



GLOBAL SNOW

GLOBAL SNOW Reference Guide



Contents

Introduction	4
How it works	4
Demo Scene	5
Quick Start.....	5
Global Snow Parameters.....	6
Scene Setup (Forward Rendering Path mode).....	6
Scene Setup (Deferred Rendering Path mode).....	7
<i>Unity standard tree and billboards (only in deferred mode)</i>	7
Quality	8
Coverage.....	9
Appearance.....	10
Features.....	11
Snow Marks on Terrain.....	12
Moving Objects (characters, vehicles, etc.).....	13
Snow Volumes	14
Mask Editor.....	15
World Manager API integration.....	16
Complete Coverage	16
VertExmotion integration.....	17
Vehicle Physics Pro integration	17
Transition/Lerping between snow profiles.....	18
Frequent Asked Questions (FAQ)	19
Some objects that get clipped when Global Snow is enabled (forward rendering)	19
How can I exclude some objects from being covered by snow?	20
How can I exclude terrain from snow in deferred rendering path?	20
There're no snow marks on some objects, like floating platforms. They only work on the terrain.	20
How can I manually add terrain marks over any place?.....	20
How can I manually add footprints over any place?	21
Can I customize footprint shape?	21
Can I change the size of the terrain marks?	21
I see terrain marks over the distance, but nothing should be causing them.	21
Is there any footfall option?	21
How to enable snow reflections on water? (forward rendering).....	22
Does it work with VR/Single Pass Stereo Rendering?.....	22
How can I manually add static snow over moving objects?	22
Unity's standard tree billboards are not covered by snow. Is it possible to use them?	22
<i>Forward rendering mode</i>	22
<i>Deferred rendering mode</i>	22
Compilation takes a long time / GlobalSnow adds many shader variants, can I reduce them?.....	23
Can I add "holes" into the terrain?.....	23
Grass appears transparent?.....	23
My FPS weapon is receiving snow. How can I exclude it?	24
Additional Support	24

Introduction

Thanks for purchasing!

Global Snow is a drag & drop solution that converts your scene into a winter landscape with a couple of clicks. Just add a single script to your camera and customize the results.

Global Snow features:

- Realistic snow with relief shading, occlusion and glitter.
- Automatic snow coverage with exclusion options (by layers, per gameobject, mask texture).
- Footprints and pack of footfalls sounds.
- Terrain marks and terrain collision detector.
- Snowfall.
- Camera frost effect.

How it works

Global Snow supports forward and deferred rendering.

- **Deferred rendering (recommended)**. In this mode Global Snow will modify the GBuffers in realtime producing a great looking effect at the best performance.
- **Forward rendering**. In this mode Global Snow uses a full-screen image effect to blend a precomputed snow layer over the landscape. In this mode, Global Snow uses two techniques: a) renders the scene using custom replacement shaders with snow, and b) renders snow in screen space using a fast shader aimed for far distances. The result is a great looking snow but the performance is not as good as in deferred rendering.
- **Decal approach**: all Global Snow customers have access to a special version of Global Snow that uses decal approach to render snow over existing objects. This version can be downloaded from our support forum on kronnect.com and is appropriate for low-end devices which can't afford full-screen blit operations.

TIP !

Deferred and forward rendering path support are provided by custom shaders located in `Workflow_Deferred` and `Workflow_Forward` inside `GlobalSnow/Resources` folder. If you're only using one of the two rendering paths (eg. deferred), then you can remove the other folder so save some resources and build time in your project.

Demo Scene

Just load the demo scene included and click “Play”. You will be able to move around using WASD and mouse. Press spacebar to jump, T to toggle on/off the snow effect or L to change daylight.

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

Quick Start

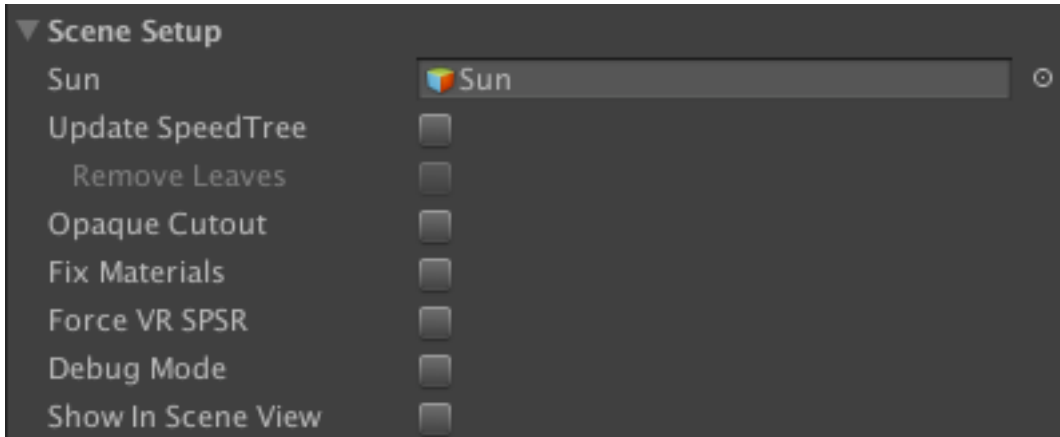
1. Add the GlobalSnow script to your main camera in your scene.
2. Check the Scene Setup parameters if needed and choose a Quality preset.
3. Optionally enable footprints, terrain marks, snowfall or camera frost features.

You can of course customize any of its parameters to match your game mood and requirements.

Please read this document to learn more about all supported features.

Global Snow Parameters

Scene Setup (Forward Rendering Path mode)



Sun: make sure you have a proper directional light assigned as the Sun. It's necessary to compute relief and occlusion effects. Global Snow will pick a directional light automatically for you but ensure this is the good one.

