

VOLUMETRIC LIGHTS

High-Definition Render Pipeline Edition

Quick Start Guide



Contents

Introduction	3
Requirements	3
Setup Instructions	3
Demo Scenes	5
Adding Volumetric Effect to existing lights	6
Adding new Volumetric Lights to the scene.....	7
Settings Description	8
Rendering section.....	9
Appearance section.....	9
Dust particles.....	10
Shadow Occlusion	10
Other Settings	10
Global Settings.....	11
Transparency and Semi-transparency support	13
Build Optimizations	14
Performance Tips.....	15
Notes	16
Orthographic Camera Support	16
Support	17

Introduction

Thanks for purchasing!

Volumetric Lights is a powerful, yet lightweight, volumetric lighting solution for Unity that adds realistic ambient to your scenes.

This version of Volumetric Lights has been created from scratch for High-Definition Render Pipeline. It won't work with other pipelines.

Requirements

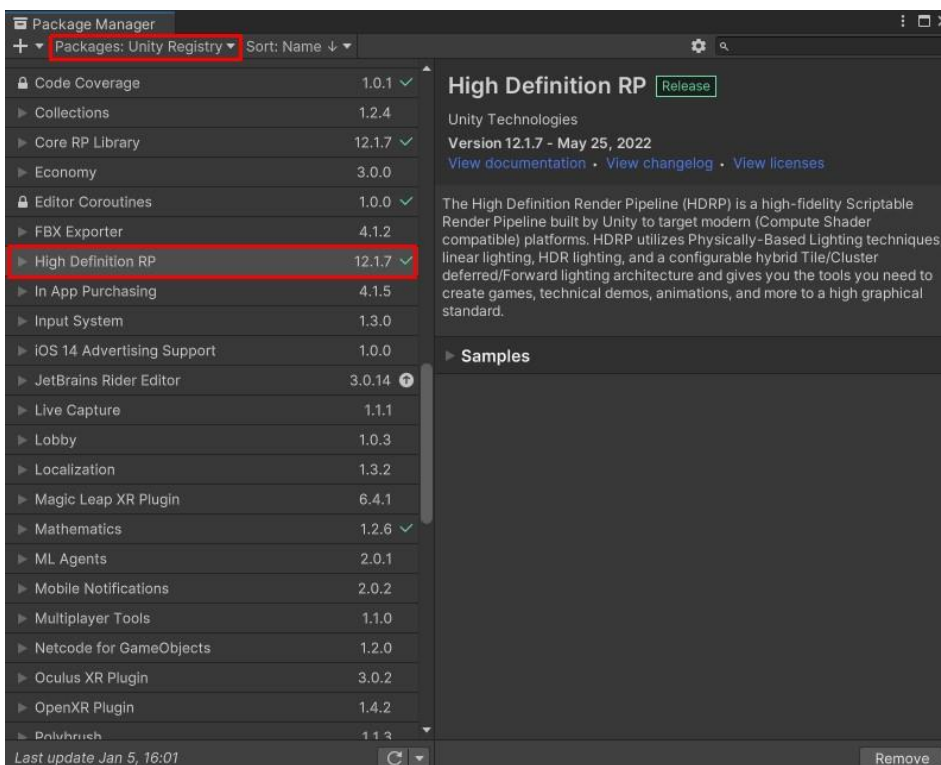
Volumetric Lights for High-Definition Render Pipeline requires:

