Documentation

2.2.0

Thank you for buying

Enviro - Sky and Weather

Getting Started 3

Basics 4

- *Profiles* 4
- *Weather Presets* 5
- *Enviro Sky Manager* 6
  - *URP/LWRP Support* 7
  - *Third Party Support* 9
- *Enviro Sky Instances* 10
  - *Rendering Setup* 11
  - *Time and Location* 12
  - *Weather Controls* 13
  - *Feature Controls* 14
  - *Audio Controls* 15
- *Enviro Zone Component* 16

FAQ and Troubleshoot 17

Scripting API 20

- *General* 20
- *Time and Date* 21
- *Seasons* 22
- *Weather* 23
- *Events* 24

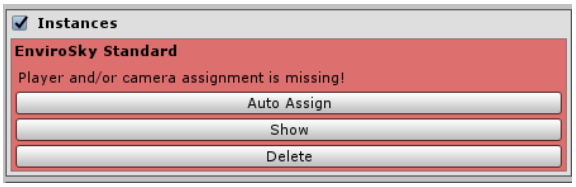
Getting Started

It's really easy to setup enviro in your project! Import the package and follow these easy steps to get started!

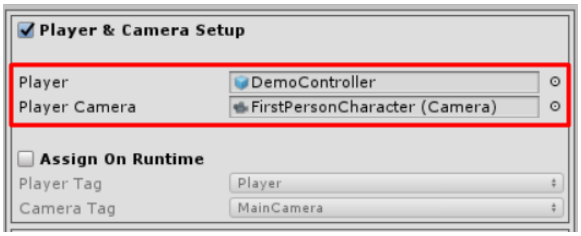
1. Click on “Assets” -> “Create” -> “Enviro” -> “Enviro Sky Manager”. This will create a new “GameObject” in your scene with the Enviro Sky Manager component.

2. Now select the “Enviro Sky Manager” object in your scene. From here you can add different preconfigured enviro instances. For example click on “Create Standard Instance”. After that take a look in “Instances” section by activating the checkbox. You should see a red box with your newly generated enviro instance. Click on “Activate” here.

3. Now we need to assign camera and if exist your player object. For a quick setup you can use the “Auto Assign” button in Enviro Sky Manager -> Instances window.



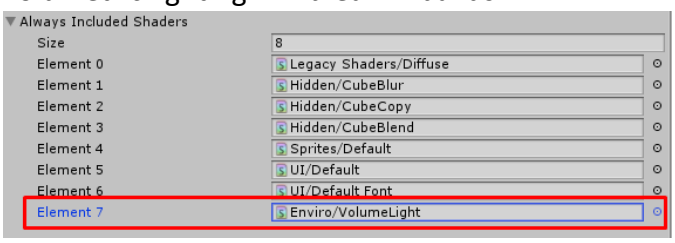
For manual or runtime assignment you have to click on the “Show” button. Then click on the “Player & Camera Setup” checkbox. Assign your "Player" and your "Player Camera” **OR** check “Assign On Runtime” and choose your tags.



4. *Disable other directional lights in your scene!*

*Only needed for “Standard” Enviro version. (Not needer for **lite** only users!):*

5. You also need to add the “Enviro/VolumeLight” shader to the “Always Included Shader” list in Unity graphic settings (Edit -> Project Settings -> Graphics). Otherwise volumetric lighting will break in builds.



Basics

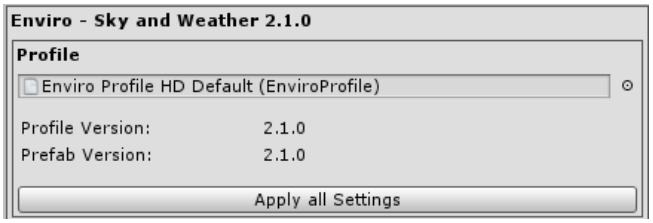
Okay now the basics! The final sky output will be controlled in two separate parts.
*First the global "**Enviro Profile**", second the current active "**Weather Preset**".*

Profiles

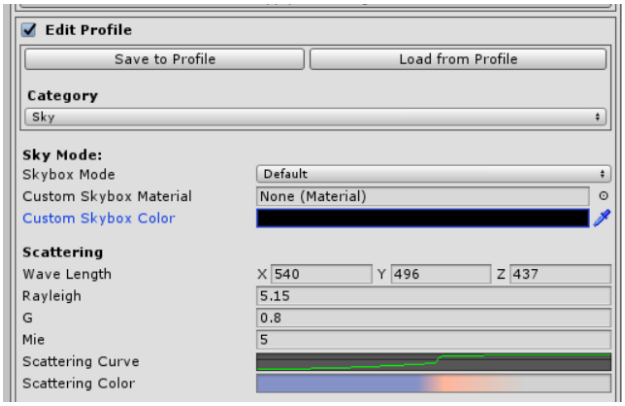
This object will hold all global information and can be saved and loaded in run- and design time!

- ***To create a new profile do a right click in your project folder and choose: Create -> Enviro -> Profile***

You can assign your profile at the top of EnviroSky inspector:



Click on "Edit Profile" in EnviroSky inspector to open the profile section.



Here you can save and load profiles with the click on the buttons.
Please note enviro never touches your profile directly!
Enviro only will overwrite your profile when you click on "Save To Profile" and only overwrite the runtime settings when you click on "Load from Profile"!

