
- Optionally assign a GameObject to the “Follow GameObject” slot. This will make the fog area to follow that game object.

Method 2: Drag & drop the fog area prefabs.

Locate the fog area prefabs inside VolumetricFog / Resources / Prefab folder:



Choose either FogBoxArea or FogSphereArea, drag and drop it into your Scene View or hierarchy:



Once the fog area is in your scene, it will automatically link to the Volumetric Fog script attached to your camera. If you need to use another camera, just change the Fog Renderer property and assign another Volumetric Fog & Mist.

Explanation: fog areas are containers for settings and storages for the fog area state values. It's the VolumetricFog script attached to the camera which is responsible of all the rendering.

**To change the dimensions of the fog area created this way, please use the transform of the fog area game object.** You can change the position and the scale of the transform to match the desired dimensions and location of the fog area.

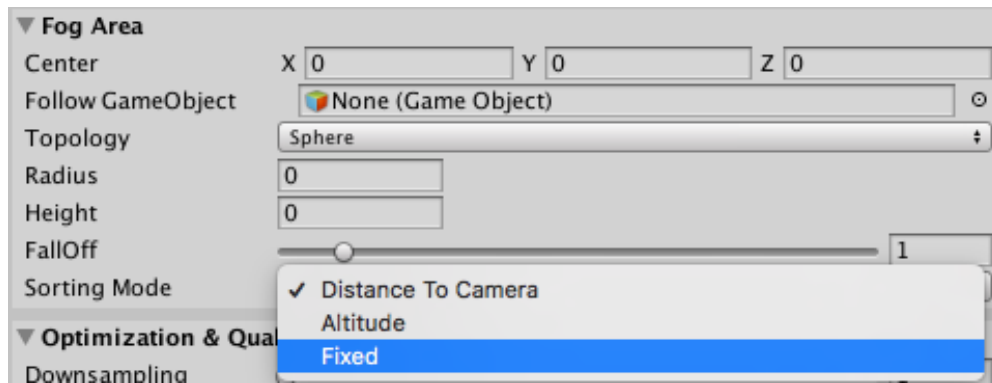
Method 3: Use the API to create fog areas.

You can also create any number of fog areas during runtime calling **fog.CreateFogArea API**. Please check demo scene 10 source code for quick examples.

Setting rendering order of fog areas

By default, fog areas are rendered from back to front respective to the distance to the camera. However, in some circumstances you may want to specify a different sorting method. This can be useful when you need a background layer of clouds or to overcome some undesired overlapping.

Go to Fog Area section and choose a different sorting method:

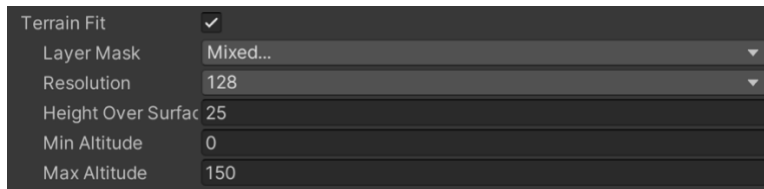


Possible options are:

- **Distance To camera** (default)
- **Altitude**: this fog area will be sorted depending on the altitude (difference in height with respect to the camera position).
- **Fixed**: this option allows you to introduce a fixed rendering order. All fog areas have a default render order of 1. If you enter a value of 2, this fog area will be rendered after all fog areas have rendered so it will show on top of every other fog area.

## Terrain Fit option for Fog Areas

The Terrain Fit option adjusts the fog vertical position to the surface altitude creating fog effects that slide over hills and mountains. This option is only available in fog areas.