Global Snow uses custom replacement shaders to render the snow over the scene geometry. By default all opaque objects will be covered by snow (see FAQ section for exclusion options). The following setup parameters will help some common shaders work with Global Snow:

Update Speed Tree: enable this option to support SpeedTree and SpeedTree billboard objects.

Remove Leaves: enables a SpeedTree variant shader that removes tree leaves. For best results, it's recommended that you use custom Winter tree models. However this option allows you to quickly prune all tree leaves if you want.

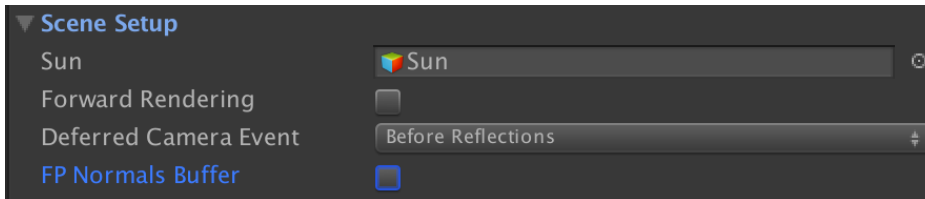
Billboard Coverage: specifies the amount of snow over tree billboards and grass objects.

Opaque Cutout: by default, opaque objects will be covered by snow. But some shaders expose holes or transparent parts of a texture and still they use the Opaque rendertype. Enabling this option will force GlobalSnow to discard those transparent pixels preventing them from being covered by snow (in this case a transparent pixel needs to have RGBA=0,0,0,0 value). Some trees may require you to enable this option so their leaves render appropriately with Global Snow.

Fix Materials: this option will force all materials to expose a RenderType tag. It's a good recommendation for shader developers to specify a RenderType per shader. However not all shaders expose this tag. Because of this, it's possible that some objects get clipped by the snow. If this happens to your scene, enable this option to force all non-transparent materials to expose an Opaque render type.

Force VR SPSR: this option is only available in Unity 5.5 or higher. Will enable compatibility with OpenVR. Don't use if your VR SDK is Oculus.

Scene Setup (Deferred Rendering Path mode)



Sun: make sure you have a proper directional light assigned as the Sun. It's necessary to compute relief and occlusion effects. Global Snow will pick a directional light automatically for you but ensure this is the good one.

Forward Rendering: enable this option to override deferred rendering and use the forward rendering mode of Global Snow.

Deferred Camera Event: this option specifies when the snow shader passes will be executed. By default this setting is BeforeReflections and that should be the "good setting". However, if you disable deferred reflections in Graphics Settings, you will need to change this to "Before Lighting".

FP Normals Buffer: enable this option to use floating-point backed normal buffer (16 bit precision) instead of normal 8-bit precision. Use it only if you see snow artifacts in the distance.

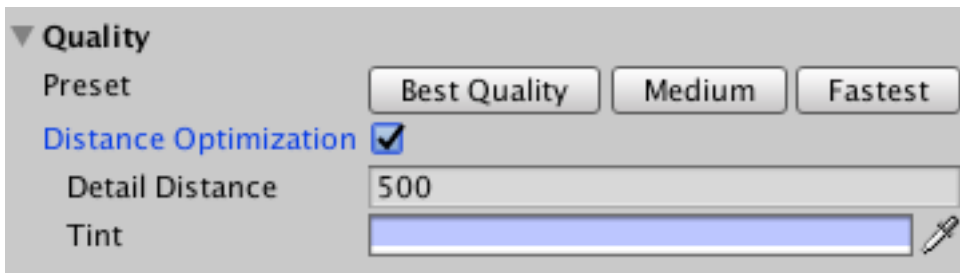
Allow Batched Meshes: this option is only used when excluding objects from snow coverage (ie. using the Layer Mask field). If the excluded objects are part of static batching, Global Snow detects this situation and uses the mesh from the collider itself. However, since the mesh collider usually is a low resolution version of the original mesh, this can produce artifacts. In this case, either disable static batching or enable this option to force Global Snow to exclude the entire batched mesh.

Unity standard tree and billboards (only in deferred mode)

- Inside the folder Resources/Workflow_Deferred/Shaders/Grass & Billboards you can find a package containing shader replacements for the default grass and tree billboards in case you need to support them.
- Also you can use the alternate "TreeOpaque Soft Occlusion Bark" shader located inside the same folder (subfolder Tree).
- SpeedTrees are supported in this mode so you don't need to change them.

Quality

This section offers main quality/performance options:



Best Quality: uses a high resolution coverage texture, relief shading, occlusion and glitter effects.

Medium: uses a smaller resolution coverage texture, relief shading, occlusion and glitter effects. It also enables Distance Optimization option with a default value of 500m.

