Documentation

V2.0.3

Thank you for buying

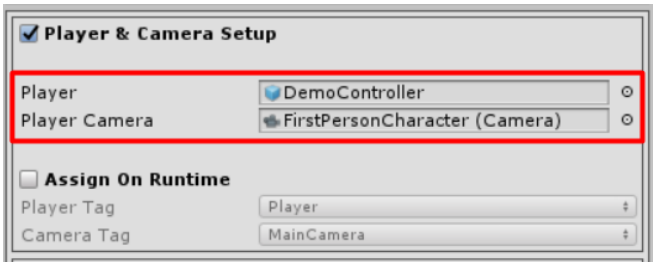
Enviro - Sky and Weather

Getting Started

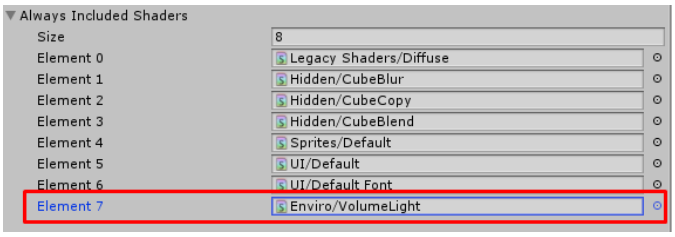
It's really easy to setup enviro in your project! Import the package and follow these easy steps to get started!

1. Drag and drop the "EnviroSky" prefab ("Enviro - Dynamic Environments" -> "Prefabs") into your scene.

2. Now select the EnviroSky object in your scene and click on the “Player & Camera Setup” checkbox to show this section. Assign your "Player" and your "Player Camera” **OR** check “Assign On Runtime” and choose your tag’s.



3. 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.



Further settings to check:

- *Disable other directional lights in your scene!*
- *For best visuals use "Linear Color Space".*
- *Enviro uses one layer for moon rendering. Default: 30*
- *Do not use Desktop and Mobile version in same project, it won’t work! This will be fixed in Mobile 2.0 release!*

Basics

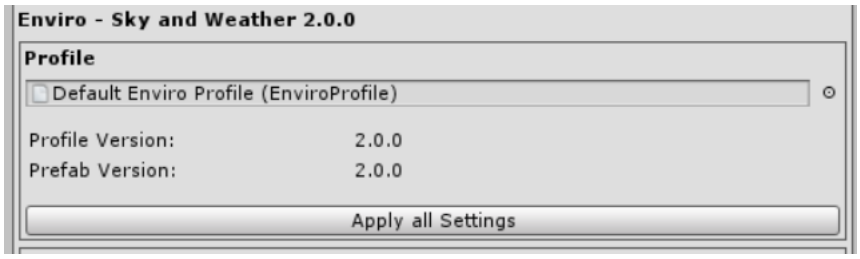
Okay now the basics! The final sky output will be controlled in two separate parts.
*First the global "**Enviro Profile**", second the current active "**Weather Preset**".*

Profiles

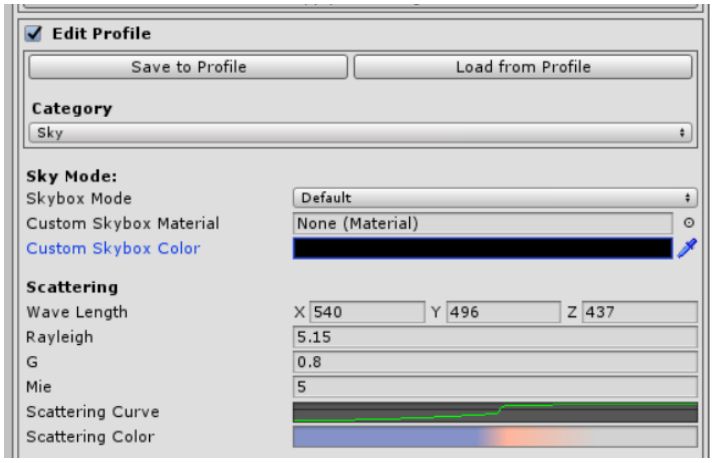
This object will hold all global information and can be saved and loaded in run- and design time!