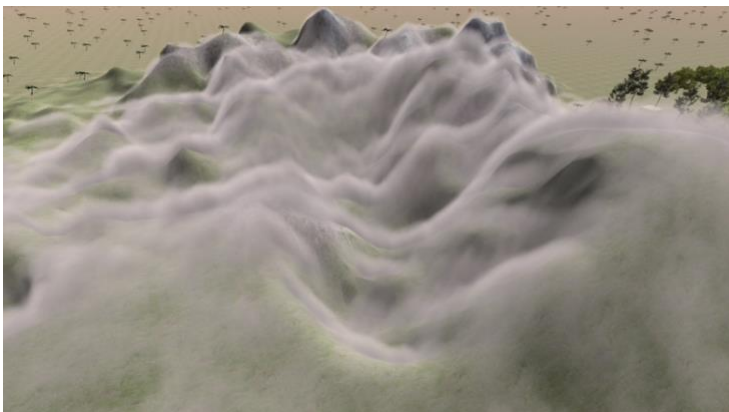


To use this option please ensure that:

- 1) The volume covered by the fog area encloses the whole surface desired. Adjust the fog area radius, height and depth accordingly.
- 2) The Height value in Fog Geometry section is high enough.

The options provided by Terrain Fit are:

- **Layer Mask:** determines which objects will be considered as ground. You can include the terrain layer but also any other solid objects.
- **Resolution:** this option performs a zenithal depth pre-pass to capture the elevation of the ground. The resolution here refers to the size of the generated depth texture.
- **Height Over Surface:** this is the height of the fog counting from the surface position.
- **Min / Max Altitude:** let you clamp the minimum and maximum altitude of the enclosing surface to render the fog.



## Fog of War

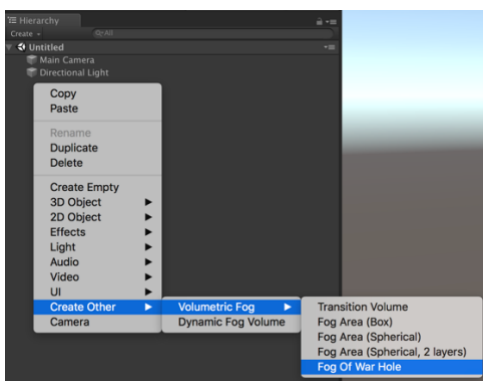
You can also set any number of void areas just calling **SetFogOfWarAlpha** method of the Volumetric script. Just pass the world space position, the radius or bounds and the desired new alpha for the fog. Just make sure the center and size of the fog of war (configured in the inspector) are properly set (by default the fog of war is centered on 0,0,0 with a size of 1024x1024). Optionally pass a duration to produce a smoother effect.

To make the cut permanent make sure Restore Delay is set to 0.

Call **ResetFogOfWar** to reset the cleared areas back to normal.

Important: Fog of War relies on z-buffer data to map world positions to the mask texture. It won't work properly if there's no geometry under the fog (it's aimed to fog over terrain or any other geometry that writes to z-buffer, transparent shaders won't work neither).

You can also place unlimited void placeholders in SceneView by placing a Fog Of War Hole. Just right click on the hierarchy panel and select Create Other -> Volumetric Fog -> Fog of War Hole:



This gameobject has a FogOfWarHole script which calls the SetFogOfWarAlpha method automatically. Use the transform position and scale to position/rescale the hole.

## Point Light support

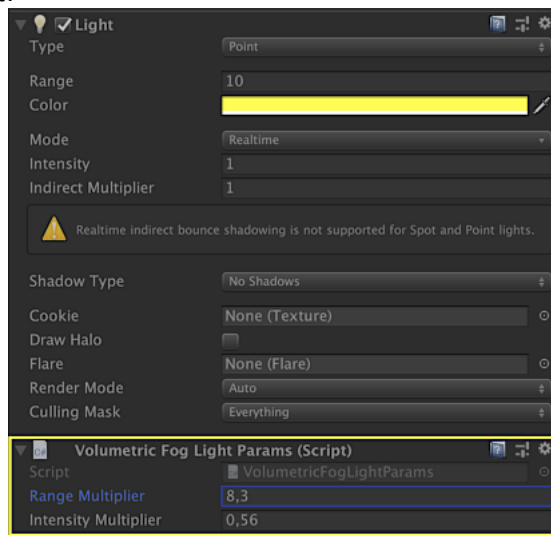
A number of point lights can be selected to illuminate the fog with different colors, ranges and intensities. The maximum number of point lights can be configured clicking on “Shader Options” button on top of Volumetric Fog & Mist inspector.