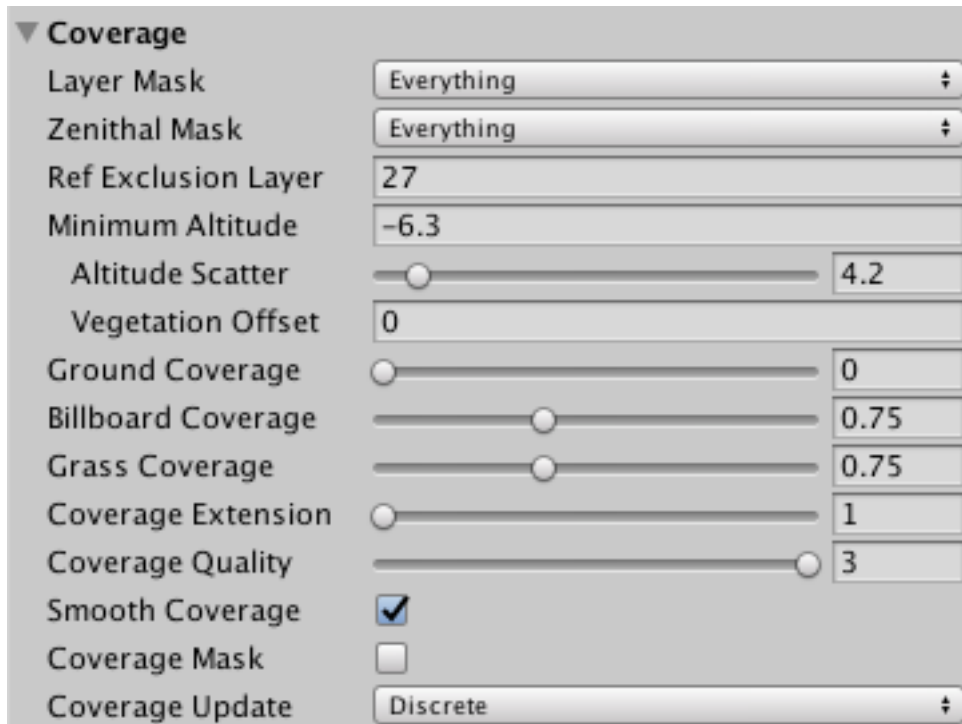
Fastest: uses a smaller coverage texture and flat shading (which still looks good!). It also enabled Distance Optimization beyond 100m.

Note that these presets just apply some factory values to other settings across the inspector. They are provided for convenience, as shortcuts to other options.

Distance Optimization: when enabled, a secondary simpler and faster algorithm is used to render the snow beyond some given distance from the camera. A “Tint” option allows you to specify the color for the far distance snow. Since this algorithm is not physically-based, the snow rendered this way is not influenced by ambient light nor global illumination. It can’t also receive shadows. Additional options are self-describing (pass mouse cursor over the option label to get a short description).

Coverage

This section allows you to customize where and how the snow will be rendered:



Layer Mask: determines which objects can be “painted” with snow. By default, all transparent objects are excluded since snow is rendered before transparent queue.

Zenithal Mask: determines which objects can occlude other beneath them. By default, any object will cancel any snow on other objects beneath them. **Make sure to exclude any particle system here both to improve performance and to reduce coverage issues.**

Ref Exclusion Layer: this is the number of the layer, used during the rendering of snow, for objects that have been marked specifically to be excluded by snow. Use a layer number that you don’t use in your project.

Minimum Altitude and Altitude Scatter: specify the starting altitude for the snow. Use Altitude Scatter to make the snow appearance more realistic. Note that above the minimum altitude, the snow will always appear. There’s no “maximum altitude”.

Vegetation Offset: specify a value to be added to the minimum altitude for trees and grass. With this option you can enforce vegetation to be covered by snow on greater altitude than terrain starts showing snow.

Coverage Extension: a value of 1 means an area of 256x256 meters around the camera position. A value of 2 means an area of 512x512 meters. This parameter along with Coverage Quality determines the final resolution of the snow coverage.

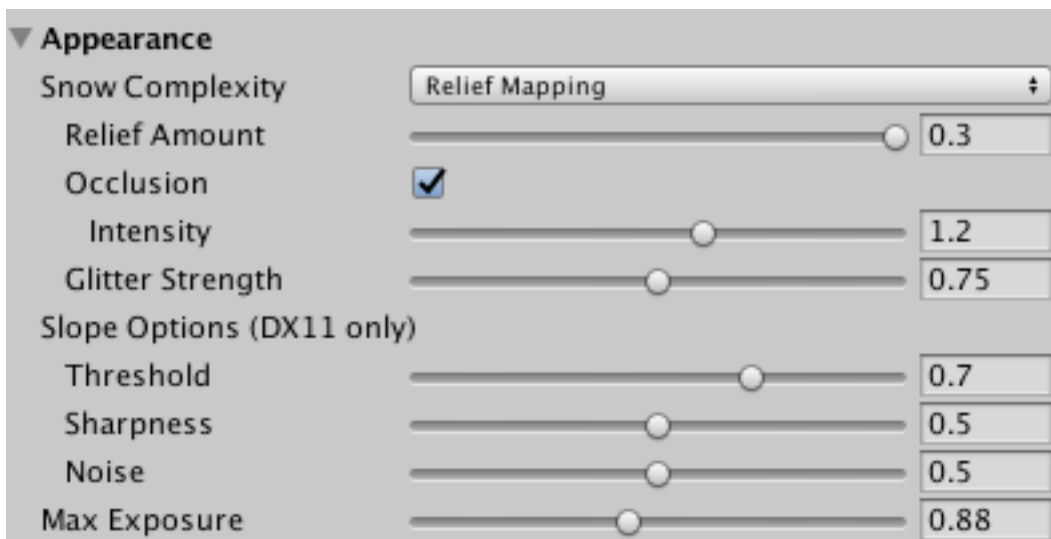
Coverage Quality: determines the size of the internal texture used for snow coverage computation. A value of 1 uses a 512x512 texture, a value of 2 uses 1024x1024 texture and a value of 3 uses a 2048x2048 texture.

Smooth Coverage: enabling this option will smooth the coverage producing more pleasant results.

Coverage Mask: you can use a texture as a mask to specify the areas of the world that will be covered by snow. This feature assumes that terrain is centered around 0,0 position. The alpha channel of the texture will be used to determine where can be snow (1=snow, 0=no snow). The world size parameter specifies the mapping size of the texture over the world. For instance, a texture of 2048x2048 pixels with a World Size of 2048x2048 will have a mapping of 1 meter per pixel (the Y-coordinate is ignored).

Appearance

This section allows you to customize the appearance of the snow itself:



Snow Complexity: Flat shading is the fastest and produces a flat and easy looking snow. Relief mapping uses a limited raytracing algorithm to simulate 3D waves over the snow.

