Real Landscapes - Valley Forest by TriForge Assets

Compatibility

Built-in Renderer (Unity 2020.3 or newer)
URP 12 (Unity 2021.3)
URP 13 (Unity 2022.1)
HDRP 10 (Unity 2020.3)
HDRP 12 (Unity 2021.3)
HDRP 13 (Unity 2022.1)

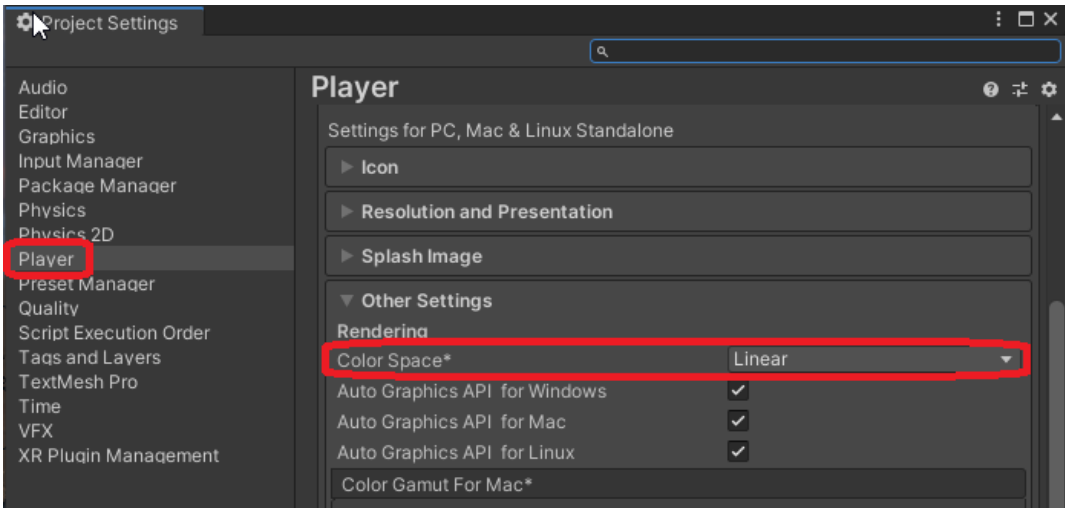
Limitations

(Built-in RP) SSAO will produce visual artifacts around geometry using Leaf Translucency shader due to Unity limitations. Using MSVO is recommended instead.

Snow controller does not affect the terrain as the pack does not include custom terrain shader.

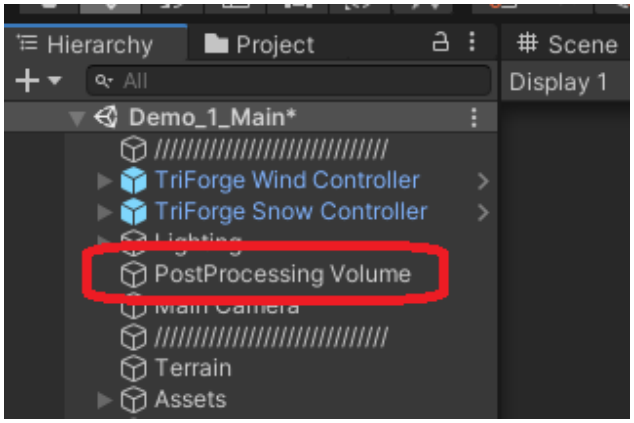
Setting up Post-Processing & visuals in Built-in Renderer

