Documentation

3.0.0

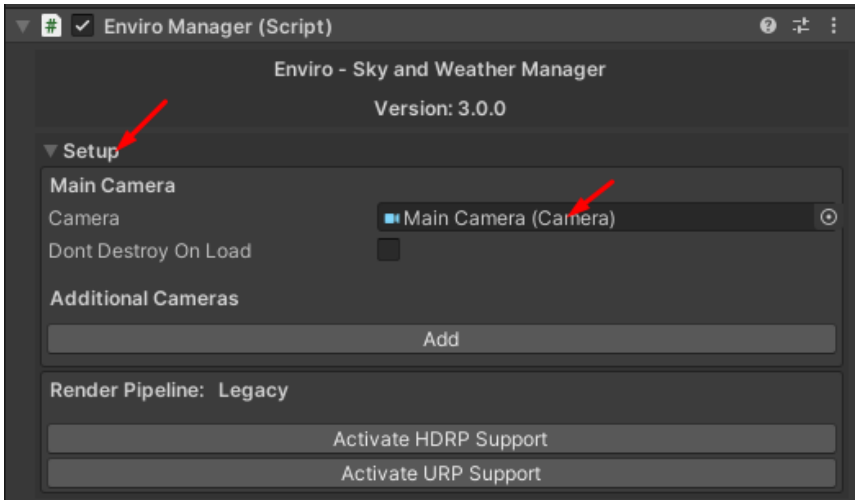
Thank you for buying

Enviro 3 - Sky and Weather

Quick Start

1. *Please take a look into the Enviro 3 – Sky and Weather folder and drag and drop the Enviro 3 prefab into your scene.*

2. *Now click on the Enviro 3 object in your scene hierarchy and select your „Main Camera“ in the „Setup“ -> „Camera“ field.*



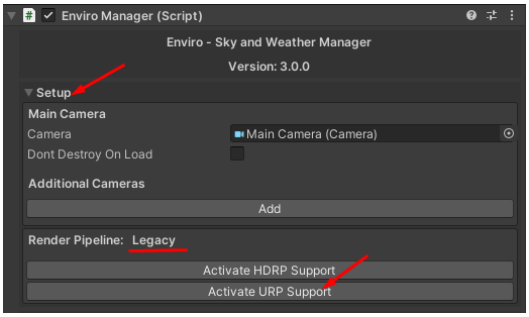
3. *You might want to check your scene for your old directional light and deactivate it.*

4. *You can start your scene now when using built-in pipeline. Otherwise check following URP and HDRP Support section.*

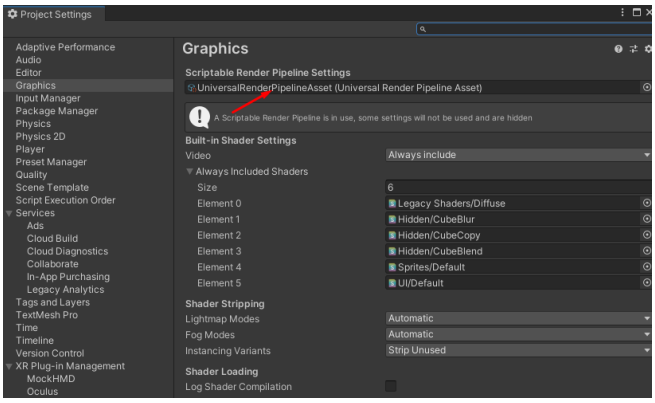
Please also check the Tonemapping! section to get best results!

URP Support

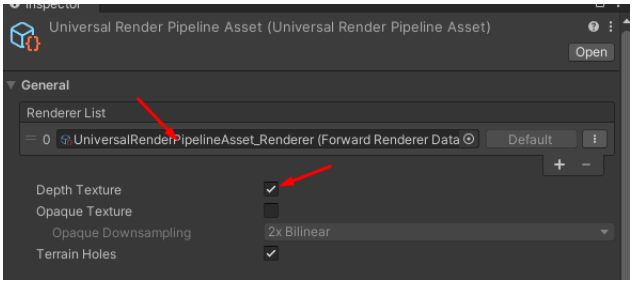
1. *Follow the „Quick Start“ section above. Then click on the „Activate URP Support“ button in your „Enviro Manager“ -> „Setup“ section. It should recompile your project now. So wait for a moment. It’s finished when it show: Render Pipeline: URP*



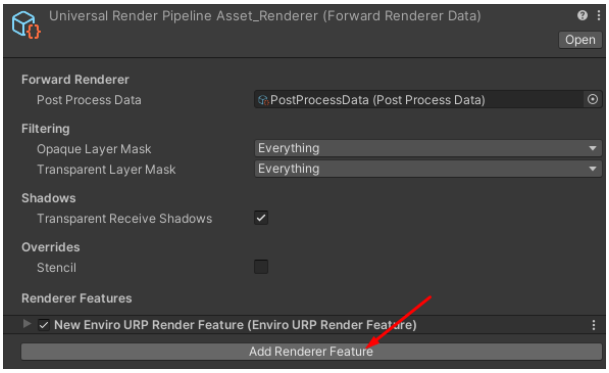
2. *Open your Project Settings -> Graphics section and double click on your Render Pipeline Asset.*



3. *Activate „Depth Texture“ option and open your assigned URP Renderer.*

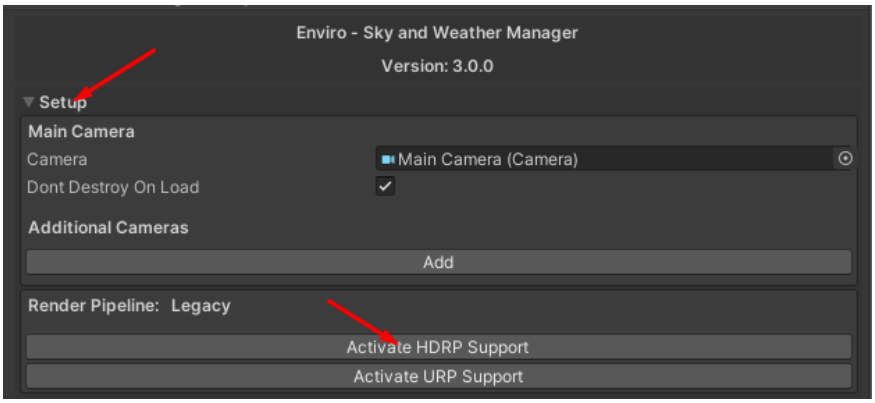


4. *Add the „Enviro URP Render Feature“ to your renderer.*

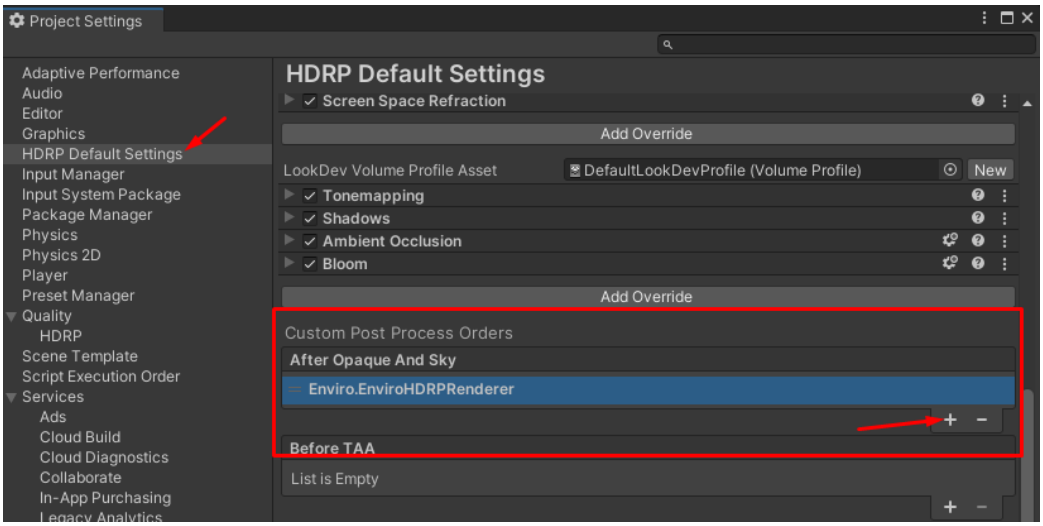


HDRP Support

1. *Follow the „Quick Start“ section above. Then click on the „Activate HDRP Support“ button in your „Enviro Manager“ -> „Setup“ section. It should recompile your project now. So wait for a moment. It’s finished when it show: Render Pipeline: HDRP*



2. *Open your „Project Settings“ -> „HDRP Default/Global Settings“ and scroll down to „Custom Post Process Order“ section. Add „Enviro.EnviroHDRPRender“ to the „After Opaque and Sky“ order section.*



Tonemapping!

Enviro does not clamp the brightness of ist effects at all. It was designed to be used with tonemapping to get best results and realistic stunning visuals.