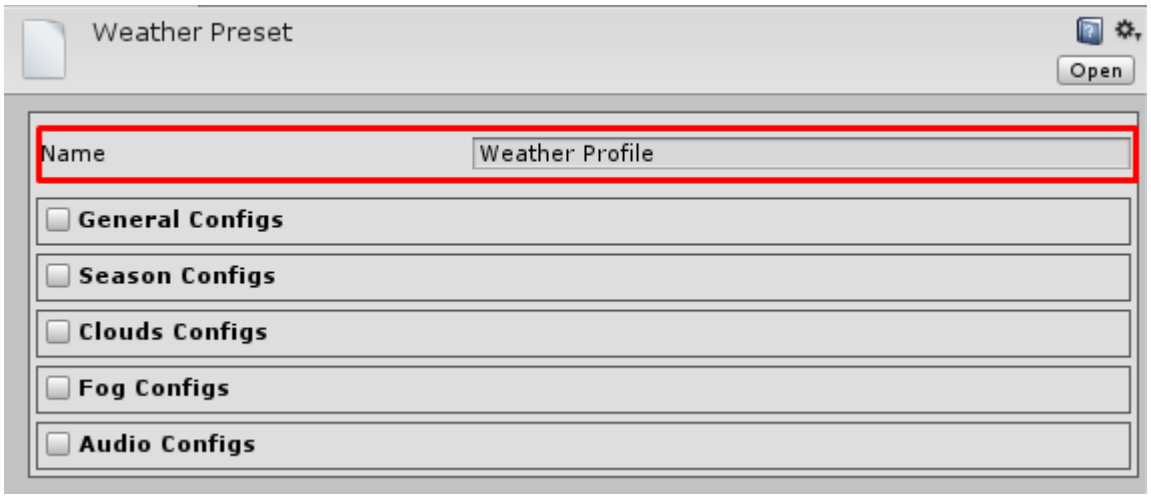
If you tweaked the settings in runtime and want them saved you have to save in runtime and load them when you are back in designtime!

*Also worth noting is that you have to use the **"Category" dropdown menu** to get access to all the different settings.*

Weather Presets

These objects will hold all information to change the look of your sky based on current weather. For example you have options to change the sky, clouds, fog and lighting. Here you also can add particle effects and audio effects to create weather that can smoothly change over time.

- ***To create a new weather preset do a right click in your project folder and choose: Create -> Enviro -> Weather Preset***

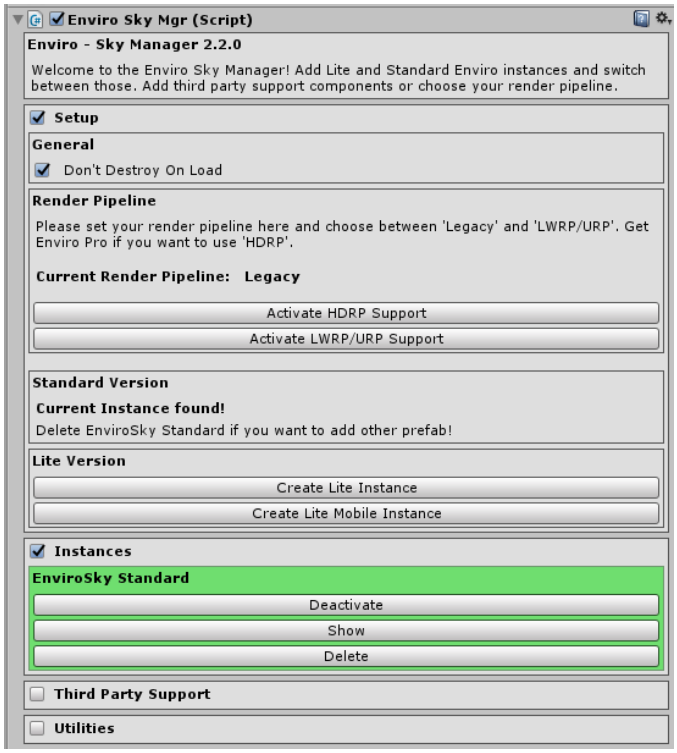


*Enviro only can use a weather preset, once it is added to a zone! To add your weather presets to the system you have to assign it in one of your **"Enviro Zone"** components. There already is a default zone on the EnviroSky object. More about zones later!*

Also note that every weather preset needs a unique name to be correctly working.

Enviro Sky Manager

Add preconfigured enviro instances, manage them and switch between standard and lite version. You also can choose your render pipeline here and activate/add third party support integrations.



General:

Activate "Don't Destroy on Load" if you use enviro in multiple scenes and you want to keep them "alive" after load.

Render Pipeline Setup:

Switch between URP/LWRP and Legacy Render Pipeline here. With the Enviro Pro upgrade you also can switch to HDRP here.

Instance Creation:

Create new instances here. You have to delete an instance to create a new one of the same type.

Instances:

Shows you Instances. You can activate or deactivate, start and stop or delete instances here. Jump to instance configuration with a click on "Show" button.

URP/LWRP Support

Activate and setup Enviro for use in LWRP/URP.

