

Intel® Graphics Performance Analyzers (Intel® GPA) Release Notes - 2024.2

ID764276

Updated 7/17/2024

Version Latest

Public

- [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 System Analyzer](#)
 - [For Graphics Trace Analyzer](#)
 - **[For Intel® GPA Framework](#)**

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

Graphics Monitor

- Enhanced UX/UI:
 - Introduced **Collect data** and **Analyze data** panels titles with quick-access links to the user guide to improve product onboarding.
 - Added separate tabs for captured **Frames and Streams** and **Traces**.
 - Now consistently displaying filenames for captured items next to the preview image, which include the full application name, capture date and time, for easier identification.
 - The **Auto-Detect Launch Applications** setting is now available as an **Auto-Detect** checkbox located near the application startup mode selection.

Graphics Frame Analyzer

- Enhanced shader analysis:
 - Added Compiler Flags used for shaders in profiled DirectX* 11 and DirectX 12 applications.
 - Added the display of ISA assembly headers, including information about SIMD width, number of used GRF registers and Spill Size, for DirectX 12 applications.
 - ISA code now automatically updates to reflect changes in shader HLSL code when applying shader code modification experiments.
- Improved handling of pipeline states:

- Added highlighting and corresponding error messages for cases where modifications cannot be applied to Pipeline States for DirectX 12 applications.
 - UX improvement: Instead of showing Pipeline State per draw call, Pipeline State is now indicated per pipeline capture handle. When multiple draw calls are selected, Pipeline State changes are applied to all of them.
- Added **Accessibility Settings** in the **Settings** panel (Frame View), including three options for the **Color Blindness Scheme** and the **Use Patterns** mode.

Shader Profiler is disabled in this release for graphics drivers versions from 31.0.101.5590 up to 31.0.101.5762 (excluding). To re-enable shader profiling, please update the graphics driver to the latest version.

Intel® GPA Framework

- Enhanced single-threaded and multi-threaded playback with up to a 50% Direct3D* 12 performance improvement, accelerating analysis for users.
- Added preliminary support for dynamic rendering extension for Vulkan* applications.
- Upgraded the Intel® GPA Framework to utilize Windows SDK version 10.0.26100, enabling support for the majority of interfaces defined in the latest Agility SDK.
- Implemented many bug fixes and improvements, significantly increasing the Intel® GPA Framework's stability.
- Support for Vulkan SDK version 1.3.280.0.

Notes

"Send Feedback" functionality was removed from all tools. The feedback and questions to GPA development team can still be send through [Community Forum](#).

Deprecation notice: We have deprecated Dependency Viewer in Intel® GPA. This functionality will be removed from the Intel® GPA releases starting with version 2024.4.

Additional Notes

Qt source code is available here: <https://archive.softwareheritage.org/> SWHID is swh:1:rev:15b7e7434fd79334a5cf071e36cee7663fe1fb45

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

Trace-cmd source code is available in <https://archive.softwareheritage.org/> hash commit is f8ea7186250942cbb2741b82f167eca5fbba9ac0

graphviz source code is available in <https://archive.softwareheritage.org/> 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 Processor Type	GPU
Intel® Arc™ graphics	integrated	Intel® Core™ Ultra processor
Intel® UHD Graphics 770	integrated	13th generation Intel® Core™ processor (Raptor Lake)
Intel® UHD Graphics 770	integrated	12th generation Intel® Core™ processor (Alder Lake)
Intel® UHD Graphics 750	integrated	12th generation Intel® Core™ processor (Rocket Lake)
Intel® X ^e HPG	discrete	Intel® Arc™ A-Series (Alchemist/DG2)
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 31.0.101.5522, 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 [Best Practices](#).

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

- Shader Profiler is disabled in this release for graphics drivers version 31.0.101.5590 and later. To re-enable shader profiling, please downgrade the graphics driver to the recommended version 31.0.101.5522.

- Frames and streams captured from different DX12 applications utilizing the Agility SDK can't be opened in Frame Analyzer simultaneously. This bug will be fixed in future releases.
- In rare cases the Resource Target Dependency Viewer may fail to load on some machines if it was used before. If the Resource Dependency Viewer shows “Loading...” on loading and it never loads, clear your browser’s cache and local storage for “localhost”. Restart Graphics Frame Analyzer, then open the Render Target Dependency Viewer.
- 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 frames captured with the new Frame [DX12] mode some features are not yet supported in Graphics Frame Analyzer: Pixel History and others. We are planning to support these in future releases.
- 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

- In the Graphics Frame Analyzer, duplicated ISA items may be displayed in the shader language dropdown menu within the Resource Viewer when the Shader Profiler is enabled. Each item with the same name displays identical ISA code for the shader.
- Shader Profiler is temporarily disabled for the DirectX11 API.
- 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.
- A few DirectX 11 applications that create a temporary Direct3D 11 device may crash if started from Graphics Monitor. To resolve this issue, please enable the System-wide Time-based GPU sampling interval option in the Metrics tab of the Graphics Monitor options.

- 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 1.3.280.0 or later.
- When profiling Vulkan applications the recommended Graphics driver version is 31.0.101.4146. **Graphics Frame Analyzer may fail to open frames with driver version 31.0.101.4369 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 System Analyzer

- Frame-based GPU metrics are disabled in the System Analyzer for DirectX 12 applications. As a workaround set **GPU Sampling interval** to any other value (400, 200, 100, 50, 25, 12 microseconds) to enable system-wide time-based metric collection (or use System View mode for profiling). To set the GPU sampling interval click the **Options** button on the Graphics Monitor Launcher screen and go to the **Metrics** tab.

For Graphics Trace Analyzer

- 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.
- 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.
- Subcapture functionality only supports Direct3D 12 at this time.
- 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.
- A modification of shaders in HLSL representation can cause the gpa-player to crash if the original shader representation was different from HLSL (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 capture/playback of Unreal* 5.3 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.
- The gpa-metrics-collector tool will write the wrong GPU identification details in the generated metadata file (*.metadata.txt) when the platform has multiple GPUs, the GPU to use is specified using the new vendor/device id options, and the associated GPU index is not 0 (the default GPU). This will be fixed in a future release.

[Archived Release Notes](#)

[Back to Release Notes' Overview](#)

[Intel GPA Main Page](#)

[Intel GPA Framework](#)