



Drop-In Sky & Weather — Realistic URP System

Documentation · v1.0.0 · Havoila Studio

Havoila Studio · Version 1.0.0 · Unity 6000.3 and newer · Universal Render Pipeline 17 and newer

Drop-In Sky & Weather gives Unity's Universal Render Pipeline a physically based sky with volumetric clouds and weather. The sky is computed from the air around a planet, lit by the sun and the moon, from dawn to the stars. Clouds drift and change with the wind, cast shadows, and light shafts fall between them. Rain and snow fall from the clouds, lightning strikes in them, and the ground gets wet or white.

There is no setup: add a sky to a URP scene and it becomes the skybox and lights the scene. There is no renderer feature to add (only the optional Soft Shadows use URP's own Screen Space Shadows feature). The time of day, a preset and a few sliders change the whole scene, and scripts, Timeline and the demo's weather cycle can drive it.

In Unity, its menus, shaders, sub graphs and scripts use the short name **Drop-In Sky** (Tools > Havoila Studio > Drop-In Sky, the `HavoilaStudio.DropIn.Sky` namespace).

1. Key features

- **Physically based atmosphere.** The sky's colours come from the air, the haze and the ozone (after Hillaire, 2020). A blue day, a white horizon, orange and red sunsets and the blue hour all follow the sun's height. Change the air to make other worlds.
- **Volumetric clouds.** Fair-weather cumulus, flat layers and towering storm clouds in a layer you set, with silver linings and light scattered inside them. Cumulonimbus rise far above the layer and spread into anvils. They drift with the wind and change shape as they go. High, thin cirrus streaks above them catch the last light of sunset, and a veil of them rings the sun and the moon with a 22° halo and sun dogs.
- **Cloud shadows and light shafts.** The clouds' shadows drift over the scene (the sun light's cookie), and shafts of sunlight fall between them.
- **Atmospheric fog.** The same air fogs the scene. Distant hills fade into the sky's colours, with mist near the ground and shafts of light behind trees and walls. It works with transparent objects, MSAA and Unity's own shaders.
- **Scene lighting.** The ambient light and the reflections follow the sky as it is now, clouds included, with nothing to bake. The sun light's colour and intensity follow the air it goes through.
- **Night exposure.** The cameras' exposure follows the light as eyes do, so evenings and nights stay readable, with a blue shift deep in the night. URP has no auto-exposure; the sky adds it.
- **Day and night.** The time of day, the date and the place put the sun, the moon and the stars where they really are, or follow the computer's clock. At night the moon lights the clouds, and the real stars and Milky Way come out: the constellations rise and set as they do at your latitude. Eclipses of the sun and of the moon come on their real dates, a city's light pollution glows on the horizon and under the clouds, and the aurora's curtains fold, ripple and break into rays towards the pole, smooth at any resolution, their arcs, folds, rays and colours yours to set. Shooting stars, satellites and airliners' lights cross the night.
- **HDRI export.** Save the sky as it is now as an EXR panorama, clouds and sun included, for other renderers or baked lighting.
- **Weather.** Rain curtains hang under distant clouds, with a rainbow where the low sun lights them, and rain and snow fall around the camera, only under the clouds. Rain stops under roofs and splashes where it lands. Lightning flashes in the clouds, lights the scene and strikes the ground, with thunder if you give it sounds. The ground gets wet and white,

and your materials show it through a Shader Graph sub graph.

- **Presets and weather cycles.** Nine presets, from clear to stormy, apply at once or blend over time. A weather cycle changes the weather by itself, the same way every time for the same clock.
- **Made for games and films.** A clock you can drive from a network time or a Timeline gives the same sky everywhere and for every scrub. There are events for sunrise, sunset and each hour, save and load, and a recording mode for clean video frames with Unity Recorder.

2. Requirements and compatibility

Unity	6000.3 (Unity 6.3 LTS) or newer. Tested with 6000.3.25f1; imports and compiles cleanly in 6000.6.0f1.
Render pipeline	Universal Render Pipeline 17 or newer. Tested with URP 17.3, Forward and Forward+. Built-in RP and HDRP are not supported.
Platforms	Platforms with compute shaders: Windows, macOS, Linux and consoles. Tested on Windows with Direct3D 12. Mobile and WebGL are not supported in this version.
XR	PC VR (OpenXR and others), single-pass instanced or multi-pass: the sky, the clouds, the atmospheric fog, rain and snow, lightning. No lens flare in the headset. Standalone headsets (Android multiview) are not supported.
URP settings	Turn on Light Cookies in the URP asset for the clouds' shadows. Nothing else: the fog asks each camera for the depth texture it needs. The optional Soft Shadows (off by default) need URP's Screen Space Shadows renderer feature on the renderer: the Inspector adds it.
Optional packages	Timeline : the Time of Day and Sky Preset tracks compile only when it is installed. Unity Recorder : the recording mode switches on by itself while it captures.
Input	The sky reads no input. The demo scripts work with both the Input System and the legacy Input Manager.
Physics	Rain stops under the colliders of the layers you choose (built-in 3D physics).
Packages	Nothing to install: the sky uses Burst for the rain's raycasts, which URP installs with itself.

3. Package contents

```
HavoilaStudio/DropInSky/  
  Runtime/           The sky component, atmosphere, clouds, fog, weather, lightning, presets, weather cycle,  
Timeline tracks  
  Editor/           Inspectors, the Scene view overlay, the preset picker, the HDRI export, Tools menu, Timeline  
clip editors  
  Shaders/          Sky, environment, fog, precipitation, aurora, soft shadows and far shadows shaders, compute  
shaders,  
                      Include/DropInSky.hlsl and SubGraphs/ for your own materials  
  Materials/         M_DropInSky (the skybox material)  
  Textures/          The night sky's all-sky maps: Night_Stars, Night_MilkyWay  
  Presets/           Clear, Fair Weather, Scattered, Cloudy, Overcast, Stormy, Hazy, High Clouds, Cirrus  
  Demo/             The demo scene (section 4) and what it is made of: the park's plants, rocks, terrain layers and  
                      shaders (Park/), its places, animals, terrain tiles and forest (Wild/), demo scripts  
(Scripts/), and  
                      the Weathered Lit shader (Shaders/), an example of the weather on a Shader Graph material  
  Documentation/    Markdown and PDF guide  
  README.txt, CHANGELOG.md, Third-Party Notices.txt
```

4. Quick start

Open the demo

Open **Demo/DropInSky_Wild.unity** and press Play. It is a park of 12 × 8 km in 24 terrain tiles, where the sky has room: forests, meadows and lakes in a granite valley, a salt lake and sand dunes in the desert, a highway with a gas station and its neon, a lake house with seaplanes, a log cabin, a lookout, a wind farm; deer, birds and rattlesnakes. Its 225,000 trees and rocks are drawn by Park Forest (culled on the GPU, drawn in a few hundred indirect draws).

- Fly with the right mouse button and **W A S D** (**Q E** down and up, **Shift** faster, mouse wheel for speed).

- Jump to viewpoints with **1 - 9**, or with the buttons in the panel.
- The panel on the right sets the time of day (and lets it pass, with the length of a day), blends to presets, sets the rain, snow and lightning, turns the weather cycle on, turns the fog on or off and sets its mist, and shows the frame time. **H** hides it.

The park's scripts and shaders are there to show the sky, not to be reused as they are, but they show how to use what it publishes (section 12): **Park Sky Fog** (Demo/Park/Shaders/ParkSkyFog.hlsl) fogs transparent objects with the sky's fog, Park Water reflects the sky's environment cube, Park Water Heights publishes the water's heights so that the rain stops at its surface, and Park Wind passes the sky's wind on to Park Foliage.

Add a sky to your scene

1. **Tools > Havoila Studio > Drop-In Sky > Create Sky** (also **GameObject > Havoila Studio > Drop-In Sky > Sky**). The sky uses the scene's sun (Lighting > Sun Source, else its first directional light, else a new one) and becomes the skybox.
2. Pick a preset in **Apply a preset...** at the top of its Inspector.
3. Set the **Time of Day**. In Play mode, turn on **Time Passes** to let the day go by.
4. In the URP asset, make sure **Light Cookies** is on, for the clouds' shadows.

