

Developer

/ Documentation

/ Unreal Engine ▾

/ Unreal Engine 5.4 Documentation

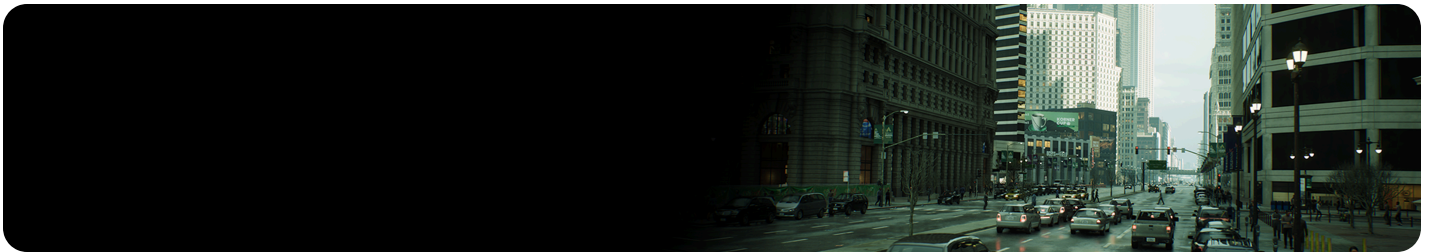
/ Designing Visuals, Rendering, and Graphics

/ Third-Party Rendering Tools and Plugins

/ NVIDIA SLI Alternate Frame Rendering

NVIDIA SLI Alternate Frame Rendering

Integration of NVIDIA's Alternate Frame Rendering technology that enables support for games using multiple GPUs with SLI.



Unreal Engine supports NVIDIA's **Alternate Frame Rendering** (AFR) for packaged games running on NVIDIA SLI configurations. Alternate Frame Rendering works by alternating frames between the linked GPUs. One frame is rendered by GPU 1 with the next frame rendered by GPU 2, before repeating the process. Using this method, image quality and performance are improved by using multiple GPUs on a single monitor.



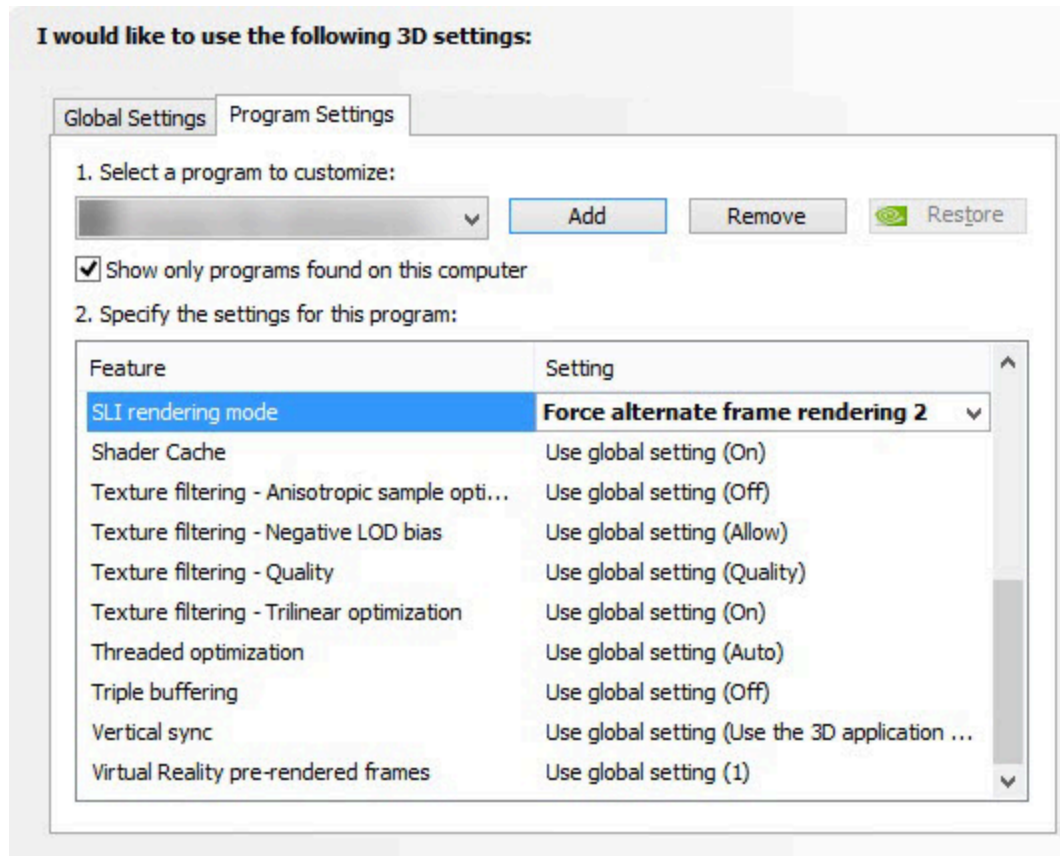
Projects that intend to ship with support for Alternate Frame Rendering will need to work directly with NVIDIA to test their games and have it automatically switch over to use this functionality where possible.

For additional details, see NVIDIA documentation on their [SLI modes, specifically Alternate Frame Rendering](#) here.

Forcefully Enabling Alternate Frame Rendering

NVIDIA Control Panel allows for some applications to be manually added which support AFR forcibly enabling the SLI rendering mode. Add applications to the NVIDIA Control Panel using the following steps.

1. Open the **NVIDIA Control Panel** from your task tray, and navigate to **Manage 3D Settings**.
2. Click on the **Program Settings** tab. Under **Select a program to customize** use the dropdown to select the program you want to add.
3. Next to **SLI Rendering Mode**, select **Force Alternate Frame Rendering**.



This feature is not guaranteed to improve the quality or performance of your applications. It is not intended for use with the Unreal Engine Editor, specifically. Should you want this for your developed and released project, you will need to talk with NVIDIA directly to set up this functionality with their drivers.