- Unity 2021.3 or later
- High-Definition Render Pipeline 12.1.7 or later

Setup Instructions

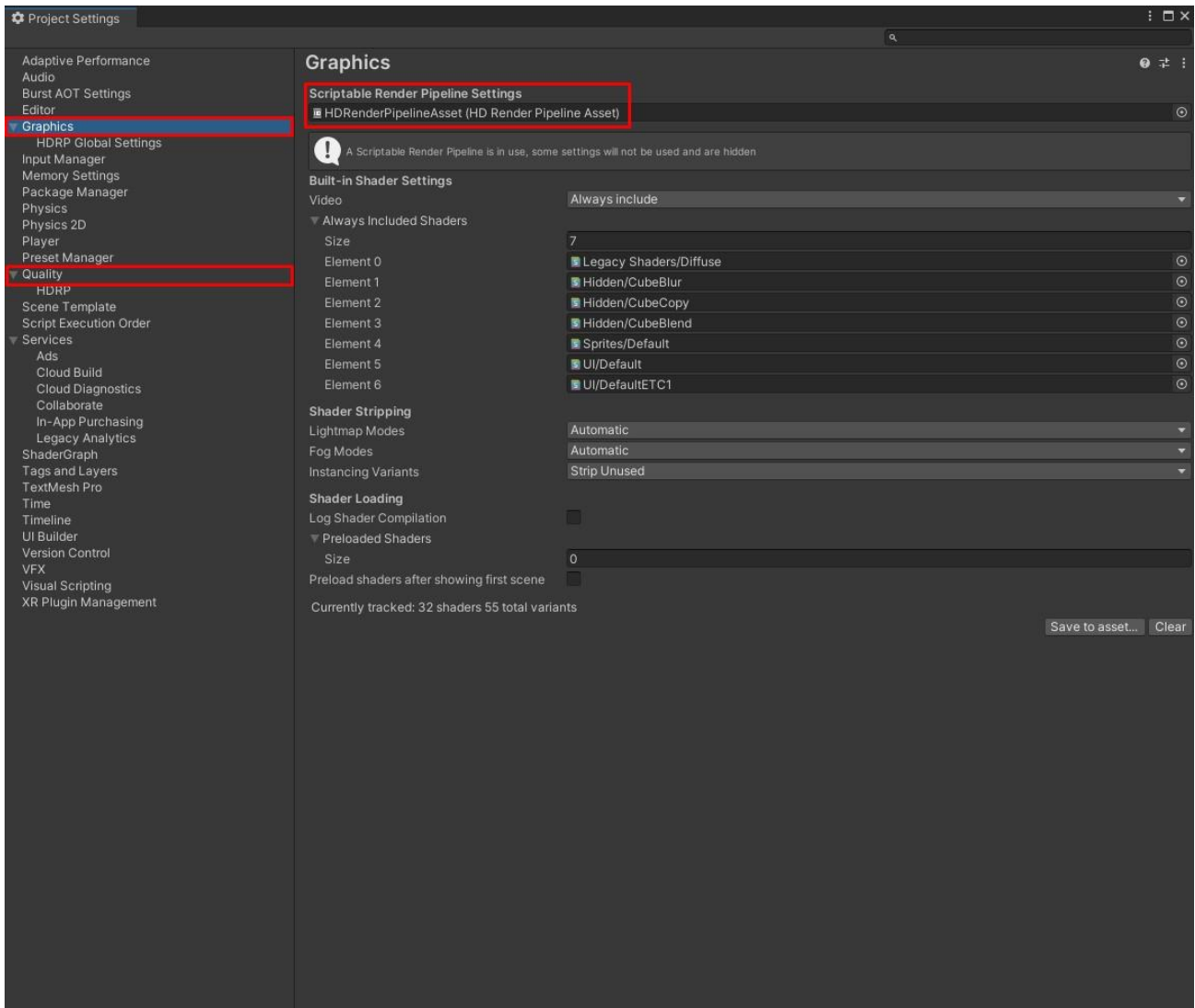
Step 1: Install High-Definition Render Pipeline

Go to Windows -> Package Manager. High-Definition RP (12.1.7 or later) and click "Install".



Step 2: Assign the High-Definition Render Pipeline Asset

Go to Project Settings -> Graphics and assign a High-Definition Render Pipeline asset.



Important note!

Check both “Project Settings / Graphics” and “Project Settings / Quality” sections since you can define HDRP settings overrides in Quality levels.

Now you can use Volumetric Lights!