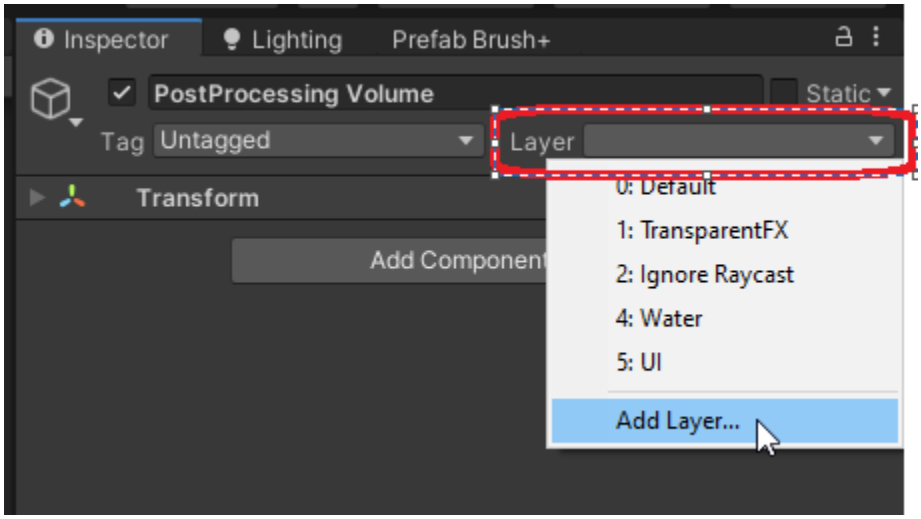
If you intend to use the Built-in Renderer follow these steps:
Because Unity Built-in Renderer does not include post-processing tools by default, the scene will not look as intended right after import and some set-up is required.
First, go to *Edit -> Project Settings -> Player* and change the Color Space from **Gamma** to **Linear**.



Then import the Post-Processing Stack from Package Manager. Go to *Window->Package Manager* then in Unity Registry search for Post Processing and hit install.

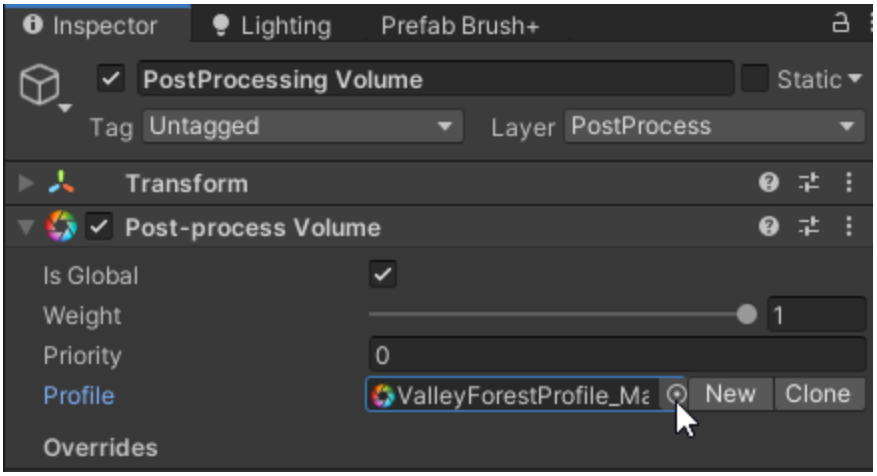
Then, select the PostProcessing Volume in the level hierarchy and in the inspector click on the Layer dropdown menu. Select *Add Layer…* and in one of the available slots type *PostProcessing*.



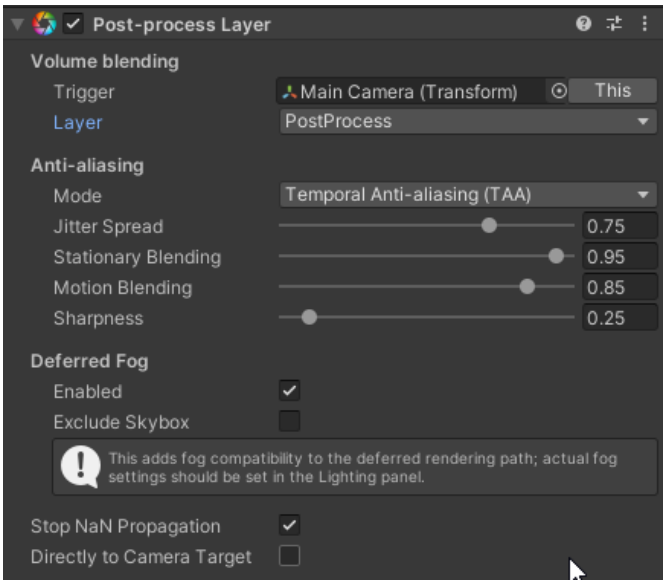


Then you must assign the newly created PostProcess layer to the PostProcessing Volume from the same menu you used to create it.

