Puget Bench
Testing Priorities & Methodology

Puget Bench Mission

Puget Bench was born out of a simple frustration with the lack of benchmarks targeted at creative professionals. CPUs and GPUs were being measured in gaming or synthetic tests, while artists and editors experienced lag, slow exports, or choppy previews in real projects. We realized there was a clear gap between what the hardware spec sheets promised and what users actually felt at their desks. Puget Bench exists to close that gap and measure performance the way professionals experience it, through **tests that reflect real-world workflows**, not artificial ones.

At its core, the mission of Puget Bench is to provide clarity. End users want to understand how hardware will perform in the software and workflows they use every day. Vendors and developers need benchmarks that validate their tools in real-world scenarios. Reviewers and media partners require standardized, credible tests they can trust. Puget Bench serves all of these audiences while maintaining neutrality, transparency, and trust, providing a reliable standard for professional performance measurement.

Core goals:

- Reflect real-world performance in professional workflows.
- Provide clarity to end users and credibility to partners.
- Maintain neutrality and trust across the industry.

How We Decide What To Test

Deciding what tests to include in Puget Bench is both an art and a science. Our priorities are shaped by a combination of real-world feedback, hard data, and emerging industry trends. The goal is simple: focus on the workflows that matter most to professionals while keeping an eye on what's coming next - all within the limits of what can be reliably measured in the host application.

We begin with the voice of the user, since, in the end, that is the core audience our benchmarks are designed for. Sales conversations, support tickets, trade show discussions, surveys, and even social media reveal the day-to-day frustrations professionals face; whether it's Photoshop brush lag, sluggish Premiere Pro playback, or slow exports in DaVinci Resolve. These insights reveal what users care about most and where performance has a direct impact on productivity.

Telemetry and usage data, often provided by ISVs, give us an additional layer of insight. Which codecs are actually being used? Which effects are most common in real projects? This data helps ensure that our benchmarks reflect how people use the software every day, not just idealized scenarios.

Finally, we consider industry direction. Hardware vendors are driving the adoption of GPU acceleration and specialized encoders, while ISVs continually release new features that push workflows in different ways. Combining this foresight with our internal expertise—years of Puget Labs testing, workflow knowledge, and consulting—allows us to craft benchmarks that are both relevant today and forward-looking for tomorrow.

Inputs we use:

- **Customer/user feedback** - Direct reports of bottlenecks and frustrations
- **Telemetry & usage data** - Real-world trends provided by the ISV
- **Industry priorities** - ISV roadmaps and hardware vendor initiatives
- **Internal expertise** - Puget Labs testing and workflow knowledge

---

Methodology Principles

Our methodology is guided by a single idea: test workflows as close as possible to the way professionals actually work. Instead of chasing peak synthetic numbers or "best-case" scenarios, we focus on real-world relevance, consistency, and transparency. Each benchmark is designed to deliver results that are repeatable, comparable, and meaningful.

The goal is to measure performance directly within the application itself, rather than through simulations or synthetic mimics. This ensures that if a specific task performs better on one hardware vendor's platform or struggles on another, our benchmarks reflect that reality, providing users with actionable insights they can trust.

Key principles:

- **Workflow-Centric** - Tests mirror real creative and professional tasks, not artificial stress tests.
- **Consistency** - Standardized projects and repeatable methods ensure fair comparisons.
- **Transparency** - Public results and open methodologies build credibility and trust.
- **Neutrality** - Tests measure reality, not marketing narratives or vendor-optimized scenarios.
- **Evolving Design** - Benchmarks adapt as ISVs release new features and as workflows change.

By adhering to these principles, Puget Bench ensures results are both reliable for end users and respected across the industry as a neutral, trusted standard. Measuring performance directly in the application allows us to highlight real advantages and limitations of hardware, providing clarity that synthetic tests alone cannot.

Key challenges

Benchmarking real-world workflows comes with inherent challenges. At the heart of many of these challenges is a simple fact: the software we test is designed for human use, not automated measurement. As a result, not every workflow can be fully scripted, and not every performance issue is easy to capture consistently. Our goal is to strike a balance between what we want to test and what is technically feasible, while staying true to our mission of reflecting the actual user experience.

Primary challenges:

- **Automation Limitations** - Some interactive workloads, like Photoshop brush lag, are not currently possible to script.
- **Software Optimizations** - Certain tasks, such as Premiere Pro live playback, incorporate built-in optimizations that can compromise the reliability and consistency of automated testing.
- **Workflow Complexity** - Real projects involve countless combinations of tasks and assets, and it is impossible to test every possible combination.

- **Maintaining Neutrality** - We must ensure benchmarks remain independent and unbiased, even when working closely with partners or vendors.

- **Data Interpretation** - Benchmark results can reveal performance issues, but isolating the root cause requires careful analysis.

