OpenEXR Project Update

April 17, 2024

OpenEXR Project Mission

The goal of the OpenEXR project is to keep the EXR format reliable and modern and to maintain its place as the preferred image format for entertainment content creation.

Major revisions are infrequent, and new features will be carefully weighed against increased complexity. The principal priorities of the project are:

- Robustness, reliability, security
- Backwards compatibility, data longevity
- Performance - read/write/compression/decompression time
- Simplicity, ease of use, maintainability
- Wide adoption, multi-platform support - Linux, Windows, macOS, and others

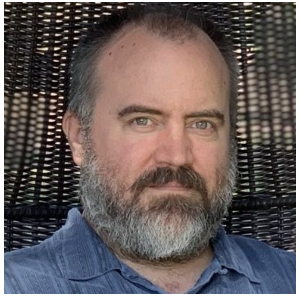
OpenEXR is intended solely for 2D data. It is not appropriate for storage of volumetric data, cached or lit 3D scenes, or more complex 3D data such as light fields.

Technical Steering Committee

Cary Phillips
Industrial Light & Magic

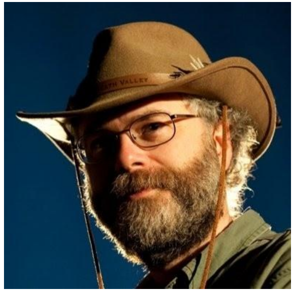
Christina Tempelaar-Lietz
Industrial Light & Magic