- ***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"!

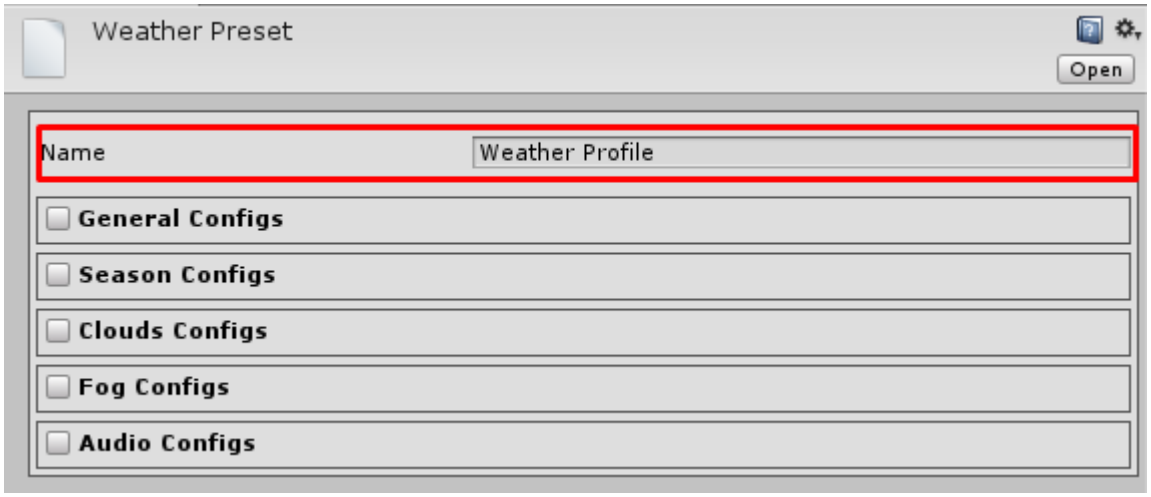
If you tweaked the settings in runtime and want them saved you have to save in runtime and load them when you are back in designtime!

*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***

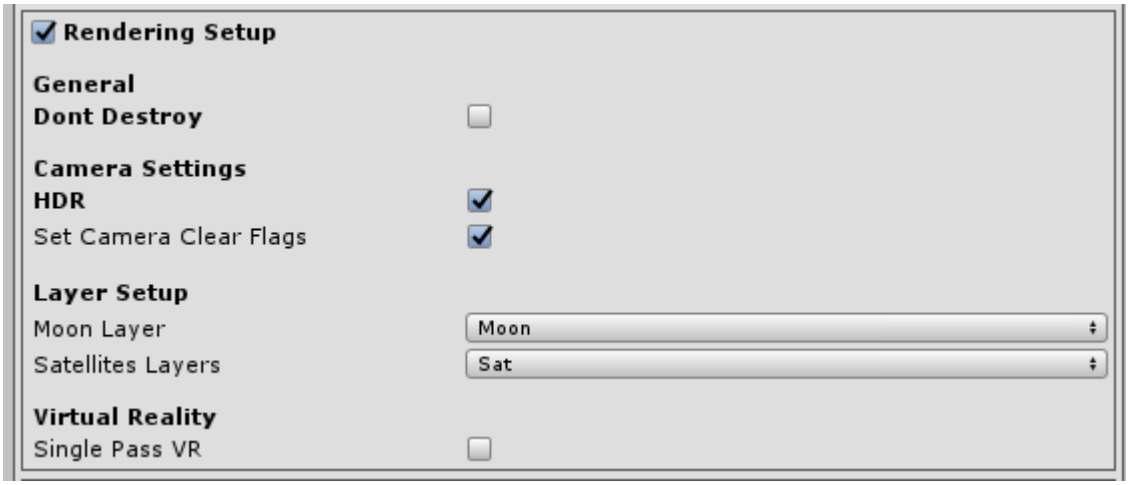


*Enviro only can use a weather preset, once it is added to a zone! To add your weather presets to the system you have to assign it in one of your **"Enviro Zone"** components. There already is a default zone on the EnviroSky object. More about zones later!*