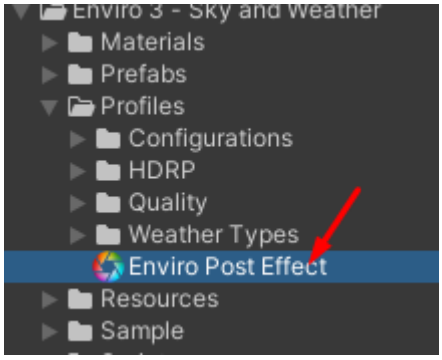
Built-in Pipeline:

Please consider to download and setup the Unity post processing package for your Enviro scenes.

Please check this link if your are new to Unity post processing:

https://docs.unity3d.com/Packages/com.unity.postprocessing@2.1/manual/Quick-start.html

Enviro already inlcudes a simple post processing profile for your scenes. Create a volume and assign the „Enviro Post Effect“.



URP:

*URP already includes a post processing system out oft he box. Just create a new „Global Volume“ with a right-click in your scene hierarchy. Then create a new profile and add the „**Tonemapping**“ module. Select „**ACES**“ mode for best results.*

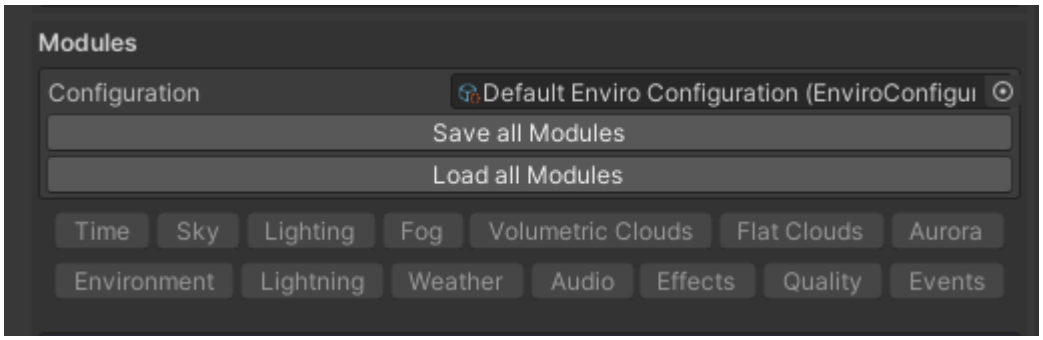
*You also need to activate „**Post Processing**“ option in your camera to make it work!*

HDRP:

HDRP already has tonemapping activated by default. If you disabled or need more control you can add the „**Tonemapping**“ module to one of your HDRP Volumes.

Modules

Enviro 3 was crafted with an modular design. So you can add and remove different modules here you might not need in your Enviro Manager.



Configuration *will hold all your settings of all your models. You can switch this out to easily load a whole different Enviro 3 setup for example.*

Save all Module*:*
A click on this button will save all the module settings in its assigned presets.

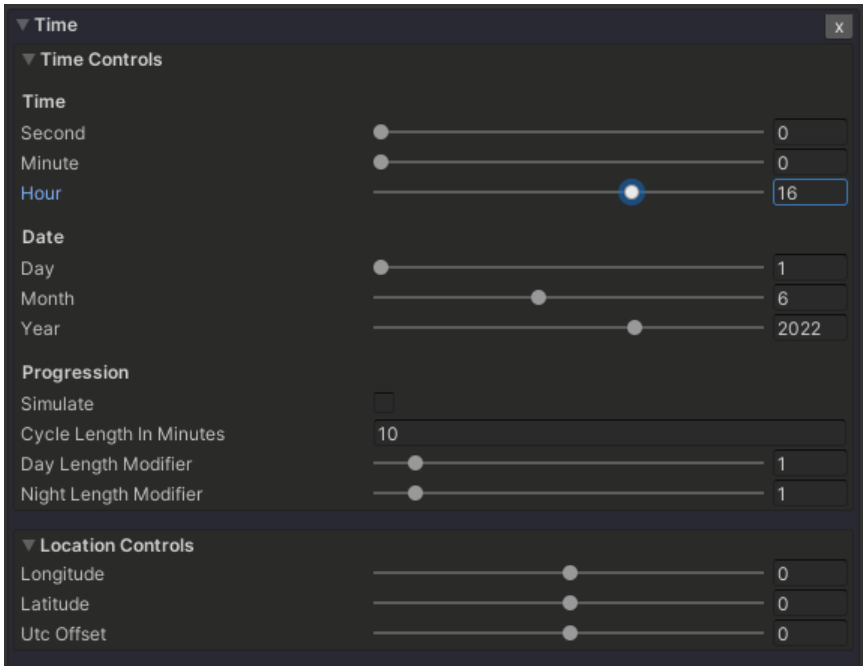
Load all Module*:*
A click on this button will load all the module settings from its assigned presets.

Add Modules:
Just click on the corresponding buttons to add a module that isn't added yet. Added ones will be shown grey and can't be added twice.

Remove Modules:
You can remove modules by clicking the small „X" on the left of each module.

Time Module

This module will control your date/time and location. If you remove the module a simple NonTime module will be shown where you can set the sun and moon positions. This is needed as other modules might need information of those to work properly.

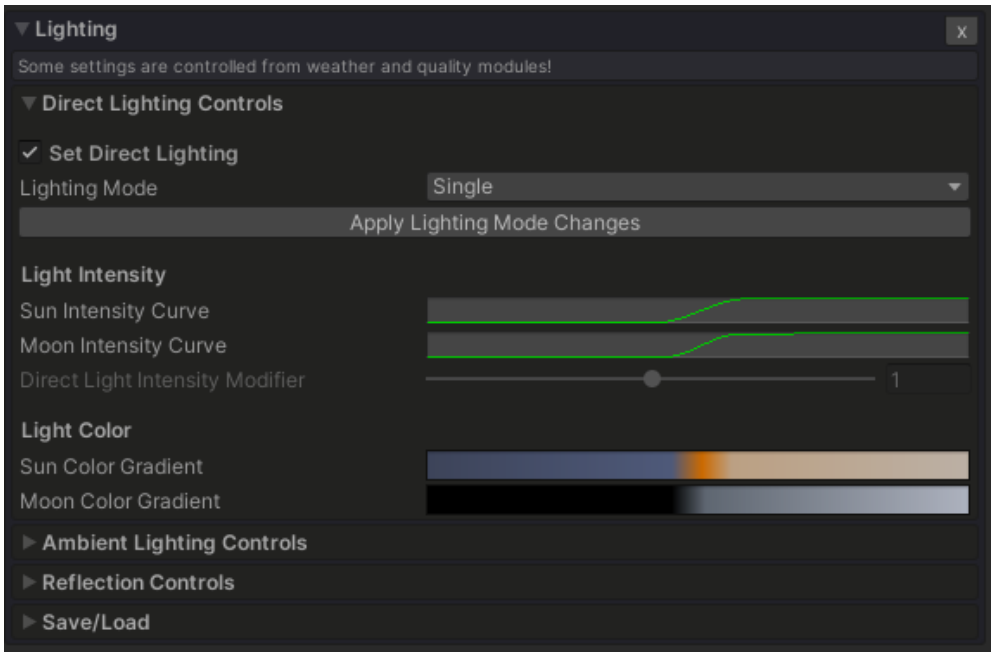


Time API

```
//Get Time of Day with UTC offset

float timeOfDay = Enviro.EnviroManager.instance.Time.GetTimeOfDay();
//Get Time of Day without UTC offset
float universalTimeOfDay =
Enviro.EnviroManager.instance.Time.GetUniversalTimeOfDay();

//Set Time of Day
Enviro.EnviroManager.instance.Time.SetTimeOfDay(12.5f);

//Get or set individual date/time
Enviro.EnviroManager.instance.Time.seconds = 1;
Enviro.EnviroManager.instance.Time.minutes = 1;
Enviro.EnviroManager.instance.Time.hours = 1;
Enviro.EnviroManager.instance.Time.days = 1;
Enviro.EnviroManager.instance.Time.months = 1;
Enviro.EnviroManager.instance.Time.years = 1;

//Enable or disable time simulation
Enviro.EnviroManager.instance.Time.Settings.simulate = true;
```

Lighting Module

This module will control your scene lighting like directional sun and moon light, ambient and reflection probe.