Demo Scenes

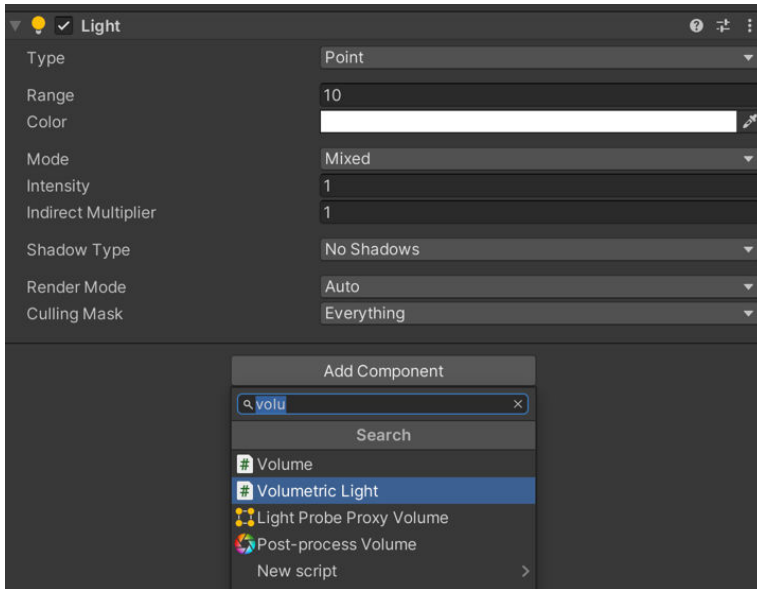
The asset includes three demo scenes to let you quickly familiarize with the different features. Please make sure you have completed the Setup explained in previous section.



Demo running on iPhone 7: https://youtu.be/s_hdefX4P9k

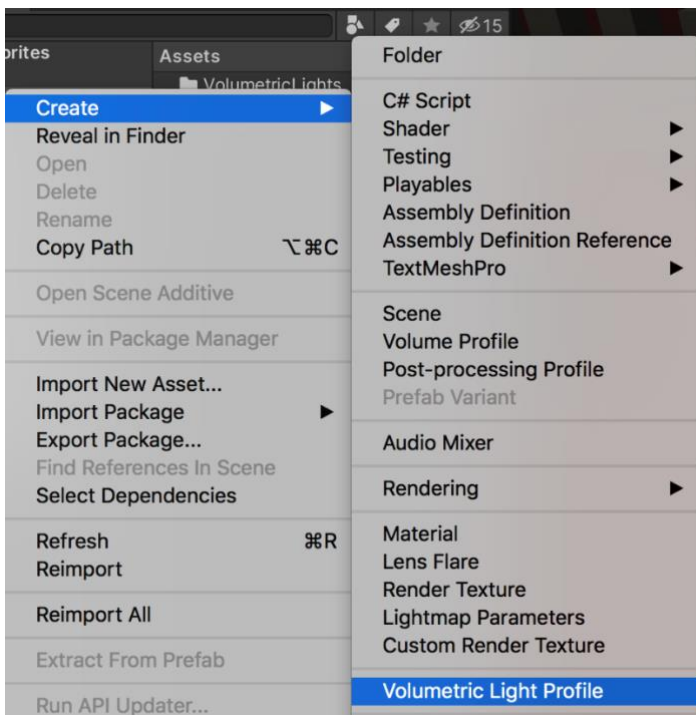
Adding Volumetric Effect to existing lights

Just add the “Volumetric Light” component to any of your lights. Please note that volumetric lights asset is currently compatible with point, spot, and area lights.



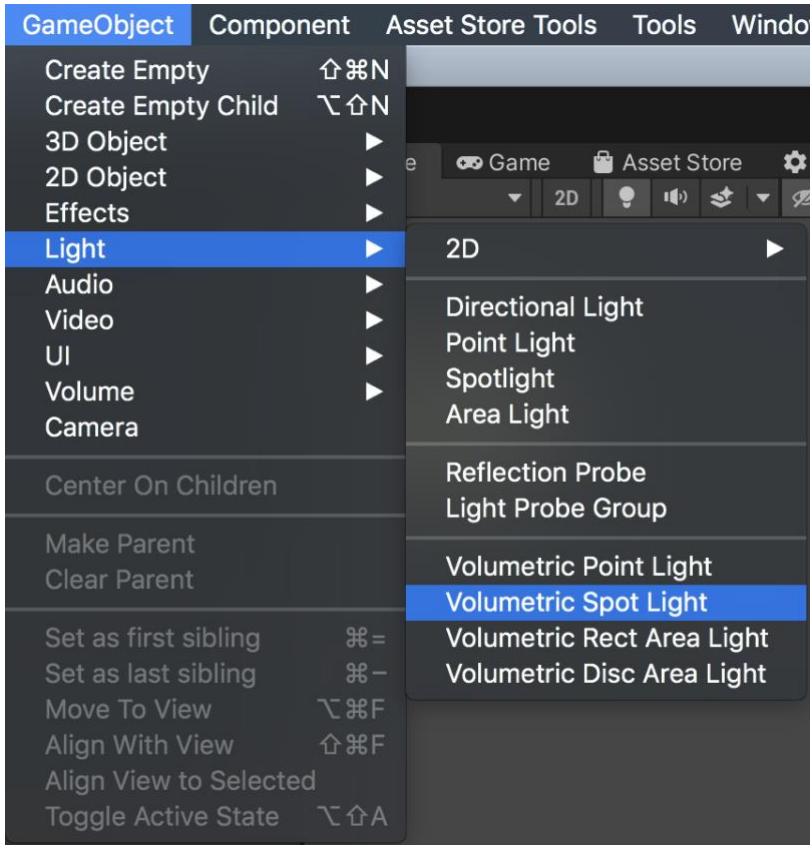
Optional: Create or assign a volumetric light profile