Joseph Goldstone
ARRI, Inc



Kimball Thurston
Wētā FX

Larry Gritz
Sony Pictures Imageworks

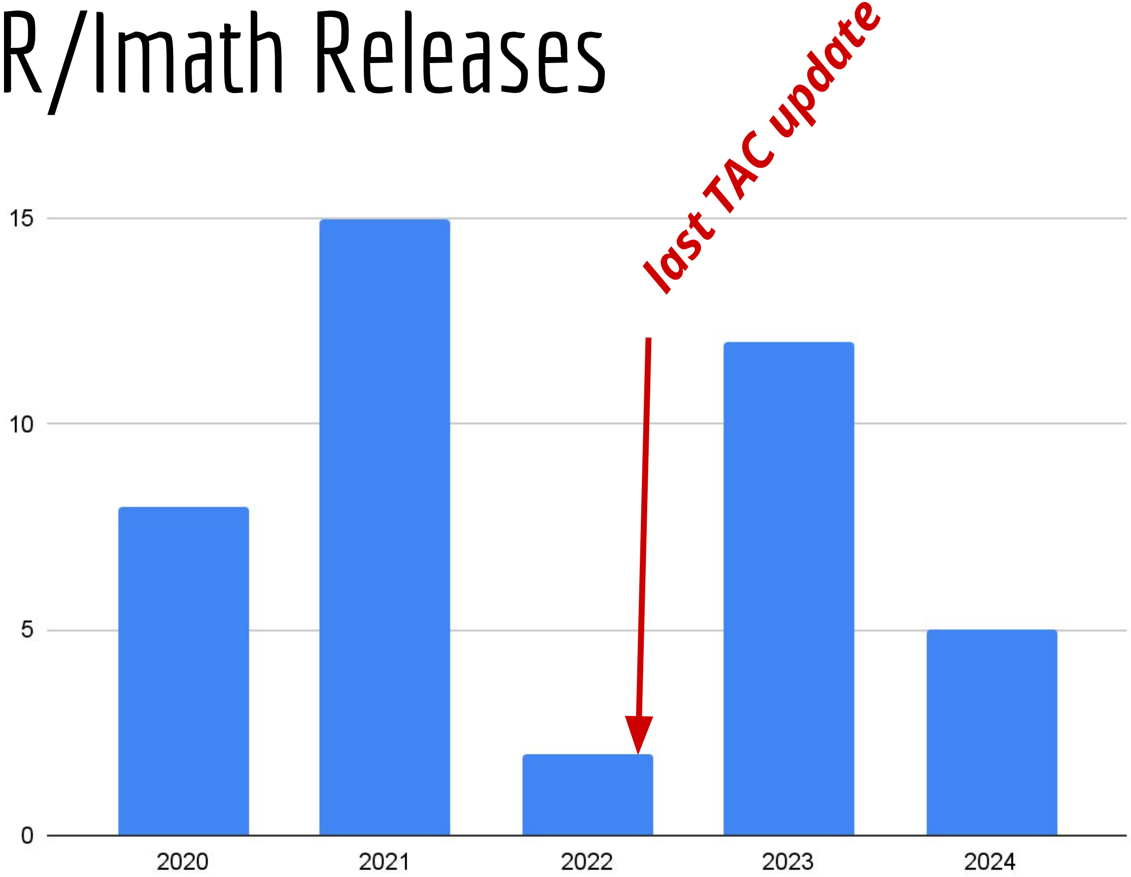


Nick Porcino
Pixar Animation Studios

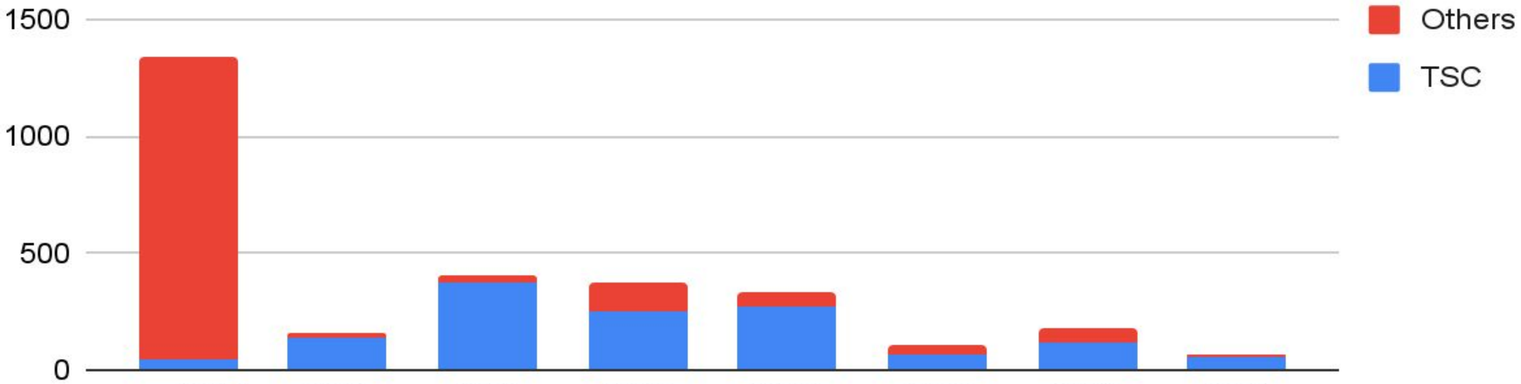
Peter Hillman
Wētā FX

Rod Bogart
Epic Games

OpenEXR/Imath Releases

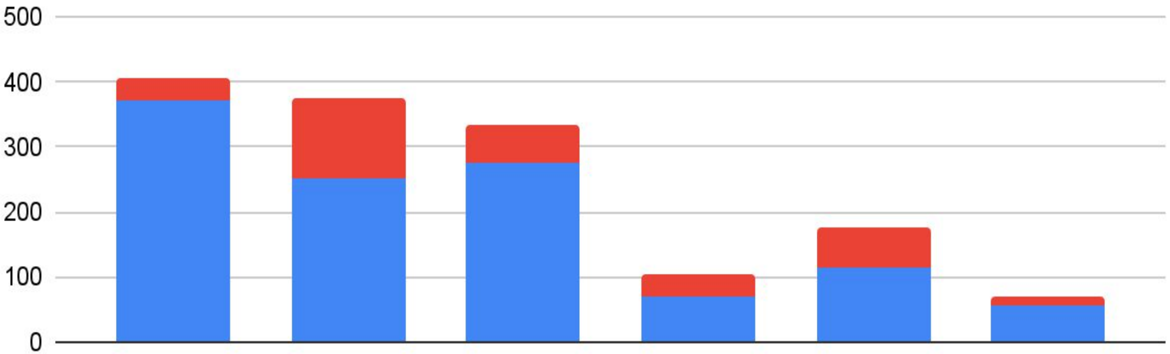


Commits: TSC Members vs. Others



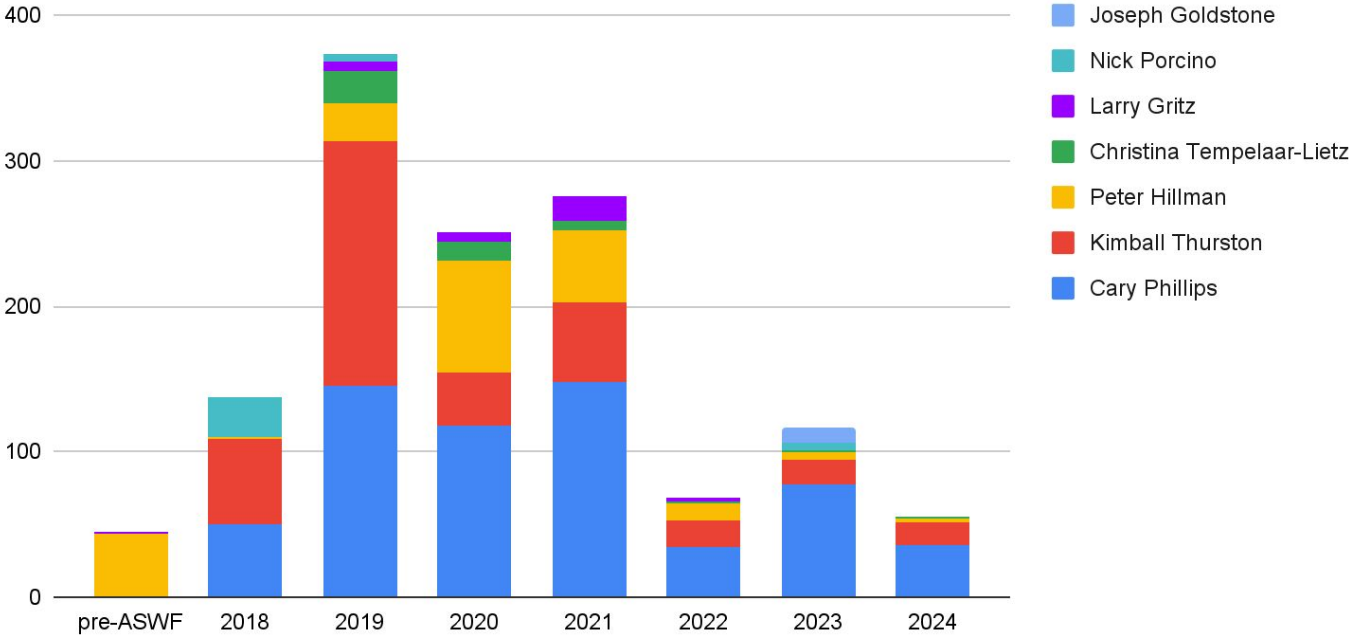
	pre-ASWF	**2018**	**2019**	**2020**	**2021**	**2022**	2023	2024
TSC	45	137	373	251	276	69	116	55
Others	1299	19	34	124	58	35	61	15
Total	1344	156	407	375	334	104	177	70
% Others	**97%**	***12%***	***8%***	***33%***	***17%***	***34%***	***34%***	***21%***

Commits: TSC Members vs. Others



	2019	**2020**	**2021**	**2022**	2023	2024
TSC	373	251	276	69	116	55