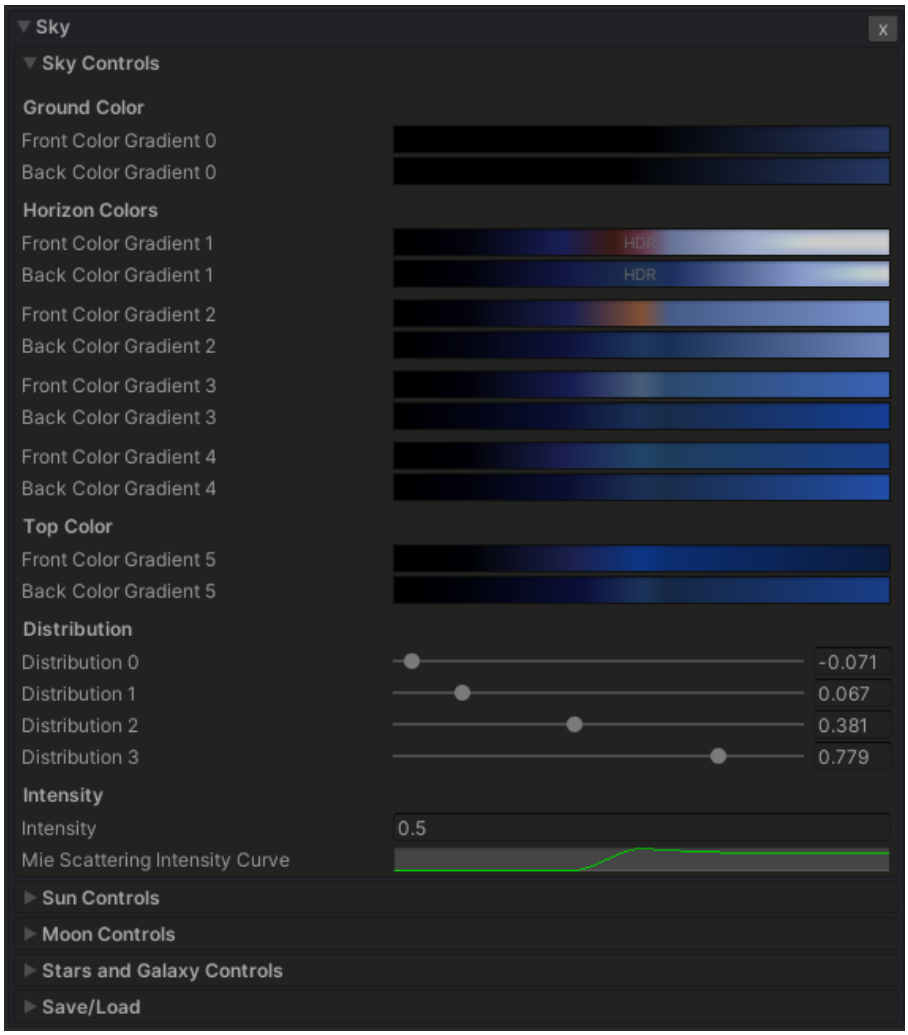


Lighting API

```
//Update the global reflection probe
Enviro.EnviroManager.instance.Lighting.UpdateReflectionForced ();

//Enables/Disables the control of Enviro over the direct lighting
Enviro.EnviroManager.instance.Lighting.Settings.setDirectLighting = false;

//Enables/Disables the control of Enviro over the ambient lighting
Enviro.EnviroManager.instance.Lighting.Settings.setAmbientLighting = false;
```

Sky Module

This module will control your sky. You can setup the colors, intensity, sun, moon, stars and galaxy visuals.

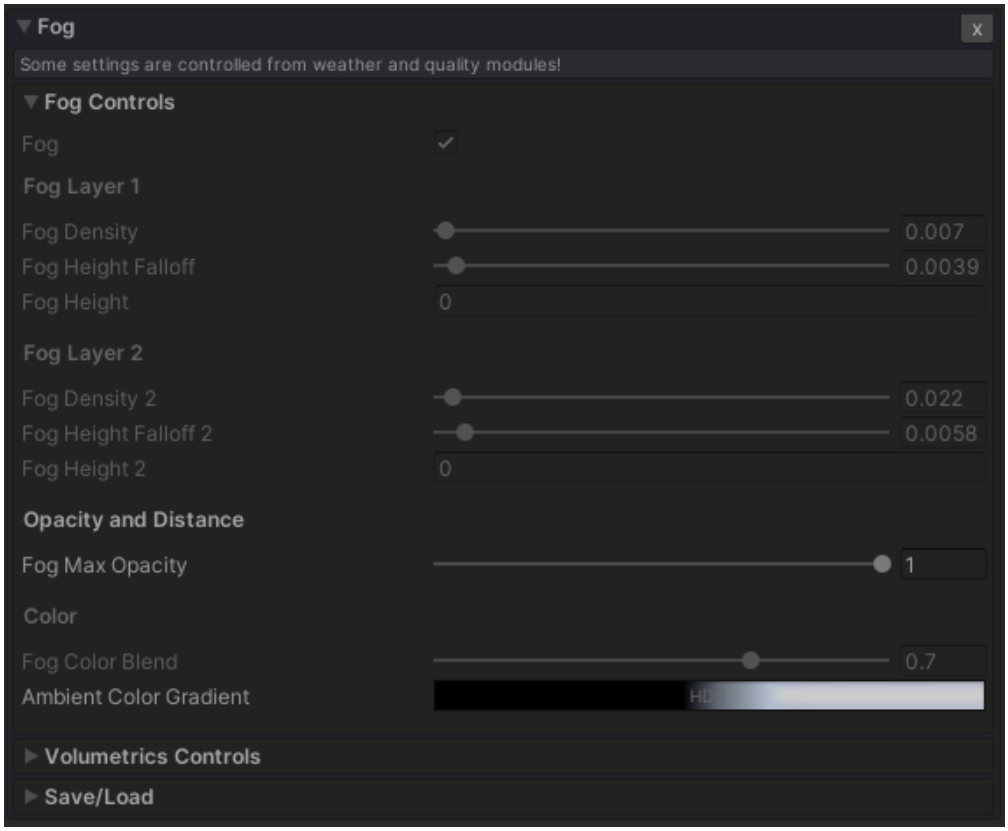


Sky API

```
//Creates and assignes the Skybox material in built-in and URP.
Enviro.EnviroManager.instance.Sky.SetupSkybox();
//Changes the intensity/brightness of the skybox.
Enviro.EnviroManager.instance.Sky.Settings.intensity = 0.5f;
//Changes the moon mode. Simple moon will have a simple position opposite of
sun.
Enviro.EnviroManager.instance.Sky.Settings.moonMode =
Enviro.EnviroSky.MoonMode.Simple;
//Changes the moon phase when using "Simple" moon mode.
Enviro.EnviroManager.instance.Sky.Settings.moonPhase = 0f;
```

Fog Module

This module will control the exponential height based fog and volumetric lighting rendering. Some settings here are greyed out when using Weather or Quality module as they will be controlled from those modules.

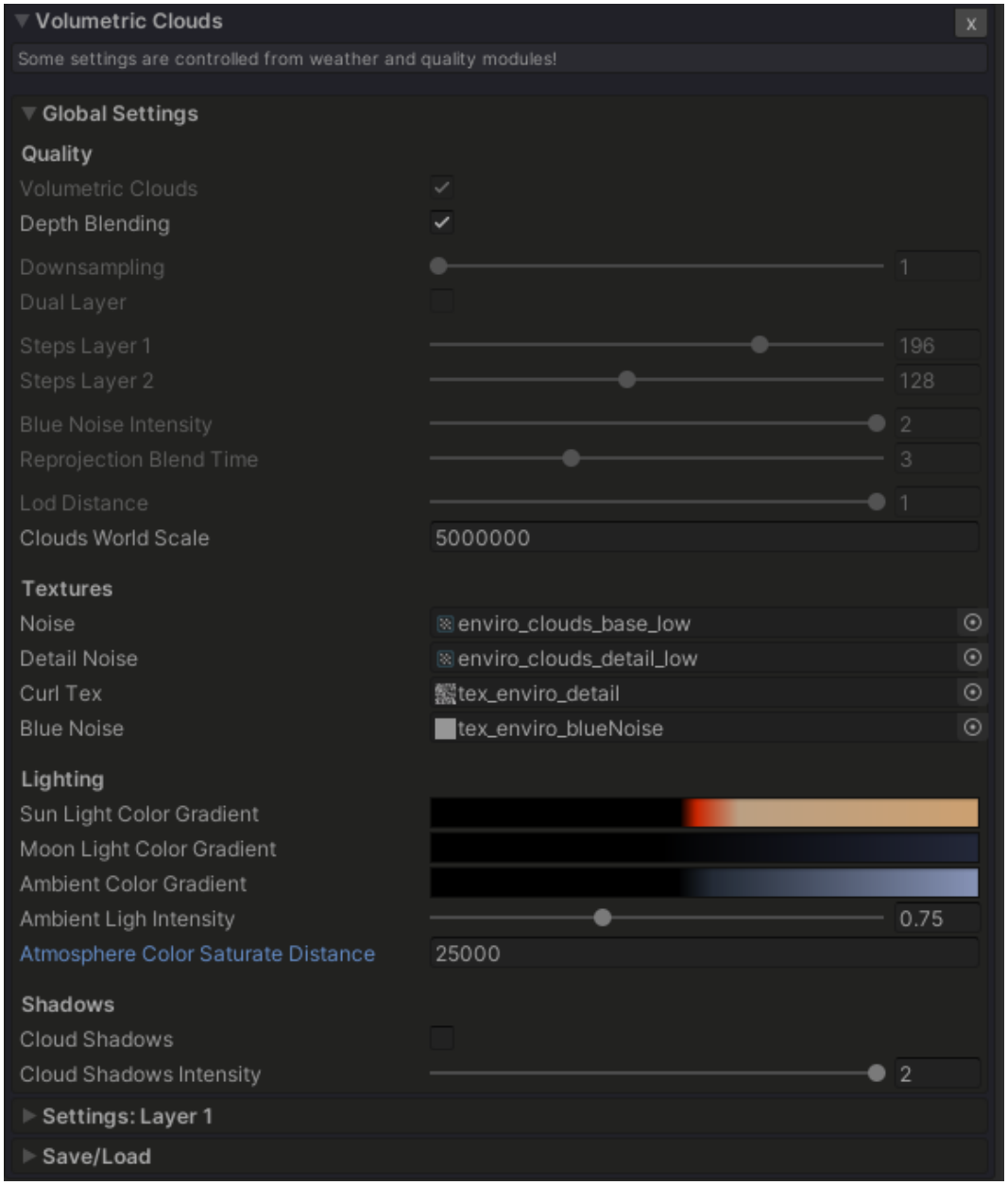


Fog API

```
//Enables or disables fog rendering when no Quality module is used.
Enviro.EnviroManager.instance.Fog.Settings.fog = true;
//Enables or disables volumetric rendering when no Quality module is used.
Enviro.EnviroManager.instance.Fog.Settings.volumetrics = true;
//Changes the quality of volumetrics rendering when no Quality module is used.
Enviro.EnviroManager.instance.Fog.Settings.quality =
Enviro.EnviroFogSettings.Quality.Medium;
```