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

Rendering Setup

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



Don't Destroy: *Enable if you are working with multiple scenes and using the same camera/player in those scenes. It will prevent Enviro from getting destroyed on scene loading.*

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)*

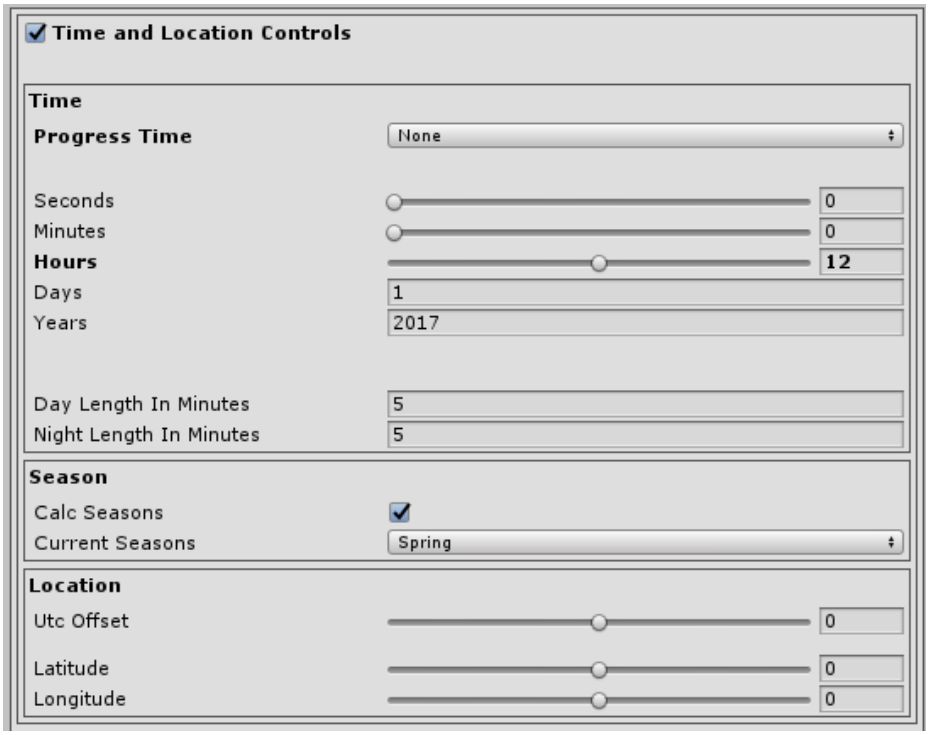
Moon Layer: *This layer is needed to correctly rendering the moon. You should create a new layer and assign it here to don't risk any conflicts.*

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.*

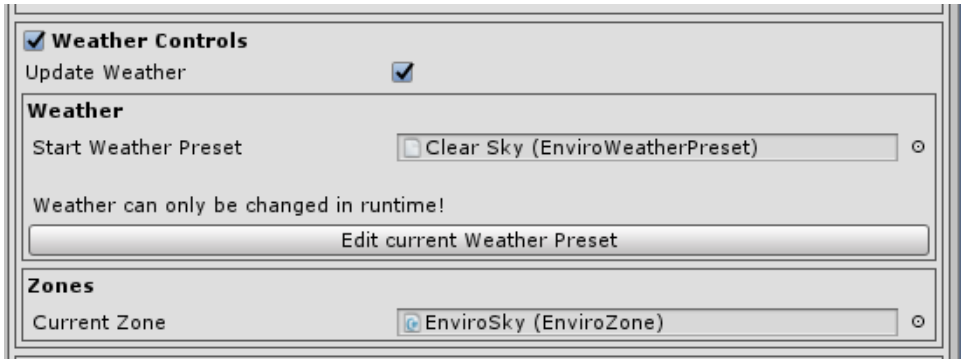
Please note! You have to set the "Progress Time Mode" to "None" if you want to change the time in runtime over inspector sliders! But you still can change time over scripting API without modifying the "Progress Time Mode".

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.