You can create a new volumetric light profile at any moment and assign it to the component. Right click in on the Project panel and select Create -> Volumetric Light Profile:



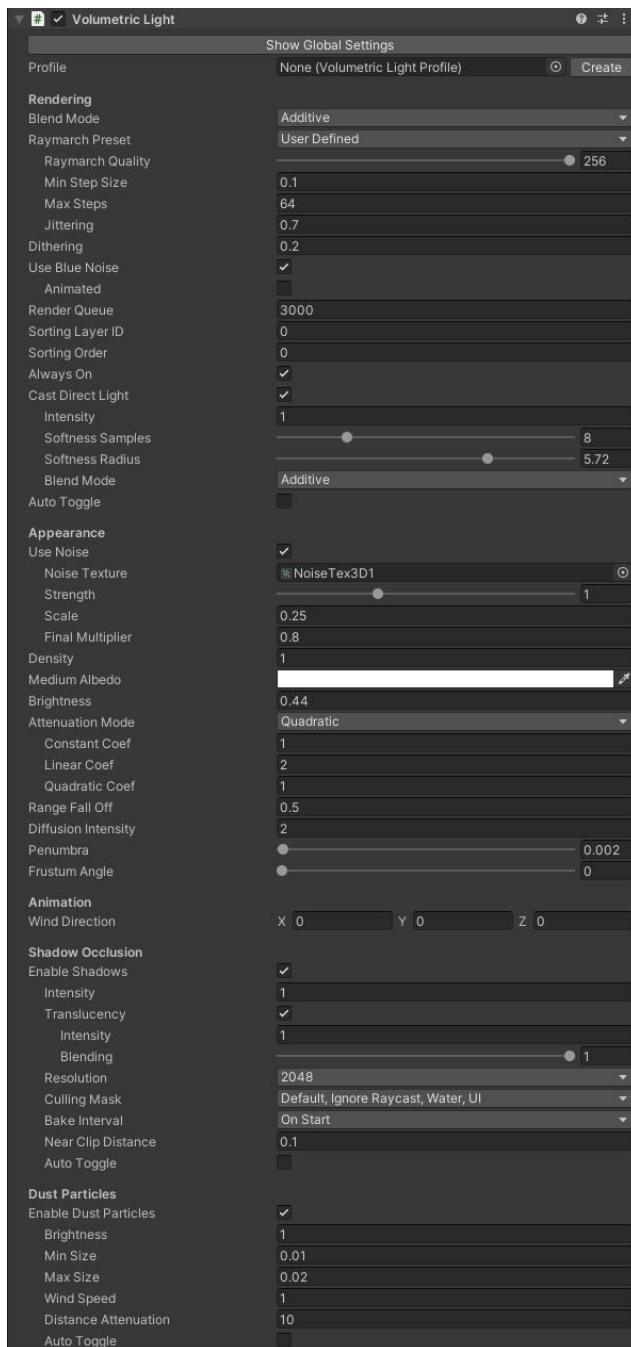
Adding new Volumetric Lights to the scene

You can create a normal light and add the “Volumetric Light” script to it as described in the previous section or you can just select the menu option **GameObject -> Light -> Volumetric *** Light** as shown in the picture below:



Settings Description

The following options are visible when you add a Volumetric Lights component:



Profile: this field shows which profile is being used.

