

Intel® Graphics Performance Analyzers (Intel® GPA) Release Notes - 2025.1

ID846010

Updated 4/3/2025

Version 2025.1

[Introduction](#)

- [What's New in This Release](#)
- [Notes](#)
- [System Requirements and Supported Platforms](#)
- [Installation Notes](#)
- [Known Issues and Limitations](#)
 - [For All Intel® GPA Components](#)
 - **[For Graphics Monitor](#)**
 - [For Graphics Frame Analyzer](#)
 - [For Graphics Trace Analyzer](#)
 - [For Intel® GPA Framework](#)

Intel® Graphics Performance Analyzers

- Improve performance by quickly identifying problem areas.
- Identify single frames of interest and profile them down to draw call level.
- View real-time CPU, GPU, and graphics API performance to quickly identify improvement areas.
- Evaluate workload performance across the CPU and GPU.

[Learn More](#)

Introduction

Intel® Graphics Performance Analyzers (Intel® GPA) provides tools for graphics analysis and optimizations for making games and other graphics-intensive applications run even faster.

The tools support the platforms based on the latest generations of Intel® Core™ and Intel Atom® processor families, for applications developed for Windows*.

Intel® GPA provides a common and integrated user interface for collecting performance data. Using it, you can quickly see performance opportunities in your application, saving time and getting products to market faster.

For detailed information and assistance in using the product, refer to the following online resources:

- [Home Page](#): view detailed information about the tool, including links to training and support resources, as well as videos on the product to help you get started quickly.
- [Get Started Guide](#): get the main features overview and learn how to start using the tools on different host systems.
- [Training](#): learn at your level with Getting Started guides, videos and tutorials.
- [User Guide](#): get details on how to analyze Windows applications.
- [Support Forum](#): report issues and get help with using Intel® GPA.

What's New in This Release

All Tools

- Added full support for Intel® Arc™ B-Series Graphics.
- Updated to address security issues.

Graphics Frame Analyzer

- Re-enabled the **Shader Profiler** for frames and streams captured from DirectX* 12 applications. We recommend updating to graphics driver version **32.0.101.6632** or later, as this version includes numerous fixes for optimal performance.
- Added initial support for displaying DirectX 12 acceleration structures. The new **Acceleration Structure** resource shows TLAS/BLAS hierarchical trees, including the properties of each node.
- Improved **Bottlenecks** Analysis: The metric values shown in the Bottlenecks tab are now collected only for the API calls involved with the bottleneck. Previously, they were aggregated for the whole frame.
- Enhanced UI/UX:

- Resource IDs are displayed in hexadecimal format to easily match the resource ID in the arguments and the resource list.
 - Enhanced syntax help for the buffer layout editor: Added highlighting for lines containing invalid instructions.
- Added GPU auto-selection feature: When opening from Graphics Monitor, Frame or Stream will be played on the GPU where it was captured. When opened from Graphics Frame Analyzer it will be played on the selected GPU.
- Added Playback GPU name to the window header for opened frame or stream, making it easy to identify the GPU used for playback.

Intel® GPA Framework

- Added experimental support for DirectStorage. This allows for capture and playback on the same machine of workloads using native DirectStorage compression types.
- Added support for Direct3D 11 video interfaces on the D3D12 runtime during deferred capture.
- Added acceleration structure information extraction support to the Direct3D 12 Metadata Extractor. This allows for more raytracing pipeline exposure when using the MetadataExtractor class, as seen in the sample.
- Added a gpa-player parameter for manual error handling control. Set --action-on-playback-error to 0 to terminate on error (default behavior), 1 to throw exception, and 2 to skip errors. This allows for more control when encountering playback issues.
- Added printing of GPA Framework-specific environment variables, such as log severity, to the help text of Intel GPA Framework binaries.
- Added support for Vulkan* SDK version 1.4.304.0.

Notes

Qt source code is available here: [Software Heritage archive](#) SWHID is swh:1:rev:90e86aee3dd3fdc502e65d5bf2916871ae9168fe

Vulkan - GitHub: Lower the barrier of entry to working with Vulkan by using developer friendly abstractions in [Intel® Graphics Performance Analyzers Utilities for Vulkan*](#)

graphviz source code is available in [Software Heritage archive](#) hash commit is 12782df8963d39f7ebaee6979567c7c1b8bb8c11

System Requirements and Supported Platforms

Minimum System Requirements

- Host processor: Intel® Core™ processor family (i9/i7/i5/i3)
- System memory: 8 GB of RAM
- Video memory: 512 MB of RAM
- Minimum display resolution for the client: 1280 x 1024
- Disk space: 5 GB for stream capture and 300 MB for frame capture

Direct installation on 32-bit Windows* systems is not supported.