Others	34	124	58	35	61	15
Total	407	375	334	104	177	70
% Others	8%	33%	17%	34%	34%	21%
# Others	15	22	23	15	25	11

Commits: TSC Members



Roadmap: 2023

- ~~ABI compatibility: C-level base types~~ *minimal progress*
- ~~Core/Multithreading in the C++ API~~ *WIP*
- Website overhaul *done!*

2023: Improvements/New Functionality:

- Bug/security/build fixes
- Documentation/website improvements
- Groundwork: compression via `libdeflate`
- OpenSSF Badge: Silver 96%, Gold 74%
- `pip install openexr`

OpenSSF Badge

- Silver: 96%
- Gold: 74%

Outstanding…

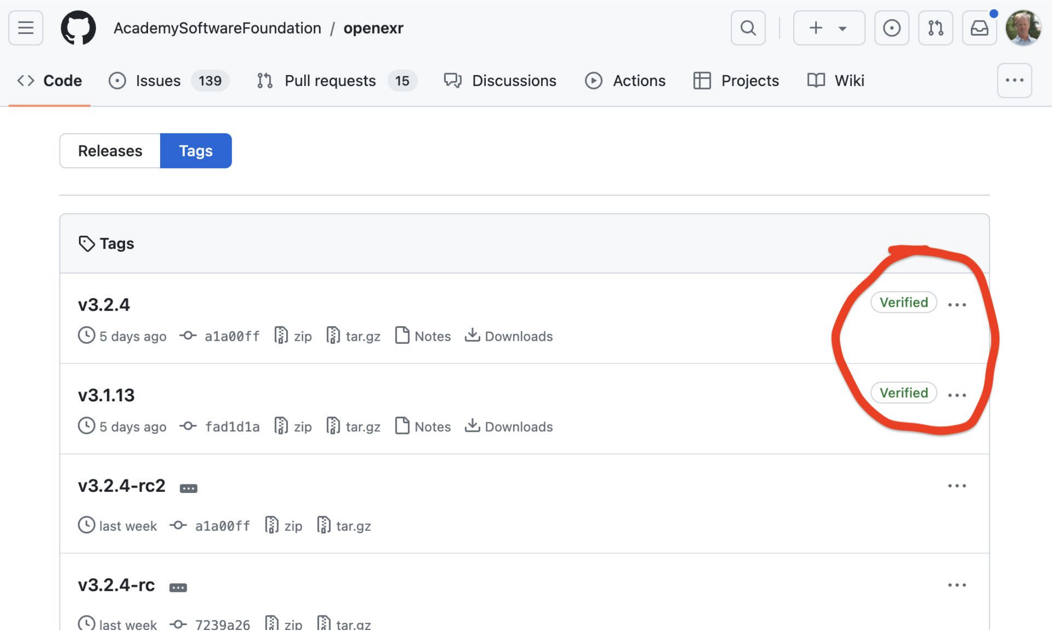
- Test coverage:
  - Currently at ~80%
- Security
  - “Secure design principles”
  - Assurance case
  - Security review
  - MITM attacks
- Reproducible build