Turn off Unity's own fog (**Lighting > Environment > Fog**): the sky's fog replaces it. The Inspector offers to do it.

The Scene view overlay

The **Drop-In Sky & Weather** overlay in the Scene view (its Overlays menu, shown by default) sets the time of day, applies a preset, sets the rain or snow and the lightning, turns the weather and the fog on or off, and selects the sky, without going to its Inspector. Without a sky, it offers to create one. Every change can be undone.

Wet and snowy materials

Use the **Drop-In Sky Weather** sub graph in your Shader Graph materials (see section 8), or start from **Demo/Shaders/Weathered Lit**.

5. How it works

The sky is a skybox material and a component, with no renderer feature. It works with reflection probes, lighting bakes and other assets that draw the sky. (The optional Soft Shadows rely on URP's own Screen Space Shadows renderer feature.)

- **Before each frame**, the sky fills small tables for the air (Hillaire's transmittance, multiple scattering and sky-view look-up tables) for the sun and the camera's altitude.
- **The clouds** are marched into a panoramic cloud map around the camera, blended over frames (or rendered whole for recordings). Their shadows go into the sun light's cookie, their edges as soft as the sun's disk makes them at the clouds' distance: a few metres at noon, a hundred metres or more at sunset, when the clouds that shade the scene stand tens of kilometres away.
- **The skybox** reads the tables and the cloud map. The same shading, without the sun's and the moon's disks, is drawn into a small cube each frame: it gives the scene its reflections and ambient light, and other assets can read it. The cube also gets the fog of the sky as the camera sees it (the air grey under the clouds, the rain's haze, the mist), so wet ground and water under a storm reflect a grey horizon, not a clear one.
- **The fog** fills a froxel volume per camera (tiles of the screen, slices in depth) and is applied before the transparent objects, with two blended draws. It never reads the scene's colour, so it works with MSAA. While there is mist, a second volume of the same size holds the mist's light, and how much mist lies in front of each pixel is worked out along its own ray: a mist a few metres thick, seen from high up, is thinner than the slices far away, and shows no rows.
- **Rain and snow** are drawn per camera, with one procedural draw and no particle system. Rays cast down around the camera find what they fall on.
- **Soft Shadows** (optional) take the place of URP's screen-space shadow texture, which opaque objects drawn with URP's shaders read for the sun's shadow. For each pixel, the casters around its place in URP's shadow map are found, their distance along the light sets the width of its shadow's edge (that distance times the angular radius of the sun's

disk), and the map is filtered over that width (percentage-closer soft shadows, Fernando 2005). A blur that keeps to the surfaces then smooths the wide edges.

Everything is computed on the GPU with compute shaders. The CPU work each frame is small: setting parameters, and, while it rains, some ten thousand raycasts every few seconds on the job system's worker threads.

6. The Inspector

The Inspector groups the settings in sections. Most headers show a short status on the right (the time, the sun's height, the cloud cover, the moon's phase, the weather, the exposure the night adds); **Expand all** and **Collapse all** are at the top, and **Documentation** opens this guide. Hover over a setting for its description.

Presets

Apply a preset..., at the top, opens the presets, each shown by a small render of the sky with it (the sky alone at 14:00, from the main camera's place; hover one for what it is for). Click one to apply it at once. The renders are made the first time, one per editor frame, and kept in the project's Library; a changed preset is rendered again, and **Refresh** renders them all again (after changing the sky's place or date, say). **Save...** saves the sky's current look as a new preset asset: its Atmosphere, Clouds and Weather groups on (switch Fog and Sun on in the asset to keep them too). See section 7.

Time of Day

- **Time** (hours) and **Date** (month, day and year; leap years count).
- **Computer's Clock**: the date and the time follow this computer's clock, in the Time Zone, in Edit mode too (Time, Date, Time Is and Time Passes are then greyed out).
- **Time Is**: **Solar** (12:00 is when the sun is at its highest) or **Clock** (the time on a clock in the **Time Zone**, in hours ahead of UTC, daylight saving included). In Clock time the sun crosses the south when it really does at the **Longitude**, a little before or after noon through the year. **This PC's** sets the Time Zone to the computer's.
- **Longitude**, **Latitude**, and **North** (the direction of north in the scene).
- **Real Moon**: the moon where it really is on the date, with its real phase (shown under it) and its path up to 5° off the sun's. Off, set the **Moon Phase** yourself (0 new, 0.5 full); it moves on as time passes.
- **Time Passes** and **Day Length** (minutes for a whole day): time goes by in Play mode, and the date and the moon with it. Off, the clouds stand still too, in Edit mode as well (with the **Game** clock).
- Turn the section off to place the sun and the moon with their lights' rotations instead.

Eclipses come with the real moon, on their real dates, where the scene's place sees them. In a solar eclipse the moon covers the sun's disk and the daylight dims with the part of the sun it hides. At totality the sky darkens, the stars come out and the corona glows around the black moon. In a lunar eclipse the earth's shadow crosses the moon: the penumbra dims it, the umbra turns it copper red, and the moonlight fades. While one is going on, the section says so under the Moon Phase, with how much of the sun or the moon it hides.

The sun, the moon and the sidereal time come from published low-precision formulas, so eclipses come within a few minutes of their real times. To see one: the total solar eclipse of 12 August 2026 is total over Reykjavik (latitude 64.15, longitude -21.94) at 17:49 UTC and over northern Spain around 18:29; the total lunar eclipse of 3 March 2026 lasts from about 11:05 to 12:05 UTC, seen from the Pacific. Set **Time Is** to Clock with a Time Zone of 0 to enter UTC.

Sun

- The header's switch shows or hides the sun's disk in the sky. Hidden, the sunlight stays (the light, and the sky it lights), and there is no lens flare.
- **Light**: the sun's directional light.
- **Drive Light** sets its colour and intensity from the air: white and strong high up, dimmer and redder as the sun gets low, off once it has set.
- **Intensity**: the sunlight above the air. The sky's brightness follows it.
- **Shadow Intensity**: how dark the sun's shadows are: the light's own Shadow Strength, edited from here. 1 lets none of the sunlight into its shadows; lower lets some through, a stand-in for the sky's light in the shade.
- **Disk Size** and **Disk Brightness**: how the sun looks.
- **Lens Flare** and **Flare Intensity**: a lens flare when the camera looks towards the sun: a glow, a faint halo and coloured ghosts, from procedural shapes (no textures), tinted by the sunlight.

- What stands in front of the sun hides it, and the clouds do too: it fades out as they cover the sun, and as the sun sets.
- Under water that publishes its surface (Drop-In Water does), it is not drawn: the sun is not seen through the water's surface. It fades out over the first 30 cm under it.
- It uses URP's data-driven lens flare, added to the sun light while the sky renders (hidden, never saved with the scene). A lens flare you put on the sun light yourself is left alone, and the sky's is not added.
- It needs the camera's post-processing, and **Data Driven Lens Flare** in the URP asset (on by default).
- It is not drawn for cameras that render to a VR headset.

Atmosphere

- The header's switch turns the air off: as with an Air Density of 0 (no blue sky, no sunset colours). The haze and the ozone stay as they are set.
- **Air Density**: the blue of the sky and the red of sunsets. 1 is the Earth's.
- **Haze, Sun Glow** (how tight the haze's glow around the sun is) and **Haze Color**: dust, pollution and droplets. Haze whitens the horizon.
- **Ozone**: keeps the sky blue at twilight.
- **Ground Color**: the ground below the horizon, which also reflects light up.
- Under **More**: the air's scattering colours and scale heights, the planet's radius and the atmosphere's thickness.
- **Exposure** and **Altitude** (the scene's height above the sea) are in the Scene Lighting section.