Volumetric Clouds Module

This module will control your volumetric clouds. You can have up to two layers of clouds with its own settings. The cloud coverage will be handled by the weather map which will be created on the fly.
*Enable the "**Cloud Shadow**" option to render shadows from clouds to your scenes.*
*Enable "**Depth Blending**" to blend clouds with your scene. For example when fly above clouds.*
*Set the "**Atmosphere Color Saturation Distance**" to blend distance clouds with your atmosphere.*

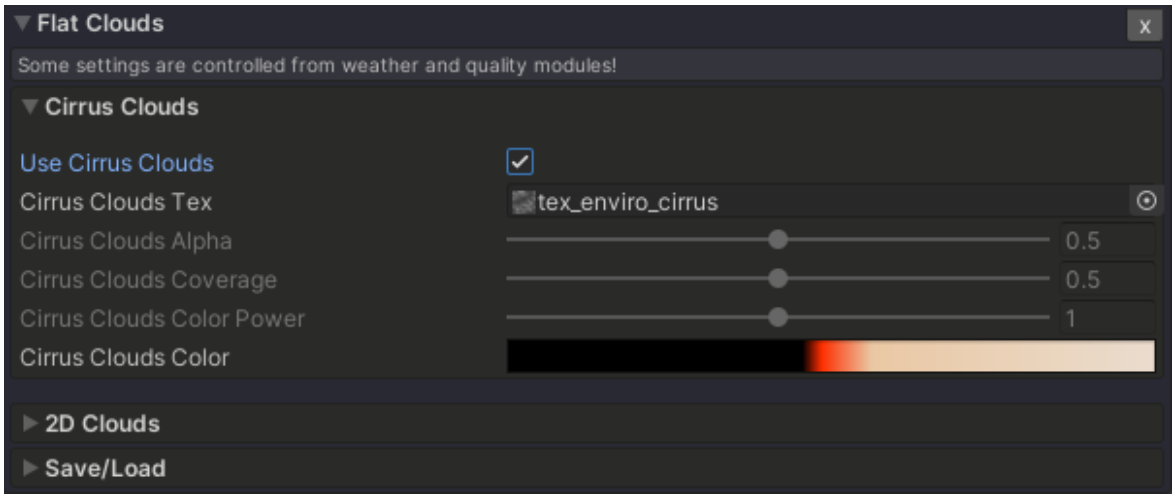


Volumetric Clouds API

```
//Enable/Disable volumetric clouds when no Quality module is used.
Enviro.EnviroManager.instance.VolumetricClouds.settingsQuality.volumetricClouds = false;
//Enable/Disable second layer when no Quality module is used.
Enviro.EnviroManager.instance.VolumetricClouds.settingsGlobal.dualLayer = true;

//Enable/Disable clouds scene blending.
Enviro.EnviroManager.instance.VolumetricClouds.settingsGlobal.depthBlending = true;
//Setup for first clouds layer.
Enviro.EnviroManager.instance.VolumetricClouds.settingsLayer1
//Setup for second clouds layer.
Enviro.EnviroManager.instance.VolumetricClouds.settingsLayer2
```

Flat Clouds Module

This module controls your Cirrus layer and 2D clouds. Both rendered directly in skybox. Using this module without the sky module would not work.

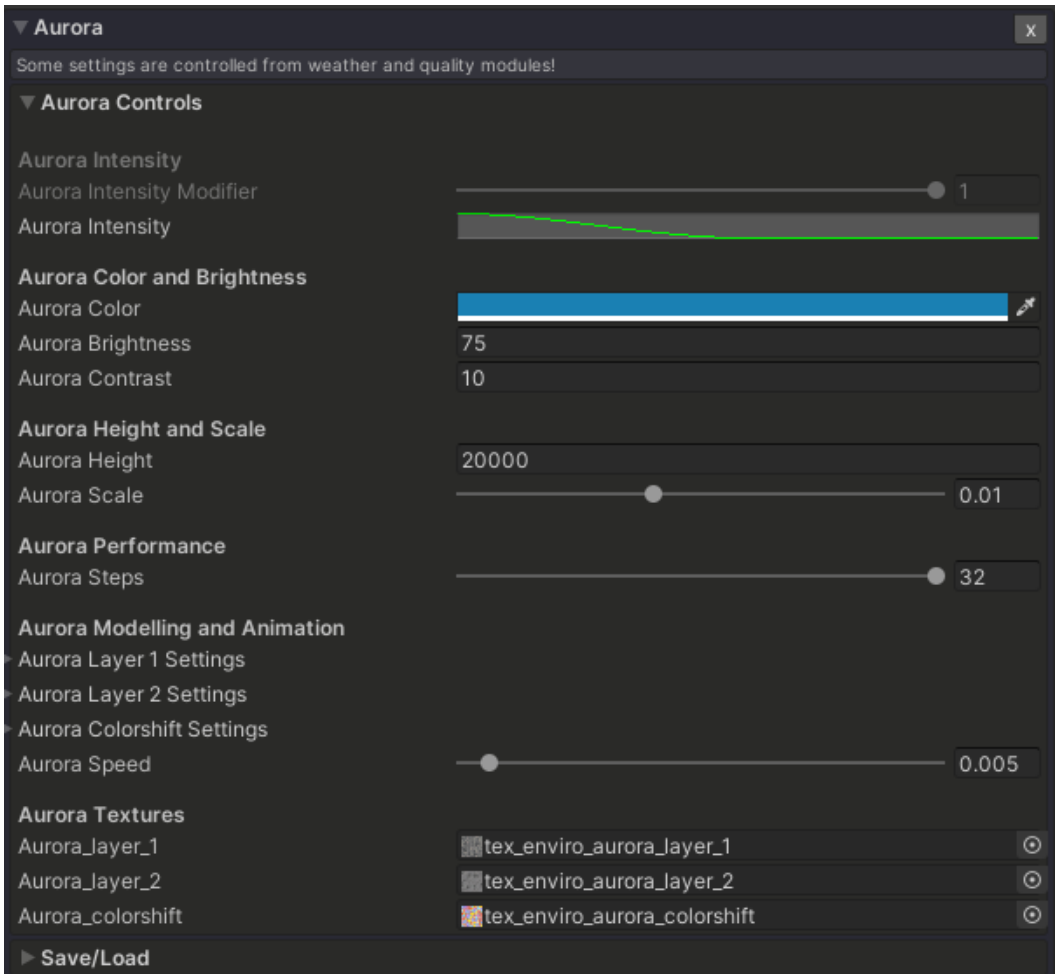


Flat Clouds API

```
//Enable/Disable cirrus clouds.

Enviro.EnviroManager.instance.FlatClouds.settings.useCirrusClouds = true;
//Enable/Disable 2D clouds.
Enviro.EnviroManager.instance.FlatClouds.settings.useFlatClouds = true;

//Sets the coverage of flat clouds when no Weather module is used.
Enviro.EnviroManager.instance.FlatClouds.settings.cirrusCloudsCoverage = 1f;
Enviro.EnviroManager.instance.FlatClouds.settings.flatCloudsCoverage = 1f;
```

Aurora Module

This module controls the aurora borealis effect. The effect will be rendered directly in skybox shader. So it has no effect when not using the sky module.