- **Evolving Software** - Application updates can change performance characteristics, and even what can (or cannot) be tested, requiring benchmarks to adapt quickly.

- **Expansive Plugin Ecosystem** - Many creative applications rely on third-party plugins, scripts, and utilities that extend functionality. These are often poorly optimized, lacking documentation, and are difficult to benchmark reliably.

Acknowledging these challenges informs our methodology and guides where we focus development effort..

Benchmark Breakdown

Puget Bench organizes testing by application to deliver results that are directly relevant to the workflows professionals use every day. Each benchmark is built around real-world tasks, not synthetic or artificial workloads, ensuring that our measurements reflect how hardware performs in practical scenarios.

It is impossible to test every combination of features, assets, or settings, so we prioritize workflows that are widely used, frequently reported as pain points, or likely to benefit from new hardware capabilities. For each application, we focus on what users care about most, how we can reliably measure it, and why it matters - helping both end users and industry partners understand performance in context.

In the sections that follow, we break down each application into the workflows and tasks important to users, noting whether they can be automated, how consistently they can be measured, and the level of demand for performance insights.

Photoshop

In Photoshop, the tasks that matter most to users are often those that involve active, interactive work — actions that take only a fraction of a second but are highly noticeable when lag occurs. These workflows are critical to the user experience but are also difficult to automate or measure reliably.

Newer AI-powered features have recently become a significant focus, with users eager to understand their performance impact. However, there is heavy confusion from users about what is processed on the cloud (where local system benchmarks are not relevant) and what is processed locally.

High-Importance, Interactive Tasks

- **General UI Responsiveness** - Quick actions like select, rotate, and move layers. Some can be automated via the Photoshop API, but not consistently.

- **AI Feature Performance** - Upscaling, denoise, and similar operations. While some features can be scripted, others, such as Neural Filters and Generative Fill, are either non-automatable or processed in the cloud, which limits local measurement.

- **Canvas Responsiveness** - Zoom, pan, and navigation. Currently, there is no reliable way to measure these interactions; even actions that can be triggered by the API lack methods to confirm when the canvas has fully refreshed.

- **Brush/Pen Tool Responsiveness** - Masking, liquify, and brush operations. A few tools (e.g., smudge) can be automated along a work path, but most brushes cannot, and brush lag remains unmeasurable.

Tasks that are less interactive, where the user waits for completion, tend to be easier to automate, even if they are less noticeable in day-to-day use.

Lower-Importance, Automatable Tasks

- **Export Performance** - Exporting project/files to JPEG, PNG, etc.

- **Filter Effect Performance** - Applying blurs, lens correction, sharpening, etc.

- **File Open/Save Time** - Saving and opening projects.

- **Application Startup Time** - Easy to measure, but can be heavily affected by installed plugins and background initialization tasks.

The current tasks in Photoshop that meet the criteria of being meaningful to users, and possible to be tested in an automated fashion are documented here.

Lightroom Classic

In Lightroom Classic, the tasks that matter most to users are those involving active interaction (such as navigating the library, selecting images, or applying edits), which often take only a

fraction of a second. These workflows are highly noticeable when lag occurs, but they are also difficult to automate or measure reliably. Performance can also vary significantly depending on the resolution and format of the photos, so testing is ideally done using high-resolution RAW files, which represent the majority of professional workflows.

High-Importance, Interactive Tasks

- **Library Responsiveness** - Scrolling through images, selecting multiple photos, and navigating folders. While many actions can be scripted, detecting when a photo is fully loaded is currently not possible. Differing settings for preview generation can also greatly affect the results.

- **Develop Module Adjustments** - Real-time edits like exposure, contrast, or color adjustments. Many adjustments can be automated via presets or API scripts, but most interactive sliders do not expose completion events for measurement.

- **Zoom and Pan in Loupe/Preview** - Navigating high-resolution images quickly. Currently, there is no reliable method to detect when a photo has fully rendered after zooming or panning.

- **Masking & Brush Tools** - Local adjustments using brushes or gradient filters. These tasks cannot be reliably automated, and there is no way to gather performance metrics.

- **AI Feature Performance** - Lightroom Classic has recently been integrating more and more AI features, many of which are highly valued by users. These can often take a significant amount of time to apply, however, especially when being done on a large number of photos.

Tasks that require the user to wait for completion tend to be easier to automate, even if they are less noticeable in daily use.

Lower-Importance, Automatable Tasks

- **Export Performance** - Exporting photos to JPEG, TIFF, or DNG formats.

- **Import Performance** - Importing large catalogs or folders of images.

- **Application Startup Time** - Launching Lightroom Classic can be measured, but results are influenced by catalog size, plugins, and background services.

In addition to the tasks themselves, the photograph resolution and format can have a significant impact on performance. The majority of Lightroom Classic users work natively with RAW photos from their camera.