This feature can be used to achieve many visual effects, from fog gradients, glow or bloom, definitely enhancing the ambiance of your scenes.

At night, if you have the fog synced with a Sun game object, the fog will be dark. You can use this feature to illuminate the surrounding area of the characters.

Note that point light are simulated in the shader, so it’s not necessary to have real point lights in the scene. But if you have them, you can simply assign them to the slots in the Point Light section or make the asset choose them for you automatically using the slider and checking “Track Point Lights”.

- If you assign a scene Point Light to a slot under Point Light section, the parameters like color, position, intensity, etc. will be driven by the Light and VolumetricFogLightParams component attached to that light:



- If you don’t assign a Point Light to a slot under Point Light section, then you can freely enter custom values like position, color, intensity, etc. to create a virtual light which can illuminate the fog.



## Light Scattering

Light scattering is the physical phenomena that makes the light reflect/refract when it crosses different densities, adding shafts of light across the scene from the light source. Volumetric Fog & Mist includes a light scattering option, which approximates that effect in screen space (that means that it only works when the light source is in front of you).

To enable this option, you need to assign a light source to the Sun property and check the Enable toggle in the inspector. Then customize the effect with the different parameters in that section.

Light Scattering section is really divided in 2 effects:

- Light Diffusion: this is a lightweight effect that will color the fog around the Sun as you look directly to it.
- Sun Shafts: simulates the effect caused when objects occlude the light of the Sun. Please note that Sun Shafts will increase the workload on the GPU, so you may have a configuration parameter in your game to enable it in addition to the fog. Also note that Sun Shafts are designed so they work better when they cross the fog.

## Depth Blur

This option blurs the color buffer before overlaying the fog to simulate additional light scattering. Try it and experiment with different depths.

## Sun Shadows (directional light + cookie support)

Enable this feature to cast Sun shadows over the fog. This option requires a directional light assigned to the Sun property in the inspector.

By default, Volumetric Fog & Mist will reuse the directional light shadow map generated by Unity. However, it's possible to force Volumetric Fog & Mist cast a custom shadow pass. To enable the custom shadow pass, click on "Shader Options" button and disable the "Native Directional Shadows" checkbox – do this only if you have a reason to perform a custom shadow pass, like choosing which objects can cast shadows on the fog. If unsure, leave the option enabled.

In the same "Shader Options" section you can enable support for the directional light cookie (if no cookie is assigned to the directional light, leave this option unchecked).

The "Shadow Cancellation" slider allows you to control the alpha transparency of the fog under shadows. If you set this value to 1, only lit fog by the assigned Sun light will be visible producing a Volumetric Light effect.

Use the "Jitter" parameter to reduce banding effects when using Sun Shadows feature.

If custom shadow pass is enabled in Shader Options, additional options will appear under Sun Shadows section:

- The "Force Update Interval" parameter can be set to a value greater than 0 to specify the interval in seconds between shadow casting shoots. If set to zero (default value) it will shoot shadows only if the Sun changes rotation (this is usually the most efficient setting). You can set this value to 0.016 to shoot every frame (if you target 60 frames/s or set even a lower value) have the shadows always updated but this will take extra ms from your framerate. It just depends if you have moving objects on the scene that clearly cast shadows over the fog and you want to ensure that shadows are always in sync with those objects.
- The "Max Distance" will determine the maximum distance from the camera for the shadows. It does not affect performance but shadow quality. The greater the value the poor quality and vice versa. The Resolution parameter also affects shadow quality in a more direct way as this value determines the size of the shadow texture that's produced in each shadow shoot. The default value of 3 uses a 4096x4096 texture, you can increase this value to 4 and it will use a 8192x8192 texture which is currently the maximum texture size allowed by Unity 5. You may experiment with lower resolutions as well and see if they work for you – it will improve performance and reduce VRAM memory usage.
- The "Bias" parameter option will offset the shadow start from the object itself to prevent self-shadowing. By default, a value of 0.1 ensures this can't happen so this value should be left as it's, unless you need to change it to create special effects.

