

# 6.1.0.1

18 Dec 2020

This is a bugfix release. For the full list of features and requirements see the [Arnold 6.1.0.0 release notes](#).

## Enhancements

- **Microsoft Azure CA update:** Updated Azure intermediate CAs for the US and Chinese sovereign clouds as of December 2<sup>nd</sup>, 2020.

## System Requirements

- Windows 7 or later, with the Visual Studio 2019 redistributable.
- Linux with at least glibc 2.12 and libstdc++ 3.4.13 (gcc 4.4.7). This is equivalent to RHEL/CentOS 6.
- OSX 10.11 to 10.15. Note that macOS 11 Big Sur is not certified.
- CPUs need to support the SSE4.1 instruction set.
- GPU rendering works on Windows and Linux only and requires an NVIDIA GPU of the Turing, Volta, Pascal, or Maxwell architecture. We recommend using the [455.38](#) or higher drivers on Linux and [457.09 \(Quadro\)](#), [457.30 \(GeForce\)](#), or higher on Windows. See [Getting Started with Arnold GPU](#) for more information.
- Optix™ denoiser requires an NVidia GPU with [CUDA™ Compute Capability 5.0](#) and above.

## Bug fixes

- [core#10125](#) Bad chromaticity detection when linear and narrow color space are the same
- [core#9979](#) Crash when missing config in color\_manager\_ocio
- [core#10140](#) Crash with quad\_light and IPR
- [core#10050 \[GPU\]](#) Crash in displacement in when using tracing shaders in the scene
- [core#10128 \[GPU\]](#) Curvature shader slows down render
- [core#10060 \[GPU\]](#) Improve handling of stack overflow exceptions
- [core#10118 \[GPU\]](#) Improve NVML detection
- [core#10196 \[GPU\]](#) Light disappears during GPU IPR after modification
- [core#9634 \[GPU\]](#) OSL texture lookup crashing on a subsequent renders
- [core#10163 \[GPU\]](#) Single channel float texture artifacts
- [core#10231 \[GPU\]](#) Texture and exception handling errors using multiple GPUs without NVLink
- [core#10146](#) Hang in crash handler
- [core#10165 \[OSL\]](#) Artifacts when using randomwalk\_bssrdf in OSL
- [core#10159](#) Random crash in subdivision after render interruption
- [core#10124](#) Surfaces with transmission depth can disappear with nested dielectrics on CPU
- [usd#592](#) Invalid face-varying primvars crash the render delegate
- [usd#596](#) Invalid USD is produced if polymesh is made of triangles and nsides is empty
- [usd#481](#) std::string, TfToken, and SdfAssetPath typed VtArrays are not converted when setting primvars