

# *Dynamic Fog & Mist*

**Kronnect**

## Contents

Introduction .....	3
How to use this asset .....	3
Demo Scenes .....	5
Fog Effect Types.....	5
Shaders used by each effect type .....	6
Fog materials .....	7
Special Features (only Image Effect) .....	8
VR and Single Pass Stereo Rendering.....	8
Fog Volumes .....	8
Fog of War .....	8
Gradient Fog.....	8
Special Features (Dynamic Fog of War).....	9
Using the Dynamic Fog of War prefab .....	9
Changing Post Processing profile values from code .....	10
Enhanced compatibility .....	11
Compatibility with Gaia .....	11
Support .....	11
FAQ.....	11

## Introduction

### Thanks for purchasing!

**Dynamic Fog & Mist** is a collection of shaders that adds live, moving **Fog**, **Mist** and **Sky Haze** to your scenes making them less dull and boring.

This asset contains:

- **A full-screen image post-processing image effect** with 7 variants.
- **15 materials with integrated fog effect** in custom shaders.
- **A fog of war prefab** optimized for top-down perspective.
- **PostProcessing Stack Extensions** so you can use the effect directly in the Unity Post Processing pipeline.

## How to use this asset

Choose one way:

1) **Traditional camera-based script:** just add "DynamicFog" or "DynamicFog Exclusive" script to your camera. It will show up in the Game view. Customize it using the custom inspector.

- Dynamic Fog & Mist: recommended script for most situations.
- Dynamic Fog & Mist Exclusive: variant of the main script with Render Scale option which results in better performance on mobile.

### 2) **Post-Processing Extensions (compatible with LWRP and URP 7.2.0)**

Support for Post-Processing Stack v2 is included. This option requires Unity 2018.1 or later (LWRP) or Unity 2018.3 or later (URP). You also need to have PostProcessing Stack V2 package installed (you can install it from the Package Manager).

Then import the package named "PostProcessingSupport.unitypackage" included in DynamicFog folder.

Important: if you opt for the post-processing extensions, don't add the script to the camera: Dynamic Fog effects will be available as pluggable effects in Post-Processing Profiles and Volumes.

Setup video for URP: <https://youtu.be/QiTM4YjLQT8>

### 3) **Custom Fog materials**

A collection of materials and shaders are provided that have the fog effect integrated in their shaders so they offer the fastest performance available! However, if you choose this path, then you need to use these materials and not the image effects or Post-Processing Extensions above.

**Important: you should use one of the options above, do not mix them.**

Explanation follows:

- The image effect is controlled by the script `DynamicFog.cs` that you add to your camera. The image effect overlays a layer of fog on top of the image every frame. The fog is computed using the depth of each pixel on the screen rendered by the camera. The farther the pixel the thicker the fog.
- The Post-Processing Extensions are the same image effect shaders but rewritten to work with Unity Post-Processing Stack. They are also compatible with SRP.
- On the other hand, the fog materials use custom shaders that integrate the fog calculation. There's no full screen blit nor full screen fog computation but instead the color of the objects are modified depending on the distance to the camera. Since there's no separate full-screen blit pass, using fog materials could be faster on old mobile devices.
- Finally, if you need a top-down view, fog of war effect that occludes game area and allows you to clear custom zones, the fog of war prefab provides you the best performance.

Only if you're really concerned about performance (eg. mobile), use the fog materials or the fog of war prefab. Otherwise use the other options (full-screen image effect or post-processing stack extensions).

Demo scenes 1-4 show examples of the full-screen image effect whereas demo scene 5 shows how to use fog materials to create a simple arcade game.

## Demo Scenes

Demo scenes for the full-screen image effect mode depends on Standard Assets' Camera package, so it needs to be imported along the Demos package (double click on it to uncompress).

Just load any included demo scene and click "Play". In FPS demos 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).

Finally there's a Fog of War demo scene that uses the fog of war prefab (only works for the top-down perspective).