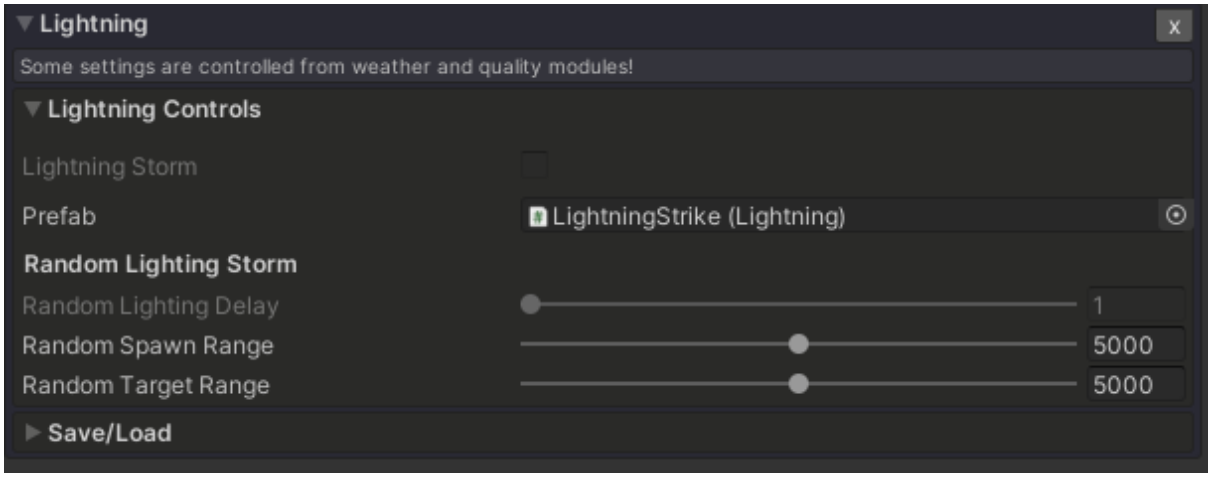


Aurora API

```
//Sets Aurora intensity when no weather module is used.
Enviro.EnviroManager.instance.Aurora.Settings.auroraIntensityModifier = 1f;
//Sets Aurora brightness.
Enviro.EnviroManager.instance.Aurora.Settings.auroraBrightness = 100f;
//Sets Aurora steps for quality control.
Enviro.EnviroManager.instance.Aurora.Settings.auroraSteps = 24;
```

Lightning Module

This module will create procedural lightning strikes randomly or with target coordinates. It also will play thunder sounds when Audio module is used.
*The "**Lightning Storm**" option will enable/disable random lightnings. It will be controlled from weather module if added to your Enviro configuration.*

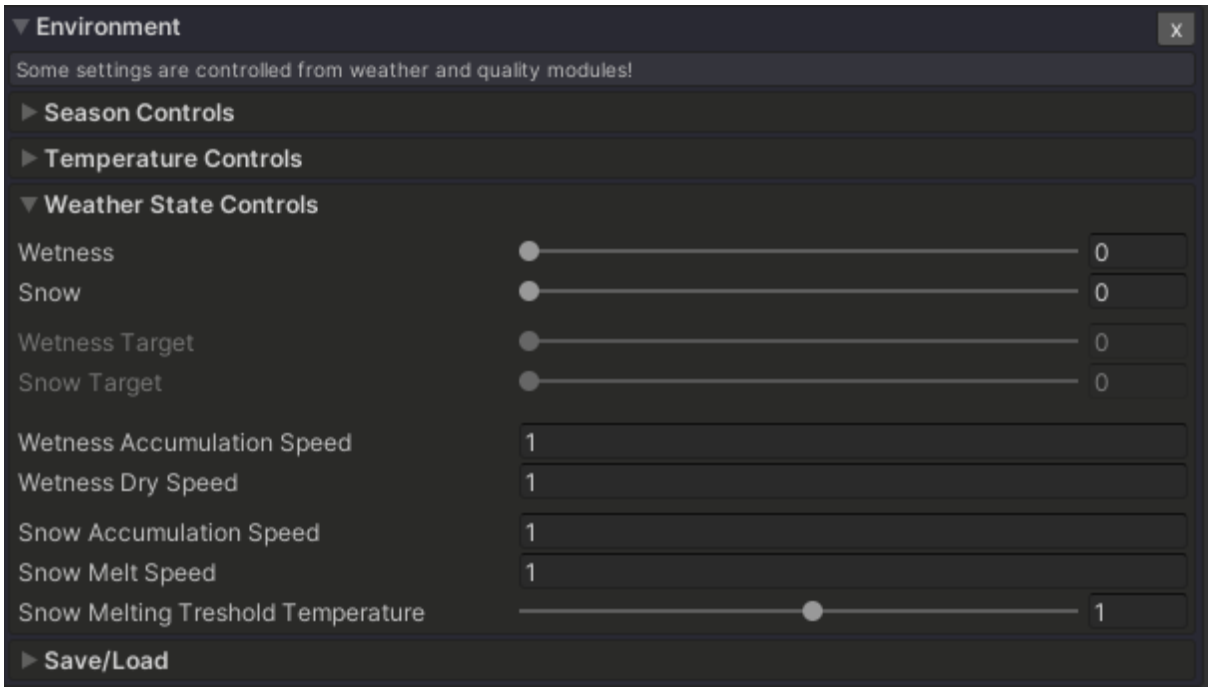


Lightning API

```
//Enable/Disable random lightnings when no weather module is used.
Enviro.EnviroManager.instance.Lightning.Settings.lightningStorm = false;

//Casts a lightning bold randomly.
Enviro.EnviroManager.instance.Lightning.CastLightningBoltRandom();

//Casts a lightning bold from a position to a position.
Enviro.EnviroManager.instance.Lightning.CastLightningBolt(Vector3 from,
Vector3 to);
```

Environment Module

This module will control your seasons, temperature and weather states like wetness/snow.



Environment API

```
//Enable or disable to let Enviro calculate the current season.
Enviro.EnviroManager.instance.Environment.Settings.changeSeason = false;
//Returns the current season.
Enviro.EnviroEnvironment.Seasons season =
Enviro.EnviroManager.instance.Environment.Settings.season;
//Changes the season.
Enviro.EnviroManager.instance.Environment.ChangeSeason(Enviro.EnviroEnvironmen
t.Seasons.Spring);

//Returns the current wetness and snow.
float wetness = Enviro.EnviroManager.instance.Environment.Settings.wetness;
float snow = Enviro.EnviroManager.instance.Environment.Settings.snow;

//Returns the current temperature.
float temp = Enviro.EnviroManager.instance.Environment.Settings.temperature;
//Set a custom temperature modifier that will be added to the current
temperature for your game logic.
Enviro.EnviroManager.instance.Environment.Settings.temperatureCustomMod = 20f;
```

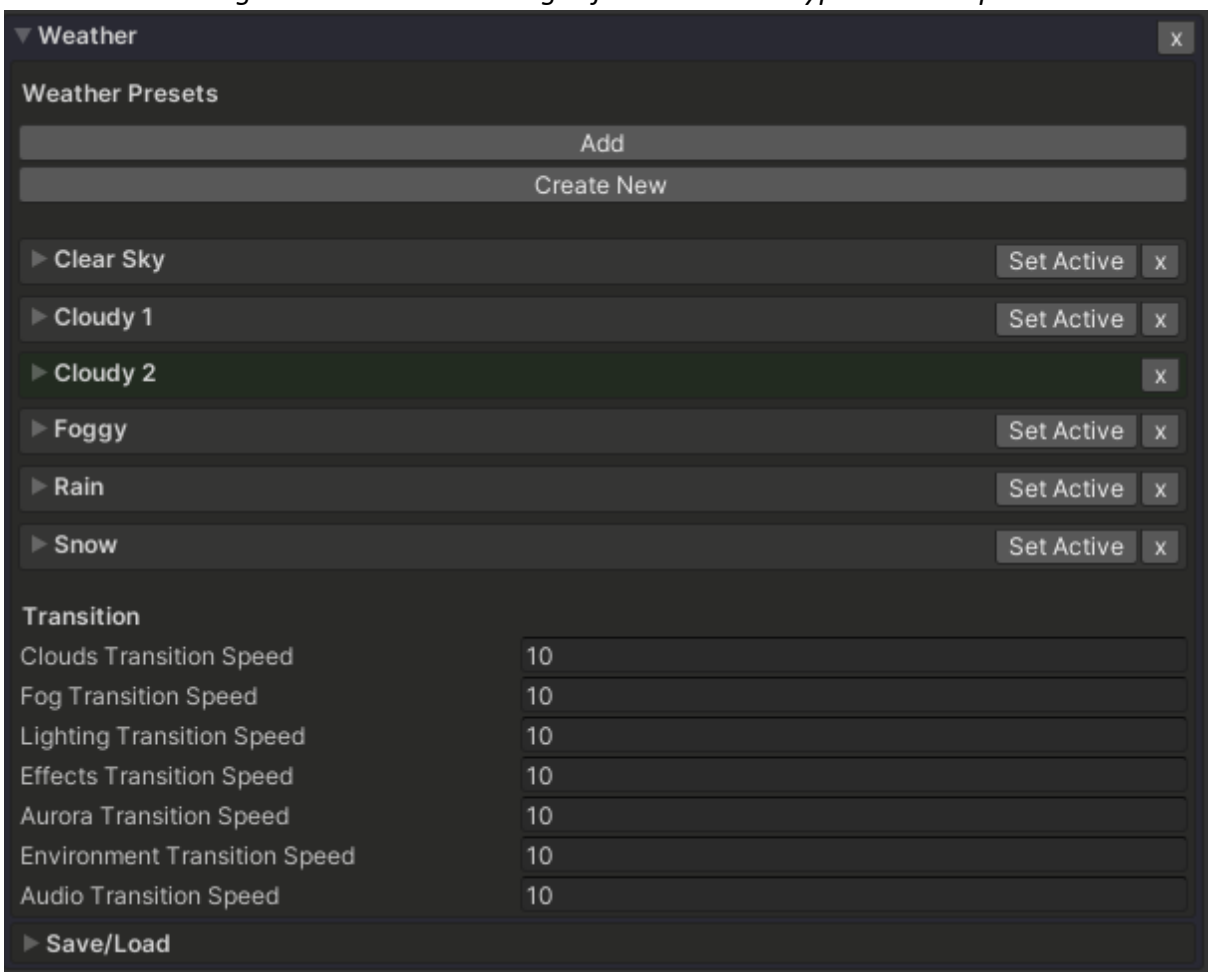
Weather Module

The weather module holds a list of weather types which will control most of the other modules settings. Weather types can be changed freely with a smooth transition.

