Creating immersive, photorealistic VR experiences with the High Definition Render Pipeline

Sebastien Lachambre, **Sebastien Lagarde** **and** **Fabien Houlmann**, March 12, 2020

Technology

> We're officially bringing virtual reality to the High Definition Render Pipeline (HDRP). In 2019.3.6f1 and after with package version 7.3.1, HDRP is now verified and can be used in VR.

This blog post takes a technical dive into using HDRP in your VR project. To learn more about all the possibilities that HDRP offers, take a look at this blog post.

VR in HDRP is designed so that:

1. All HDRP features are compatible with VR.
2. HDRP is fully supported with the new Unity XR Plugin Framework.
3. Single-pass (instancing) is the default and recommended rendering solution for VR.

Feature highlights

Using HDRP for a VR project, you can take advantage of all the render pipeline's features to create experiences bound only by your imagination. With its state-of-the-art rendering techniques, HDRP can deliver stunning, photorealistic visuals at a quality rarely seen before in virtual reality environments.

Here's a very quick overview of the features available for your VR projects:

- Deferred and Forward rendering
- All light types, shadows, decals, and volumetrics
- Screen space effects
  - Ambient Occlusion (AO)
  - Screen Space Reflection (SSR)
  - Subsurface Scattering (SSS)
  - Distortion and Refraction
- Post-processing

    - Color Grading, Anti-aliasing, Depth of Field, etc.
- All VFXs from Visual Effect Graph

Supported platforms

VR for HDRP is currently available for the following platforms and devices:

- Oculus Rift & Rift S (Oculus XR Plugin, Windows 10, DirectX 11)
- Windows Mixed Reality (Windows XR Plugin, Windows 10, DirectX 11)
- PlayStationVR

OpenVR: Valve is currently developing their OpenVR Unity XR Plugin for 2019.3 and beyond, and this will be available soon.

Stereo rendering techniques

A native VR implementation will process everything twice – once for each eye. We call this solution *multipass rendering*. HDRP supports multipass rendering, however, we do not recommend this method because your application will use twice as much CPU power for rendering, essentially doubling your number of draw calls. On top of that, shadows will be rendered twice and could consume a significant portion of your GPU budget.

That said, there are some cases where using multipass is appropriate:

- If your system has a small amount of GPU memory, multipass uses less memory for render targets than single pass.
- If for some reason you need to render vastly different viewpoints for each eye.

A faster solution is to use single pass (instanced) rendering. In this mode, every draw call is simultaneously rendered for both eyes. This is accomplished by using a texture array for the render targets and instanced draw calls. Furthermore, culling and shadows are processed only once per frame.

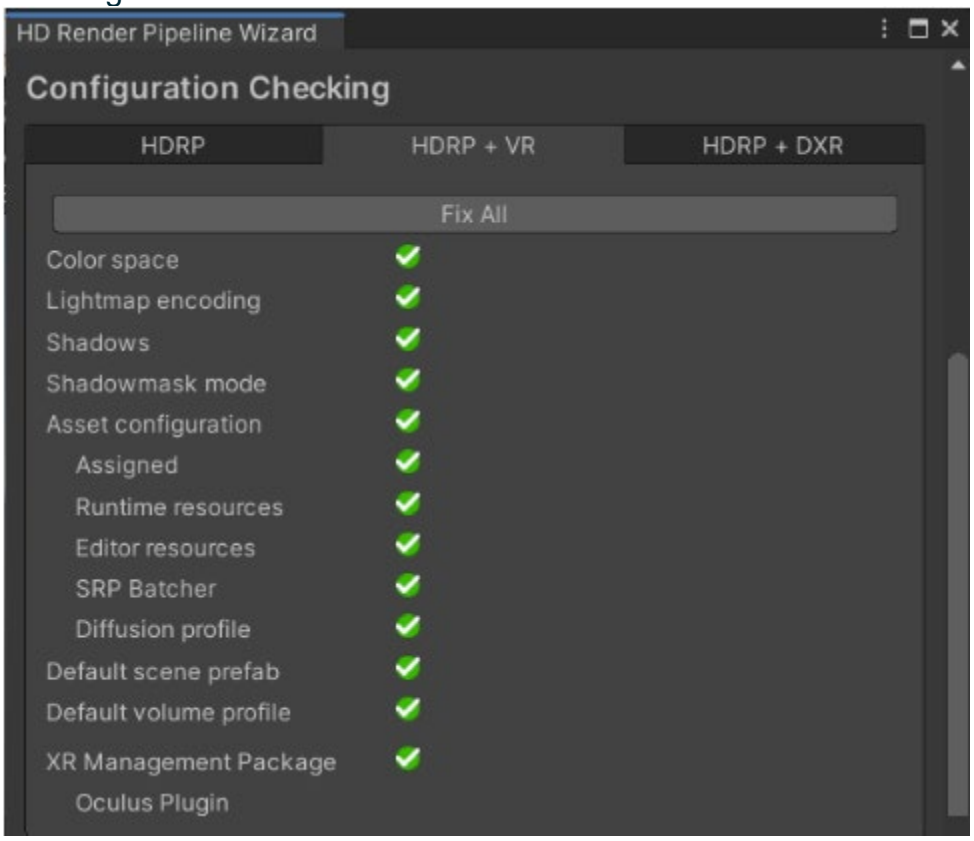
HDRP has been designed so that all features are compatible with VR and optimized for single pass rendering.

The key design decision was to use texture array for all render targets (even when you’re not creating for VR). This decision, coupled with shader macros, has allowed us to author shaders that are automatically compatible with VR, apart from a few special cases (e.g., light list generation, indirect tile deferred shading, volumetric lighting, and camera-relative rendering).

Note that single pass rendering for double-wide textures is not supported by HDRP because of the additional complexity and overhead required for all full-screen passes and effects.

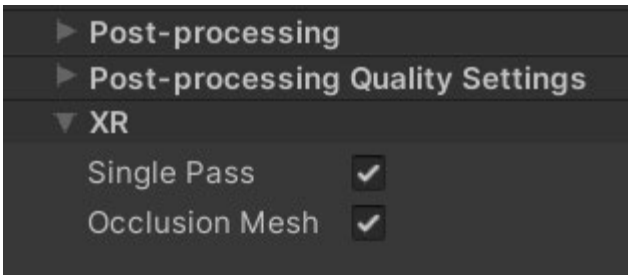
Start with HDRP VR

To get started with HDRP VR, check out the VR Overview section in Unity's documentation. To help you to set up HDRP with VR, we also provide an HDRP Wizard, which validates your settings and can help you to fix any settings the Wizard identifies as being incorrect.



The HDRP VR Wizard

To configure your project for VR manually using the new XR plugin framework, please refer to the documentation. To set up single pass rendering, you must have *both* Project Settings set to Single-Pass Stereo Rendering mode *and* the HDRP asset settings set to Single Pass. HDRP will default to multipass if either one of those two options isn't enabled for single pass.



Enable Single Pass rendering with the Oculus plugin in Project Settings and in HDRP asset settings.

Anti-aliasing

Reducing aliasing is extremely important in VR in order to create a great user experience and avoid breaking the virtual environment's immersiveness. HDRP provides several solutions to help with anti-aliasing.

The camera anti-aliasing modes are described in-detail in Unity's documentation. These options include:

- Multisampling Anti-Aliasing (MSAA) is supported with forward rendering. You can balance quality versus performance by choosing the number of samples (2x, 4x, 8x). This technique can achieve great results, but it can also be expensive.
- Temporal Anti-Aliasing is likely the best solution for most applications. It's very effective at reducing aliasing, but it can blur out some details. You can counterbalance this blur with the included sharpen filter control.
- Somewhat cheaper solutions include Fast Approximate Anti-Aliasing (FXAA) or Subpixel Morphological Anti-Aliasing (SMAA).
- It's also possible to combine MSAA with TAA, FXAA, or SMAA. This technique improves visual quality, but the cost is cumulative.
- There is also additional shading anti-aliasing available per material with Geometric Specular Anti-Aliasing, which you can tweak directly on materials. Its use is recommended for smooth and dense surfaces.

Performance

VR rendering is extremely demanding due to the higher refresh rate and resolution required to display to both eyes. Make sure to disable any features you don't need in the HDRP asset settings. Features like Volumetric aren't suited for VR applications since their performance doesn't meet the required 90 fps despite being supported.

Frequently monitoring and profiling performance will help you to identify any bottlenecks in your project.

Note that by default, the precision of volumetric effects (z slices) in VR will be halved to keep GPU performance more acceptable. In addition to volumetric lighting, it's highly recommended to **disable HDRP Area Light support** when doing a VR project. Unlike other features, Area Light must be disabled via the shader configuration files.

There are two rendering methods available in HDRP, which also impact performance: Lit Shader Mode Forward and Deferred. To learn about the differences between those two modes, please see the documentation. Choosing the right mode for VR depends on the project's requirements. Forward rendering lets you enable MSAA and reduce memory usage, while Deferred rendering is more efficient for projects with a large number of lights, but it also consumes more memory.

Another factor that influences GPU performance is the resolution of the rendering buffer. This resolution is initially set by the XR display plugin and depends on the headset you're using. You can then adjust the resolution in your application or use the dynamic resolution feature to drive resolution depending on the current scene's context. For example, resolution could be adapted based on the current GPU frame time.

For more tips, check out this HDRP VR talk from Unite 2019 Copenhagen.

Technical dive into HDRP VR

To support VR in HDRP, we've added a set of shader macros to help handle the view instancing and texture array usage for the render target. For example, you can declare a texture in a shader with the following code:

TEXTURE2D_X(MyTexture);

On platforms that support texture array, this macro will expand to TEXTURE2D_ARRAY. If the platform does not support texture array or if the setting in ShaderConfig.cs is disabled, the macro will expand to regular TEXTURE2D. Similar features are available for texture sampling.

On the shader side, the proper view constants (view matrix, projection matrix, etc.) are stored in the array and indexed based on the eye index, which is derived from the instanceID of the primitives. In the case of compute shaders, the z dispatch dimension is used to identify each eye. The macro UNITY_XR_ASSIGN_VIEW_INDEX is usually used to assign the proper eye index.

The future of HDRP VR

Future versions of HDRP VR will focus on:

- Improving performance with new hardware options like Variable Rate Shading
- Improving platform support with Vulkan and DX12
- Improving devices support
- Extending single pass to support more than two views

Let us know what you think

You can start taking advantage of HDRP VR today. We'd love to hear your feedback in the HDRP forum as we continue to make improvements.

Related posts

Previous

Escape the witch's cottage in our new VR learning experience

Kat Woolley
February 27, 2020•6
Technology

HDRP: Out of Preview in 2019.3

Sebastien Lagarde
February 24, 2020•13
Technology

Realizing rapid conceptual design with kitbashing

Alitt Khaliq +2
August 11, 2020•9
Asset Store

Learn to create AR and VR apps in Unity

Estefany Delgado
May 1, 2020•3
Manufacturing Technology

High Definition Render Pipeline real-time ray tracing is now in Preview

Anis Benyoub
March 6, 2020•7
Technology

Creating safer construction projects with virtual reality

Tony Faccenda
March 5, 2020•6
AEC

Escape the witch's cottage in our new VR learning experience

Kat Woolley
February 27, 2020•6
Technology

HDRP: Out of Preview in 2019.3

Sebastien Lagarde
February 24, 2020•13
Technology

Realizing rapid conceptual design with kitbashing

Alitt Khaliq +2
August 11, 2020•9
Asset Store

Learn to create AR and VR apps in Unity

Estefany Delgado
May 1, 2020•3
Manufacturing Technology

Next

- 1
- 2
- 3

- 4
- 5
- 6

7 replies on “Creating immersive, photorealistic VR experiences with the High Definition Render Pipeline”

Panagiotis Kontogiannissays:

March 24, 2020 at 8:41 am
Will it be compatible with the new one PS5?

AlexGSsays:

March 16, 2020 at 8:18 am
Not a single mention to Steam VR? and whats with Oculus Quest?

Xylo Klapasays:

March 15, 2020 at 9:45 am
I am so excited

fuzzy3dsays:

March 14, 2020 at 1:35 pm
Hi Alejandro, we have great results with HDRP+ VR on Oculus Quest with Link (tethered Quest). I think no HDRP is possible on Android now.

Alejandro Castansays:

March 13, 2020 at 12:55 pm
Hi. Great news Unity !!! Is this High Definition Render Pipeline for Oculus Quest?