Rendering section

- **Blend Mode:** blend or additive. Blend does the traditional alpha blending with the background while additive adds the light intensity to the existing pixels.
- **Raymarch Quality:** influences the number of iterations executed in the raymarching loop. The lower value, the better performance.
- **Min Step Size:** determines the minimum step size. Increase to improve performance (accuracy will be reduced as well).
- **Max Steps:** determines the maximum number of steps or samples taken along each ray. This is a maximum value which can be lowered to improve performance by ensuring there're no rays that take too many samples.
- **Dithering:** reduces banding effect.
- **Jittering:** reduces banding effect (especially useful when shadows are enabled).
- **Use Blue Noise:** enables blue noise for jittering computation. It adds an additional texture fetch but reduces moiré pattern caused by regular jittering algorithm. Use only if you get a noticeable improvement.
- **Render Queue / Sorting Layer / Sorting Order:** the volumetric effect renders in world space in the transparent queue. These settings allow you to customize the order of the rendering.
- **Always On:** enabling this option will force the volumetric effect to be visible regardless of the light enable state. This option is useful to create fake volumetric lights, where only the volumetric effect is visible but the light component itself doesn't add any overhead since you can disable it.
- **Cast Direct Light:** this option can only be used in deferred rendering mode (enable it on the camera). It will illuminate the objects reached by the volumetric effect and can be used along other effects like translucency (see the Church demo scene for an example).
- **Auto Toggle:** enables or disables the light in function of the camera distance.

Appearance section

- **Use Noise:** uses an internal noise texture instead of assuming a homogeneous medium.
- **Noise Texture:** the asset will use a default 3D noise texture located in Resources/Textures folder. You can find more noise textures in Demo/NoiseTextures folder. You can also provide your own textures.
- **Density:** density of the medium used during the raymarching loop. The greater value, the opaquer effect. A low-density value used in a very large light space can impact performance as the ray march loop needs to run many steps.
- **Diffusion Intensity:** controls the intensity of the diffusion of the light when looking directly to the source through the medium.
- **Range Falloff / Border:** controls the falloff of the volume. Border has no effect on point lights.
- **Attenuation Mode:** sets the decay curve applied to the light intensity with distance to the emission point. Choose between "Simple" or "Quadratic". Use quadratic for a more realistic result in which you can configure 3 operands: constant, linear and quadratic coefficients of the formula.

Dust particles

The component will use a particle system with a custom particle shader that's aware of the lighting volume geometry.

Shadow Occlusion

This option adds shadow support to the effect.

- **Shadow Intensity:** amount of light cancelling due to shadows.
- **Shadow Resolution:** resolution of the shadow map. Usually, you don't need a very high value to create a good effect. Try to use the lower texture size as possible.
- **Shadow Culling Mask:** specifies which objects cause shadows.
- **Shadow Bake Interval:** specifies if shadow map is computed during start only or on every frame. Note that if the light is moved or rotated, the shadow map will be automatically recomputed.
- **Shadow Bake Mode** (only for point lights): point light shadows are expensive – this option let you choose between computing shadows on a half sphere oriented to player (faster) or a cubemap (slower). If cubemap option is selected, you can choose to spread the rendering across several frames (1 face per frame) or render all 6 faces every frame.
- **Shadow Orientation** (only for point lights): baked shadows captured in a half sphere (180°) for point lights requires orienting the camera. This option let you specify a fixed direction for the alignment of such half sphere or automatically orient it to the player camera.
- **Shadow Near Distance:** specifies the near clip plane of the camera used to create the shadow map. This setting allows you to skip objects that are too near to the light.
- **Translucency:** when enabled, colors from transparent objects that have the "Volumetric Lights Translucency" script attached will be used in the effect. Important: the object must be visible from the light point of view (common error include using a single sided window that's not visible from exterior).