Occlusion: adds extra local shadows to the snow. Only available when relief shading is used. This option is useful when the Sun is high in the sky because from the zenith the snow will look very bright and white without occlusion.

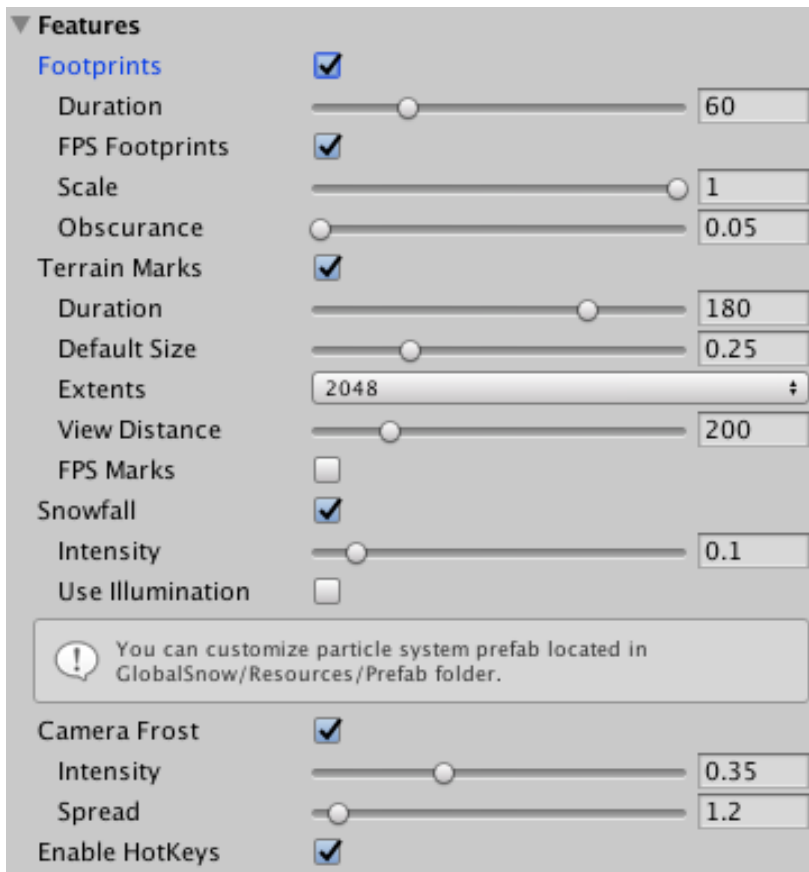
Glitter Strength: controls the amount of glittering over the snow. Glitter tends to glow stronger when facing Sun.

Slope Options: customize the borders of the snow-covered areas, useful for adjusting the look of snow covered cliffs and mountains. Note that this feature only works on DX11 and OpenGLCore compatible graphics.

Max Exposure: let you limit the amount of light that the snow will reflect avoiding over exposure when Sun is very high, or its light intensity is greater than 1. Note that this is an artificial change and can produce odd results if abused when compared with the rest of non-covered geometry.

Features

This section allows you to unlock special Global Snow features:



Footprints: footprints are human-shaped footprints decals that are printed over the snow. This feature is interesting to help the player walk over large distance and be able to return over the same path. Although Global Snow only supports adding footprints at the camera position, it could be modified to add custom footprints over other areas.

- Enable **FPS Footprints** to add automatically footprints after your FPS character moves.
- Use **Duration** parameter to customize the fade out duration of the footprints.

Terrain Marks: this feature allows you to cause small or medium impacts over the snow.

- Choose the **default size** in meters for the radius of the mark.
- The maximum **Extents** is the size of the internal control texture that's used to render marks over the terrain – increase to have a greater area of non-repetitive marks.
- **View distance** allows you to control the rendering distance of the terrain marks.
- Enable **FPS Marks** to leave automatic footprint tracks using terrain marks.

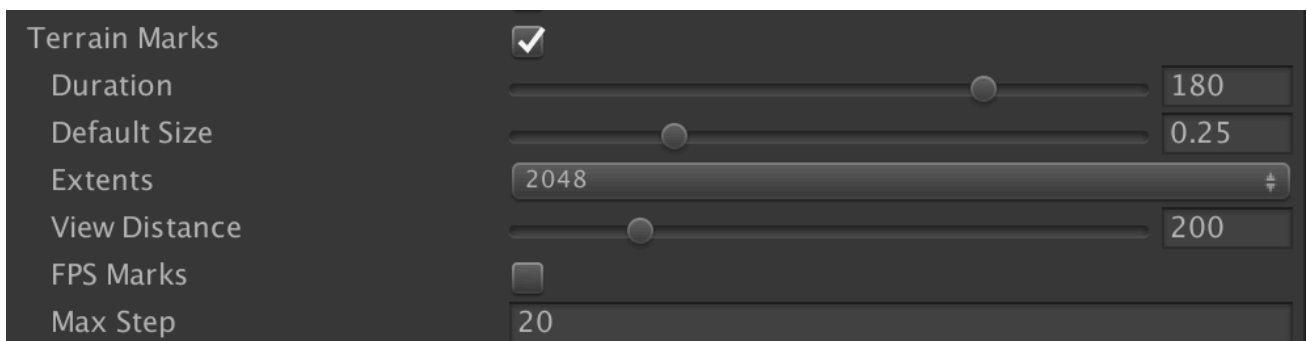
Snowfall: enable this parameter to instantiate the SnowParticleSystem prefab located in Resources/Prefabs. This particles system will emit snowflakes at **Intensity** rate. You can customize the look and behaviour of the snow particle system modifying the prefab itself. Enable “**Use Illumination**” to change the particle material to a Standard Shader material which will be affected by Global Illumination.

Camera Frost: adds a screen-space frosting effect which can be customize by intensity and screen spread. Useful to add a temperature sense to the player.