Clouds

- The header's switch turns the volumetric clouds off (the high clouds have their own).
- **Coverage**: 0 clear, 1 overcast.
- **Type**: 0 flat layers, 0.5 puffy cumulus, 1 towering storm clouds.
- **Variation**: how much the coverage and the type change across the sky.
- **Altitude** and **Thickness** of their layer, their **Density**, the **Size** of the shapes, and **Detail**, the erosion into wisps.
- **Weather Size**: the size of the areas of clear sky and dense clouds.
- **Weather Fronts**: wide stretches of more and of fewer clouds that pass with the wind, so the sky clears and clouds over as they go by. **Front Size** is the distance from one stretch of clouds to the next (300 km by default). In a 10 m/s wind a front takes hours to pass, so speed up the time of day to see it. 0 is none.
- **Storm Clouds**: cumulonimbus in some parts of the sky. Their towers rise far above the other clouds, the strongest up to the **Storm Top** (11000 m by default; the tropopause, from some 8000 m near the poles to 16000 m in the tropics). There they spread into a flat anvil blown downwind, thick next to the tower and thinning towards its far edge; **Anvil Spread** sets how far (up to some 20 km past the tower). Weaker storms stop lower and grow no anvil, so the cloud tops vary from place to place. The storms are cells of the weather map about a third of the Weather Size apart; they drift with it, and rain falls under them with the Weather's Precipitation. 0 is none, 1 a storm in most places. The clouds then take up to 1.6 times their usual time, marched through the taller layer.
- **Add Cloud Area** places a Cloud Area 6 km ahead of the Scene view's camera (below); the row also counts the areas in the scene.
- **Wind: Speed** and **Direction** (where it comes from). **Evolution**: how fast the clouds change shape as they drift.
- **Silver Lining** (the bright rims of clouds in front of the sun, 0 to 0.95), **Ambient** and **Multiple Scattering** tune the light in them; for the last two, 1 is physically based.
- **Seed**: another pattern of clouds.
- **Shadows** (a box with its own switch): the clouds' shadows on the scene, their **Strength** and the **Distance** (km, 12 by default) they cover around the camera.

Cloud Areas

A **Cloud Area** (the button in Clouds, **GameObject > Havoila Studio > Drop-In Sky > Cloud Area**, or the component on any object) adds clouds above an area of the world or clears them from it, for art direction: a storm cloud standing over a valley for key art, a bank of clouds over the sea, or a hole of blue sky over the scene.

- **Mode:** **Add Clouds** (the coverage raised towards overcast) or **Clear Clouds** (towards a clear sky).
- **Radius** (m) along the object's X and Z. Its scale stretches the area into an ellipse and its rotation about the vertical turns it. The Scene view draws the area on the ground and at the clouds' base.
- **Amount:** 1 is covered, or clear, in the middle.
- **Softness:** 0 gives a sharp edge; 1 fades all the way from the middle, which makes a single rounded mass.
- **Set Type** and **Type** (Add Clouds): the kind of clouds in the area, tallest in the middle. For a cumulonimbus, use Type 1 with a Softness of 1, and give the clouds' layer a **Thickness** of some 8000 m, since clouds grow no taller than their layer.

The clouds' noise still drifts through the area with the wind, so their shapes change while the area stays; set the Wind Speed to 0, or leave **Time Passes** off, to keep a shape for a still. Rain falls under added clouds as under any other. Up to 8 areas count.

High Clouds

Cirrus, far above the other clouds. The header's switch turns them on or off; they are drawn even with the volumetric clouds off.

- **Coverage** (a few wisps to a veil), **Opacity**, and **Streaks** (a milky layer to long threads).
- **Size**, **Altitude**, their own **Wind**, and **Ambient**.
- **Halo:** the ring 22° around the sun and the moon (red on the inside, the sky a little darker within it) and a faint one at 46°. It shows clearest through a thin, smooth veil: high Coverage, low Opacity and few Streaks. 1 is as it is usually seen, 5 a vivid display.
- **Sun Dogs:** bright spots at the sun's height on either side, just beyond the halo, with white tails away from the sun. They move out as the sun rises and are gone once it is higher than about 60°. The moon gets them too. 1 is as they are usually seen, 5 a vivid display.

Weather

- The header's switch turns all the weather's effects on or off: rain or snow (around the camera, in the curtains, in the haze), lightning and thunder, and the ground's wetness and snow (dry at once when off). The settings stay as they are, and presets and the weather cycle go on setting them.
- **Precipitation** (0 none, 1 a downpour) and **Snow** (0 rain, 1 snow, sleet between). It falls beneath the clouds, the thicker the more: in curtains under distant clouds, around the camera where their shadows fall (at night, with no sunlight to cast them, all around the camera). Nothing falls without the volumetric clouds, and the ground stays dry.
- **Rain:** **Density** (how many drops, against a downpour's), **Streak Length** (a camera's exposure: 1 is 1/30 s; longer reads as heavier rain), **Drop Width** (mm), **Fall Speed** (m/s: 4 to 9 for rain, 2 to 4 for drizzle) and **Opacity**.
 - **Rainbow:** where the sun shines on falling rain, with the sun behind you and lower than 42°: in the rain you stand in (a whole arc, against clouds or clear sky alike) and in the rain curtains under distant clouds, wherever the sun gets through to them. The primary bow is red on the outside. The secondary bow, beyond it, is red on the inside, less than half as bright, and shows only in heavy rain (Precipitation above 0.7), so lighter rain gives a single bow. The rainbow fades with the rain: none in rain too thin to see. Its colours come from water's dispersion at 12 wavelengths. 1 is nature's brightness; snow makes none.
 - **Rainbow Width:** 1 is nature's width; more gives broader, softer bands. The colours spread over the same angle whatever the width.
 - **Secondary Bow:** 1 is nature's, 0 never shows it (always a single rainbow). It fades in from a Precipitation of 0.7 and is full at 0.8.
- **Snow:** **Density**, **Flake Size** (cm), **Fall Speed** (about 1 m/s; more for wet snow), **Sway** (m) and **Opacity**.
- **Rain and Snow:**
 - **Draw Distance** (m): how far around the camera drops and flakes are drawn; beyond it, the fog's haze stands for them. Beyond some 10 m they are drawn in layers, each twice as far out as the one before, sparser, and each drop there drawn a little stronger for the drops it stands for. Each doubling of the distance draws twice as many as the one before (at 40 m, some 70,000 far drops in a downpour; 400,000 at most). Drops, flakes and splashes are seen through the fog, so they fade into the haze and mist with distance.

- **Wind:** how much the wind carries them and slants the distant curtains (1: the clouds' wind, a third of its speed near the ground).
- **Brightness** of the drops, flakes and splashes.
- **Haze:** the haze they make in the air under the clouds (0: none).
- **Distant Curtains:** how dense the grey curtains under distant clouds are.
- **Stopped By:** the layers whose colliders stop rain and snow, such as roofs, porches and bridges. Rays are cast down around the camera. **None** lets it fall through everything. Water that publishes its surface (Drop-In Water does) stops them too, with no collider: the drops end at the surface instead of falling to the bottom, and splash there: droplets thrown up, without the ring they leave on the ground, since the water draws its own rings. They splash at the water's rest level, so on high waves some float a little above the troughs.
 - **Moving objects**, meaning colliders on a Rigidbody (dynamic or kinematic: vehicles, crates, lifts, platforms), are found again every frame instead of by the rays, which would lag behind them. The drops stop on their tops where they are now, and nothing splashes on the ground beneath them. The rain splashes on their upward faces open to the sky, and each splash rides along with the object it landed on. Colliders without a Rigidbody, and terrain, are treated as standing still.
- **Splashes:** rain splashing where it lands, with their **Splash Size**, **Splash Amount**, **Splash Opacity** and **Splash Distance** (m, 25 by default). The rings the rain makes in the puddles of weather materials (section 8) follow the Splash Amount and fade out at the Splash Distance. Drops, flakes and splashes stay in place in the world as the camera moves. Beyond the drops and flakes near the camera, a sparser layer fills the distance (to the Draw Distance), so the fall keeps its direction when the camera moves.
- **Wind Zone:** a Wind Zone to follow the clouds' wind, with a **Strength**. Trees, grass and particles then sway with the weather.
- **Lightning: Frequency** (0 none, 1 strikes every few seconds) and **Brightness**. It needs the volumetric clouds.
 - **Thunder:** your sound clips, played after the delay of the strike's distance, quieter and duller when far.
 - **Thunder Volume.**
- **Wetting Time:** minutes the ground takes to get soaked in heavy rain, or white in heavy snow. It dries or thaws three times slower (snow faster in rain). In Edit mode the ground follows the weather at once. The section shows how wet and white the ground is now.
- **Add Weather Cycle** adds a Sky Weather Cycle (section 7).