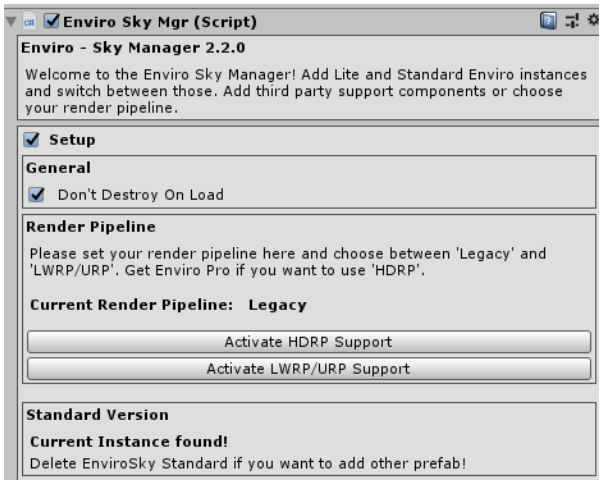
Requirements:

Unity 2019.1+ with LWRP/URP 5.X or higher.

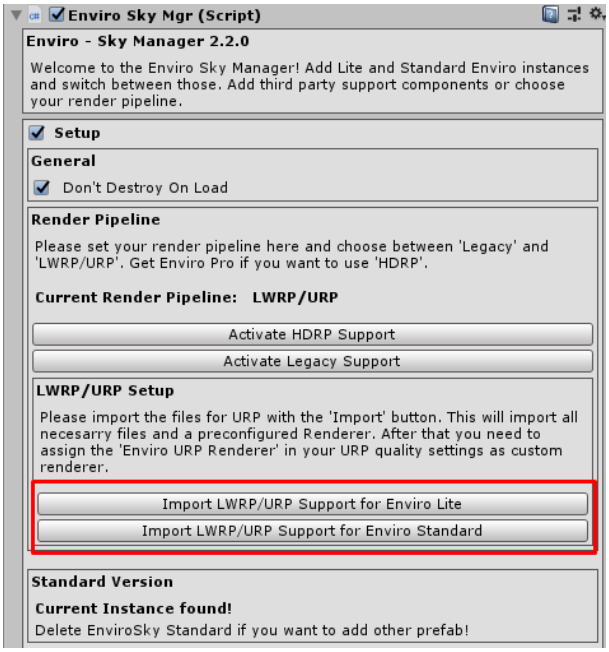
Setup:

1. *Setup enviro described in "Getting Started" section.*

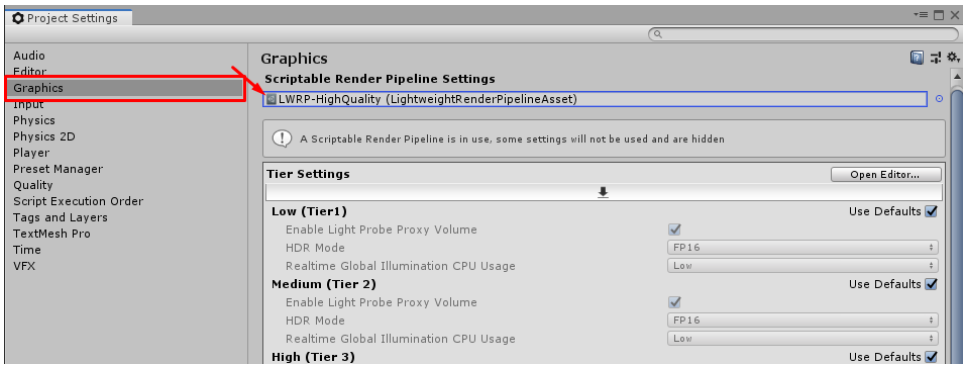
2. *Click on Enviro Sky Manager in your scene and activate LWRP/URP Support in Render Pipeline section.*



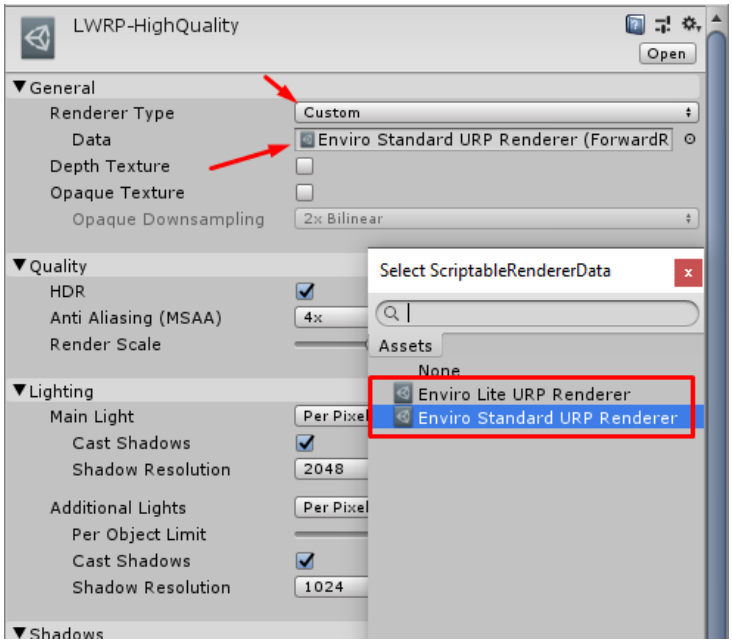
3. *Import the LWRP/URP support files for the enviro version you would like to use.*