The weather module will control other module settings. These will be greyed out in its own module.

*Click on the "**Add**" button to add a new already existing weather type to your configuration.*
*A click on the "**Create New**" will create a new weather type object in your project.*
*Remove a weather type with a click on the "**x**".*
*Click on "**Set Active**" to change to that weather type.*

You also can change and tweak the settings of each weather type in this inspector.



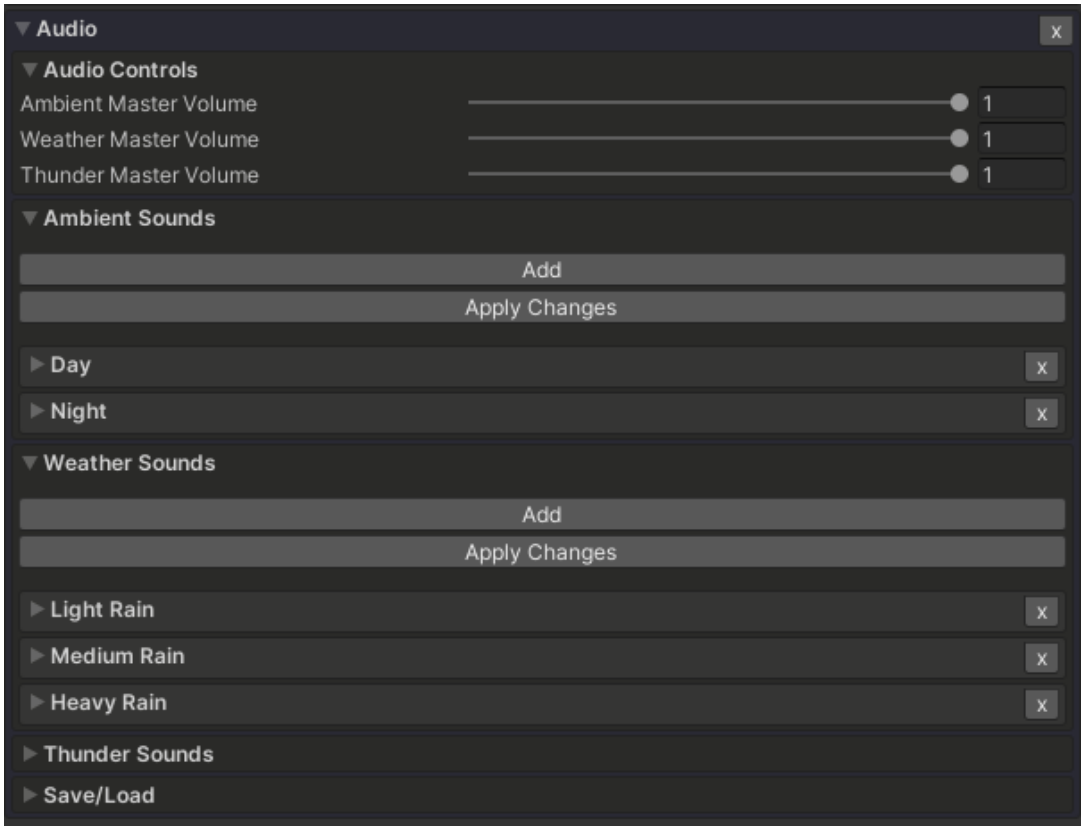
Weather API

```
//Changes current active weather with a transition. This only works when player isn't in a Enviro Zone.
Enviro.EnviroManager.instance.Weather.ChangeWeather("Weather Type Name");
//Changes current active weather without a transition. This only works when player isn't in a Enviro Zone.
Enviro.EnviroManager.instance.Weather.ChangeWeatherInstant(Enviro.EnviroWeatherType weatherType);
//Get the current active weather type.
Enviro.EnviroWeatherType weatherType = Enviro.EnviroManager.instance.Weather.targetWeatherType;
//Get the current active weather zone.
Enviro.EnviroZone zone = Enviro.EnviroManager.instance.Weather.currentZone;
```

Audio Module

The audio module will control ambient, weather and thunder sound effect. Add your own sound effects here. Those can be controlled from the weather system to play only selected audio. For example rain sounds or different ambient sounds for day/night.
You also can set a master volume for all three category here.

*Click on the "**Add**" button to add a new sound. Make sure to set unique names to be able to select those in the weather types.*
*The "**Apply Changes**" button will recreate the audio sources for all the sounds.*



Audio API

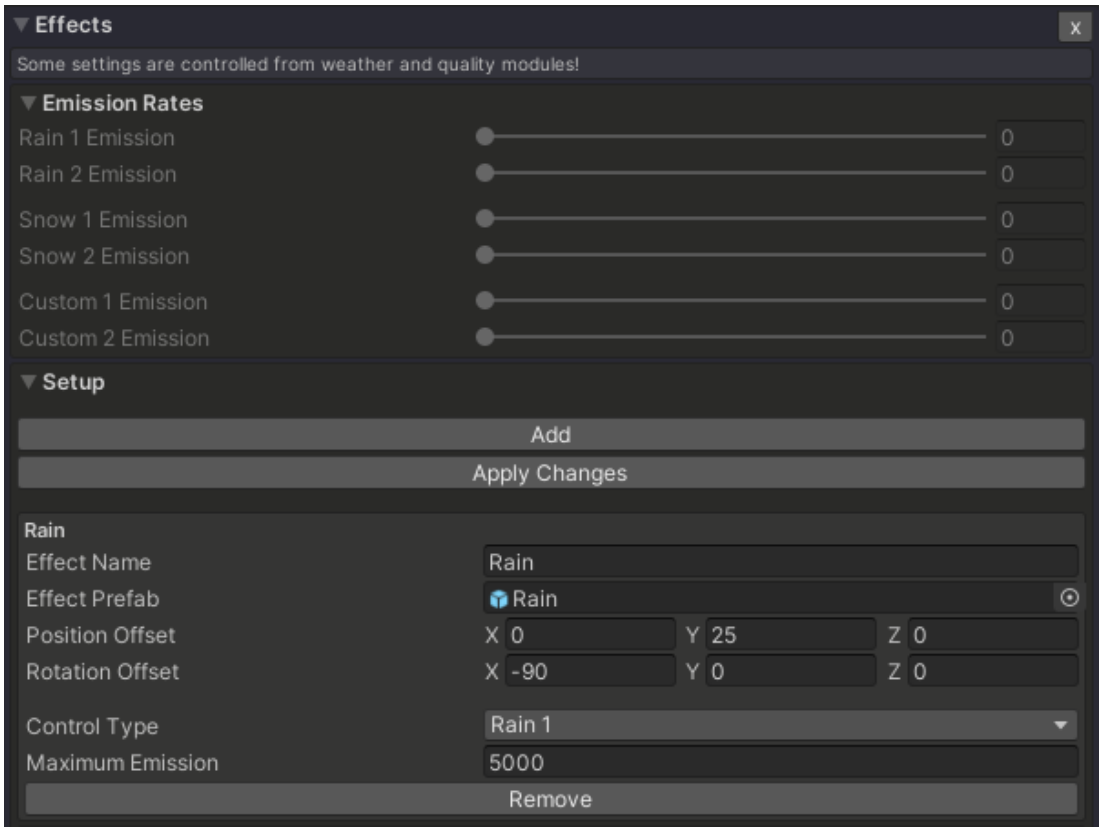
```
//Set the master volume for each category of sounds.
Enviro.EnviroManager.instance.Audio.Settings.ambientMasterVolume = 1f;
Enviro.EnviroManager.instance.Audio.Settings.weatherMasterVolume = 1f;
Enviro.EnviroManager.instance.Audio.Settings.thunderMasterVolume = 1f;
//Plays a random thunder sound once.
Enviro.EnviroManager.instance.Audio.PlayRandomThunderSFX();
```

Effects Module

The effects module adds support for particle effects into Enviro. Control the emission rates of those with the predefined emission sliders. Those will be controlled from weather module if added.