Operating Systems and Graphics APIs supported by this release

- Target platform: Where your game runs
- Host platform: Where you run the analysis
- Target application: Supported applications that run on the target system

Target System (the system where your game runs)	Host System (your development system where you run the analysis)	Target Application (types of supported applications running on the target system)
Windows 10, 11	Windows 10, 11	Direct3D* 11 and 12; Vulkan* 1.1, 1.2, 1.3 OpenGL, OpenCL, Media SDK and oneVPL (Trace mode only)

OpenGL: Although the tools may appear to work with the latest OpenGL API on the Windows platform in Frame mode (Graphics Frame Analyzer), this API is not officially supported for Graphics Frame Analyzer. Some features and metrics may not be available.

Supported Targets for Windows Devices

Intel® GPA supports Intel® Graphics devices found in these Intel® processor generations as targets for analyzing Windows workloads. All these targets have enhanced metrics support.

Target Graphics	Target Graphics Type	Processor	GPU
Intel® Arc™ Graphics	integrated	Intel® Core™ Ultra processors	
Intel® Arc™ Dedicated Graphics	discrete		Intel® Arc™ B-Series Graphics
Intel® UHD Graphics 770	integrated	Intel® Core™ processors (14th gen)	
Intel® UHD Graphics 770	integrated	13th generation Intel® Core™ processors (Raptor Lake)	
Intel® UHD Graphics 770	integrated	12th generation Intel® Core™ processors (Alder Lake)	
Intel® UHD Graphics 750	integrated	12th generation Intel® Core™ processors (Rocket Lake)	
Intel® Arc™ Dedicated Graphics	discrete		Intel® Arc™ A-Series Graphics
Intel® Iris® X ^e MAX graphics	discrete		Intel® Arc™ (DG1)
Intel® Iris® X ^e graphics	integrated	11th generation Intel® Core™ processors	

Although the tools may appear to work with other graphics devices, because they are unsupported some features and metrics may not be available.

Graphics Driver Requirements

When running Intel® GPA, the tools generally require the latest graphics drivers for proper operation. We have tested with the driver version 32.0.101.6632, other drivers are not guaranteed. You may download and install the latest graphics drivers from the Intel [Download Center](#).

Intel® GPA inspects your current driver version and notifies you if your driver is out-of-date.

Installation Notes

Installing Intel® GPA

Download the Intel® GPA installer from the [Intel® GPA Free Downloads](#).

If you use the product in a host/target configuration, install Intel® GPA on both systems. For more information on the host/target configuration, refer to [User Guide](#).

To profile Vulkan applications, make sure to download and install the latest [Vulkan SDK and runtimes](#).

For DirectX applications, download and install the [DirectX End-User Runtime](#).

Technical Support and Troubleshooting

For technical support, including answers to questions not addressed in the installed product, visit the [Support Forum](#).

Known Issues and Limitations

For All Intel® GPA Components

- To collect stable metrics on third party graphics solutions, enable *Developer Mode* in Microsoft Windows OS settings.

For Graphics Monitor

- In Graphics Monitor when capturing in Stream mode, if the application is closed with Alt+F4 or via the "stop" button, the stream file produced may have corrupted metadata. Information reflected in Graphics Frame Analyzer may be incorrect for the following: API name, Resolution reported, Device name and/or GPU name. However, the stream can be opened for profiling in most cases.

For Graphics Frame Analyzer

- The message about a GPU driver version mismatch might appear in an open file dialog even when the stream was captured with the same driver.
- For capture-time compression, framerate may drop when compression is enabled, hence a much smaller stream may be the result of both compression and fewer frames being captured.

For DirectX Applications

- DirectStorage currently has only experimental support. If a crash occurs during capture or when opening a frame/stream, try disabling DirectStorage in the analyzed application.