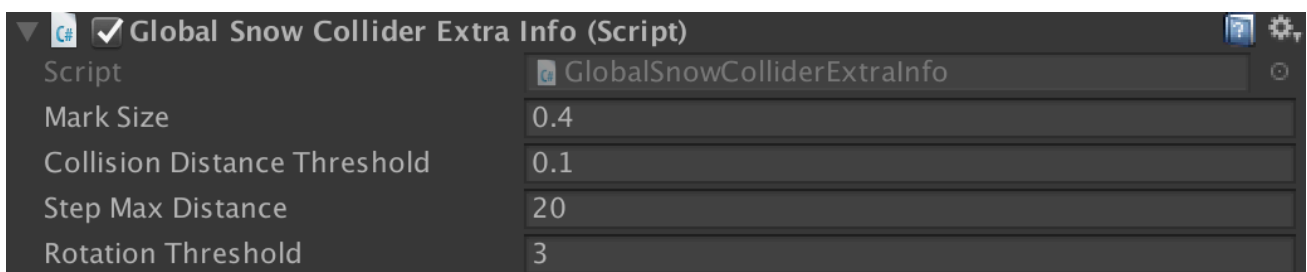
Snow Marks on Terrain

Global Snow detects collisions with the terrain and takes care of snow marks. It does by attaching a **GlobalSnowCollisionDetector** script to your terrain gameobjects. If you have custom terrain objects or create them at runtime or you want snow marks on other kind of surfaces, just add the collision detector script to those objects using `AddComponent<GlobalSnowCollisionDetector>()`.

Once a collision occurs, Global Snow will render the mark on a special texture used during the screen space effect. By default, Global Snow will use the settings specified in the Global Snow inspector:



You can also customize the marks effect per object. Add a **GlobalSnowColliderExtraInfo** script to your gameobject:



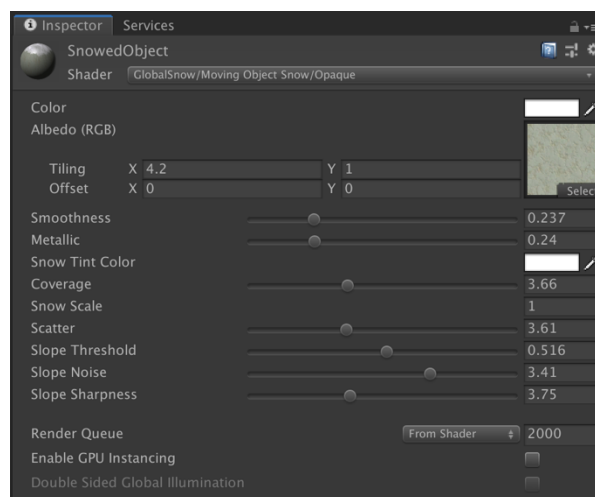
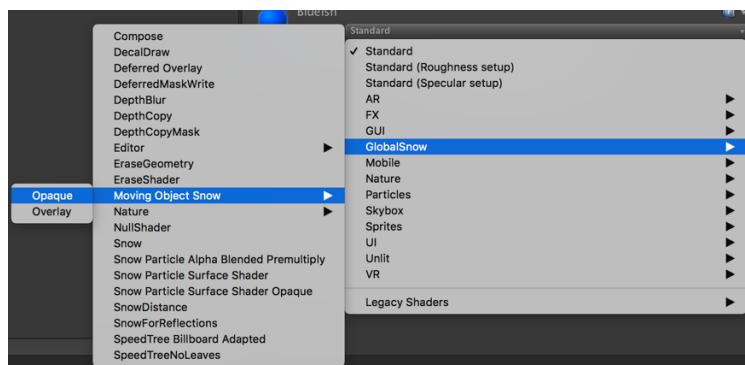
The settings of the Collider extra info script override the default settings from Global Snow inspector.

Moving Objects (characters, vehicles, etc.)

Global Snow works with g-buffers and by default it doesn't differentiate moving objects from static objects. Snow on moving objects can look strange on certain moving objects since snow will show alterations as the object moves. To avoid this issue, you have two options:

Step 1) Exclude snow on the moving object. You can do so by assigning the object or prefab to an exclusion layer in Global Snow coverage section. This way the object won't receive the normal snow that Global Snow renders over the scene. Another way to exclude gameobjects from Global Snow is adding the Global Snow Ignore Coverage script to the gameobject itself (note that when using the exclusion options, the result might not be accurate in SceneView due to how CommandBuffers work – always verify the result in the GameView).

Step 2) Use a snow custom shader. If you want the moving object to show some snow, use the GlobalSnow/Moving Object Snow/Opaque shader included in the asset. This shader is a Standard shader-based shader that integrates snow features on it. The benefit of this shader is that the rendered snow on top of the moving gameobject doesn't change its shape when the object or character moves, which produces a more realistic effect.



This shader includes a few appearance options as you can check in the screenshot above, including Snow Tint Color, Coverage (amount of snow), Snow Scale, Scatter, etc. Experiment with the different settings until you get the desired result.

Step 3) If the moving object is a vehicle and you want Global Snow to leave snow tracks after its wheels, add the Global Snow Collider Extra Info to the root of the vehicle gameobject. This script takes care of checking the contact points of wheels and ground and report those hits to Global Snow.

Snow Volumes

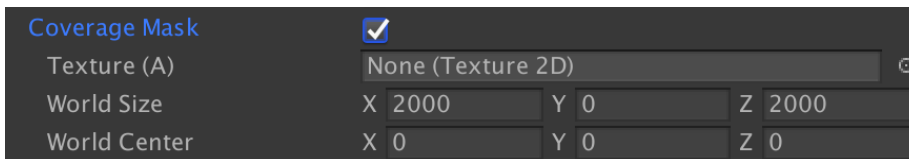
Global Snow volumes are special areas where snow can be completely disabled. When the player enters a snow volume, the snow disappears and when he exists the area, the snow script is re-enabled automatically.