*Click on the "**Add**" button to add a new particle effect. Give it a unique name and assign your Unity particle effect prefab to it. Now set the "**Control Type**" and "**Maximum Emission**" value.*

*Click on "**Apply Changes**" to recreate the particle effects.*



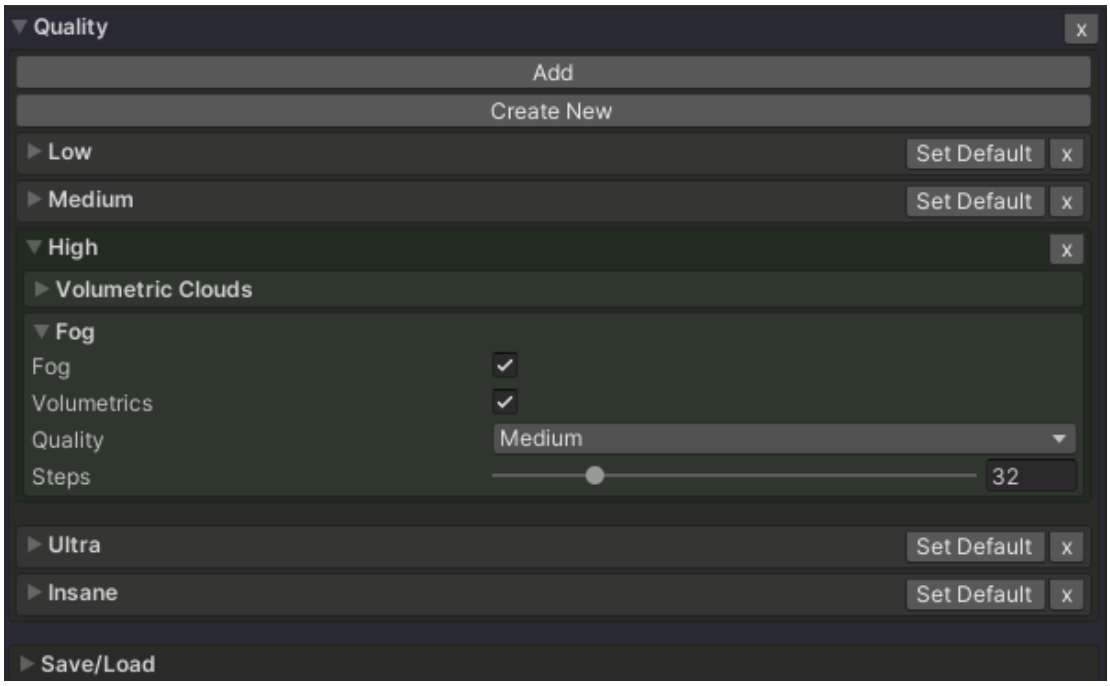
Effects API

```
//Set the emission rates when not using the weather module.
Enviro.EnviroManager.instance.Effects.Settings.rain1Emission = 0.5f;
Enviro.EnviroManager.instance.Effects.Settings.rain2Emission = 0.5f;

Enviro.EnviroManager.instance.Effects.Settings.snow1Emission = 0.5f;
Enviro.EnviroManager.instance.Effects.Settings.snow2Emission = 0.5f;

Enviro.EnviroManager.instance.Effects.Settings.custom1Emission = 0.5f;
Enviro.EnviroManager.instance.Effects.Settings.custom2Emission = 0.5f;
```

Quality Module

The quality module will add the feature of predefined quality presets that control the most important performance related settings. Use it to give your users the ability to change the quality of your sky rendering to match their hardware.

*Click on "**Add**" to add an already existing EnviroQuality preset.*
*Click on "**Create New**" to create a new EnviroQuality preset in your project.*
*Click on "**Set Default**" to select a quality that should be used for the moment.*

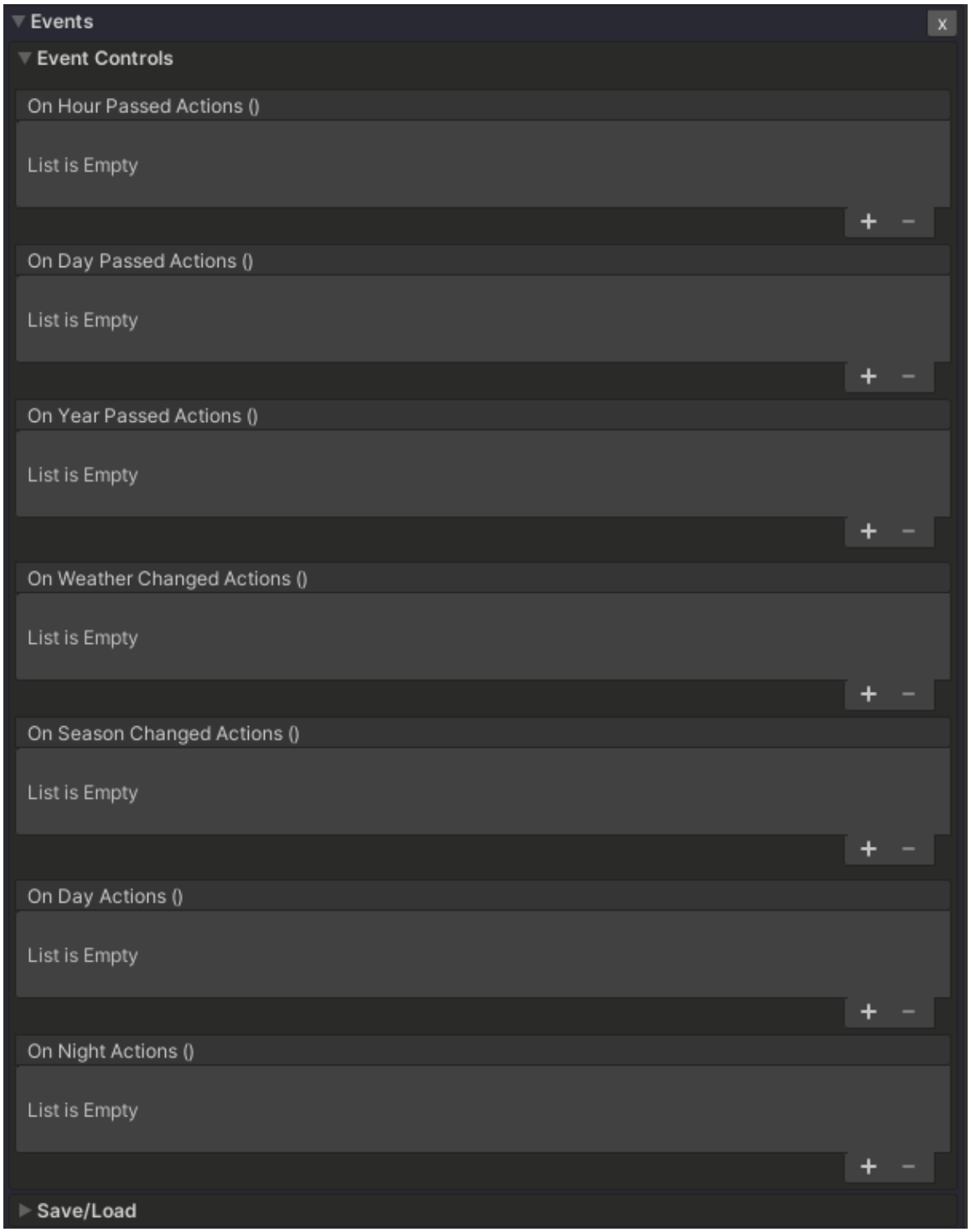


Quality API

```
//Set a new default quality.
Enviro.EnviroManager.instance.Quality.Settings.defaultQuality =
Enviro.EnviroQuality newDefaultQuality;
//Get the list of all qualities.
List<Enviro.EnviroQuality> allQualities =
Enviro.EnviroManager.instance.Quality.Settings.Qualities;
```

Events Module

Use this module for your game logic to take action when different events fired.
You can assign your own functions in this module inspector or use the API to hook to them.