Next step is to add global volume to the PostProcessing Volume. Click on the Add Component button and add Post-process Volume component. Then tick *Is Global* box and assign the *ValleyForestProfile_Main* (or ValleyForestProfile_Overcast if you are configuring the overcast scene) profile.



Finally, we need to enable Post-processing on our camera. Select the Main Camera in the level hierarchy, click Add Component and add Post-process layer component. Switch the Layer option on the component to PostProcessing. Enable deferred fog and, if you wish so, you can untick the Exclude Skybox option. There, you can also set Anti-aliasing.



It is also recommended to set your project's LOD bias to 2 in the *Edit -> Project Settings -> Quality*.

URP & HDRP Compatibility

By default the pack is set up for Built-in Renderer and needs to be converted to work with URP or HDRP. The conversion patches are located in *!HDRP and URP Conversion Patches* folder.

HDRP Compatibility

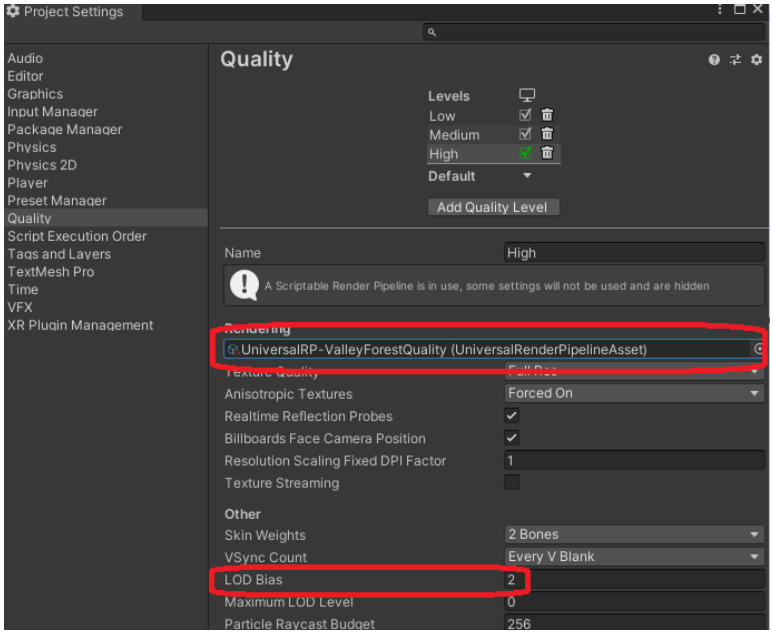
Import the *RLVF HDRP 10 Conversion Patch* file into your HDRP project (either double-click on it or right-click in the Project window and select Import Package->Custom Package) and all relevant files, prefabs, materials and shaders will be replaced with HDRP-compatible versions. Also a new folder will be added into the Scenes folder, containing HDRP variants of the demo scenes.

It's also recommended to change the LOD Bias settings in HDRP Default Settings (HDRP10) or HDRP Global Settings(HDRP 12) from Low to High.

URP Compatibility

Import the *RLVF URP 12 Conversion Patch* file into your URP project (either double-click on it or right-click in the Project window and select Import Package->Custom Package) and all relevant files will be replaced with URP-compatible versions. Also a new folder will be added into the Scenes folder, containing URP variants of the demo scenes.

Then, replace the default URP quality settings file with UniversalRP-ValleyForestQuality in the *Edit -> Project Settings -> Quality* tab and set the LOD Bias to 2.

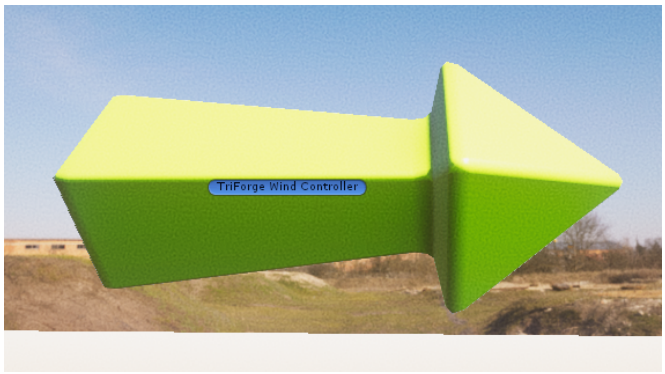


Note: URP lacks many features that Built-in or HDRP have and it may not always be possible to achieve the same visual results as in the screenshots in the store page. A notable example would be lack of any interaction between fog and the skybox, which is very visible in the overcast demo scene example.