Fog

- The header's switch turns the fog off: the scene keeps its colours to the horizon; the sky and the clouds keep their own air.
- **Distance Scale:** how far away the scene looks. 1 is true to scale, and hills turn blue over tens of kilometres. Use more for scenes smaller than real landscapes. 0 leaves only the mist. Seen from high up, where the horizon dips below the level and the ground beyond the scene shows past its edge, that ground fades into the same haze as the scene's. It fogs the scene, not the sky and the clouds, which have their own air. What is far away fades into the colour of the air in front of it, close to the horizon's: over the sea, which reflects the horizon, it shows less than on land.
- **Fog Color:** tints the light the air adds. White keeps the sky's colours (blue by day, orange towards a low sun).
- **Mist** and **Mist Height:** white mist near the ground. 0.2 is a light haze, 0.5 a mist, 1 a thick fog. With a large Mist Height (a few hundred metres) it fills the scene: a fog bank.
- **Mist Color:** tints the mist's light (smoke, dust, a coloured haze). It hides what lies behind it as much, whatever its colour.
- **Shafts of Light: From the Clouds** (the clouds' shadows in the air; the clouds' **Shadows** must be on) and **From Objects** (the scene's shadows in the air near the camera), with a **Strength** for all of them.
- **From Far Away:** the shadows of what stands far away (mountains, hills, tall buildings) in the air. When the sun or the moon has set behind a mountain, the air in front of it gets none of their light, and dark rays fall from the peaks. In the air, it works only with **From the Clouds** on.
 - **On the Scene:** these shadows on the ground, the water and everything that reads URP's shadows too, within URP's shadow distance. URP's own shadow map holds only the casters close to what the camera sees: at sunrise and sunset, a ridge kilometres away would leave a lake in its shadow lit, with the sun's glint on it. The sky writes the

mountains URP leaves out into URP's shadow map, each frame, and publishes its far shadows for other assets: Drop-In Water reads them beyond URP's shadow distance too (section 12). What stands only a few texels above the light's path is left out (some ten metres at sunrise, more with the sun high): the maps are coarse.

- It uses coarse shadow maps seen from the sun and from the moon (the moon's at night only, 16 MB each): 2048 texels across twice the **Far Distance** (km, 20 by default) around the camera, and only as tall as the landscape across the light (its terrains' heights). With a low sun, that makes their texels a few metres tall where they are tens of metres wide, so the ridges' shafts at sunrise move smoothly as the sun rises. Their shadows are softened over a couple of texels, as the sun's disk softens shadows far from what casts them. They are rendered again when the light has turned by a fraction of a texel or the camera has moved enough.
- Both faces of the shadow casters block the light, so a landscape made of one-sided surfaces doesn't let light through from below. For URP's own shadows near the camera, set such a mesh's **Cast Shadows** to **Two Sided**.
- **Far Casters** sets the layers that cast these shadows. **On the Scene**, **Far Distance** and **Far Casters** show while **From Far Away** is on.
- A Scene view far from the game's camera gets maps of its own (another 16 MB, 32 at night), so that neither view draws them again for the other every frame.
- The fog's resolution is set in the Quality section.

Moon

- The header's switch shows or hides the moon. Off, it gives no light, and there are no eclipses.
- **Light**: an optional directional light. With **Drive Light**, the sky sets its colour and intensity and makes it the main light while it outshines the sun.
- **Intensity**, **Disk Size**, **Disk Brightness** and **Color**.

The moon shows its real near side. The seas (maria) are at their places: Imbrium, Serenitatis, Tranquillitatis and Crisium, Oceanus Procellarum, Nubium and the others. There are bright young craters with their rays (Tycho, Copernicus, Kepler, Aristarchus) and dark-floored Plato and Grimaldi. When the disk is drawn large, small craters show too, with their sides lit or shaded near the terminator. Its north points to the celestial pole, so it appears upside down from the southern hemisphere, as it should. The full moon is as bright at its edge as in its middle, as the moon's dusty ground makes it. The disk keeps its brightness whatever the Night Exposure adds, so its face is not washed out.

Night Exposure

URP has no auto-exposure. With this section on, the sky sets the cameras' exposure from the light the scene gets, as eyes and cameras adapt. Evenings and nights come out brighter, so the scene stays readable, and the day is left as it is.

The light the scene gets is the sun's and the moon's lights (the sun's through the clouds around the camera) and the ambient light. When it falls more than an f-stop below a clear day's, the exposure goes up by part of the difference.

- **Night Brightness**: how much of the darkness is made up for. 0 leaves nights as dark as their light; 1 makes them as bright as the day.
- **Max Boost**: the most the exposure goes up, in f-stops (1 doubles the brightness).
- **Adaptation**: how long the exposure takes to follow the light in Play mode (dusk in a time-lapse, going into a cloud's shadow). In Edit mode it follows at once.
- **Blue Shift**: deep in the night, colours fade and turn blue, as the eyes see in the dark.

It works through the cameras' **Post Exposure** (Color Adjustments): the sky adds its own hidden global volume, which is never saved. The boost is added to the Post Exposure your volumes give where the main camera is, and the blue shift to their Saturation and Color Filter, so your settings still count. The cameras need **Post Processing** on. The header shows the boost now.

Night Sky

The night sky comes from two all-sky maps of the real sky (**Textures/Night_Stars** and **Night_MilkyWay**, in right ascension and declination). The sky turns them with the time of day, the date and the latitude, so the constellations rise and set where they really do: the galactic centre low in the south on a summer night in the north, the Southern Cross from the southern

hemisphere.

- **Stars:** their brightness. **Star Density:** how many of the faint ones show. 1 shows all of the map's; less shows only the brighter ones, as under a hazy or a city sky; more brightens the faint ones. The brightest always show.
- **Twinkle:** each star twinkles on its own, more near the horizon.
- **Milky Way:** the brightness of its glow and dust lanes.
- **Airglow:** the darkest the sky gets.
- **Light Pollution:** a town's lights at night. A warm glow lies low in the sky all around, the clouds' bases are lit from below, and the scene's ambient light and mist get warmer. It comes on at dusk and hides the fainter stars. 0 is a dark site, 1 the edge of a city, 2 its centre.

Shooting stars, satellites, airplanes and the aurora each have a box with a switch in its header. Switched off, they don't show, and their settings are kept (greyed out). Switched on while their amount is 0, they get a starting one, so they show at once: a **Frequency** of 0.1 for the shooting stars, an **Amount** of 0.3 for the satellites and 0.2 for the airplanes, a **Brightness** of 1 for the aurora. The shooting stars are on by default, the others off.

Shooting Stars (on by default): meteors streaking across the night sky, mostly heading down. Each one flares up as it enters and streaks on, slowing a little. The path behind it glows as a tail that grows and cools from the head's colour to the tail's. Towards the end the tail draws back into the head, which flares one last time and goes out last.