Security

- Policy statement, CVE reporting
- OSS-fuzz
- Signed releases/verified tags
- Snyk scan
- OpenSSF Scorecard

Signed/verified tags:

```
% git tag -s v3.2.4
```



Tags

v3.2.4 Verified
5 days ago a1a00ff zip tar.gz Notes Downloads

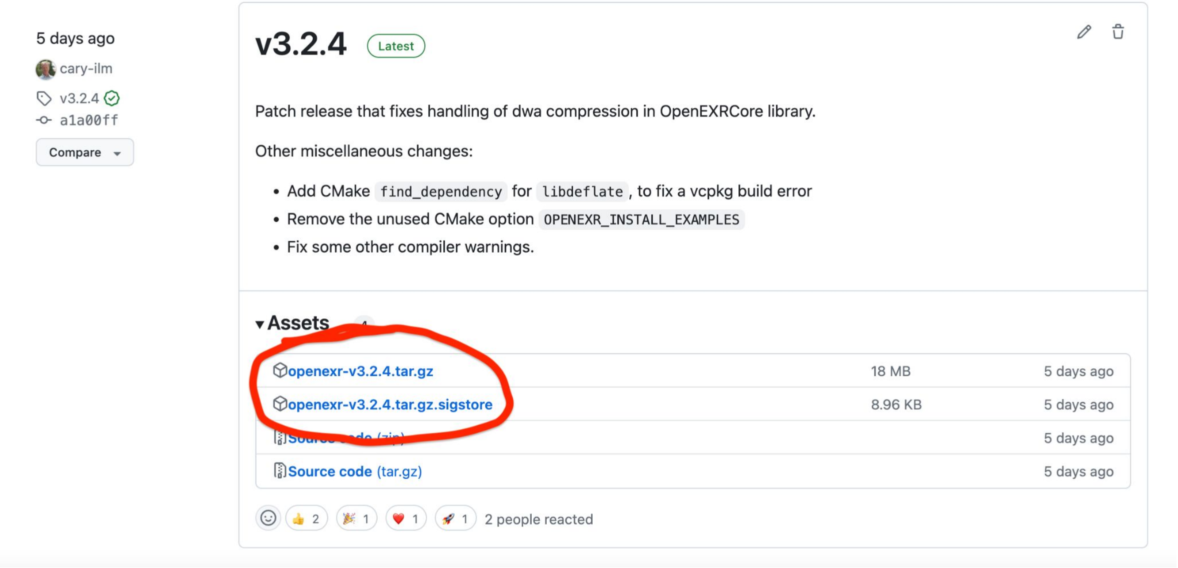
v3.1.13 Verified
5 days ago fad1d1a zip tar.gz Notes Downloads

v3.2.4-rc2
last week a1a00ff zip tar.gz

v3.2.4-rc
last week 7239a26 zip tar.gz

Signed releases

`.github/workflows/release-sign.yml`



5 days ago

cary-ilm

v3.2.4 ✓

a1a00ff

Compare ▾

v3.2.4 Latest

Patch release that fixes handling of dwa compression in OpenEXRCore library.

Other miscellaneous changes:

- Add CMake `find_dependency` for `libdeflate`, to fix a vcpkg build error
- Remove the unused CMake option `OPENEXR_INSTALL_EXAMPLES`
- Fix some other compiler warnings.

▾ Assets 4

openexr-v3.2.4.tar.gz	18 MB	5 days ago
openexr-v3.2.4.tar.gz.sigstore	8.96 KB	5 days ago
Source code (zip)		5 days ago
Source code (tar.gz)		5 days ago

👍 2 🎉 1 ❤️ 1 🚀 1 2 people reacted

Documentation Improvements:

- New-ish website (sphinx, readthedocs)
    - [now builds on windows/macOS...]
- Standard Attributes
- Scene-linear

Coming soon.. (hopefully):

- Compression via zstd
- C++ interface to OpenEXRCore
- Rewrite OpenEXR python bindings in pybind11
- Finish Imath port to Pybind11

Project Weaknesses/Needs…

- Windows support
- Hardware-we-don’t-have support
- Widening the contributor community
- Transition planning

EOF