4. Open your LWRP/URP Quality Settings. (Edit -> Project Settings -> Graphics)



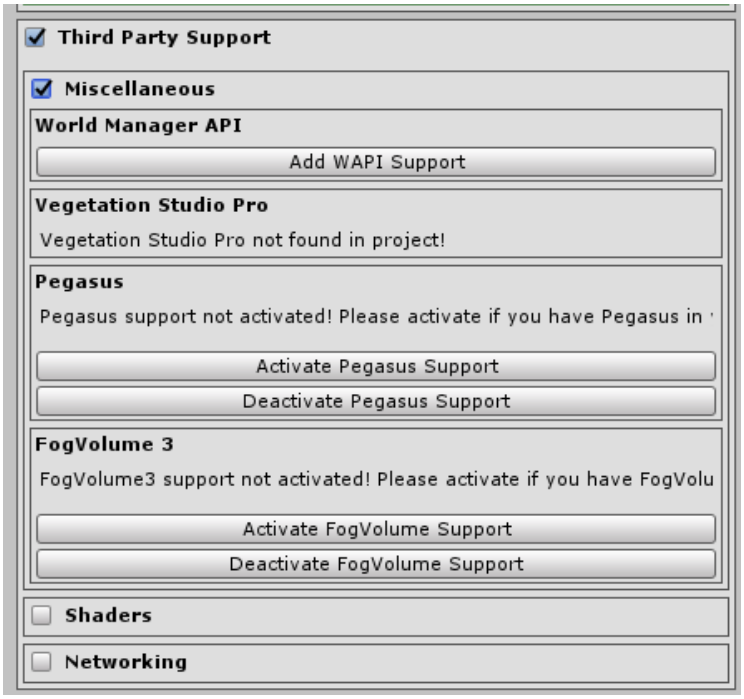
5. *Set "Render Type" to "Custom" and assign the "Enviro Standard URP Renderer" or "Enviro Lite URP Renderer".*



6. *Repeat step 5 with your other LWRP/URP quality settings. You're done now and can use Enviro in LWRP/URP.*

Third Party Support

Activate and deactivate third party support here.



Click on "Activate" and project will recompile with integration component active.

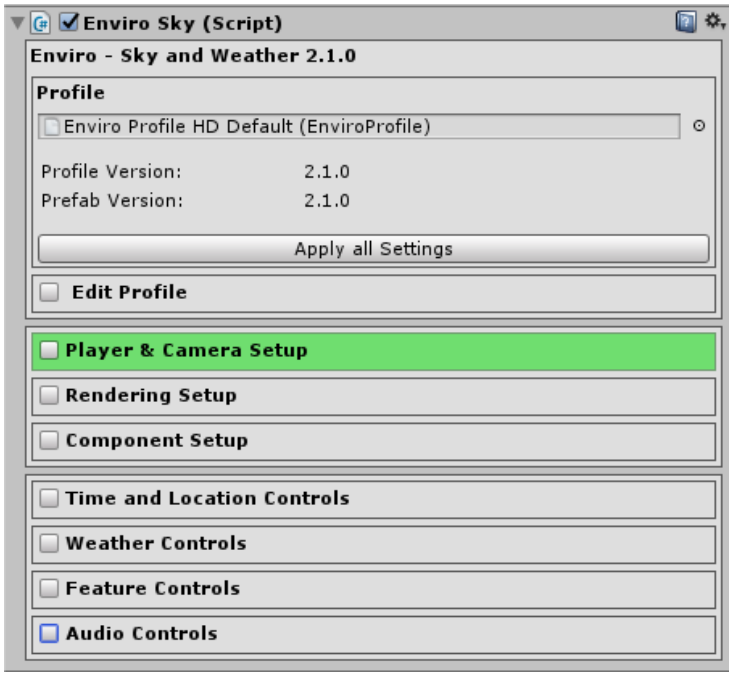
After that you will see a new button to actually add the integration component.

Please note:
Some Integrations will auto detect if third party asset is installed, so you don't need to activate them by hand.

Attention!
*If you activate an integration you don't have in your project you will get **errors**! Click on **"Deactivate"** to fix these and import the third party asset before you are activating enviro support for it.*

Enviro Sky Instances

This is the heart of each enviro sky instance where you can configure visuals, control time and weather or activate and deactivate features.



Edit Profile:

You can assign a profile at the top here and work on that with a click on "Edit Profile". Please check the "Profile" section for more information.

Player and Camera Setup:

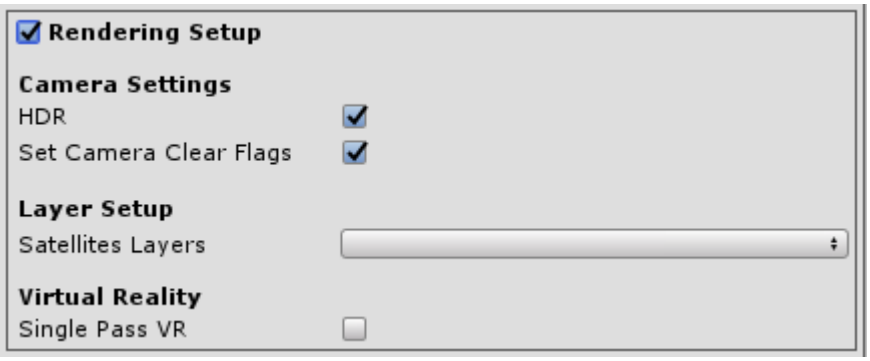
Assign your camera and player here. If you already done setup nothing special to do here anymore.