To add a Global Snow Volume to your scene, just drag & drop the GlobalSnow Volume prefab from the GlobalSnow/Resources/Prefabs folder.

Position the volume in your scene as you do with a normal object. Change the collider scale to match the desired area size.

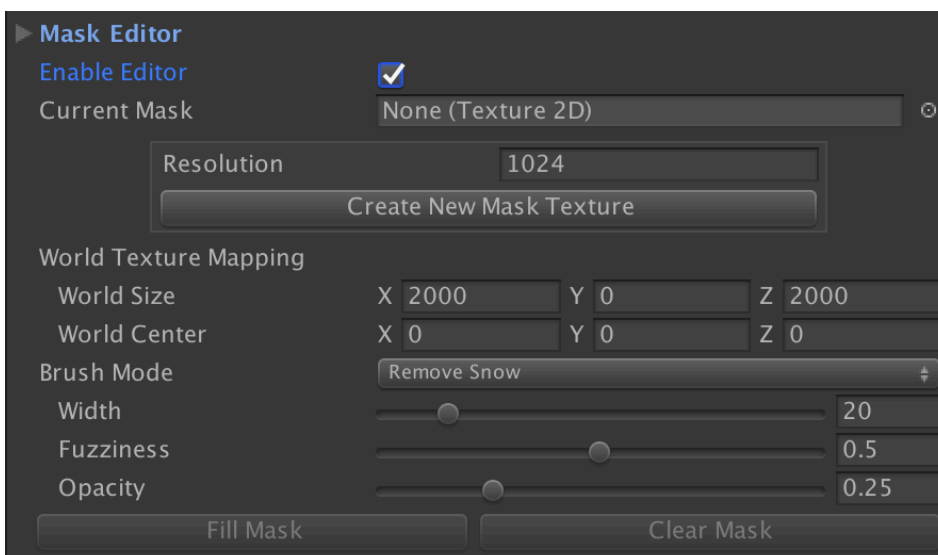
Mask Editor

Another way to remove/add snow on custom areas is by using a coverage mask:



A coverage mask is a special texture whose alpha component determines the opacity of the snow at a given location. The texture is overlaid over the world space at “World Center” with extents of “World Size” (so a size of 2000 means a coverage of -2000 to 2000).

You can provide your own texture mask or use the integrated SceneView editor. This mask editor is enabled using the provided section in the inspector:



The Mask Editor allows you to create a new mask texture from scratch. Select a proper resolution according to the detail level you need. For gross results, a texture of 1024x1024 may be enough. If you need finer coverage details use a bigger texture (ie. 2048, 4096 or even 8192 pixels).

The brush mode, width, fuzziness and opacity controls the behaviour of the paint tool. When the Mask Editor is enabled you can remove or add snow into the SceneView directly!

Click “Fill Mask” to reset the mask texture to fully snowed. “Clear Mask” will remove all snow from the covered area.

Please note that if you use Unity standard tree or grass you also need to edit the GlobalSnowForwardOptions.cginc or GlobalSnowDeferredOptions.cginc file (depending on the active camera rendering path) and enable the #define GLOBALSNOW_MASK line.

World Manager API integration

World Manager API is a component developed by Adam Goodrich that enables syncing some atmospheric parameters across different assets.

You can find more information on GitHub: <https://github.com/adamgoodrich/WorldManager>

To enable World Manager API integration, make sure the corresponding checkbox is checked at the bottom of Global Snow inspector:



(Of course WMAPI asset needs to be present in the project for the integration to work).

The current integration allows snow altitude to change depending on the minimum snow altitude and season parameters of WMAPI. Also the snow fall intensity is synced with WMAPI properties.

Complete Coverage

It's possible to disable the zenithal coverage computation if your setup requires/allows that. Normally, objects in the scene will occlude snow beneath them (no snow below). This kind of occlusion is implemented by using a second zenithal camera that takes discrete depth shots from above. That information is used in the main snow shader to ensure no snow is rendered below any other object.

Currently you can have the "zenithal depth shots" to occur every frame, at discrete intervals or in manual mode. Discrete intervals means that a new zenithal depth capture will be taken every 50 meters as the player moves across the scene. Manual means that the shot is taken only with a call to `UpdateSnowCoverage()` method.

If you don't need the zenithal depth pass, then you can set Coverage Update to Manual and edit `GlobalSnowDeferredOptions.cginc` or `GlobalSnowForwardOptions.cginc` file and comment out the line below:

```
// Comment out to disable zenithal depth
#define USE_ZENITHAL_DEPTH
```


VertExmotion integration

If you exclude some characters from snow coverage that use VertExmotion asset, you need to enable the integration to avoid artifacts.

- In deferred rendering path locate and edit the file **GlobalSnowDeferredOptions.cginc** inside Resources/Workflow_Deferred folder.
- In forward rendering path the file is named **GlobalSnowForwardOptions.cginc** inside Resources/Workflow_Forward folder.

Uncomment the following two lines and make sure the path in the include line is correct:

```
#define ENABLE_VERTEXMOTION_INTEGRATION
#include "Assets/VertExmotion/Shaders/VertExmotion.cginc"
```

Vehicle Physics Pro integration

Global Snow works out of the box with assets like Vehicle Physics Pro which implements wheel-based collision on the ground to leave snow tracks as covered in the “Moving Objects” section.