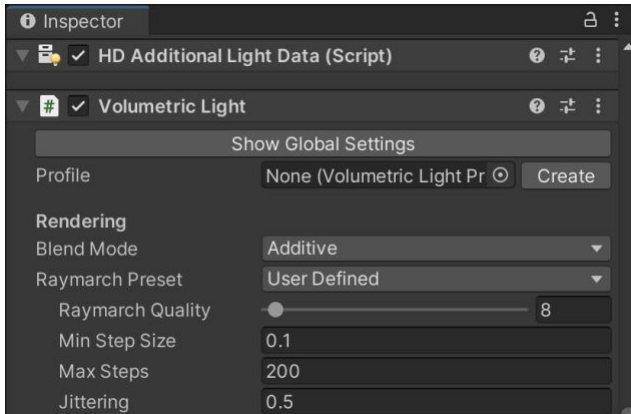
Pro tip! When **Shadow Bake Interval** is set to **OnStart**, you can force shadows update anytime calling `ScheduleShadowCapture()` method of the `VolumetricLight` component attached to the light.

Other Settings

- **Use Custom Size:** let you override light properties like range or size.
- **Use Custom Bounds:** let you customize the boundary of the light effect effectively clipping the volumetric effect. This is useful to clamp the volumetric effect to certain corners for example. When this option is enabled, you will see the actual bounds in the Scene View and you can use the handlers of the cube to adjust it size.

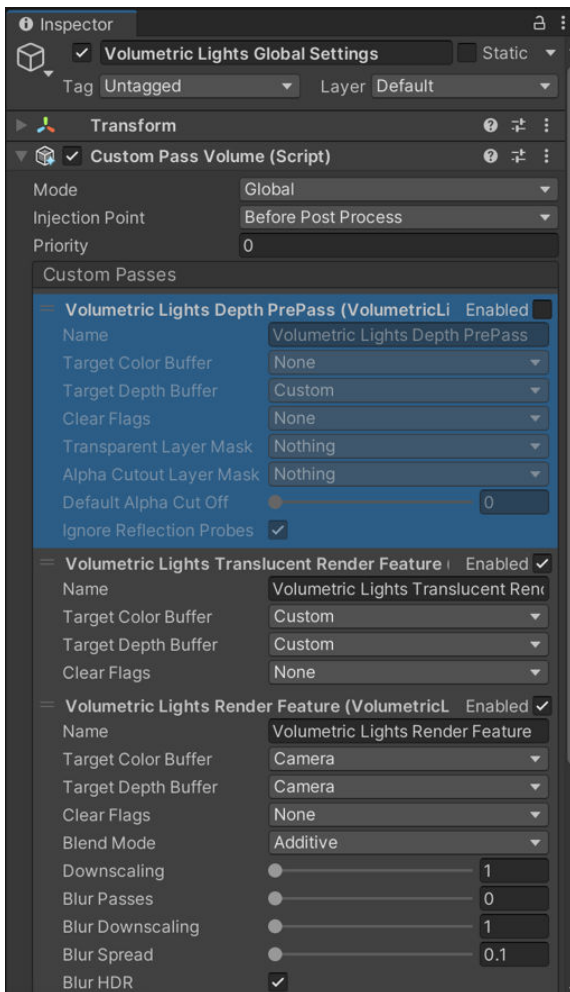
Global Settings

Some global settings (common to all volumetric lights in the scene) can be accessed by clicking on the “Show Global Settings” button at the top of any Volumetric Light inspector:



The “Global Settings” is a special volume which contains 3 custom passes:

- **Volumetric Lights Depth PrePass:** performs a custom depth pass to compute transparent objects depth (useful if you need to treat some transparent objects like opaque, like lakes). This custom pass is disabled by default.
- **Volumetric Lights Translucent Render Feature:** required by the translucent shadow maps. This custom pass is enabled by default but will only be executed on lights that have the “Translucency” option enabled in the inspector.
- **Volumetric Lights Render Feature:** this custom pass lets you customize the result of the volumetric effect by choosing the blending mode, smoothing the result (reduces noise) and/or applying a downscaling factor (which improves performance if required).



Volumetric Lights Depth PrePass options:

- Target Color Buffer/Target Depth Buffer/Clear Flags: keep the default values.
- Transparent/Alpha Cutout Layer Mask: use them to specify which objects would be included in this pass. For cutout shaders, the default Alpha Cutoff will be used only if the material does not expose a “_CutOff” property.