Common Cameras and Image Formats

- Canon (.CR2 / .CR3)
- Nikon (.NEF)
- Sony (.ARW)
- Panasonic (.RW2)

The current tasks in Lightroom Classic (and image formats) that meet the criteria of being meaningful to users, and possible to be tested in an automated fashion, are documented here [COMING SOON!].

After Effects

In After Effects, the tasks that matter most to users are those that affect interactive performance, such as how smoothly the timeline and viewport respond during creative work. These workflows are highly noticeable when performance drops, as every frame is rendered rather than skipped to maintain a target FPS.

Testing After Effects can be challenging due to the innumerous combinations of layers, effects, and assets, each with vastly different performance characteristics.

High-Importance, Common Tasks

- **Viewport Preview FPS** - Real-time playback performance while previewing compositions. This is consistently rated as the top priority by users and is relatively straightforward to test in an automatic fashion. The primary challenge is that performance can vary depending on the user's display resolution and the size of the preview window.

- **Timeline Responsiveness** - Scrubbing, trimming, and playback delay. While scripting can simulate some edits, measuring how quickly the UI responds or playback begins is not currently possible.

- **Export / Render Speed** - Rendering compositions to final delivery formats (PNG, MOV, MP4, etc.). This is fully automatable and already a core part of the Puget Bench suite.

- **Effect & Transition Performance** - Applying and previewing GPU- or CPU-based effects. This can be measured via render tests, although there is added complexity when multiple effects are stacked on a layer, and multiple layers active in the composition.

These high-priority tasks can be accurately represented by export/render tests, as the primary performance consideration in After Effects is the raw time it takes to render each frame.

Lower-Importance, Less Common Tasks

- **Tracking & Rotoscoping** - These tasks can be automated, although they often require creative uses of API functions to do so. This is already a core part of the Puget Bench suite.

- **AI Feature Performance** - Rotobrush 3.0 (part of rotoscoping) is the only commonly used AI feature currently in After Effects. It can be automated, although, like many AI features, there is potential for different hardware to yield different results, making apples-to-apples comparisons more challenging.

- **Project Save / Load Time** - Measurable via scripting, though results depend on project complexity.

- **Application Startup Time** - Easy to measure, but can be heavily affected by installed plugins and background initialization tasks.

The current tasks in After Effects that meet the criteria of being meaningful to users and possible to be tested in an automated fashion are documented here.

Premiere Pro

In Premiere Pro, the workflows that matter most to users center on timeline performance and playback responsiveness—how smoothly the application responds during editing. These tasks have the greatest day-to-day impact on professional editors but are also the most challenging to measure consistently due to real-time optimizations applied by the software.

While Premiere Pro includes several AI-driven features, most locally processed tools (such as text-based editing, scene edit detection, and auto reframe) are used as more of a supplement to workflows and are not typically considered major performance bottlenecks by end users.

High-Importance, Interactive Tasks

- **Timeline Responsiveness** - Scrubbing, trimming, and playback delay. These tasks are key to editing fluidity but are currently unsupported by the Premiere Pro API. In some cases, they can be triggered, but there is no mechanism to measure performance.

- **Timeline Playback FPS** - Real-time playback performance with different codecs and effects. Frame rate can be logged indirectly by recording the number of dropped frames; however, playback FPS has incredibly high run-to-run variation due to background optimizations. These optimizations are great for end users to achieve the highest FPS possible, but they result in inconsistent performance that is not conducive to a performance benchmark.

- **Export / Render Speed** - Rendering sequences to delivery formats (H.264, HEVC, ProRes, etc.). Fully automatable and already a key focus of Puget Bench testing.

- **General UI Responsiveness** - Opening panels, switching sequences, and adjusting settings. These tasks impact the overall feel but are not possible to measure.

- **Effect & Transition Performance** - Applying GPU-accelerated effects and transitions during editing or playback. Can be tested by rendering/exporting specific timelines with the applied effects/transitions, although most common effects are so lightweight that meaningfully measuring performance requires configurations that are bordering on synthetic (for example, using 30x Lumetri Color effects).

Tasks that are less critical to perceived interactivity are generally easier to automate and provide consistent results across systems.

Lower-Importance, Automatable Tasks

- **Project Save / Load Time** - Straightforward to measure but highly dependent on media quantity and project size.

- **MultiCam Editing Performance** - Important for some workflows, but difficult to standardize due to back-end performance optimizations.

- **Proxy Generation** - Can be easily measured by performing a render/export with the appropriate source and destination codec/resolution/FPS. The main difficulty is the number of possible combinations of source/destination settings.

- **AI Features** - While Premiere Pro offers a range of AI features, few of them can be triggered or measured using the current API functions. In addition, they tend to have high levels of caching, which makes reliable and consistent benchmarking difficult.