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

## Fog Effect Types

The effect type selector in the inspector allows you to select one of the shader variants:

- **Desktop Fog + Sky Haze:** this effect provides a fast fog effect plus sky haze options for desktop platforms. It uses two different noise textures to produce a soft animated fog effect.
- **Desktop Fog Plus + Sky Haze:** this effect provides a fog effect using a volumetric approach improving the fog coverage when there's no geometry in front of the camera. It also uses sky haze.
- **Mobile Fog + Sky Haze:** optimized version of Desktop Fog + Sky Haze for mobile devices. Uses a single noise texture.
- **Mobile Fog (No Sky Haze):** same than Mobile Fog but without sky haze effect.
- **Mobile Fog (Simplified):** uses a volumetric approach and no noise textures, hence, no animation nor turbulence is supported in this mode. This can be the best choice for fast static fog.
- **Mobile Fog (Basic):** uses a even simpler version of Mobile Fog (Simplified) aimed for the low end mobile devices.
- **The Orthogonal** versions are variants that computes depth and height separately and mix them. They are intended for isometric or games with special perspectives.

## Shaders used by each effect type

Fog variant	Perspective camera	Orthographic camera
<b>Desktop Fog Plus With Sky Haze</b>	DFGDesktopPlus	DFODesktopPlus
<b>Desktop Fog Plus Orthogonal</b>	DFGDesktopPlusOrthogonal	DFODesktopPlusOrthogonal
<b>Desktop Fog + Sky Haze</b>	DFGDesktop	DFODesktop
<b>Mobile Fog (Simplified)</b>	DFGSimple	DFOSimple
<b>Mobile Fog + Sky Haze</b>	DFGWithSky	DFOWithSky
<b>Mobile Fog (No Sky Haze)</b>	DFGOnlyFog	DFOOnlyFog
<b>Mobile Fog Orthogonal</b>	DFGOrthogonal	DFOOrthogonal
<b>Mobile Fog (Basic)</b>	DFGBasic	DFOBASIC

Note: you can remove unused materials and shaders from the Resources folder.

## Fog materials

**To use Dynamic Fog & Mist fog materials (not to be combined with the image effect! – see demo scene 5):**

1. Select and duplicate any fog material inside DynamicFog/Resources/Materials (see below for names, they start with DFM).
2. Customize the material properties (color, texture or metallic/glossiness, depending on the chosen material).
3. Assign the material to your game objects (you can mix different material types).
4. Create an empty game object and add “DynamicFogManager.cs” script to it. This script controls the shaders of the fog materials and allows you to customize the fog properties for all materials from a centralized point.

The current fog materials available in DynamicFog/Resources/Materials are:

- **DFMStandardShader**: a derivation of the Standard Shader which integrated our fog effect.
- **DFMStandardShaderNormal**: same but adds bump map.
- **DFMStandardShaderNormalOcclusionMap**: same but adds occlusion map.
- **DFMStandardShaderNormalOcclusionIntegrated**: in this case the alpha component of the main texture is used for occlusion.
- **DFMStandardShaderSolidColor**: same but does not use textures, just single color (faster).
- **DFMLambert**: a fog shader that uses simple lighting (faster).
- **DFMLambertSolidColor**: same but does not use textures (even faster).
- **DFMUnlit**: does not use lighting, just texture, color and fog effect (faster yet!)
- **DFMUnlitSingleColor**: same but does not use texture (the fastest shader!)

These materials are also available with a reflection texture support:

- **DFRStandardShader**: a derivation of the Standard Shader which integrated our fog effect.
- **DFRStandardShaderSolidColor**: same but does not use textures, just single color (faster).
- **DFRLambert**: a fog shader that uses simple lighting (faster).
- **DFRLambertSolidColor**: same but does not use textures (even faster).
- **DFRUnlit**: does not use lighting, just texture, color and fog effect (faster yet!)
- **DFRUnlitSingleColor**: same but does not use texture (the fastest shader!)

It's possible to assign a reflection texture to these materials (using `mat.SetTexture(“_ReflectionTex”, texture)`) to blend reflections along with the fog. The free asset Realtime Reflections provide a simple way to expose the reflection textures to these materials: <https://www.assetstore.unity3d.com/en/#!/content/21730>

## Special Features (only Image Effect)

### VR and Single Pass Stereo Rendering

Dynamic Fog & Mist fully supports VR and Single Pass Stereo Rendering. It has been tested with Oculus DK2 headset and both Oculus and OpenVR SDKs.