Volumetric Lights Render Feature:

- **Blend Mode:** the blending mode for the final composition.
- **Brightness:** global brightness for the final composition (this value multiplies to the actual brightness of each individual light).
- **Downscaling:** reduces the resolution of the intermediate light buffer. This option will improve performance.
- **Blur Passes:** number of blur iterations. The higher number, the softer the result.
- **Downscaling:** reduces the size of the blur buffers. This option improves performance at quality expense.
- **Blur Spread:** increases the kernel radius of the blur pass. Can produce softer results in combination with the blur passes and downscaling.
- **Blur Edge Preserve:** makes the edges of the objects sharper. This setting is very useful when the blur used to make the shadows look smooth reduces the definition of the objects.

- **Blur HDR:** uses a high range buffer during blur passes. Disable this option to improve blur operations performance but it can affect the visuals of the volumetric light.

Transparency and Semi-transparency support

Volumetric Lights relies on the scene depth buffer to compute where objects are. Since transparent objects do not write to depth buffer, they might get overdrawn by the volumetric light effect.

There're two ways to solve this issue:

- 1) **Render queues:**

You can change the render queue of the volumetric light, so it has a lower value than your transparent objects. Note that transparent queue starts at 3000. With this solution, volumetric light effect will render always before your transparent objects, regardless the position in the scene.

If this option works for you, perfect! This solution is fast and easy to implement.

- 2) Alternatively, you can use the "Volumetric Light Depth Pre-Pass Render Feature" as explained in the previous section (see "Global Settings").

Build Optimizations

Volumetric Lights uses a few shader keywords to optimize the execution of the effects. They're needed to enable the desired functionality. However, each keyword multiplies the number of shader variants so you may want to disable or remove some of them manually if you want to optimize your build. As a result, building the shaders for the first time can take a few minutes. Once it's built, Unity caches the compilation so next builds are much faster.

To reduce the number of shader variants and speed up the build, you can remove some keywords related to VolumetricLights options that you're not using in your project.

Locate and edit the file VolumetricLightHDRP.shader inside VolumetricLights/Resources/Shaders folder. The following lines declare the keywords. Feel free to comment out or remove any of these lines or keywords to reduce the number of shader variants:

```
#pragma multi_compile _ VF2_DEPTH_PREPASS // used only if depth prepass is enabled
#pragma multi_compile_local _ VL_NOISE // used if Noise option is enabled
#pragma multi_compile_local _ VL_BLUENOISE // used if Blue Noise option is enabled
#pragma multi_compile_local _ VL_PHYSICAL_ATTEN // used by Quadratic attenuation
#pragma shader_feature_local VL_CUSTOM_BOUNDS // used by Custom Bounds option
```

The following keywords are required per light type. If you don't use a specific light type, you can remove that keyword:

```
#pragma multi_compile_local VL_SPOT VL_SPOT_COOKIE VL_POINT VL_AREA_RECT
VL_AREA_DISC
```

If you don't use shadows in the volumetric lights, you can remove this line. Or if you don't use the cubemap based shadows (only for point lights), you can remove the VL_SHADOWS_CUBEMAP keyword:

```
#pragma multi_compile_local _ VL_SHADOWS VL_SHADOWS_TRANSLUCENCY
VL_SHADOWS_CUBEMAP
```

Important: if you upgrade Volumetric Lights to a newer version, any changes done to the VolumetricLightHDRP.shader will be lost! Remember to remove any unwanted keywords again after upgrading.

Performance Tips

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

- Reduce Raymarch Quality: reducing this value will indeed reduce the number of texture fetches per pixel.
- Increase “Min Step Size” value.
- Decrease “Max Steps” value.
- If Shadow Occlusion is used, make sure the shadow culling mask only includes objects that you want to cast shadows. By default, shadow culling mask doesn’t include Transparent FX layer as volumetric lights are set to that layer to avoid being rendered again by the occlusion cameras.
- Use the Downscaling option in Global Settings to improve overall performance.
- Disable “Use Blue Noise” option.
- Reduce the range of the volumetric effect by using the “Use Custom Size” and specifying new range or size
- Set Shadow Bake Interval to “On Start” whenever possible.

Notes

Orthographic Camera Support

To enable fully support of orthographic camera, locate and edit the Options.hsl file and uncomment this line:

```
//#define ORTHO_SUPPORT
```


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.