- Running Advanced Profiling Mode on Intel Arc B-Series Graphics (formerly code-named Battlemage) requires administrative privileges. Please run Graphics Frame Analyzer as an administrator when you want to identify bottlenecks using Advanced Profiling Mode on Intel Arc B-Series Graphics.
- Advanced Profiling Mode is not available on Intel® Core™ Ultra processors (Series 1) GT1 configurations (GPU Device IDs 0x7D40 and 0x7D45).
- Graphics Frame Analyzer shows increased metrics values for Direct3D 12 ExecuteIndirect events whose CommandSignature contains resource binding commands which may change vertex buffer, index buffer, constant buffer, shader resource and unordered access views.
- Graphics Frame Analyzer's Shader Editor doesn't display HLSL source code if shaders are compiled offline without debug information.
- Graphics Frame Analyzer does not display the Direct3D 12 Pipeline Statistics metrics for frames that contain ClearState or SetProtectedResourceSession calls.
- Stream capture for 32-bit applications is not supported.
- Metrics in DXR Shader execution table don't take into account continuation DXR shaders when Intel Graphics Compiler splits shader into multiple parts.

For Vulkan Applications

- To profile Vulkan titles, make sure to download the Vulkan runtimes and SDK, version 1.4.304.0 or later.
- In the case where Vulkan applications run on multi-GPU machines choose to render on a non-default adapter, one outside of the list of explicitly enumerated adapters, stream files may be produced that cannot be analyzed in the Profiling View of Graphics Frame Analyzer.
- Metrics will not show in Graphics Frame Analyzer for applications created in Unity that use the Vulkan renderer.

For Graphics Trace Analyzer

- Flip queues are not supported on Windows 11 24H2 version. The support will be added in a future release.
- The process name in Unicode is not properly shown in CPU context switch tracks when the name doesn't match the "Language for non-Unicode programs" setting in Windows.

- Tasks in OpenCL Execution tracks may be incorrectly aligned with other tracks. Task durations and relative positions inside the track are correct.
- DirectX 11 Parallel Execution View in Graphics Trace Analyzer is not working properly on Intel® X^e-HPG platforms.

For Intel® GPA Framework

- Intel® GPA Framework has been released for Direct3D 12. Vulkan and Direct3D 11 support is released as Beta, and is subject to change.
- DirectStorage support is currently limited to workloads using native DirectStorage compression types. If the workload uses DirectStorage to open files, the streams will be playable only on the machine they were captured on.
- Performing a subcapture of any Vulkan stream by the subcapture tool is not supported. This will be addressed in a future release.
- When using the screenshot layer, if a large number of screenshots are requested in close frame-intervals (every or every other frame) it is suggested to use the `async=false` layer flag to prevent unusually high memory consumption and out of memory errors. This may result in a slight decrease in performance since screenshot cut and writing will be not run in async processes.
- Frame 0 or the setup frame cannot currently be specified as a repeat frame in gpa-player's range repeat functionality.
- GPA CPP Generator does not support source-level generation of calls beyond the last frame delimiter in a stream.
- User input for restore points and deferred stream capture may not be registered in certain applications. Please use time-interval restore points and capture layer parameters for deferred stream capture if user input fails.
- At present, the Intel® GPA Framework contains only Release builds (no debug builds of libraries).
- The capture layer supports only 64-bit applications. However, the injection mechanism will work with 32-bit processes for the purpose of allowing the capture of applications that require 32-bit launchers.
- Changing the shader representation language from anything other than the original language can cause the gpa-player to crash (Vulkan).

- Support is not provided yet for collecting query-based metrics from Vulkan transfer queues. When used, they are ignored by the MetricsExtractor when the graphics API pipeline statistics based query managers are used, but are accepted by the Intel Hardware Metrics Discovery library query manager which may return metrics results. No validation of such results has occurred.
- Crashes have been observed in some DXR/DXIL shader profiling scenarios, and the data obtained has had a zero duration.
- Deferred and Continuous capture/playback of Unreal 5.1+ titles using raytracing may show visual corruption.
- Link errors occur with use of the gpa-cpp-generator generated gpa-cpp/CMakeList.txt. To work around this limitation, replace `'${GPA_FRAMEWORK_LIBS}'` in that file with `'${GPA_FRAMEWORK_LIBS}${ENV{INTEL_GPA_FRAMEWORK}/lib/*'.`
- Playback of titles using XeSS with PSO filtering disabled (by supplying `--disable-pso-filtering` argument to gpa-player) will produce black or corrupted final framebuffers.
- Playback of the workloads utilizing Xess might cause sporadic playback problems.

[Intel GPA Main Page](#)

[Intel GPA Framework](#)