Component Setup:

Just all needed components are referenced here. Normally you don't need to change here anything as everything should be setup already. However if something goes very wrong you can reassign components here.

Rendering Setup

The "Rendering Setup" section includes a few additional setup settings that you may want to change for your projects.



HDR: *That option will enable HDR rendering in your camera and enviro effects.*

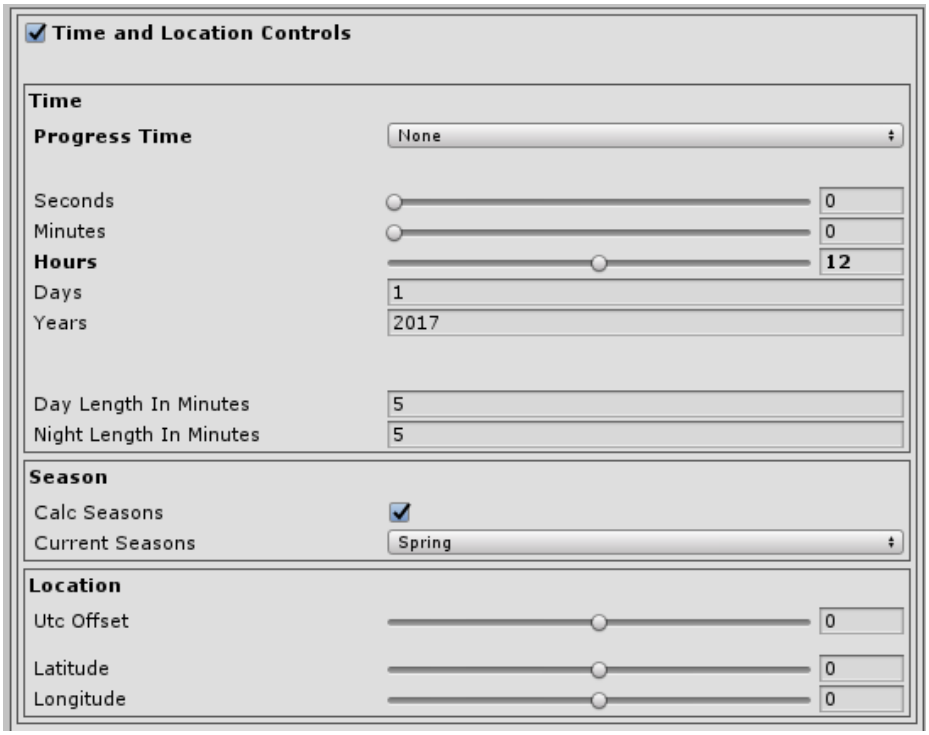
Set Camera Flags: *Disable when you want to set your own camera clear flag. (Not recommended)*

Satellites Layers: *This layer will only be used if you added additional satellites. (Deactivated by default)*

Single Pass VR: *If you are working on a **VR project** and using **"Single-Pass Rendering"** you have to set this to **enabled**! Disable if you are not working on VR or using the "Multi-Pass Mode".*

Time and Location

In this section you can control the time progressing, current time of day and date, simulation speed and location based on longitude and latitude.



Progress Time: *- **"Simulated":** This mode will progress time based on the Day and Night Length in Minutes settings.*
*- **"One Day":** Works like "Simulated" but will not progressing days or years.*
*- **"System Time":** Uses the user system time and keep them in sync.*
*- **"None":** Disables time progressing.*

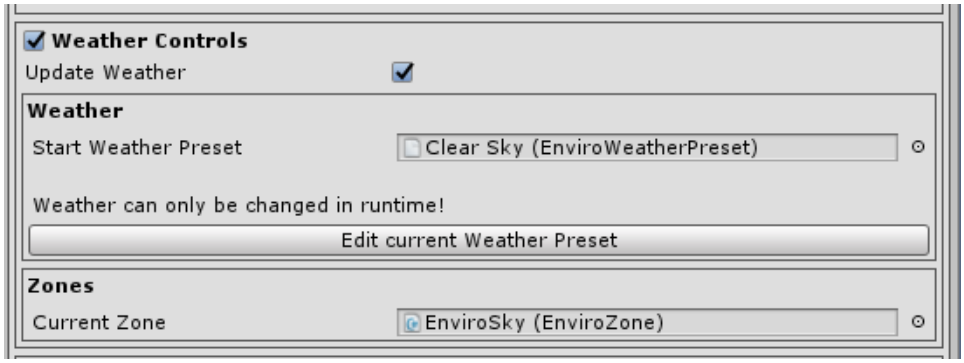
Please note! You have to set the "Progress Time Mode" to "None" if you want to change the time in runtime over inspector sliders! But you still can change time over scripting API without modifying the "Progress Time Mode".

Seasons: *Seasons will be used by various utility components like vegetation growth or season based material/gameobject swap components.*
If you enable "Calc Season" the current season will be chosen based on day of year. Please check the Profile -> Season settings to configure season lengths.