- **Frequency:** 0 is none, 0.1 (the default) about one every half minute where the camera looks, 1 a meteor storm.
- **Brightness.**
- **Head Colour** and **Tail Colour:** bluish white and orange by default.
- **Length, Width** and **Duration:** each meteor's is picked at random between the two values. Length is how far it streaks (degrees of sky), Width how thick it is (arcminutes; thinner than a pixel draws it fainter rather than thinner), Duration how long it lasts (seconds).
- **Head Size:** how big each meteor's head is against its streak's width, picked at random between the two values (0.05 to 3 by default). 1 is a bright point a little wider than the streak; more makes a larger fireball, its glow larger too; less a dimmer head.

Satellites (off by default): points of sunlight crossing the sky slowly and steadily. They shine only while the sun still lights them, above the earth's shadow: many in the first hours after dusk and before dawn, few or none in the middle of the night. Most are faint; a few are as bright as the brightest stars.

- **Amount:** 0 is none (the default), 1 up to 24 above the horizon at once, most of them low.

Airplanes (off by default): airliners at cruising height, seen only by their lights: red on the left wingtip, green on the right, white at the tail, a red beacon blinking about once a second and white strobes flashing twice every second or so. One passing overhead crosses the sky in a few minutes.

- **Amount:** 0 is none (the default), 1 up to eight in the sky at once, most of them far away and low.

Shooting stars, satellites and airplanes show only at night, come from the sky's clock (a recording or a Timeline shows the same ones every time) and are hidden by the clouds.

Aurora (off by default): the northern or southern lights. Curtains of light hang some 100 km up, in arcs along an oval around the magnetic pole. Each curtain is green from its sharp lower border up, breaks into vertical rays, and turns red towards its top; the bright stretches have a thin pink fringe along their border, and a diffuse red glow lies above the arcs. The curtains fold and curl, drift and ripple, and stretches of them brighten and fade. They are brightest where the eye looks along them, edge-on, as at their folds. They are drawn as smooth geometry over the sky, so they stay sharp and steady at any resolution, and they show in the scene's reflections and ambient light and in an exported HDRI too. The air near the horizon dims and reddens them, and the clouds hide them. They show towards the pole: the north one at a northern **Latitude**, the south one at a southern.

- **Brightness:** 0 for none to 1 for a bright display (up to 4).
- **Placement:**

- **Real:** place them as the real aurora is, in the oval around the magnetic pole (taken at the geographic one), from the scene's **Latitude** and the **Activity**. They are then out of sight at lower latitudes. Off (the default), they stand at the **Height** below.
- **Height:** how high the curtains' lower border stands above the horizon towards the pole, in degrees (90 is overhead), wherever the scene is (Real off).
- **Activity:** the sun's activity: a livelier display, brighter, with taller curtains and a wider glow above them. With **Real** on, how far from the pole the oval reaches too. At 0 (quiet) it is overhead at latitude 68, and seen low towards the pole down to about 60. At 1 (a great storm) it is overhead at 45, and from farther away mostly its red glow shows, low towards the pole.
- **Direction:** turns where the display lies around the horizon, in degrees clockwise from the pole's direction (0 is towards the pole). It turns Real's oval too: it brings the aurora into a view without changing the scene's **North**.
- **Span:** how far across the sky the curtains reach, in degrees of azimuth, roughly as seen from the camera. Their ends fade out. Over 160 they reach on around the oval, all of it at 360: seen from outside the oval (always the case without **Real**), its far side lies low or below the horizon; from inside it (with **Real** on, nearer the pole than the oval), the curtains reach all around the sky.
- **Shape:**
 - **Arcs:** how many arcs of curtains stand side by side, 1 to 6 (3 by default). The first is where the display is placed; the others stand in turn beyond it (towards the pole) and before it, each fainter.
 - **Arc Spacing:** how far apart the arcs are, in km.
 - **Curtain Height:** how tall the curtains stand above their lower border, in km (150 by default). The **Activity** changes it a little: lower when quiet, taller in a storm.
 - **Folds:** how much the curtains fold and curl. 0 gives straight arcs.
 - **Fold Size:** the size of the folds, in km: how far apart their bends are along the curtains, and how deep they go.
- **Look:**
 - **Lower Colour** and **Upper Colour:** oxygen's green at the bottom, and its red towards the tops and in the glow, by default.
 - **Fringe** and **Fringe Colour:** how strong the thin fringe along the lower border of the bright stretches is, and its colour (nitrogen's pink by default).
 - **Rays:** how strongly the curtains break into vertical rays. 0 gives smooth bands; bright rays reach higher than the rest.
 - **Ray Density:** how many rays. More gives finer ones, closer together.
 - **Glow:** the diffuse red glow above the arcs (in the Upper Colour).
- **Motion:**
 - **Speed:** how fast the curtains move, with the sky's clock, so a recording or a Timeline shows the same aurora every time.
 - **Pulse:** how much stretches of the curtains brighten and fade over time. 0 keeps them steady.
 - **Seed:** another arrangement of the folds, the rays and the bright stretches.

On moonless nights the clouds are lit faintly by the night sky's own glow (the airglow, and the aurora's while it stands above the horizon), so they show as dark shapes against the stars.

Scene Lighting

- **Environment Lighting:** the scene's reflections and ambient light from the sky as it is now. Objects lit by baked probes or lightmaps keep their baked light.
- **Set Skybox:** make this sky the scene's skybox.
- **Sky Material:** the skybox material it sets (**Materials/M_DropInSky** by default).
- **Exposure:** the sky's brightness, without changing the sunlight.
- **Altitude (m):** the scene's height above the sea.
- **Export HDRI...** opens the HDRI export (below).

Exporting an HDRI

Tools > Havoila Studio > Drop-In Sky > Export HDRI... (or the button in Scene Lighting) saves the sky as it is now as a latitude-longitude EXR panorama, in linear HDR, from the main camera's place. It holds the atmosphere with the fog of the sky, the clouds, the stars and, with **Sun Disk**, the sun's and the moon's disks, the sun's as bright as the sunlight it gives, so the HDRI lights a scene as the sun did.

- **Resolution:** 1024 to 8192 wide (twice the height). Up to 4096 it is saved in 32-bit float, at 8192 in 16-bit float.
- **Cloud Detail x2** renders the clouds at twice the sky's resolution for the export, and **Cloud Samples** averages several renders of them (less noise).
- Saved inside the project, the EXR is imported as a cubemap (Latitude-Longitude layout), ready for a skybox or a reflection probe. It follows Unity's panorama layout, so Skybox/Panoramic reads it the right way round too.

Quality

- **Clouds:** the cloud map's resolution and march steps (Low, Medium, High), **Update Spread** (every frame, or the clouds rendered over 2, 4 or 8 frames for less time per frame), and **Draw Distance** (km, 160 by default).
- Two switches for more realistic clouds, off by default, each for more GPU time:
 - **Detailed Lighting:** clouds lit as solid volumes. They are bright white where the sun strikes them and grey on their far side and underneath, as real cumulus are, rather than glowing evenly from within. The billows shade each other, the crevices between them are darker (the cloud nearby hides the sky's light), and the edges are darker where the sun lights the side facing the camera. About a third more time for the clouds.
 - **Detailed Shapes:** firmer billows. The outlines are sharper and the cloud denser right inside them, finer billows show down to some 10 m, and wisps swirled by turbulence (curl noise) hang near the clouds' base. About 15 % more time for the clouds. Thin clouds (a low **Density**) let the eye see into them and look soft whatever their shape: use a Density of 1 or more for firm cumulus.
- **Fog:** Low, Medium or High (screen tiles of 16, 10 or 8 pixels and 48, 64 or 96 depth slices). The scene's shadows in the air are read up to 2, 4 or 8 times along each slice, one read every 2 m or so: smooth shafts from trees and rocks, with no blotches that crawl as the camera moves.
- **Soft Shadows** (Off, Low, Medium, High; off by default): the sun's and the moon's shadows as soft as real ones. The light comes from a disk about half a degree across, not from a point: a shadow is sharp where an object meets the ground, and its edge widens by about a centimetre for each metre between them. The long shadows of a low sun blur toward their tips, and those of thin things, such as a palm's trunk, lose their dark core after a few tens of metres. The width is physical by default (the real sizes of the sun and the moon, not the disks drawn in the sky); the quality sets how many samples are taken. **Softness** scales how fast the edges widen with distance (1: the real sun; 2 or 3 for the softer shadows of a hazy low sun; under 1 for sharper ones), and **Max Blur** caps how wide an edge may grow, from fully lit to fully dark (2 m by default; wider edges are found and filtered in coarser parts of URP's shadow map). **Fade** (off by default) makes shadows fade out with their distance from what casts them, between the two distances of **From – To** (60 to 200 m by default), as haze in the air between them fills them with light: the far ends of a low sun's long shadows disappear. Shadows of nearby things (a bench, a rock) keep their full strength. They need URP's **Screen Space Shadows** renderer feature on the renderer, which the shaders read the shadow from: the Inspector warns when it is missing and adds it (**Add Screen Space Shadows to the Renderer**). Set back to Off, the Inspector offers to remove it (**Remove Screen Space Shadows from the Renderer**): without Soft Shadows it only resolves URP's own shadows on screen, and costs a depth prepass. They apply to opaque objects drawn with URP's shaders (Lit, Simple Lit, Complex Lit, Terrain, SpeedTree, Shader Graph, deferred lighting); transparent objects keep URP's shadows.
- **Recording**, with its **Mode** (Automatic, Always, Never) and **Samples:** every frame complete in itself, for videos (section 11).