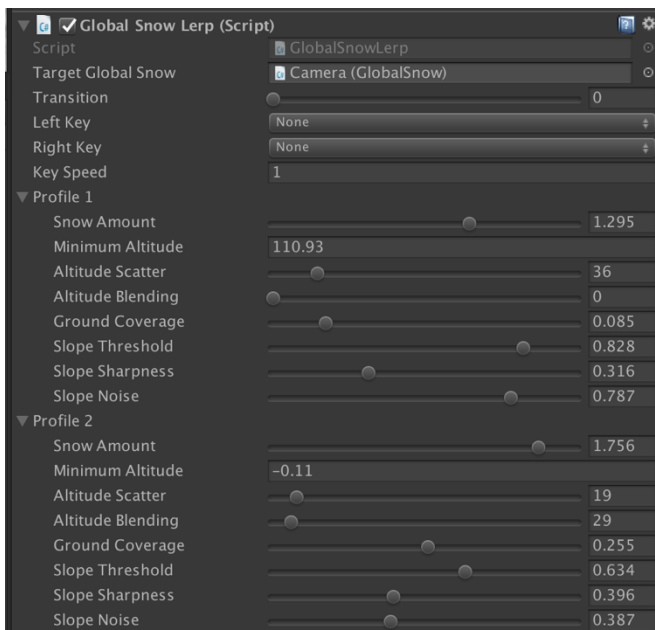
Step 1: Ensure your vehicle reports correct ground hits adding “Global Snow Collider Extra Info” script to the root of your vehicle prefab or gameobject. This script is necessary to detect wheel collisions and report them to Global Snow.

Step 2: Since the vehicle is a moving object, you will want to exclude it from Global Snow coverage (add it to a exclusion layer or add the Global Snow Ignore Coverage script to the root of the vehicle). Then, use the Global Snow/Moving Object Snow/Opaque shader to the materials of the vehicle in order to add static snow coverage on top of the vehicle parts.

Transition/Lerping between snow profiles

All global snow properties are animatable but also the asset includes a Global Snow Lerp script which allows you to quickly lerp between two sets of values.

Just create a gameobject and add the “Global Snow Lerp” script to it (you can also add this script to the camera):



The main settings are:

- **Target Global Snow:** this will be automatically picked by the script upon start but lets you assign a different object (useful if you instantiate the player camera after scene is loaded).
- **Transition:** value from 0 to 1. A value of 0 will use the settings of profile 1 while a value of 1 will use the settings of profile 2. Any value in between will mix the values from profiles 1 and 2 producing a smooth transition of the effect.
- **Left / Right Key:** allows you to specify keys which will decrease/increase the transition value when pressing them in playmode.
- **Profile1 and Profile2:** settings for global snow.

Frequent Asked Questions (FAQ)

Some objects that get clipped when Global Snow is enabled (forward rendering)

Global Snow uses replacement shaders to render a snow-covered version of the geometry. This snow layer is rendered before transparent objects as a screen-space effect.

Objects that render during the transparent queue won't be clipped by Global Snow, so this issue should only happen when some opaque shaders don't expose the `RenderType` appropriately (which is a recommendation). **Try using Fix Materials option in Scene Setup section of the inspector.**

Another solution to this issue involves editing that object shader and set the `RenderType` tag to one of the following:

- **"Opaque"**: this render type will cover the geometry with snow, and also allow footprints and terrain marks. It can also expose holes or transparent parts not covered by snow using alpha channel IF the `"OpaqueCutout"` parameter is enabled in Global Snow inspector.
- **"TransparentCutout"**: same than before but will also take into account the alpha channel to discard those pixels REGARDLESS of the `OpaqueCutout` parameter.
- **"TerrainOpaque"**: same that `"Opaque"` but will ALWAYS ignore alpha channel. Commonly used for full opaque objects.
- **"AntiSnow"**: will cause the geometry to prevent any kind of snow covering. Useful for water or liquid objects that render in the opaque queue or anything you want to exclude at shader level. Note that you can also exclude objects from GlobalSnow using the Layer Mask parameter under Coverage section in the inspector or adding the `GlobalSnowIgnoreCoverage` script to that gameobject.

When you edit the shader, the change should read like this (other tags should not be touched!):

```
SubShader {  
    Tags { "RenderType"="Opaque" ... }  
    ...  
}
```

Other render types supported are:

- **"TreeOpaque"**: used for covering opaque parts of standard Unity tree.
- **"SnowedSpeedTree"**: used as a replacement for the SpeedTree shader. This tag is used when you enable the `"Update SpeedTree"` parameter in GlobalSnow inspector.
- **"SnowedSpeedTreeBillboard"**: same but for the billboard trees.
- **"Grass"**: used to cover standard Unity grass objects.
- **"GrassBillboard"**: used to cover standard Unity grass billboard objects.

How can I exclude some objects from being covered by snow?

You have 4 options here:

- 1) Add those objects to a custom layer and exclude that layer in the Layer Mask parameter of Global Snow (Coverage section).
- 2) ... or simply add "GlobalSnowIgnoreCoverage" script to those objects (then no need to change their layers).
- 3) ... in deferred rendering path, change the shader and add "exclude_path: deferred" at the end of the #pragma surf line. This directive informs Unity to render that object in forward rendering path so Global Snow won't cover the object with snow.
- 4) ... in forward rendering path, edit the shader of those objects and set the RenderType to "AntiSnow".

Note: in deferred rendering path, terrain can't be excluded using the above methods. See next question for a solution.

How can I exclude terrain from snow in deferred rendering path?

This is a 2-step process:

- 1) The terrain material needs to be switched to the TerrainWithoutSnow material provided by Global Snow. This material can be found in GlobalSnow/Resources/Common/ExcludedTerrain folder.
- 2) Edit global snow shader and uncomment the lines related to stencil. This shader is located in GlobalSnow/Resources/Workflow/Deferred/Shader/SnowOverlay.shader file. Once you open the file, you will immediately see the commented lines. Remove the comment tags (/* and */) to enable the stencil filter.

There're no snow marks on some objects, like floating platforms. They only work on the terrain.

