

4.0.4.0

Milestone 4.0.4

Enhancements

- **EXR zips compression support:** The compression parameter in the driver_exr node now accepts a zips (single-scanline ZIP) value. zips is a more efficient compression format for Nuke. (#2770)
- **Optimized scenes with many nodes:** A few optimizations have made handling of scenes with millions of nodes much faster. In particular, node creation (specially instances), node destruction, and iteration over the nodes in general has been optimized. The memory footprint has also been reduced for these scenes. (#2725, #2778)
- **Reuse main thread for rendering:** The main Arnold thread is now also used for rendering buckets, so instead of firing N+1 threads, we now fire N threads. In some specific situations, such as with older versions of Linux, this can result in consistently faster threading scalability. (#2761)
- **Added spherical_camera node:** A new camera type spherical_camera has been added to help create environment maps in the lat-long format easily. To get the full spherical range, the camera's screen window must be set to [-1,-1] to [1,1]. Note that the same mapping could be achieved in the cyl_camera with careful tweaking of the horizontal_fov, vertical_fov and projective parameters, but this was too convoluted to be practical. (#2785)
- **Support for multiple license servers:** We now allow for the license check to span multiple license servers. For this we added support for new environment variables so, in addition to the existing Arnold-specific ARNOLD_LICENSE_HOST and ARNOLD_LICENSE_PORT, we now also recognize the RLM-standard variables RLM_LICENSE and solidangle_LICENSE. Licenses will be searched for in the following order: (#2691)
 - First, from the contents of solidangle_LICENSE, which can contain a list of multiple servers/ports in the following format:

```
server1:server2:server3:...:serverN <--- linux/osx
server1;server2;server3;...;serverN <--- windows
```

where serverN can be port@host or host@port (the port is optional and defaults to 5053).
 - Second, from the contents of RLM_LICENSE, with the same format as above.
 - Finally, from the contents of ARNOLD_LICENSE_HOST/ARNOLD_LICENSE_PORT, which together define a single license server target. (it wasn't possible to specify multiple servers in these variables and it's still not possible - that is why we have added support for the above variables)
- **Upgraded OIIO to 0.10.7:** There have been several bugfixes and performance improvements since the last OIIO version we shipped with Arnold (which was 0.10.4) including: (#2766)
 - Threading performance improvement in the texture system, which should help when there are many texture file opens per second.
 - Fix for degenerate derivatives that could corrupt the filter footprint calculations in anisotropic filtering, resulting in an infinitely long major axis.
 - More efficient subpixel filtering for very narrow anisotropic footprints when on the highest-res MIP level.

API additions

- **AiLicenseInfo():** This API function has been added as a replacement for the older AiLicenseGetInfo() as we now get license information from all available license servers at the same time. (#2691)

```
AI_API int AiLicenseInfo(std::vector<AtLicenseInfo>& licenses);
```

Incompatible changes

- **Color scheme in utility shader has changed:** This affects color modes obj, prims and floatgrid. The hashing of values to RGB colors has been modified to avoid producing black or very dark colors. In particular, the object and primitive with index 0 would previously always map to black. (#2773)
- **Deprecated AiLicenseGetInfo():** This older function has changed behaviour in that it will now list license information for all license servers found, including those in the new environment variables mentioned above, in addition to the license server set via the host and port arguments. These function arguments are therefore useless and that's why this API has been deprecated, and a new AiLicenseInfo() API that has no extra arguments has been added instead. (#2691)

```
AI_API int AiLicenseGetInfo(const char* host, unsigned int port, std::vector<AtLicenseInfo>& licenses);
```

Bug fixes

Ticket	Summary	Component	Owner	Priority	Version	Created
#2767	parallel BVH build hangs with degenerate triangles	arnold	thiago	major	3.3	11 days