

Developer

/ Documentation

/ Unreal Engine ▾

/ Unreal Engine 5.4 Documentation

/ Designing Visuals, Rendering, and Graphics

/ Third-Party Rendering Tools and Plugins

/ NVIDIA Nsight Aftermath for GPU Pipeline Debugging

NVIDIA Nsight Aftermath for GPU Pipeline Debugging

Integration of NVIDIA Nsight™ Aftermath C++ library that enables post-mortem GPU crash analysis on NVIDIA GPUs.



Unreal Engine supports NVIDIA's **Nsight™ Aftermath** C++ library for Windows-based developers, which provides additional data from NVIDIA GeForce-based GPUs after a crash occurs. This data contains crucial information about what the GPU was doing when the crash occurred, enabling you to track down what's happening with the GPU in your own projects.

Aftermath is a lightweight tool that is unobtrusive and reduces the performance footprint needed by some debugging tools. In fact, it's lightweight enough that it can even ship with finished games to provide the data you need from your customers' machines. Aftermath enables programmers to insert markers into their code that assist in tracking the root causes of crashes. This is now being used in the Unreal Engine Editor to track down and fix reported or captured issues.

For additional information and how to use it in your own project, visit [NVIDIA's Nsight™ Aftermath documentation](#) page.

Enabling NVIDIA Nsight™ Aftermath

Add the following console variable to your `ConsoleVariables.ini` configuration file to enable NVIDIA Nsight™ Aftermath in your project:tion file:

```
r.GPUCrashDebugging=1
```

 Copy full snippet

Or, you can enable it by passing it as a command line argument:


```
-gpucrashdebugging
```

 Copy full snippet

Logs

Once you've enabled NVIDIA Aftermath, you should see the following output in your log:

```
LogD3D11RHI: [Aftermath] Aftermath enabled and primed
```

 Copy full snippet

You can access your log in Unreal Engine 4 by going to **File Menu > Window > Developer Tools > Output Log** or by opening the log text file from your project folder.

When a crash happens, the log displays something similar to:

```
1 LogD3D11RHI: Error: Result failed at X:[Project
  Folder]\Engine\Source\Runtime\Windows\D3D11RHI\Private\D3D11Viewport.cpp:290
  with error DXGI_ERROR_DEVICE_REMOVED DXGI_ERROR_DEVICE_HUNG
2 LogRHI: Error: [Aftermath] Status: Timeout
3 LogRHI: Error: [Aftermath] GPU Stack Dump
4 LogRHI: Error: [Aftermath] 0: Frame2769
5 LogRHI: Error: [Aftermath] 1: FRAME
6 LogRHI: Error: [Aftermath] 2: Scene
7 LogRHI: Error: [Aftermath] 3: ComputeLightGrid
8 LogRHI: Error: [Aftermath] 4: Compact
9 LogRHI: Error: [Aftermath] GPU Stack Dump
```

In this particular example, the GPU has crashed, and the resulting lines with the `[Aftermath]` preceeding indicate the current status and where in the frame the problem is so that you can investigate the root cause.

Considerations

For an average project that has approximately 200 to 300 markers, Aftermath is fast enough but slowdowns can happen when you have certain things like lots of per-object shadows that increase its cost. For this reason, the feature is not enabled by default.