- **Application Startup Time** - Measurable but affected by plugins and background services.

Current Premiere Pro testing focuses heavily on export performance, as it is consistent, reliable, and serves as a reasonable proxy for many other aspects of workflow. Expanding automation for timeline and playback responsiveness remains a major development goal, but would require significant investments by Adobe.

In addition to the tasks themselves, both the source and destination resolution/codec can have a significant impact on performance. The type of codec used varies depending on the type of user, with hobbyists, freelancers, and influencers tending to use highly compressed formats like

H.264/HEVC, while more professional-level users tend to use edit-friendly codecs like ProRes or work with RAW footage.

Common Source Formats

- H.264
- HEVC
- ProRes
- RAW (R3D, ARRIRAW, X-OCN, etc.)
- DNx

Common Export Formats

- H.264
- ProRes
- HEVC
- DNx

The current tasks in Premiere Pro (and source/destination codecs) that meet the criteria of being meaningful to users, and possible to be tested in an automated fashion are documented here.

DaVinci Resolve

DaVinci Resolve spans multiple creative domains, including editing, color grading, motion graphics, VFX, audio, and AI-driven workflows. As with other post-production tools, the tasks that matter most to users are those that affect real-time interactivity and playback responsiveness. These workflows have the greatest day-to-day impact, yet they are also the most challenging to test consistently due to the wide variety of project types, codecs, and effects used across different workflows.

Resolve provides fairly robust API and scripting support, but the main challenge is focusing tests on specific aspects of such a broad application, while preventing the benchmark from taking too long to run, and still reflecting meaningful real-world performance.

High-Importance, Interactive Tasks

- **Timeline Responsiveness** - Scrubbing, trimming, and playback delay. These tasks are key to editing fluidity but are currently unsupported by the DaVinci Resolve API. In some cases, they can be triggered, but there is no mechanism to measure performance.

- **Timeline Playback FPS** - Real-time playback performance with different codecs and effects. This cannot currently be triggered or measured by the API.

- **Export / Render Speed** - Rendering sequences to delivery formats (H.264, HEVC, ProRes, etc.). Fully automatable and already a key focus of Puget Bench testing.

- **Effect & Transition Performance** - Applying GPU-accelerated effects and transitions during editing or playback. Can be tested by rendering/exporting specific timelines with the applied effects/transitions, although most common effects are so lightweight that meaningfully measuring performance requires configurations that are bordering on synthetic (for example, using 30x Color nodes).

- **Fusion (VFX & Motion Graphics)** - Compositing, motion graphics, and visual effects. Certain Fusion tasks can be automated and benchmarked, but most often have to rely on the final render/export to measure performance.

- **AI Features** - DaVinci Resolve has a wealth of AI features, and nearly all of them can be triggered and measured with the current API functions. Caching can cause some problems, but generally can be accounted for.

Tasks that are less critical to perceived interactivity are generally easier to automate and provide consistent results across systems.

Lower-Importance, Automatable Tasks

- **Project Save / Load Time** - Straightforward to measure but highly dependent on media quantity and project size.

- **MultiCam Editing Performance** - Important for some workflows, but difficult to standardize due to back-end performance optimizations.

- **Proxy Generation** - Can be easily measured by performing a render/export with the appropriate source and destination codec/resolution/FPS. The main difficulty is the number of possible combinations of source/destination settings.

- **Fairlight (Audio)** - Multitrack audio playback and real-time processing. While audio export can be tested, interactive latency and playback stability are difficult to capture.

- **Application Startup Time** - Measurable but affected by plugins and background services.

Current DaVinci Resolve testing emphasizes export performance as it is highly consistent and reliable, as well as being a decent stand-in for many other aspects of users workflow. Expanding reliable testing for timeline and playback responsiveness remains a major development goal, but would require significant investments by Blackmagic.

In addition to the tasks themselves, both the source and destination resolution/codec can have a significant impact on performance. The type of codec used varies depending on the type of user, with hobbyists, freelancers, and influencers tending to use highly compressed formats like H.264/HEVC, while more professional-level users tend to use edit-friendly codecs like ProRes or work with RAW footage.

Common Source Formats

- H.264
- HEVC
- ProRes
- RAW (R3D, BRAW, ARRIRAW, X-OCN, etc.)
- DNx

Common Export Formats

- H.264
- ProRes
- HEVC
- DNx

Additional Considerations
DaVinci Resolve exists in two versions: the paid Studio version with full features, and the free version, which limits timeline resolution and access to OpenFX and AI tools. Benchmarks are designed to reflect performance in the Studio version, which represents the professional workflows our testing prioritizes.

The current tasks in DaVinci Resolve (and source/destination codecs) that meet the criteria of being meaningful to users, and possible to be tested in an automated fashion are documented here.