**Important:** if you enable Single Pass Stereo Rendering in Player Settings, make sure to review Dynamic Fog & Mist inspector on your camera and ensure the checkbox “Single Pass Stereo” shown at top of inspector is marked. This checkbox is automatically set based on the Player Settings so no need to change it, just ensure it reflects current player settings. Reason for this checkbox is that Unity does not provide a way to check if Single Pass Stereo is enabled at runtime (from scripts) so this redundant checkbox was added. Just make sure it's in line with your Player Settings.

### Fog Volumes

You can define special zones (fog volumes) where fog alpha will automatically change. Create a fog volume from the menu `GameObject / Create Other / Fog Volume`. Position the fog volume over the desired area, edit the collider bounds and set the desired fog alpha and transition duration in the inspector.

### Fog of War

You can also set any number of void areas just calling **SetFogOfWarAlpha** method of the DynamicFog script. Just pass the world space position, the radius 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).

Texture Size defines the resolution of the fog of war transparency texture. Increase this size to improve the precision of the clearings.

Call **ResetForOfWar** to reset the cleared areas back to normal. Also **GetFogOfWarAlpha** gives you the value at any world space position.

You can also add fog of war holes by creating special “Dynamic Fog Of War Hole” objects. Right click in the hierarchy or select top menu `GameObject -> Create Other -> Dynamic Fog -> Dynamic Fog of War Hole`.

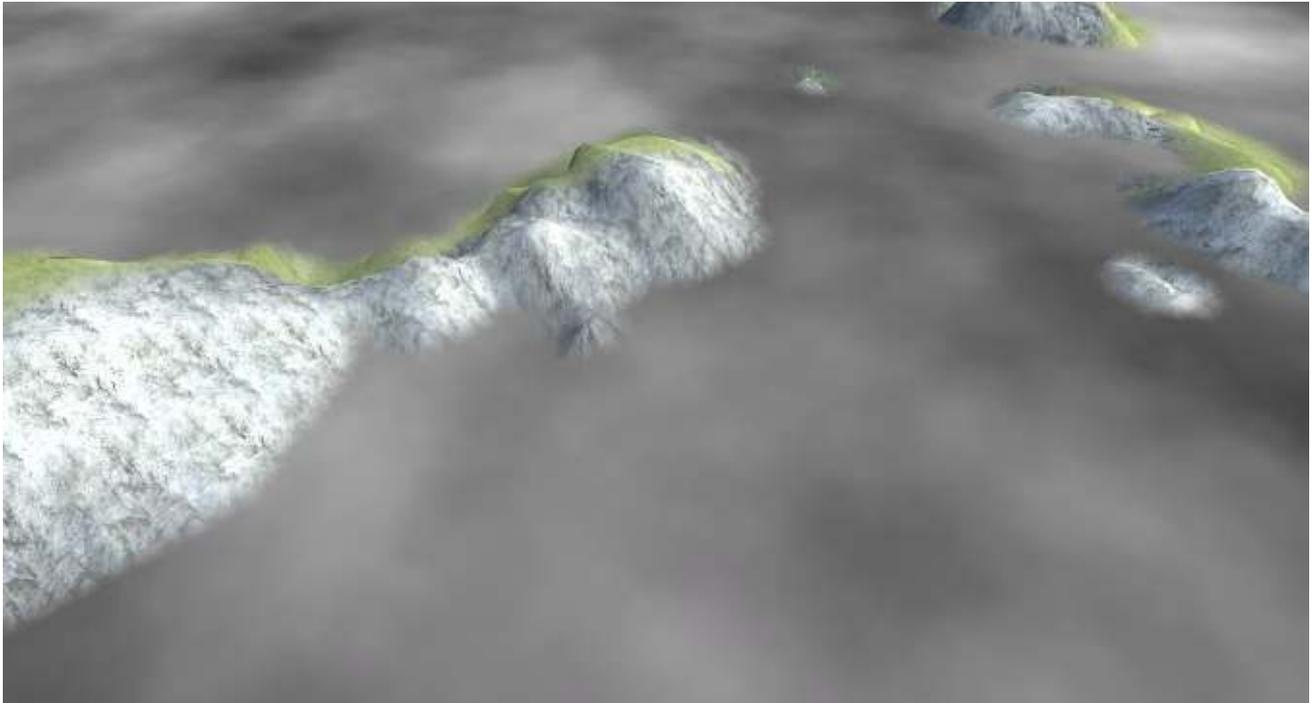
### Gradient Fog

Dynamic Fog & Mist exposes two color selectors to create artistic fog gradients. Just experiment with them!