Clock

- **Source:** what the sky's clock follows (section 9).
 - **Game:** the game's time; in Edit mode, the editor's.
 - **Custom:** a time your script sets.
- While a Timeline drives the clock, the section shows it.

Events

UnityEvents for **Dawn**, **Sunrise**, **Sunset**, **Dusk**, each **Hour** and each **Day**, in Play mode. They fire as the time of day moves, by itself, from a script or from a Timeline, and also when the time jumps.

7. Presets and weather cycles

A **Sky Preset** is an asset holding a look.

- It holds any of these groups, each with its own switch:
 - the atmosphere;
 - the clouds' look and motion (not their quality, distances or seed);
 - the weather (rain or snow, lightning);
 - the fog (not its quality, nor its far shadows);
 - the sunlight's intensity and the exposure.
- Create one with **Save...** in the Inspector, or **Create > Havoila Studio > Drop-In Sky > Sky Preset**.
- The built-in presets set the atmosphere, the clouds and the weather. Stormy rains and strikes; the others are dry.

From a script:

```
sky.ApplyPreset(preset);           // at once
sky.BlendToPreset(preset, 30f);    // over 30 seconds (Play mode; follows the sky's clock)
```

Sky Weather Cycle

Add it with **Add Weather Cycle** in the Weather section. It moves the sky from preset to preset, by itself, in Play mode:

- **Sky**: the sky it drives. None: the one on its GameObject, else the scene's.
- **Weathers**: presets with a **Weight** (how often each comes). It starts with the built-in presets, from clear to stormy: fair, scattered and cloudy skies most often, clear and overcast less, stormy least.
- **Spell Length**: minutes of the sky's clock a weather lasts on average, from a third of it to one and two thirds.
- **Change Time**: minutes a change takes.
- **Seed**: another succession of weathers.

The weather comes from the sky's clock and the seed only, never from what came before. The same clock always gives the same weather: in a recording, with a Timeline, in a saved game, and for every player of a multiplayer game sharing the clock. The Inspector shows the weather now and when the next change begins. Between changes, you can still adjust the sky by hand.

Give the presets of a cycle the same groups. A change blends each group from one preset's values to the next's.

8. Weather on your materials

The sky publishes the weather for your shaders: rain and snow falling, how wet the ground is and how much snow lies. Rain and snow build up over **Wetting Time** and dry or thaw after.

Shader Graph

Add the **Drop-In Sky Weather** sub graph (Havoila Studio > Drop-In Sky) between your material's values and the Master Stack:

Input	
Base Color, Smoothness, Normal	Your surface's values (Normal in tangent space; leave it at (0, 0, 1) without a normal map).
Porosity	How much water it soaks up, darkening it. 0 for metal, glass and plastic, 1 for soil, cloth and rough wood.
Puddles	How much water its flat parts gather once soaked. 0 for roofs and walls, 1 for the ground.
Snow	How much of the lying snow settles on it. Faces turned up take it first.

Its outputs, **Base Color**, **Smoothness** and **Normal**, go to the Master Stack.

- A wet surface gets darker (by its porosity) and glossier.
- Once soaked, flat ground gathers puddles, with rings where the raindrops fall.
- Snow covers what faces up first, then steeper faces as it gets deeper, with a broken edge.
- Without a sky, the sub graph leaves the surface as it is.

Demo/Shaders/Weathered Lit is a ready example: URP Lit with a Base Map, a colour and the three weather sliders.

The **Drop-In Sky Cloud Shadow** sub graph gives the sunlight the clouds let through at a point (0 to 1). It is for shaders that do not read the sun light's cookie, such as grass, particles and custom lighting.

HLSL

Include `Shaders/Include/DropInSky.hlsl` after URP's `Core.hlsl`:

```
#include "Assets/HavoilaStudio/DropInSky/Shaders/Include/DropInSky.hlsl"

// albedo, smoothness, normalTS: in and out; porosity, puddles, snow: 0..1
DropInSkyWeatherSurface(albedo, smoothness, normalTS, positionWS, normalWS, porosity, puddles, snow);
// or with a Wet Gloss (0..1): how much of the rain's gloss the surface takes (puddles apart)
DropInSkyWeatherSurface(albedo, smoothness, normalTS, positionWS, normalWS, porosity, puddles, snow, wetGloss);
float cloudShadow = DropInSkyCloudShadow(positionWS);
```

Limits

Surfaces under a roof get wet too; the sky cannot tell what is sheltered. Pass 0 as Puddles and Snow there, or scale them by a mask of your own (vertex colours, a texture).

A surface whose wet film beads up or soaks in (leaves, grass, sand) looks like a mirror of the overcast sky with the full gloss of the rain. In HLSL, pass a Wet Gloss below 1 (0.15 to 0.4) to keep it darker and only a little glossier; puddles keep their own gloss. The sub graph has no Wet Gloss: give such surfaces a Puddles of 0 and a lower Porosity.

The weather costs little where there is none: the puddles' and the snow's noise is worked out only where the ground is soaked or snow lies.

9. Scripting

What the Inspector shows is in public fields of the **DropInSky** component (namespace `HavoilaStudio.DropIn.Sky`), but for the **Sky Material** (`sky.Material`) and the sun's **Shadow Intensity** (its light's own `Shadow Strength`). `DropInSky.Active` is the sky in use.

```
var sky = DropInSky.Active;
sky.time.timeOfDay = 18.5f;           // hours
sky.AdvanceTime(0.25);                // a quarter of an hour on (the date and the moon follow)
sky.weather.precipitation = 0.8f;
sky.clouds.coverage = 0.9f;
float sunHeight = sky.SunElevation;    // degrees
bool night = sky.IsNight;              // the sun more than 6° below the horizon
float phase = sky.MoonPhase;           // 0 new, 0.25 first quarter, 0.5 full
float eclipse = sky.SolarEclipse;      // how much of the sun the moon hides here (0..1)
float red = sky.LunarEclipse;          // how much of the moon is in the earth's umbra (0..1)
float boost = sky.ExposureBoost;       // the Post Exposure the night exposure adds now (f-stops)
```

`SkyAstronomy.Compute(sky.time)` gives where the sun, the moon and the stars are for any time, with the eclipses then, without changing the sky.

HDRI at runtime

```
var panorama = new RenderTexture(4096, 2048, 0, RenderTextureFormat.ARGBFloat);
sky.RenderPanorama(panorama, sunDisk: true, cloudSamples: 8);
```

It renders one frame first, from the main camera's place (or the camera you pass as `from`), for the sky to be up to date there, and costs a few times the sky's usual frame. `cloudDetail` renders the clouds at a multiple of the sky's resolution.