Location: *Here you can setup your location based on latitude and latitude and choose your time-zone.*

Weather Controls

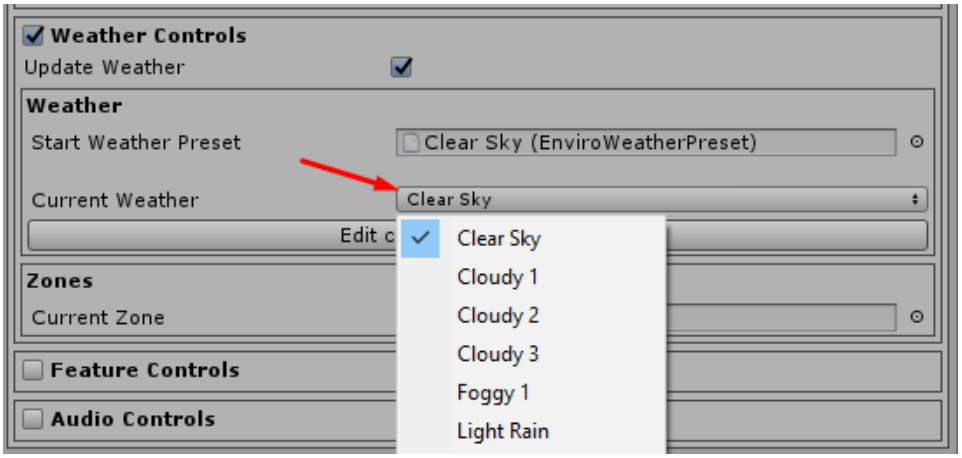
In this section you can control the current active weather, start weather and quickly edit the active weather preset. You also can disable automatic weather changes for all your zones here (Update Weather option).



Tipps:

- *If you enabled "Scene Preview" you can change the preview by assigning different weather presets in "Start Weather Preset". This will allow you to work on your scenes with enviro effects enabled to better showcase how your scene will look in runtime.*

- *In runtime you also can quickly change the current weather in current zone here!*



- ***"Edit current Weather Preset"** will select the currently playing weather preset for quick edits!*

Feature Controls

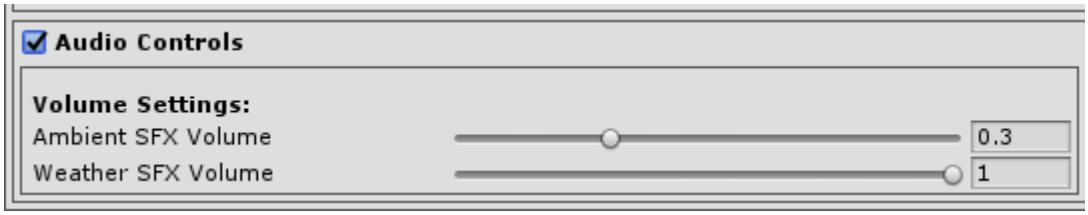
In "Feature Controls" section you can quickly enable and disable enviro effects like clouds, volumetric lighting and lightshafts.

Settings should be self-explaining. But I want to point out the "Scene View Preview".

Enable or disable these to show or hide enviro effects in your scene view while working on your scene. Sometimes you may want to disable these to have better editor performance, or if you don't need to have your scene fogged or clouds in sky.

Audio Controls

In “Audio Controls” section you can set the volume of sound effects used by enviro for ambient and weather.

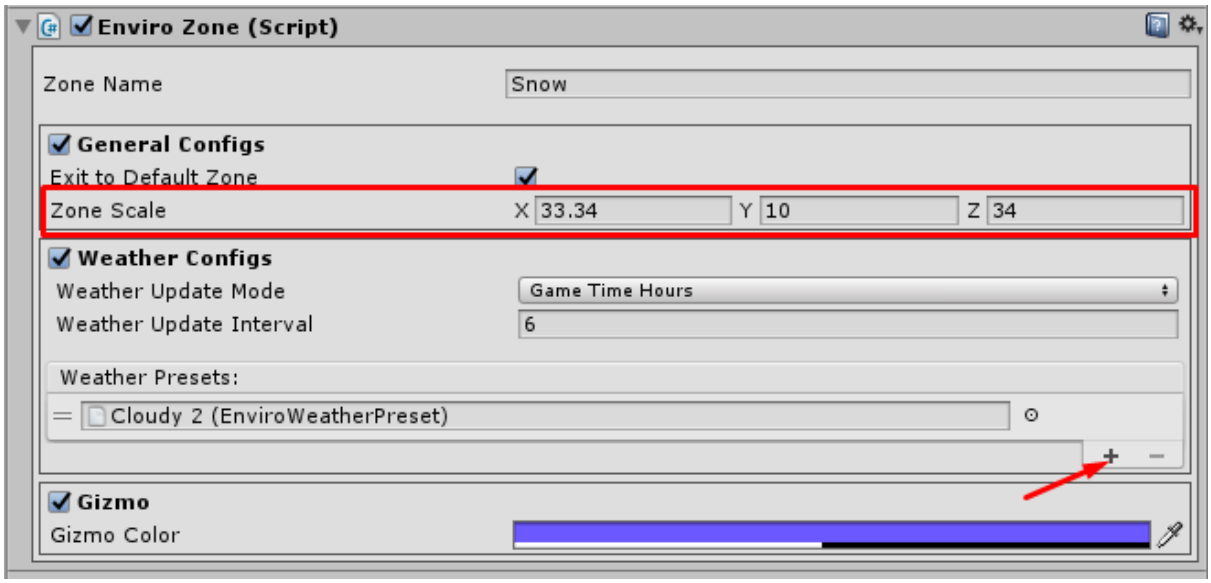


Nothing special here, but you may want to globally set the volume of enviro ambient and weather sfx for your project here.