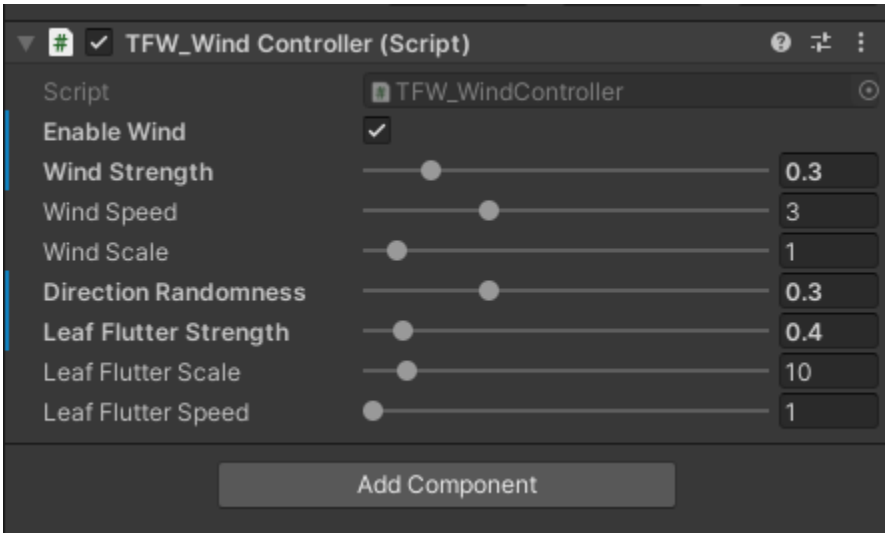
Wind and Snow Controllers

TriForge Wind Controller and TriForge Snow Controller are used to control the vegetation and prop materials. The prefab can be found in *Real Landscapes - Valley Forest/Content/Wind & Snow Controllers* folder.

Wind Controller



TriForge Wind Controller in the viewport



TriForge Wind Controller in the inspector

Direction of the wind is controlled directly by the rotation of the wind controller prefab and is indicated by the attached arrow model.

Wind controller allows you to shape the wind with following properties:

Enable Wind - turns the wind animation on or off.
Wind Strength - this is the main property and ideally the only one you would need to modify. The bigger the value the stronger the trees will bend and twigs flutter.
Wind Speed - speed of the wind animation, high values may produce unrealistic results. This value should always be constant at runtime.

Wind Scale - scale of the wind noise mask. Higher values will make the wind movement appear more synchronized across big patches of vegetation, while smaller will make the movement more “random”.
Direction Randomness - Maximum value will make the trees completely disregard the direction of the wind controller prefab and will cause the trees to rotate according to their own local forward vector.
Leaf Flutter Strength - controls the intensity of twig movement
Leaf Flutter Scale & Speed - identical to Wind Speed and Wind Scale, but affecting only twigs/leaves.

Snow Controller

The snow controller uses only a single property - 0 means no snow; 1 - full snow coverage.

Nature Renderer Compatibility

First, make sure you have Nature Renderer properly imported into your project and that you have the *Nature Renderer.templatex* file located at *Assets/Visual Design Cafe/Nature Shaders/Integrations*.
A compatibility patch for Nature Renderer is available in the *Real Landscapes - Valley Forest/Asset Compatibility Packs* folder. Importing it will add a new set of grass prefabs compatible with Nature Renderer’s procedural instancing. *Only Built-in Renderer is supported at this time.*

Vegetation Studio Pro Compatibility

Since version 1.0.2 the vegetation shaders for Built-in Renderer now support VSP Indirect Instancing out of the box, so there is no need to apply any compatibility patch. HDRP and URP should work, but with regular instancing only.