Preset previews

```
var preview = new RenderTexture(256, 144, 24);
sky.RenderPresetPreview(preset, preview, timeOfDay: 14f);
```

It renders how the sky looks with a preset (the sky alone, looking away from the sun), then puts the sky back as it was, nothing changed on screen: thumbnails for a preset picker in your game's menus.

The clock

The clouds drift and change with the sky's clock, `ClockTime` (seconds). In Play mode, the time of day and blends to presets move on with it too. With the **Game** clock and the time of day on, the clouds stand still while **Time Passes** is off; a **Custom** clock moves them as you set it.

- With the **Game** clock it follows `Time.time`.
- Setting `ClockTime` switches the clock to **Custom**, and the sky is then the same whenever this time is the same. Use it for multiplayer (a synchronised network time: every player sees the same clouds and the same weather) or for replays.
- `DriveClock(seconds)` drives the clock for one frame. Timeline's Time of Day track uses it. When the calls stop, the sky's own clock comes back.
- `CloudClockTime` gives the clouds a clock of their own (seconds; `null`, the default, follows `ClockTime`). For a time-lapse, run it as fast as the hours go by: the clouds race across the sky while the stars, the night sky and the weather keep the sky's clock. It is not saved with the scene.

Events

```
sky.Sunrise += () => lampPost.enabled = false;
sky.Dusk += () => lampPost.enabled = true;
sky.HourChanged += hour => Debug.Log($"{hour}:00");
```

`Dawn`, `Sunrise`, `Sunset`, `Dusk`, `HourChanged` and `DayChanged` fire in Play mode. Dawn and dusk are when the sun crosses 6° below the horizon (civil twilight); sunrise and sunset are when its disk crosses the horizon. The same events are UnityEvents in the Events section.

Save and load

```
string json = JsonUtility.ToJson(sky.GetState());
// ... later
sky.SetState(JsonUtility.FromJson<SkyState>(json));
```

The state holds:

- the time, the date (with its year) and the moon's phase;
- the look: atmosphere, clouds, fog, the weather (rain or snow, lightning), the sunlight's intensity and the exposure;
- how wet and white the ground is;
- the clock, if it is Custom. With the Game clock, the clouds go on from the game's time.

Weather and lightning

- `Wetness` and `SnowCover` (0 to 1) tell how wet the ground is and how much snow lies.
- `LightningFlash` (0 when nothing flashes, about 1 at a strike's brightest) and `LightningDirection` let you flash your own lights with the lightning.

10. Timeline

With the Timeline package installed, two tracks bind to the sky (**Havoila Studio > Drop-In Sky**):

- **Time of Day**: each clip goes from a start time to an end time of day, with a speed for the clouds (for time-lapses). The sky's clock follows the track, so scrubbing back shows the same clouds. The track's **Cloud Clock Start** (seconds) sets the clouds' clock at the timeline's start: another value gives other clouds.
- **Sky Preset**: each clip applies a preset. Overlapping clips blend.

The clips are named after what they do, for example "06:00 → 18:00 · clouds ×60" or the preset's name.

11. Recording videos

The clouds are normally rendered over several frames and blended with earlier ones: smooth, and cheap. A recording needs each frame complete in itself instead, with no trails even in a fast time-lapse.

- **Recording** (Quality section) does this:
 - the clouds are rendered whole several times per frame (**Samples**) and averaged;
 - the ambient light is that of the frame.
- **Automatic** switches it on in Play mode while Unity Recorder captures (anything that sets `Time.captureDeltaTime`). **Always** keeps it on.
- It costs Samples times the clouds' time per frame.

12. For other assets: the published values

The sky publishes global shader properties for other assets, so they can use it without referencing its code. Drop-In Water reads them, and so do the park's shaders in the demo (Demo/Park/Shader: ParkSkyFog.hlsl for the fog, Park Water for the environment). Each is valid while its `...Params.x` is 1 (the mist's, while `_HavoilaSkyFogParams.x` is 1 and its own `x` above 0).

Property	
<code>_HavoilaSkyEnvironment</code> , <code>_HavoilaSkyEnvironmentParams</code>	The sky's environment cube (clouds and the fog of the sky included, no sun or moon disk), mips filtered for roughness like URP's probes, linear HDR. Published while Environment Lighting is on.
<code>_HavoilaSkyFog</code> , <code>_HavoilaSkyFogMatrix</code> , <code>_HavoilaSkyFogParams</code>	The fog volume (rgb in-scattered light, a transmittance), for transparent shaders to fog themselves. Valid while the camera draws its transparent objects. The matrix takes a world position to the camera's clip space (OpenGL-like, not jittered); the volume is read at $uvw = (\text{clip.xy} / \text{clip.w} * 0.5 + 0.5, \ln(\text{clip.w} / y) * z)$. Params <code>y</code> , <code>z</code> : the slices' depths (the first slice's, $1 / \ln(\text{far} / \text{first})$); <code>w</code> : their number.
<code>_HavoilaSkyFogMist</code> , <code>_HavoilaSkyFogMistParams</code>	The mist, apart from the volume: Params <code>x</code> is its extinction at the world's $y = 0$ (1/km; 0 = no mist), <code>y</code> is 1 / its height (1/m), <code>z</code> is the floor it thickens down to (m). How much of it lies in front of a point is worked out along the point's own ray (a mist a few metres thick, seen from high up, is thinner than the volume's slices); the texture, laid out as the volume, gives its light (rgb). The comments of <code>AtmosphericFog.cs</code> and <code>Shaders/Include/Mist.hlsl</code> give the formulas.
<code>_HavoilaSkyFarShadow</code> , <code>_HavoilaSkyFarShadowMatrix</code> , <code>_HavoilaSkyFarShadowParams</code>	The far shadows of the main light (the sun, or the moon while it leads), while Fog > From Far Away > On the Scene is on (and the fog): a shadow map seen from the light, compared as URP's (<code>SAMPLE_TEXTURE2D_SHADOW</code>). The matrix takes a world position to its texture space (<code>u</code> , <code>v</code> , <code>depth</code>). Params <code>y</code> : the bias to add to the depth (towards the light); <code>z</code> : how far above a point to read it (m: its texels are metres to tens of metres tall). For transparent shaders beyond URP's shadow distance: Drop-In Water's sun glint is gone there behind a ridge at sunrise.
<code>_HavoilaSkyWeather</code>	<code>x</code> : rain falling, <code>y</code> : snow falling, <code>z</code> : how wet the ground is, <code>w</code> : how much snow lies (0 to 1). <code>_HavoilaSkyWeatherParams.y</code> : how far raindrops are seen landing (m, the Splash Distance); <code>.z</code> : how many splash (the Splash Amount).
<code>_HavoilaSkyCloudShadows</code> , <code>...U</code> , <code>...V</code>	The clouds' shadow map (<code>r</code> : the sunlight let through, 0 to 1) and its mapping: $uv = (\text{dot}(\text{float4}(\text{positionWS}, 1), U), \text{dot}(\text{float4}(\text{positionWS}, 1), V))$. <code>_HavoilaSkyCloudShadowsParams.y</code> : the strength the map was made with (its darkest is $1 - y$), while it is also the sun light's cookie (0 when it is not). A shader can take it back out of the cookie where the sun's disk itself shows: Drop-In Water does, so the sun's glint is gone under a thick cloud.

The sky reads one property that water can publish (Drop-In Water does), so rain and snow stop at the water's surface, and the lens flare is not drawn for a camera under it:

Property	
<code>_HavoilaWaterHeights</code> , <code>_HavoilaWaterHeightsParams</code>	A map of the water surface's height around the camera, seen from above (<code>r</code> : height in m, below -10000 where there is no water). Params <code>x</code> , <code>y</code> : world <code>x</code> , <code>z</code> of the map's corner; <code>z</code> : 1 / its span (m); <code>w</code> : 1 while valid. The lens flare reads it on the CPU, when it is a readable <code>Texture2D</code> .