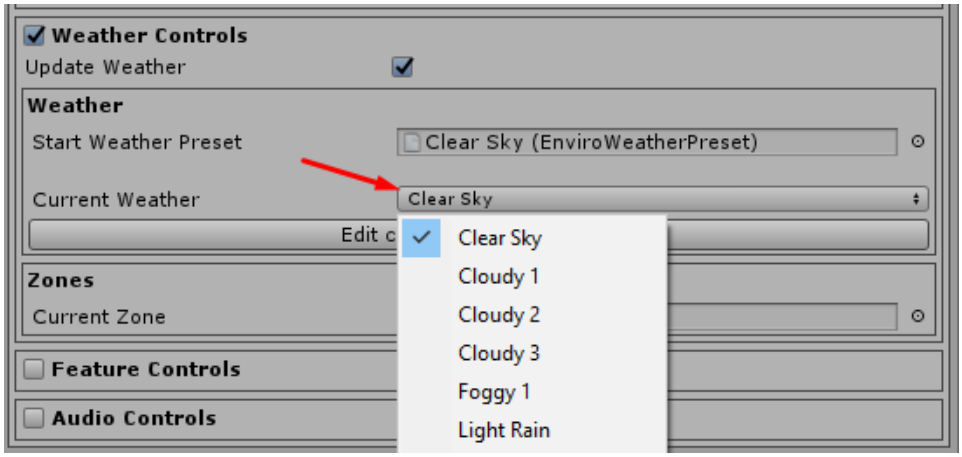
Location: *Here you can setup your location based on latitude and latitude and choose your time-zone.*

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).



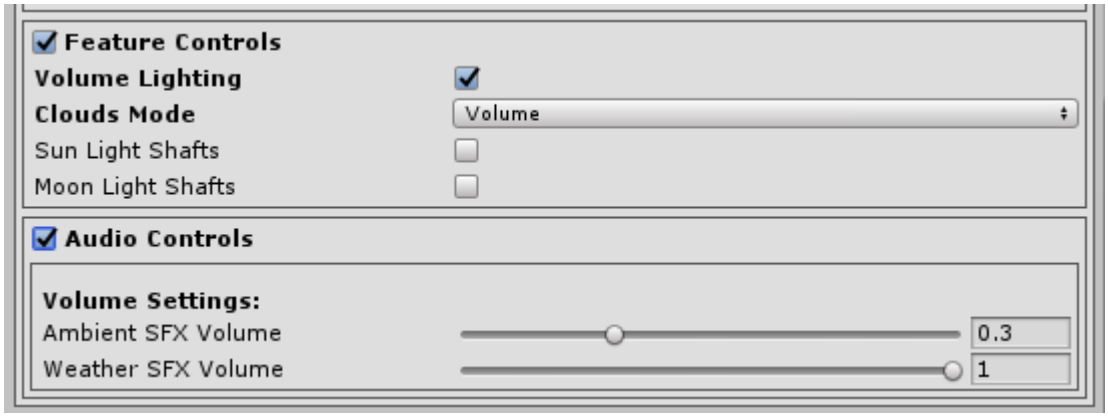
In runtime you also can quickly change the current weather in current zone here!



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

Feature and Audio Controls

In "Feature Controls" section you can quickly enable and disable enviro effects like clouds, volumetric lighting and lightshafts. In "Audio Controls" section you can set the volume of sound effects used by enviro for ambient and weather.