Events API

```
            EnviroManager.instance.OnHourPassed += () =>
            {
                Debug.Log("Hour Passed!");
            };

            EnviroManager.instance.OnDayPassed += () =>
            {
                Debug.Log("New Day!");
            };

            EnviroManager.instance.OnYearPassed += () =>
            {
                Debug.Log("New Year!");
            };

            EnviroManager.instance.OnDayTime += () =>
            {
                Debug.Log("Day!");
            };

            EnviroManager.instance.OnNightTime += () =>
            {
                Debug.Log("Night!");
            };

            EnviroManager.instance.OnSeasonChanged +=
(EnviroEnvironment.Seasons s) =>
            {
                Debug.Log("Season changed to: " + s.ToString());
            };

            EnviroManager.instance.OnWeatherChanged += (EnviroWeatherType w)
=>
            {
                Debug.Log("Weather changed to: " + w.name);
            };

            EnviroManager.instance.OnZoneWeatherChanged += (EnviroWeatherType
w, EnviroZone z) =>
            {
                Debug.Log("Weather changed to: " + w.name.ToString() + " in
zone:" + z.name);
            };
```

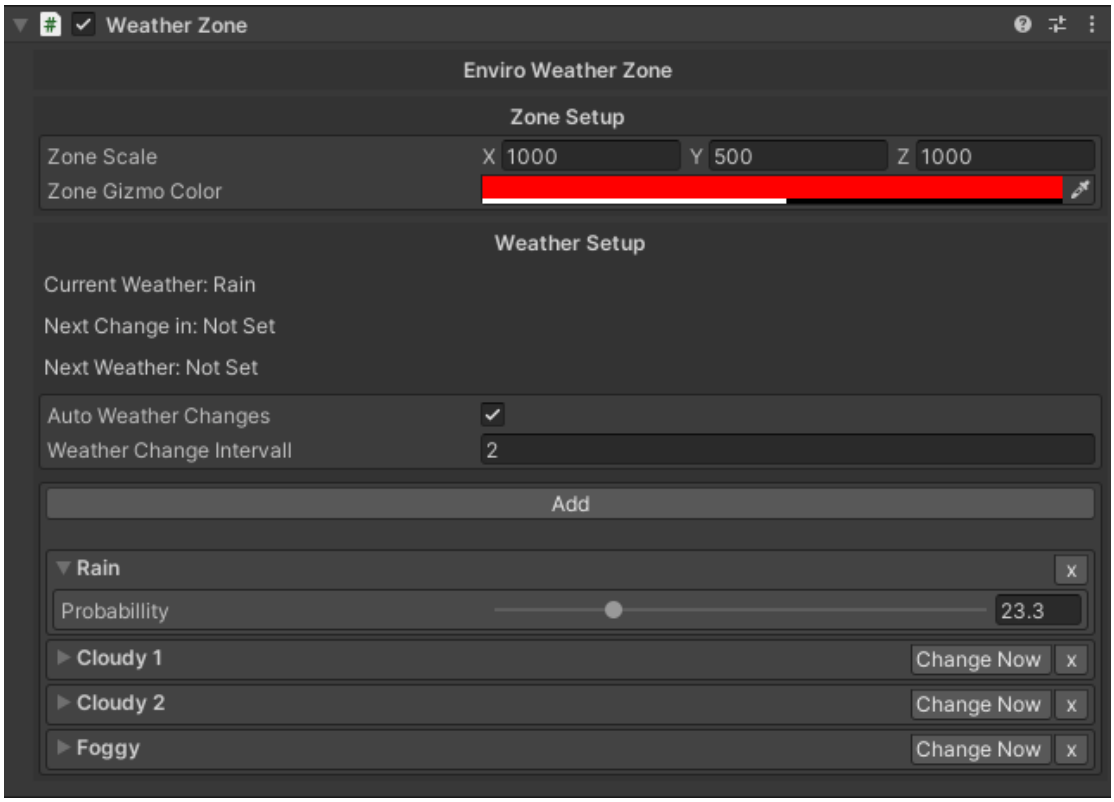
Components

There are a few other components shipped with Enviro that give you some additional functionality.

Enviro Weather Zone

*The „**Enviro Weather Zone**“ component can be used for regional auto changing weather.*

You can add it through the (Components -> Enviro 3 -> Weather Zone) menu to a GameObject.



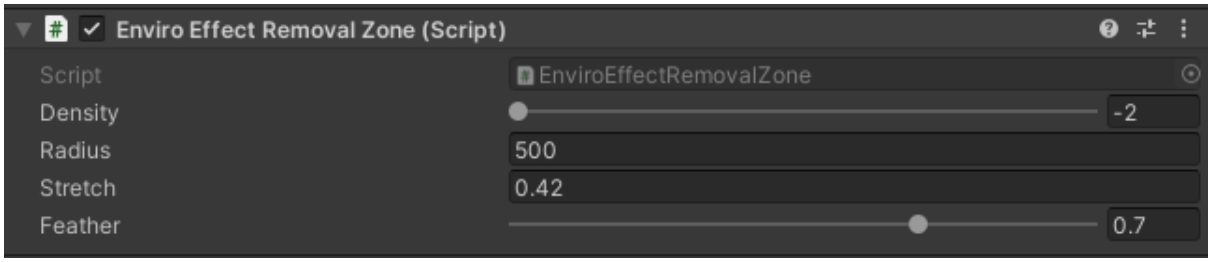
*Set the „**Zone scale**“ and move it in your scene to the wished location. Once your main camera enters the area the weather will smoothly change to the current active zone weather.*

*Click on „**Add**“ button to add other weather types to this zone. Also do not forget to set the „**Probabillity**“ settings! If „Auto Change Weather“ is enabled each „Weather Intervall“ game time hours the system will try to change the weather based on the weather types „**Probabillity**“ settings.*
*You also can click on the „**Change Now**“ button to change the weather by manually.*

Effect Removal Zone

This zone can disable fog and particle effects with the „Enviro Weather Particle“ shader in a performant way.

You can add it through the (Components -> Enviro 3 -> Effect Removal Zone) menu to a GameObject.

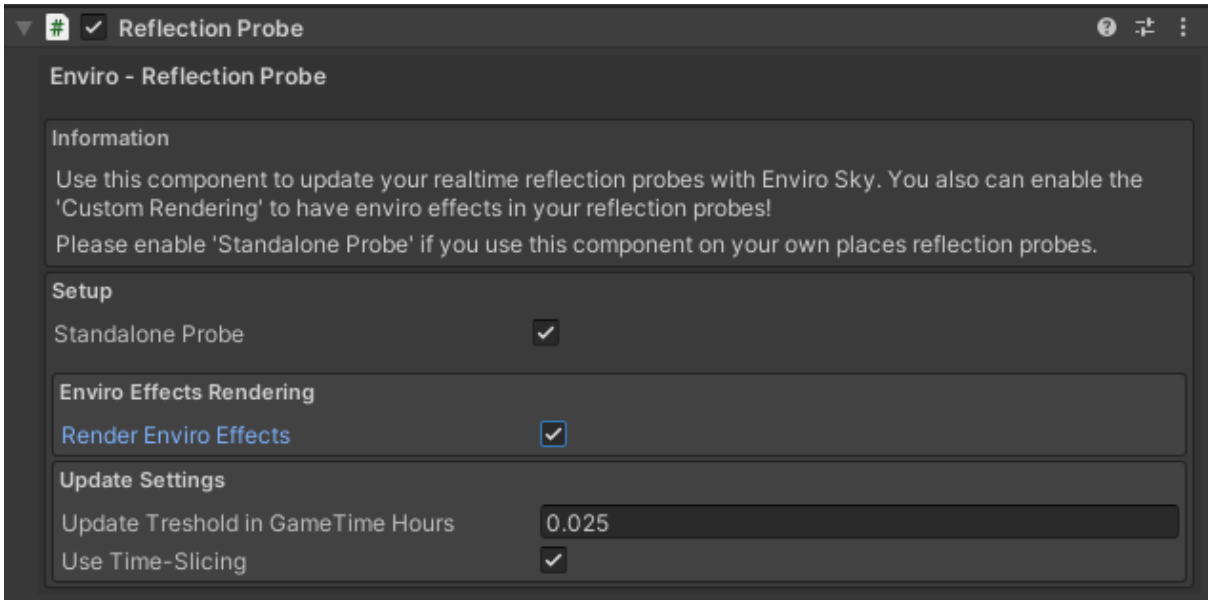


*Set the „**Radius**“ and „**Stretch**“ value to increase/decrease the zone influence. You also can move around the GameObject to place it in the area you need it.*

*Lower the „**Density**“ setting to start removing fog and particle effects.*

Enviro Reflection Probe

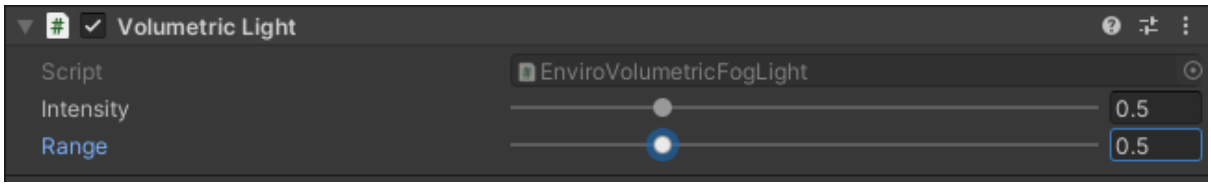
This reflection probe script can render your reflection probes based on Enviro time progression. It also allows you to render volumetric clouds to your reflection probes in Built-In renderpipeline.



*Enable „**Standalone Probe**“ when you use this component on manually placed probes.*

Enviro Volumetric Light

Add this component to your spot and point lights to make them illuminate the fog and render a volumetric effect for those. This component only work when fog module with activated volumetric option is used in your Enviro Manager.



*You can set the „**Intensity**“ to increase/decrease the effect brightness and also change the „**Range**“ to increase/decrease the influence area.*

Gradients and Curves

Enviro makes heavy use of gradients and curve. Those are evaluated based on the solarTime and lunarTime with a range from 0 -> 1. Those describe the position of sun/moon in the sky. Please take a look into this graphic: