

AMD FidelityFX Super Resolution 1.0

Unreal Engine 4 Plugin

AMD FidelityFX Super Resolution (FSR) is an open source, high-quality solution for producing high resolution frames from lower resolution inputs. It uses a collection of cutting-edge algorithms with a particular emphasis on creating high-quality edges, giving large performance improvements compared to rendering at native resolution directly. FSR enables “practical performance” for costly render operations, such as hardware ray tracing.

This guide will help you navigate the process of installing and configuring the FSR Plugin for Unreal Engine 4.

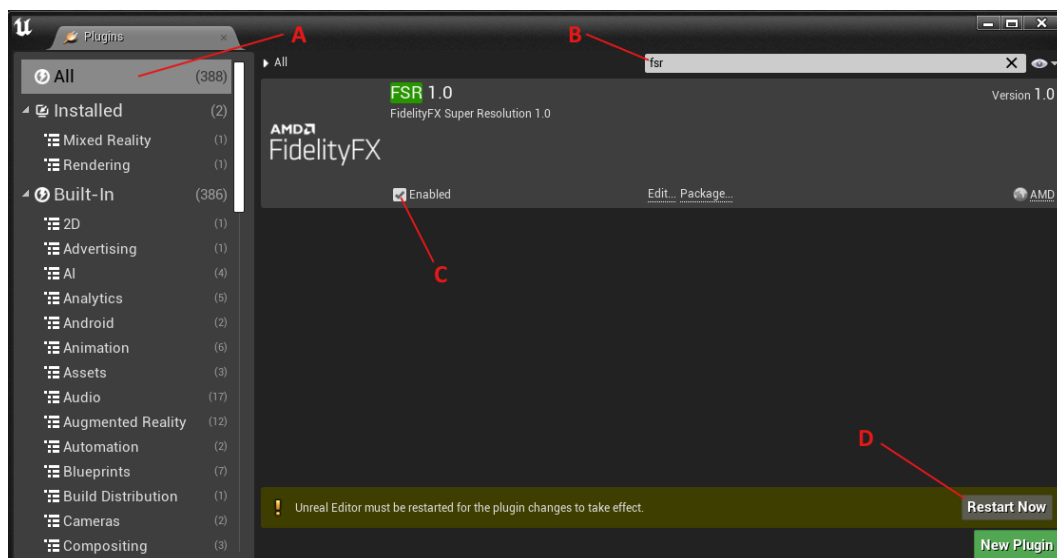
Plugin Installation

This plugin is intended for [Unreal Engine 4.27.1](#)* or later.

*If you are not a registered Unreal Engine developer, you will need to [follow these instructions](#) to register for access to this link.

Installation Procedure

1. Locate the *Engine\Plugins\Runtime* directory of your Unreal Engine 4 installation.
2. Unzip *FSRPlugin.zip* to *Engine\Plugins\Runtime\AMD*.
3. Launch Unreal Engine 4. If Unreal Engine 4 is already running, close and restart it.
4. Open your Unreal Engine 4 project.
5. Navigate to **Edit > Plugins** in the Unreal Engine 4 toolbar.
6. Within the plugin dialog...
 - a. Ensure that **All** is selected on the left side.
 - b. Type **fsr** into the search box in the top right corner.
 - c. Select the **Enabled** checkbox for the **FSR 1.0** plugin.
 - d. When prompted, click **Restart Now** to apply changes, and restart Unreal Engine 4.



Plugin Configuration

Usage

FSR 1.0 can be enabled or disabled with the CVar `r.FidelityFX.FSR.Enabled`.

Upscaling percentage can be tuned with the CVar `r.ScreenPercentage`. We recommend the following values:

- **Ultra Quality** : `r.ScreenPercentage 77`
Produces an image with quality virtually indistinguishable from native rendering. It should be selected when the highest quality is desired.
- **Quality** : `r.ScreenPercentage 67`
Produces a Super Resolution image which is representative of native rendering, with a sizeable performance gain.
- **Balanced** : `r.ScreenPercentage 59`
Produces a Super Resolution image approximating native rendering quality, with a major performance gain compared to native.
- **Performance** : `r.ScreenPercentage 50`
Visibly impacts image quality and should be only be selected in situations where needing additional performance is critical.

We strongly recommend ensuring that `r.ScreenPercentage` meets or exceeds a minimum value of 50 at all times.

Other Configurations

CVar	Default Value	Valid Values	Details
<code>r.FidelityFX.FSR.UseFP16</code>	1	0, 1	Enables half-precision floating point arithmetic, improving performance without visible artifacts.
<code>r.FidelityFX.FSR.EnableFP16OnNvDX11</code>	0	0, 1	Our testing revealed that certain models of NVIDIA GPUs may not produce correct results when running DirectX 11 in conjunction with the FP16 variant of the FSR shaders. If this is resolved in the future, then you may want to use this CVar to re-enable the use of the 16-bit version of FSR on those GPUs.
<code>r.FidelityFX.FSR.RCAS.Enabled</code>	1	0, 1	Robust Contrast Adaptive Sharpening Filter. Sharpens images after upscaling is complete.
<code>r.FidelityFX.FSR.RCAS.Denoise</code>	0	0, 1	RCAS Denoising. Consider enabling for grainy inputs, such as when dithering or film grain effects are applied prior to FSR.

r.FidelityFX.FSR.RCAS.Sharpness	0.2	[0.0, inf)	RCAS Sharpness tuning. <ul style="list-style-type: none"> • 0.0: sharpest • 1.0: 1/2 as sharp • 2.0: 1/4 as sharp • 3.0: 1/8 as sharp etc...
r.FidelityFX.FSR.HDR.PQDitherAmount	1.0	[0.0, 1.0]	HDR-Only Dither amount to apply for PQ->Gamma2 conversion, reducing color banding when the output device is ST2084/PQ.
r.FidelityFX.FSR.Post.FilmGrain	1	0, 1	Applies UE4 FilmGrain effects later in the post-processing chain, to avoid conflicts with FidelityFX Super Resolution.
r.FidelityFX.FSR.Post.ExperimentalChromaticAberration	0	0, 1	Be careful! <i>This feature is Experimental and may produce undesirable results in some situations!</i> Applies ChromaticAberration effects later in the post-processing chain, to avoid conflicts with FidelityFX Super Resolution.
r.FidelityFX.FSR.Debug.ForcePS	0	0, 1	If enabled, runs FSR and post-FFX ChromaticAberration passes in VS-PS pipelines instead of CS.

Project Configuration

FidelityFX CAS

FidelityFX Super Resolution 1.0 contains a built-in sharpening pass called RCAS that can be configured through the CVar `r.FidelityFX.FSR.RCAS` and is enabled by default. If your project has already integrated [FidelityFX-CAS](#), be sure to completely disable FidelityFX CAS – including any in-game menu options – while FidelityFX Super Resolution 1.0 is enabled. This will prevent over-sharpening your final renders and improve integration results.

Hybrid Upscaling

If Temporal Upscaling is requested via the CVar `r.TemporalAA.Upsampling`, either the default Unreal Engine 4 **Temporal Anti-Aliasing** with **Upsampling** (TAAU) algorithm or a replacement Temporal Upscaling algorithm installed via plugins will always serve as the Primary Upscaler regardless of any further FSR settings. Under these conditions, `r.ScreenPercentage` will apply to the Primary Upscaler and will have no effect on FSR. However, FSR can still be used as a Secondary Upscaler to meet project-specific needs!

If `r.TemporalAA.Upsampling` and `r.FidelityFX.FSR.Enabled` are both requested, this **Hybrid Upscaling** mode is selected automatically. In Hybrid Upscaling mode, upscaling intensity can be tuned to meet project requirements using the CVar `r.SecondaryScreenPercentage.GameViewport`.

Mip Bias Selection

With upscaling enabled, rendered resolutions are frequently much lower than displayed resolutions. When combined with mipmap optimizations, this can limit the fidelity of texture samples relative to the displayed resolution. In these situations, specifying a **negative mip bias** can improve the quality of final renders.

When **Hybrid Upscaling** is in use, mip biasing guidelines for the Temporal Upscaler should be followed in lieu of these instructions. In the case of the Unreal Engine 4 default **TAAU** implementation, all mip bias selection is handled automatically.

When FSR is the Primary Upscaler, the developer should consider specifying a custom mip bias. This can be easily done in Unreal Engine 4 by CVar, using `r.MipMapLODBias`. The desired mip bias varies with the ratio of (rendered resolution) : (displayed resolution), which is also represented as `r.ScreenPercentage`. For upscaling with values of `r.ScreenPercentage` in the range `[0, 100)`, we recommend the following formula:

$$r.MipMapLODBias = \log_2(r.ScreenPercentage/100)$$

For the recommended values of `r.ScreenPercentage`, these values are recommended:

- **Ultra Quality** (`r.ScreenPercentage 77`) : `r.MipMapLODBias = -0.3765`
- **Quality** (`r.ScreenPercentage 67`) : `r.MipMapLODBias = -0.5771`
- **Balanced** (`r.ScreenPercentage 59`) : `r.MipMapLODBias = -0.7606`
- **Performance** (`r.ScreenPercentage 50`) : `r.MipMapLODBias = -1.0000`

Panini Projection

The FidelityFX Super Resolution 1.0 plugin is currently incompatible with [UE4 Panini Projections](#). Panini Projection will be explicitly disabled whenever FSR 1.0 is enabled.