You need to add the GlobalSnowCollisionDetector script to any geometry (except for the terrain) if you want it to support terrain marks when collisions occurs. Demo scene contains some geometry that allows terrain marks over them using this approach. You can also add custom terrain marks using code, see below.

How can I manually add terrain marks over any place?

You can use this code to add any number of marks to the terrain or geometry that's covered by snow:

```
GlobalSnow snow = GlobalSnow.instance;  
snow.MarkSnowAt(position, size);
```

Where position is a Vector3 in world space coordinates and size is a value between 0 and 1.

[How can I manually add footprints over any place?](#)

You can use this code to add footprints at any position:

```
GlobalSnow snow = GlobalSnow.instance;  
snow.FootprintAt(position, direction);
```

[Can I customize footprint shape?](#)

Yes, the footprint feature uses the Footprint texture located in Resources/Textures folder. Feel free to use another texture for your footprints 😊

Alternatively, you can use the Terrain Marks feature to generate footprints-like marks as your FPS character moves (enable the FPS Marks option).

[Can I change the size of the terrain marks?](#)

You can specify a default mark size using the GlobalSnow inspector or if using the API you can pass a value between 0 and 1. You can also add GlobalSnowColliderExtraInfo script to any object and set the mark size individually in this script.

[I see terrain marks over the distance, but nothing should be causing them.](#)

Increase the **Extents parameter** or reduce the **View Distance parameter** in the Terrain Marks section of Global Snow inspector.

[Is there any footfall option?](#)

You can find 3 snow footfalls audio clips inside GlobalSnow/Resources/Footfalls folder. Assign them to your First Person Controller script (footsteps sounds property).

Note that there's currently no way to determine if there's snow under players feet. You could use a RayCast from above the player position to check if it's under cover and switch between normal footfalls and snow footfalls.

[How to enable snow reflections on water? \(forward rendering\)](#)

Support for snow reflections in Unity Water Pro system from Standard Assets is included. For instructions on setting up, please check the README file inside GlobalSnow/Extras/WaterReflections folder.

[Does it work with VR/Single Pass Stereo Rendering?](#)

Yes, it has been tested with Oculus and OpenVR SDKs. For OpenVR you need to use Unity 5.5 or higher and enable the option "Force VR SPSR" available in the Scene Setup section.

It's recommended to use deferred rendering since the performance will be better.

[How can I manually add static snow over moving objects?](#)

Check page 12 for more details.

[Unity's standard tree billboards are not covered by snow. Is it possible to use them?](#)

Forward rendering mode

Standard Tree billboards render as transparent objects hence you need to make a couple of changes:

- 1) Edit GlobalSnow.shader and uncomment the last SubShader section related to TreeBillboard.
- 2) Edit GlobalSnow.cs script and comment out [ImageEffectOpaque] just before OnRenderImage method.

Deferred rendering mode

Locate the GrassAndBillboardsReplacementShaders package inside the folder Resources/Workflow_Deferred_Shaders/Grass & Billboards and import it. Once imported, they should automatically override the default shaders (it's possible that you need to reload the scene).

Compilation takes a long time / GlobalSnow adds many shader variants, can I reduce them?

If you use deferred mode, you can rename GlobalSnow.shader file to something like "GlobalSnow.shader.old" inside the folder Resources/Workflow_Forward/Shaders. This way, Unity won't waste time compiling this complex snow shader.

If you use forward rendering mode, edit GlobalSnow.shader file and look for these groups of keywords:

```
#pragma multi_compile __ GLOBALSNOW_FLAT_SHADING GLOBALSNOW_RELIEF GLOBALSNOW_OCCLUSION
#pragma multi_compile __ GLOBALSNOW_OPAQUE_CUTOUT
#pragma multi_compile __ GLOBALSNOW_FOOTPRINTS
#pragma multi_compile __ GLOBALSNOW_TERRAINMARKS
```

For example, if you really want to use terrain marks, you could replace this line:

```
#pragma multi_compile __ GLOBALSNOW_TERRAINMARKS
by
#define GLOBALSNOW_TERRAINMARKS 1
```

This change will enforce the use of terrain marks without multiplying the number of shader variants while also reducing compilation time.

Can I add "holes" into the terrain?

If you use a custom terrain shader that support holes, you can use the "CircularHole" or "QuadHole" prefabs located in GlobalSnow/Resources/Prefabs folder to erase the snow over the entrance of the hole.

These prefabs are just normal geometry what use a custom material "GlobalSnowEraser" and have the "GlobalSnowIgnoreCoverage" script attached. You can use any other geometry, assign the same material and script and it will act as a stencil object – snow will not appear behind that geometry.

Grass appears transparent?

If you have issues with the grass or grass billboard included in the folder Global Snow / Resources / Workflow_Deferred/Shaders / Grass & Billboards in deferred rendering path try the following:

Option 1:

Edit WavingGrass.shader and WavingGrassBillboard.shader and replace "Queue" = "Geometry+200" with "Queue" = "Geometry+1200"

Option 2:

Edit WavingGrass.shader and WavingGrassBillboard.shader and do not modify the Queue tag but instead comment out this pragma: // exclude_path:deferred

My FPS weapon is receiving snow. How can I exclude it?

To exclude objects from snow, usually the exclusion options detailed in this manual apply. However, since FPS weapons render very close to the camera near clip plane, we can just exclude any snow coverage if the position of the pixel in the depth buffer is very small. This trick only works for deferred rendering path.

Edit GlobalSnowDeferredOptions.cginc file and uncomment EXCLUDE_NEAR_SNOW macro. Save the changes and verify the result in your scene. You may need to increase the NEAR_DISTANCE_SNOW value if the weapon still receives snow.

Additional Support

Please visit kronnect.com's forum for questions, support and more info.

If you have any issue using GlobalSnow you can also reach us at contact@kronnect.me