13. Performance

Measured at 1920 × 1080 on an NVIDIA GeForce RTX 5080, in the Unity Editor (Play mode), in the demo (DropInSky_Wild) at its Valley View viewpoint, a clear afternoon: GPU time of each part, from the Profiler's samplers.

	Low	Medium	High
Clouds (Update Spread: every frame)	0.44 ms	0.83 ms	1.82 ms
Clouds, under a storm's thick clouds and rain curtains	0.75 ms	1.37 ms	2.77 ms
Fog (with shafts from the scene and from far away)	0.30 ms	0.52 ms	1.15 ms
Recording, 4 samples (instead of the clouds' usual time)		3.5 ms	7.7 ms

The rest costs the same at every quality:

Environment lighting: the cube, its filtering and the ambient light, and the fog of the sky drawn into it	about 0.1 ms
The clouds' shadows (the sun light's cookie)	0.02 ms
The atmosphere's tables (only when the sun, the moon or the camera's height has moved)	0.02 ms
The skybox	0.02 ms
Far shadows On the Scene , over URP's 8192 × 8192 shadow map	0.31 ms

So the whole sky takes some 1.2 ms (Low), 1.8 ms (Medium) or 3.4 ms (High) of a frame here, the far shadows on the scene included. At night the clouds and the fog cost less (0.62 and 0.27 ms at Medium): the aurora adds about 0.2 ms when it is on, shooting stars, satellites and airplanes about 0.1 ms all together.

On the CPU, the sky's own work is about 0.2 ms of the main thread per frame, with hardly any memory allocated.

- **On the Scene** runs over URP's main light shadow map once per camera and frame: its cost follows the map's size (a quarter of the above at 4096 × 4096, a sixteenth at 2048 × 2048). Most of the map is decided from two of its four taps.
- The far shadows' maps are rendered again when the sun has turned by a quarter of their texels' height as seen 10 km away (about 0.005° at sunrise, so every frame or two of a day that passes in minutes, and less often with the sun high), when the camera has moved a tenth of the Far Distance, and every frame while recording. Each time, they draw the Far Casters' shadow casters once. In the demo's park (24 terrains and some 1,500 other casters), that takes about 1.3 ms, mostly the other casters' draw calls; the terrains alone take about 0.4 ms. Restrict **Far Casters** to the landscape in heavy scenes. In the Profiler, look for **Drop-In Sky: far shadows**.
- Every part has its sampler in the Profiler: **Drop-In Sky: cloud map, cloud shadows, atmosphere tables, environment cube, environment fog, fog, far shadows on the scene, aurora, soft shadows**.
- Drops, splashes and snowflakes cost too little to measure apart from the rest. Lightning costs a few operations in the sky's shader, only while a flash is on screen.
- Materials using the weather (section 8) work out the puddles' and the snow's noise only where the ground is soaked or snow lies: on a dry day they cost a few operations.
- The night sky's maps take about 32 MB of video memory on PC (the stars 8192 × 4096 in BC1, the Milky Way 4096 × 2048 in BC7, with their mips). Lower their **Max Size** in the import settings for smaller targets: the stars get softer, the Milky Way hardly changes.
- Moving objects (colliders on a Rigidbody) near the camera cost a few rays each per frame, on the main thread (one per half-metre cell they cover, 4096 at most), plus their splashes (a few dozen on a car in a downpour).
- Rain casts rays down around the camera when it has moved 3 m, or every 2 seconds: some 12,500 at the default Splash Distance, 25,600 at most. They run on the job system's worker threads, compiled by Burst, in under a millisecond; the main thread only uploads the result (under 0.1 ms).

- **Soft Shadows**, when on: some 0.5 to 1.5 ms at 2560 × 1440 (Low to High; more where many casters stand close together), on top of URP's Screen Space Shadows feature, which makes URP draw a depth prepass of the opaque objects (unless something else, such as SSAO, already does).

Profile a build on your target hardware for exact numbers.

To make it cheaper:

- **Update Spread** renders the clouds over 2, 4 or 8 frames. It costs a little latency in their changes.
- **Detailed Lighting** and **Detailed Shapes** off (Quality section): each costs GPU time for finer clouds.
- A lower **Quality** for the clouds and the fog: the fog's High reads the scene's shadows eight times per slice, for the smoothest shafts.
- A shorter **Draw Distance** for the clouds.
- **Environment Lighting** off, if the scene's lighting is baked.
- **On the Scene** off, or a smaller URP shadow map, if no mountain stands between your scene and a low sun.
- The fog off, for scenes without a far view.
- **Soft Shadows** at Low, or off.

14. Limitations

- **Cameras stay below the clouds.** The clouds are rendered as a panorama seen from the ground. Flying through or above them is not supported.
- **Compute shaders are needed.** Mobile and WebGL are not supported in this version.
- **XR.** The fog's volume is made once for both eyes, seen from the point between them: fog very close to the camera (under a metre) can differ slightly between the eyes. The lens flare is not drawn in the headset (a flare drawn on each eye's image does not sit where the eyes see the sun).
- **One sky per scene.** The last sky enabled is used.
- **Rain and roofs.** Rain and snow stop under colliders, not under meshes without them. Materials under a roof still get wet (section 8).
- **Far shadows reach the scene within URP's shadow distance only.** Beyond it, distant mountains shade the fog and Drop-In Water (it reads `_HavoilaSkyFarShadow`), not the ground: URP's shaders read no shadows there.
- **Thunder.** No thunder sounds are included; give the sky your own clips.
- **Rainbows are in the sky only.** They show in the rain around the camera and in the distant curtains, over the sky, not in front of the scene's own hills.
- **Soft Shadows** soften what URP's shadow map holds: opaque objects drawn with URP's shaders, within URP's shadow distance (transparent objects, such as water and particles, keep URP's own shadows), on desktop and consoles but not OpenGL, and not in XR (the headset keeps URP's shadows). URP's **Normal Bias** thins the casters in its shadow map, more in its far cascades: very thin objects may cast no shadow there; lower the Normal Bias if they disappear. Where casters stand at very different heights around a point, their mean distance sets one edge width for all of them.

15. Troubleshooting

Problem	What to do
No cloud shadows on the scene	Turn on Light Cookies in the URP asset. Check the clouds' Shadows . A cookie of your own on the sun light is left alone, and then the clouds' shadows are not set.
The scene is fogged twice	Turn off Unity's fog in Lighting > Environment . The Inspector offers to do it.
Nothing falls	Rain and snow fall beneath the volumetric clouds only. Check that the clouds are on, and that Precipitation is above 0.
Rain falls through a roof	Give the roof a collider, on a layer of Stopped By .
Clouds leave trails in a video	Set Recording to Automatic (with Unity Recorder) or Always.
Changes in Play mode are lost	Unity restores the scene after Play mode. Save a preset (Save...) to keep a look.
The night exposure does nothing	Turn on the camera's Post Processing . A camera whose Volume Update Mode is Via Scripting only takes volume changes when your script updates it.
The sky is black in a new scene	Check that the sky is enabled and Set Skybox is on, and that the camera's Background Type is Skybox.
Soft Shadows change nothing	Add URP's Screen Space Shadows renderer feature to the renderer (the Inspector offers to), and turn on Main Light shadows in the URP asset. The main light must be the sky's sun or moon. They are not drawn in XR or on OpenGL.
Some objects lose their shadows with URP's Screen Space Shadows on	Their shaders do not read screen-space shadows: add <code>_MAIN_LIGHT_SHADOWS_SCREEN</code> to their main light shadow <code>multi_compile</code> (as URP's own shaders have it).

16. Support

- Email: support@havoila.com
- Website: <https://havoila.com>

Please include your Unity and URP versions, your GPU, and a screenshot or short video of the issue. **Tools > Havoila Studio > Drop-In Sky > Contact Support** opens an email with the version filled in, and **Documentation** opens this guide.

17. Version history

1.0.0

- First release.