## Special Features (Dynamic Fog of War)

### Using the Dynamic Fog of War prefab

Dynamic Fog also includes a custom prefab optimized for Fog of War usage:



This prefab does not use any image effect but a custom shader that works much faster on mobile. Please check demo scene 7 for a full example.

To use this feature in your game, just locate and drag the prefab `FogOfWarLayer` located in `DynamicFog/Resources/Prefabs` folder to your scene. Position the plane over the terrain or area to cover. Then, the following API is available:

```
DynamicFogOfWar fog = DynamicFogOfWar.instance;
```

```
fog.SetFogWarTerrainBoundary(terrain, borderSize)
```

This function smoothly cuts the fog crossing the terrain. This call is optional and only useful if you want to produce a smooth cut like in the image above.

```
fog.SetFogOfWarAlpha(worldPosition, radius, alpha)
```

Call this function to clear or set a custom fog transparency on any world position.

```
fog.ResetFogOfWar()
```

Call this function to restore fog of war to fully opaque.

```
fog.fogOfWarTextureSize
```

This property determines the resolution of the internal alpha control texture.

## Changing Post Processing profile values from code

If you're using the Post Processing Stack v2 version of the asset (see "How to Use this Asset" section), you can use this code to change the values of the profile at runtime.

```
using UnityEngine;
using DynamicFogAndMist;
using UnityEngine.Rendering.PostProcessing;

namespace DynamicFogAndMistDemos {
    public class DynamicFogDistanceColor : MonoBehaviour {

        DynamicFogPPS _fogSettings;
        PostProcessVolume _volume;

        /// <summary>
        /// Returns a reference to the Post Processing Volume which contains Dynamic Fog
effect.
        /// </summary>
        /// <value>The volume.</value>
        public PostProcessVolume volume {
            get {
                if (_volume == null) {
                    _volume = FindObjectOfType<PostProcessVolume>();
                }
                return _volume;
            }
        }

        /// <summary>
        /// Returns a reference to the settings of Dynamic Fog & Mist in the Post
Processing Profile
        /// </summary>
        /// <value>The shared settings.</value>
        public DynamicFogPPS fogSettings {
            get {
                if (_fogSettings != null) return _fogSettings;
                PostProcessVolume v = volume;
                if (v == null) return null;

                bool foundEffectSettings =
v.sharedProfile.TryGetSettings<DynamicFogPPS>(out _fogSettings);
                if (!foundEffectSettings) {
                    Debug.Log("Cant load Dynamic Fog settings");
                    return null;
                }
                return _fogSettings;
            }
        }

        // Update is called once per frame to transition the color from white to blue in a
loop
        void Update() {
            float t = Mathf.PingPong(Time.time * 0.1f, 1f);
            fogSettings.color.value = Color.Lerp(Color.white, Color.blue, t);
            fogSettings.color.overrideState = true;
        }
    }
}
```

## Enhanced compatibility

### Compatibility with Gaia

Dynamic 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 Dynamic Fog & Mist in just one click.

## Support

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

## FAQ

### How can I render the fog behind particles?

Just edit DynamicFog.cs and add [ImageEffectOpaque] before OnRenderImage method.

### At night, fog still is showing in same color, how can I make it react to ambient light?

Assign a directional light or Sun gameobject to the Sun property in the inspector (eg. assign the Sun gameobject of Time Of Day plugin). Once assigned, DynamicFog will lerp between fog color and black depending on the Y-axis angle of the directional light.

### What's Dynamic Fog Exclusive script for?

Starting version 6, Dynamic Fog & Mist includes two scripts:

- Dynamic Fog (normal usage, recommended for most situations).
- Dynamic Fog Exclusive. This new script is a variant of the normal script optimized for performance. It includes a "Downsampling" parameter in the inspector that allows you to reduce quality to improve speed.