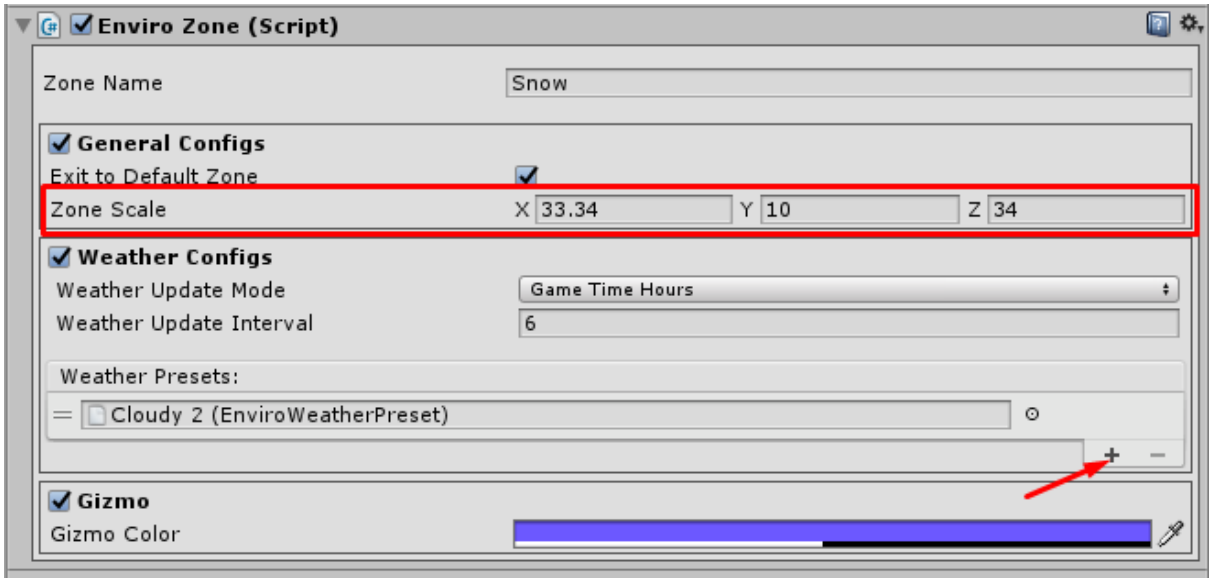
Settings should be self-explaining. But I want to point out the different cloud modes!

"None"	*Disables all cloud rendering.*
"Volume"	*Only rendering the advanced volumetric clouds.*
"Flat"	*Renders only flat clouds. You should use these for slower target hardware, best performance or Virtual Reality.*
"Both"	*Enables volume and flat clouds at the same time.*

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!***



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:

```
EnviroSky.instance.AssignAndStart (GameObject player, Camera Camera);
```

Start Enviro in headless server mode:

```
EnviroSky.instance.StartAsServer();
```

Change camera and player in runtime:

```
EnviroSky.instance.ChangeFocus (GameObject player, Camera Camera);
```

Change clouds mode:

```
EnviroSky.instance.cloudsMode = EnviroSky.EnviroCloudsMode.None;
```

Change clouds quality:

```
EnviroSky.instance.cloudsSettings.cloudsQuality = EnviroCloudSettings.CloudQuality.Low;
```

Enable/disable volumetric lighting:

```
EnviroSky.instance.volumeLighting = false;
```

Time and Date

Get current Time:

```
int currentSecond = EnviroSky.instance.GameTime.Seconds;

int currentMinute = EnviroSky.instance.GameTime.Minutes;

int currentHour = EnviroSky.instance.GameTime.Hours;

int currentDay = EnviroSky.instance.GameTime.Days;

int currentYear = EnviroSky.instance.GameTime.Years;

float timeOfDay = EnviroSky.instance.GetUniversalTimeOfDay();
```

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

```
float solarTime = EnviroSky.instance.GameTime.solarTime;

float lunarTime = EnviroSky.instance.GameTime.lunarTime;
```

Get a time string to be used in your UI:

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

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

Set time of enviro:

```
EnviroSky.instance.SetInternalTimeOfDay(float internalHour);

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

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

Set time progress mode of enviro:

```
EnviroSky.instance.GameTime.ProgressTime = EnviroTime.TimeProgressMode.None;
```

Seasons

Season enum:

EnviroSeasons.Seasons.Spring

EnviroSeasons.Seasons.Summer

EnviroSeasons.Seasons.Autumn

EnviroSeasons.Seasons.Winter

Activate/deactivate auto season changes:

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

Get current season:

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

Set current season:

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

Weather

Get current active weather:

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

Get a list of all available weather presets:

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

Get current active weather zone:

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

Get snow and wetness intensity:

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

Set current active weather with smooth transition:

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

Set current active weather without transition:

```
EnviroSky.instance.SetWeatherOverwrite(int id);
```

Events

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

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

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

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

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

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

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