Enviro Zone Component

The "Enviro Zone" component handles your weather presets and initiate the weather changing. There always is one zone added to the "EnviroSky" object. That one is your default zone and will be used whenever your player/camera is in no other manual added zones. "Manual added zones"? Yes, you can add as many zones as you want to your game world, each with their own weather list and current active weather!
Once your player/camera enters an additional zone the weather will smoothly change to the current active weather in that zone. Use this to have multiple biomes in your game.

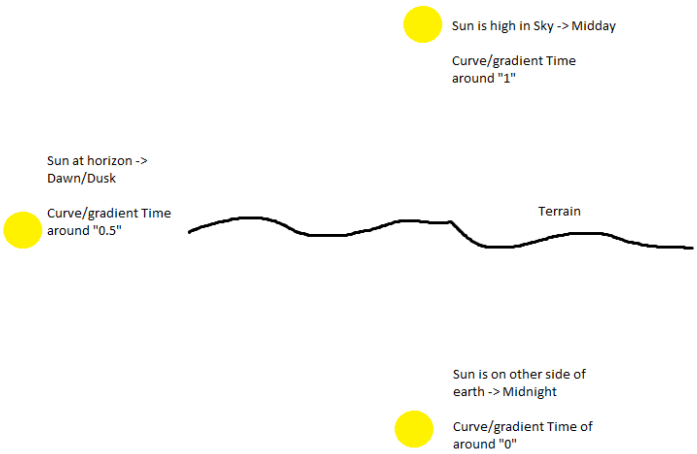
- ***To create a new zone create an empty GameObject (Right-click in hierarchy -> "Create Empty") and add the "Enviro Zone" component to It ("Components" -> "Enviro" -> "Weather Zone").***
- ***Now just set the scale and add your weather presets. That's all!***



FAQ and Troubleshoot

Short answers on frequently asked questions and issues that may happen.

- ***How to tweak enviro curves and gradients?***
  Most curves and gradients are evaluated based on sun altitude in sky.
  It ranges from 0 -> midnight, ~0.5 -> dawn/dusk, 1 -> midday.

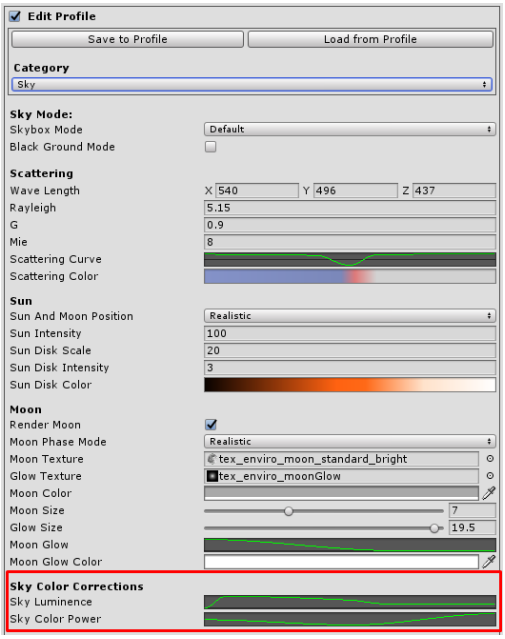


- ***How to make nights brigher or darker? (scene lighting)***
  Select your Enviro Sky Instance -> Edit Profile -> Lighting category:

Set left half of gradients and curves marked in screenshot darker/lower or brighter/higher.

- ***How to make nights brigher or darker? (Sky)***

  Select your Enviro Sky Instance -> Edit Profile -> Sky category:



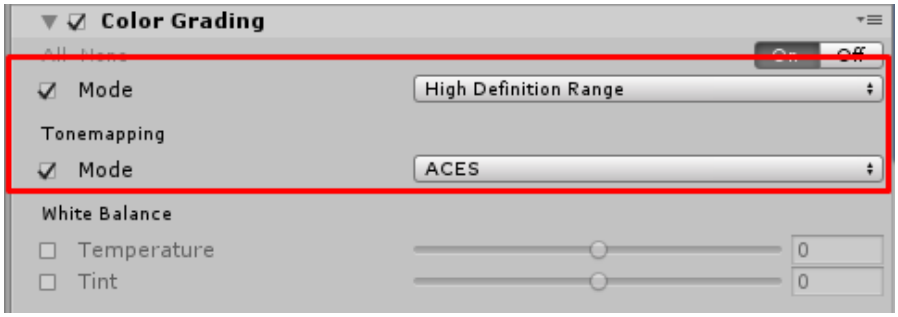
  Tweak left half of marked curves. Higher values in "Sky Luminance" will make sky brighter. Higher values in "Sky Color Power" will make sky darker while colors stronger.

- ***Clouds and Sky are over brighten and look bad.***

  *With default shipped settings Enviro does not limit the colors brightness. This is needed for some post effects to work correctly. For example Bloom. And it also will give you best results and HDR compability. However you **need** to use some tonemapping to get best results. For example using the Unity Post Processing -> Color Grading tonemapping.*

  So download and import Unity Post Processing through the Unity Package Manager and setup in your scenes.
  https://docs.unity3d.com/Packages/com.unity.postprocessing@2.1/manual/Quick-start.html

  Now in your post processing volume profile, add the Color Grading module and activate Tonemapping here. Personally I use the "ACES" mode here.