## Geometry Mask

Geometry masks use the depth information from scene objects to mask out the fog in the scene. This feature is similar to fog areas but has two key differences:

- Geometry Mask affects entire fog rendering. While each fog area can have different appearance and behaviour attributes, the geometry mask affects the entire fog (including any other fog area!).
- Geometry Mask can include any number of objects which allows to make very complex shapes.

The workflow for using geometry mask follows:

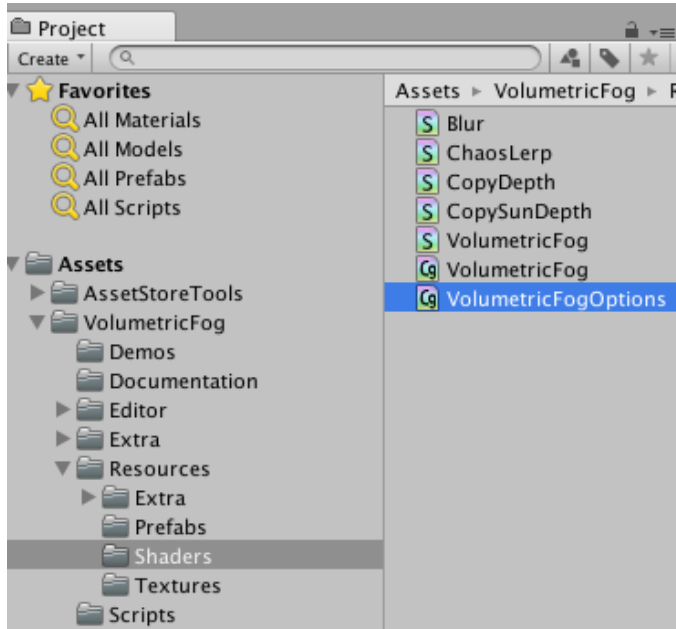
- 1.- Select the main camera and click “Shader Options” button located on top of Volumetric Fog inspector.
- 2.- Scroll down until you see the Fog Mask option. There’re two geometry mask modes. Normal and inverted. The inverted mode makes the fog only visible outside the mask while the normal mode does the opposite (shows fog only inside the objects only). Enable only one of the two options.
- 3.- Close the Shader Options section and scroll down to the bottom of inspector until you see “Geometry Mask” section.
- 4.- Select which layers include the objects that will make the mask.
- 5.- Click “Refresh” to quickly list which objects are found. If no object is found, verify that the objects belong to the same layer.
- 6.- Optionally use downsampling option to increase performance (only if needed).

### Important:

- Geometry Mask uses the meshes from scene objects to create complex and compound volumes to render the fog inside.
- Those objects should be used only for the mask. For example if you have a room and you want fog inside the room only, then you can add a cube and scale it inside the room. Assign that cube to a custom layer for the mask purposes.
- In order to allow the fog visible inside the objects assigned to the geometry mask, their mesh renderers need to be disabled. This is expected as the mask objects are only meant to provide their geometry for the mask.
- You can combine any number of objects or meshes to create complex and irregular shapes.
- If you can achieve the results using a geometry mask instead of a fog area, stick to the geometry mask. Using both geometry mask and fog areas is possible although each additional feature costs performance.

## Orthographic Camera support

To enable orthographic camera support, you need to edit VolumetricFogOptions.cginc file located in VolumetricFog/Resources/Shaders folder:



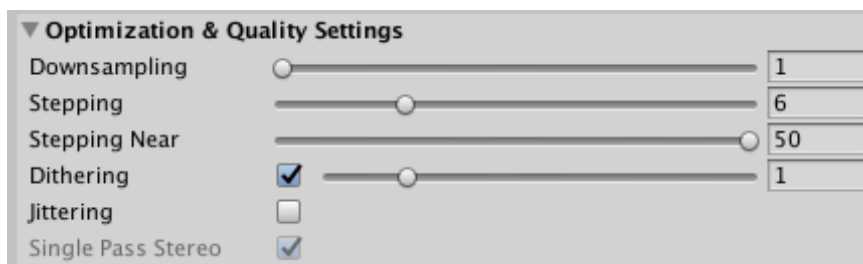
This file contains a global macro called “FOG\_ORTHO”. To enable orthographic camera support, just change the 0 to 1 so it reads:

```
#define FOG_ORTHO 1
```

## VR and Single Pass Stereo Rendering

Volumetric Fog & Mist is fully compatible with VR in both “Multi” and “Single”-Pass Stereo Rendering. It has been tested with Oculus DK2 with both Oculus and OpenVR SDKs.

Important: if you enable Single Pass Stereo Rendering in Player Settings, make sure you verify that it’s reflected in Volumetric Fog & Mist inspector on your camera. Check this new option which is automatically set based on the Player Settings when the inspector is shown.



To disable XR usage, edit VolumetricFog.cs script and comment out the macro define at the top:  
// #define ENABLE\_XR

## Scripting Support

### Accessing global 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 VolumetricFogAndMist;  
...  
VolumetricFog fog = VolumetricFog.instance;
```

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

```
fog.color = new Color(1,0,0);  
fog.speed = 0.5f;  
fog.windDirection = Vector3.right;
```

Although you can set any property using this way, you can also assign a whole profile:

```
fog.profile = myprofile; // my profile is a VolumetricFogProfile scriptableobject.
```

### List of useful runtime Volumetric Fog properties

Unless specified, all properties can return or set a value. For example, fog.profile will return the currently assigned profile while fog.profile = ... will assign a new profile and apply its settings immediately.

Please note that the VolumetricFog script exposes many more properties which usually are set either in the inspector or through a profile. Here's a reduced list of most frequent used properties at runtime.

Property name	Type	Description
preset	FOG_PRESET	Gets or sets a legacy factory preset (ie: WindyMist, SeaClouds, etc.)
profile	VolumetricForProfile	Gets or sets a fog profile with predefined settings. See "Profiles" in Special Section of this document for more details.
useFogVolumes	bool	Specifies if fog volumes can be detected and used. Fog Volumes are special areas where a custom profile can be applied producing a smooth transition. Check "Fog Volumes" section for more details.

sun	GameObject	Gets or sets the gameobject used as Sun. This is usually the directional light.
distance	float	Start distance of the fog in meters.
distanceFalloff	float (0-5)	Controls the gradient transition between camera and the starting distance.
height	float	Height of the fog.
heightFalloff	float (0-1)	Controls the gradient vertical transition.
baselineHeight	float	Specifies the altitude in meters at which the fog starts.
baselineRelativeToCamera	bool	If true, the baselineHeight is relative to the camera Y coordinate. Otherwise the baselineHeight is relative to 0 altitude.
density	float (0-1.25)	The fog density.
alpha	float (0-1.05)	Transparency of the fog.
color	Color	Albedo color of the fog.
specularColor	Color	Specular color of the fog.
copySunColor	bool	If Sun color and intensity should be used. If false, the lightColor and lightIntensity properties are used instead.
lightColor	Color	The color of the directional light.
lightIntensity	float	The intensity or brightness of the directional light.
speed	float (0-1)	Speed of the wind animation.
windDirection	Vector3	Normalized direction of the wind animation (only X & Z axis are used).
useRealTime	bool	If the wind animation timing should use realtime clock or not. If true, any change in speed will be smoother.
turbulence	float (0-10)	Turbulence amount. 0 = disables this effect.
skyColor	Color	The color of the sky haze effect.
skyHaze	float	The altitude of the sky haze effect (0 = disables it completely).
<b>FOG VOID RELATED</b>		
character	GameObject	The character or gameobject to be followed by the fog void.
fogVoidPosition	Vector3	The position of the fog void if not following a gameobject (character property is null).
fogVoidTopology	FOG_VOID_TOPOLOGY	If the fog void is spherical or boxed.
fogVoidRadius	float	Radius of the fog void.

fogVoidDepth	float	Depth of the fog void (if boxed).
fogVoidHeight	float	Height of the fog void (if boxed).
<b>FOG AREA RELATED</b>		
fogAreaCenter	GameObject	The character or gameobject to be followed by this fog area.
fogAreaFollowMode	FOG_AREA_FOLLOW_MODE	If the fog area follows in 3D or 2D (X/Z plane only).
fogAreaPosition	Vector3	The position of the fog area if not following a gameobject (fogAreaCenter is null).
fogAreaTopology	FOG_AREA_TOPOLOGY	If the fog area is spherical or boxed.
fogAreaRadius	float	Radius of the fog area.
fogAreaDepth	float	Depth of the fog area (if boxed).
fogAreaHeight	float	Height of the fog area (if boxed).
<b>OTHER</b>		
Downsampling	int	1 = no downsampling (native screen resolution). Increasing downsampling improves performance at quality expense.
sunShadows	bool	If directional light shadows should be used.

## Fog of War methods

The following fog of war methods are available:

### **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.

### **SetFogOfWarAlpha(...)**

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

## Enhanced compatibility

### Enhanced Sprite Compatibility

Standard Unity sprite shaders are not compatible with Image Effects as they are not rendered as 3D objects and don't write to ZBuffer on real world positions. That forces Image Effects to be rendered before the transparent queue so at least sprites (as particles) are not covered by the Image Effect. This cause that sprites look floating over the scene as the fog makes them clearly separated from rest of the scene.

To avoid the problem described above, Volumetric Fog & Mist includes two custom sprite shaders found in Resources/Extras folder. You will find two materials with names SpriteFogDiffuse and SpriteFogUnlit, which you may just assign to your Sprite Renderers to make Sprite fully compatible with the Image Effects.

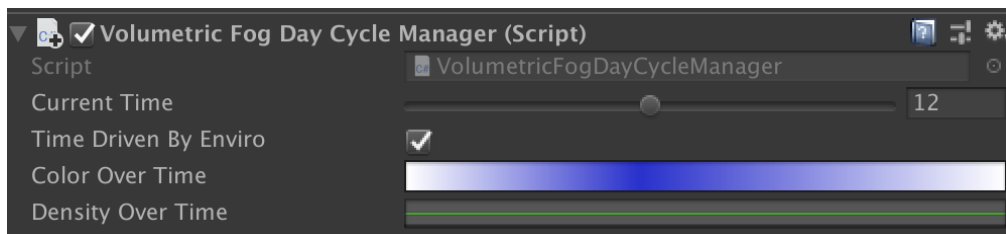


## Compatibility with Time of Day

If you use Time of Day asset, you may assign the Sun game object found under Sky Dome to the Sun field in the Volumetric Fog & Mist inspector. That will sync the direction and color of the Sun light with the fog.

## Compatibility with Enviro

If you use Enviro asset, you can add “VolumetricFogDayCycleManager” script to your any object in your scene (ie. the same camera where you have Volumetric Fog) to modify colors and densities during the day according to the time of day provided by Enviro.



## Compatibility with other Sky Managers

The compatibility with Sky managers in Volumetric Fog & Mist affects two parameters:

- Depth of skybox: many sky managers use a “Sky Dome” to render the sky. This sky dome is a big sphere that surrounds the scene. To allow sky haze work with sky domes, reduce the sky depth parameters under the sky haze section.
- Sun orientation: fog lighting will adapt automatically to the Sun orientation. Just drag and drop any gameobject that reflects current Sun direction onto the Sun property in the inspector.

## Compatibility with Gaia

Volumetric Fog & Mist is also available from Gaia’s Extension Manager. You will find a list of convenient buttons that configures and select the different presets of Volumetric Fog & Mist in just one click.

## Compatibility with Horizon(ON)

Volumetric Fog & Mist has been tested with recent versions of Horizon(ON). It will only work with the real terrain though, so you will probably need to reduce the max distance factor and combine Volumetric Fog & Mist with the fog effect provided by Horizon(ON) for the long distance.

## Compatibility with other shaders (surface shaders)

Although Volumetric Fog & Mist provides methods to render before/after transparent objects, it's also possible to embed the fog effect into other complex shaders so the fog is rendered in the same pass. This solution solves some problems related with transparency.

To integrate Volumetric Fog & Mist in an existing surface shader, please follow these steps:

**Step 1:** Apply the following changes to your shader:

a) edit the surface shader and locate the line starting with `#pragma surface ...`. Add **"noinstancing finalcolor:xxx"** at the end of the same line (without quotes) where xxx can be `overlayFog` (regular surface shaders), `overlayFogStandard` (standard surface shader) or `overlayFogStandardSpecular` (standard surface shader with specular setup).

If **"fullforwardshadows"** is present, you may want to reduce it to improve compilation time

b) insert the following lines just before the `#pragma surface` line:

```
// Insert only if you want to use the distance fog option
#pragma multi_compile_local __ FOG_DISTANCE_ON

// Insert only if you want point lights to affect the material
#pragma multi_compile_local __ FOG_POINT_LIGHTS

// Insert only if you want shadows over the fog in the material
// (compilation can take a very long time!)
#pragma multi_compile_local __ FOG_SUN_SHADOWS_ON
```

c) locate the Input struct declaration and add the following fields if they're not already present:

```
struct Input {
    ...
    float4 screenPos;
    float3 worldPos;
}
```

d) add this line just after the Input struct declaration:

```
#include "Assets/VolumetricFog/Resources/Shaders/VolumetricFogOverlay.cginc"
```

(if you have moved VolumetricFog to another location please include the appropriate path)

e) Save changes.

**Step 2:** Add “**VolumetricFogMaterialIntegration**” script to the Camera where Volumetric Fog is attached.

**Step 3:** Finally, drag & drop the game objects that use the surface shader to the Materials list of the Volumetric Fog Material Integration inspector. **Important:** it's only necessary to have one object per modified shader added to the list. For example, if you have 2 lakes or rivers with same material, just add one of them as they will be sharing the same material hence the same properties.

### Compatibility with other shaders (vertex/fragment shaders)

Although Volumetric Fog & Mist provides methods to render before/after transparent objects, it's also possible to embed the fog effect into other complex shaders so the fog is rendered in the same pass. This solution solves some problems related with transparency.

To integrate Volumetric Fog & Mist in an existing vertex/fragment shader, please follow these steps:

**Step 1:** Apply the following changes to your shader:

a) insert the following lines just after any other #pragma line:

```
// Insert only if you want to use the distance fog option
#pragma multi_compile_local __ FOG_DISTANCE_ON

// Insert only if you want point lights to affect the material
#pragma multi_compile_local __ FOG_POINT_LIGHTS

// Insert only if you want shadows over the fog in the material
(compilation can take a very long time!)
#pragma multi_compile_local __ FOG_SUN_SHADOWS_ON
```

b) add this line just before the vertex shader declaration:

```
#include "Assets/VolumetricFog/Resources/Shaders/VolumetricFogOverlay.cginc"

(If you have moved VolumetricFog to another location please include the appropriate path)
```

c) Save changes.

**Step 2:** Add “**VolumetricFogMaterialIntegration**” script to the Camera where Volumetric Fog is attached.

**Step 3:** Finally, drag & drop the game objects that use the surface shader to the Materials list of the Volumetric Fog Material Integration inspector. **Important:** it’s only necessary to have one object per modified shader added to the list. For example, if you have 2 lakes or rivers with same material, just add one of them as they will be sharing the same material hence the same properties.

## Performance Tips

Volumetric Fog uses an extremely optimized ray-marching algorithm to provide “volumetric sense” fog in front of your player. This great effect comes at a performance price that makes Volumetric Fog & Mist not suitable for mobile devices (at least mobile devices to date).

However, we have added a few optimization parameters to provide you with more control regarding the performance vs quality:

- **Compute Depth:** disable this option unless it’s required in your scene as this will add an additional rendering stage (it’s optimized but nevertheless it’s an additional hidden camera rendering stuff like billboard trees).
- **Max Distance:** reduce the max distance property to a value that matches your scene/requirements so no extra fog is calculated unnecessarily.
- **Sky Haze:** reduce to 0 to improve performance.
- **Transparency Mode:** set to None to avoid extra overhead.
- **Downsampling:** increase this value to improve performance. A high value will produce visible artifacts around objects. A x2 value usually works well. A x4 or higher value works better with elevated/cloud fog. You can enable the “Edge improvement” option to reduce fog bleeding/pixelization around geometry borders.
- **Stepping:** controls the step of the ray-marching algorithm. Reduce this value to improve performance.
- **Stepping near:** additional factor for the ray-marching step applied only to close distances. Increase to improve the fog effect when stepping is reduced.
- **Distance** (starting Distance in Fog Geometry group): reduce to 0 to improve performance.
- **Height:** try to set the minimum height for the look you need.

Optional features that you may check they're either on (because you want them) or off (because you don't want them and want to make sure they are really disabled):

- **Point Lights:** reduce the number of point lights that can illuminate the fog (Point Light section). Make sure the range for non-used slots is zero.
- **Light Scattering:** uncheck the Enable toggle to disable the effect and increase performance. Make sure that if you enable it, the effect is visible when facing towards Sun, otherwise you will be wasting lot GPU cycles!
- **Fog Void / Fog of War:** disable (either setting the fog void radius to zero) or disabling fog of war feature (toggle in its section) if you're not using them to reduce GPU work.

## Common Issues (FAQ)

### When I generate a build, it stuck or takes lot of time!

This is a known issue with Unity when compiling shaders with many keywords.

You may want to reduce the shader complexity removing some keywords that you don't use by using the "Shader Options" section on top of the Volumetric Fog inspector. This is the recommended solution.

If you want to further optimize the shader, you can edit it and modify the VolumetricFog.shader file and look for lines that contains #pragma multi\_compile sentences:

- A) I don't need that feature: just remove the corresponding keyword or the entire #pragma line.
- B) I need that feature: you may remove the #pragma line AND remove all #if related statements in the shader so that code will always execute.

`#pragma multi_compile __ FOG_DISTANCE_ON`

Used when you set Starting Distance > 0.

`#pragma multi_compile __ FOG_VOID_SPHERE FOG_VOID_BOX`

Used when you enable fog void option.

`#pragma multi_compile __ FOG_AREA_SPHERE FOG_AREA_BOX`

Used when you enable fog area option..

`#pragma multi_compile __ FOG_HAZE_ON`

Used when you Sky Haze setting is > 0.

`#pragma multi_compile __ FOG_OF_WAR_ON`

Used when you enable Fog of War feature.

`#pragma multi_compile __ FOG_POINT_LIGHTS`

Used when the range setting of the optional point light setting is > 0.

`#pragma multi_compile __ FOG_SCATTERING_ON`

Used for light scattering (god rays, sun shafts).

`#pragma multi_compile __ FOG_SURFACE`

Used for terrain fit feature.

If you want to further optimize shader keywords at project level, consider using Shader Control asset (check it on the Asset Store).

### **I see banding artifacts, how can I remove them?**

To reduce banding artifacts (ie. circles on the sky), make sure your camera has HDR enabled and use the Dithering and Jittering options found in the Optimization & Quality settings section. You can customize the dither effect using the slider next to the toggle. For low density fog, you may need to increase the dithering intensity.

### **I see lot of noise, how can I smooth the effect?**

To reduce noise, you can use the Blur option under the Optimizations section at the bottom of the inspector. You may need to enable “Force Composition” to make the blur option visible.

Also, you can enable “Blue Noise” option in Shader Options as well.

### **Can I extend the maximum distance far beyond the 2000 limit?**

You can edit the custom inspector to increase that cap. However, increasing that value will impact performance. If you need to cover vast or far areas with fog, consider adding Dynamic Fog & Mist in addition to Volumetric Fog & Mist. Use Volumetric Fog & Mist for short distance effects and Dynamic Fog & Mist for the background.

### **When I enable transparency pass (Blend) I see strange colors or the particles don't show up. What is it?**

In this case, make sure Volumetric Fog & Mist camera scripts are first on the camera stack of components, before any other image effect.



## Additional Support

Please visit [kronnect.com](https://kronnect.com) for questions, support and more info.

## Dynamic Fog & Mist

You will find a copy of Dynamic Fog & Mist included in this package located in the Extras folder.

Dynamic Fog & Mist is somewhat less impressive visually than Volumetric Fog, but it will provide better performance.

If you're not interested in using Dynamic Fog & Mist, remember to remove the Extra folder completely.