- ***Scene renders black in combination with Unity Post Processing (Unity 2019).***

  *Please take a look in your Post Processing Layer component on your camera. Make sure that **"Directly To Camera Target"** option is **disabled**. That option will break legacy image effects that using the "OnRenderImage" function.*

Scripting API

Usefull functions and api calls to communicate with enviro in your own scripts.

General

Assign your Camera and Player and start Enviro systems:

EnviroSkyMgr.instance.AssignAndStart (GameObject player, Camera Camera);

Start Enviro in headless server mode:

EnviroSkyMgr.instance.StartAsServer();

Change camera and player in runtime:

EnviroSkyMgr.instance.ChangeFocus (GameObject player, Camera Camera);

Activate or deactivate features:

EnviroSkyMgr.instance.useVolumeLighting = true;

EnviroSkyMgr.instance.useVolumeClouds = true;

EnviroSkyMgr.instance.useFlatClouds = true;

EnviroSkyMgr.instance.useParticleClouds = true;

EnviroSkyMgr.instance.useSunShafts = true;

EnviroSkyMgr.instance.useMoonShafts = true;

EnviroSkyMgr.instance.useDistanceBlur = true;

Change clouds quality:

EnviroSky.instance.ApplyVolumeCloudsQualityPreset(EnviroVolumeCloudsQuality qualityPreset);

Time and Date

Get current Time:

```
int currentSecond = EnviroSkyMgr.instance.Time.Seconds;

int currentMinute = EnviroSkyMgr.instance.Time.Minutes;

int currentHour = EnviroSkyMgr.instance.Time.Hours;

int currentDay = EnviroSkyMgr.instance.Time.Days;

int currentMonth = EnviroSkyMgr.instance.GetCurrentMonth();

int currentYear = EnviroSkyMgr.instance.Time.Years;

float timeOfDay = EnviroSkyMgr.instance.GetTimeOfDay();
```

Get current sun and moon "time" (0-1). Usefull to use in curves and gradient to evaluate values:

```
float solarTime = EnviroSkyMgr.instance.Time.solarTime;

float lunarTime = EnviroSkyMgr.instance.Time.lunarTime;
```

Get a time string to be used in your UI:

```
string timeString = EnviroSkyMgr.instance.GetTimeString();

string timeStringWithSeconds = EnviroSkyMgr.instance.GetTimeStringWithSeconds();
```

Set time of enviro:

```
EnviroSkyMgr.instance.SetTimeOfDay(float tod);

EnviroSkyMgr.instance.SetTime(int year,int day,int hour,int minute,int second);

EnviroSkyMgr.instance.SetTime(System.DateTime dateTime);
```

Set time progress mode of enviro:

```
EnviroSkyMgr.instance.Time.ProgressTime = EnviroTime.TimeProgressMode.None;
```

Seasons

Season enum:

```
EnviroSeasons.Seasons.Spring
EnviroSeasons.Seasons.Summer
EnviroSeasons.Seasons.Autumn
EnviroSeasons.Seasons.Winter
```

Activate/deactivate auto season changes:

```
EnviroSkyMgr.instance.Seasons.calcSeasons = false;
```

Get current season:

```
EnviroSeasons.Seasons season = EnviroSkyMgr.instance.Seasons.currentSeasons;
```

Set current season:

```
EnviroSkyMgr.instance.ChangeSeason(EnviroSeasons.Seasons.Spring);
```

Weather

Get current active weather:

```
EnviroWeatherPreset weather =
EnviroSkyMgr.instance.Weather.currentActiveWeatherPreset;
```

Get a list of all available weather presets:

```
List<EnviroWeatherPreset> presets = EnviroSkyMgr.instance.Weather.weatherPresets;
```

Get current active weather zone:

```
EnviroZone zone = EnviroSkyMgr.instance.Weather.currentActiveZone;
```

Get snow and wetness intensity:

```
float currentWetness = EnviroSkyMgr.instance.Weather.curWetness;
float currentSnow = EnviroSkyMgr.instance.Weather.curSnowStrength;
```

Set current active weather with smooth transition:

```
EnviroSkyMgr.instance.ChangeWeather(int weatherID);
EnviroSkyMgr.instance.ChangeWeather(EnviroWeatherPreset weatherPreset);
```

Set current active weather without transition:

```
EnviroSkyMgr.instance.ChangeWeatherInstant(int id);
EnviroSkyMgr.instance.ChangeWeatherInstant(EnviroWeatherPreset weatherPreset);
```

Events

```
EnviroSkyMgr.instance.OnWeatherChanged += (EnviroWeatherPrefab type) =>
{
Debug.Log("Weather changed to: " + type.Name);
};

EnviroSkyMgr.instance.OnSeasonChanged += (EnviroSeasons.Seasons season) =>
{
Debug.Log("Season changed");
};

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

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

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

EnviroSkyMgr.instance.OnDay += () =>
{
Debug.Log("Day!");
};

EnviroSkyMgr.instance.OnNight += () =>
{
Debug.Log("Night!");
};
```