

Proposal for more frequent Rust Arrow release process

May 1, 2021

Summary

This document proposes regularly releasing minor and patch releases of the three crates (arrow, parquet, and arrow-flight) in the [arrow-rs repository](#), following the [semver](#) versioning scheme used by the Rust ecosystem.

Problem Statement

Many of the projects¹ that use the Rust implementation of Arrow do not use the published versions of arrow crate in [crates.io](#) (the main rust ecosystem package manager) but instead directly refer to the git sha (or fork the project and use that fork). Releases of Arrow are currently in lockstep with other languages and happen every 4 months.

This results in the following problems:

1. If an user needs a bug fix, they have to pull all changes in from master (including potentially breaking changes)
2. The number of changes (both breaking and bug fixes) between major releases is substantial (e.g. see the [Rust feature list](#) in 4.0)
3. Users of arrow-rs and related crates do not automatically get bug fixes / new compatible features via `cargo update` and have to manually update git shas regularly
4. Users of dependent crates such as DataFusion must also use the exact git version of Arrow that DataFusion uses
5. Users of arrow-rs via Git references cannot publish their crates to crates.io

Not covered

Note this document does not cover the projects in the arrow-datafusion repo, only the crates in arrow-rs. The release schedules of those projects would also likely benefit from a difference cadence, and doing so will be helped by a regular arrow release.

Proposal

- Aim for a minor or patch release releases every two weeks
- Keep the same major release version cadence as the main arrow project (once every 3 months, in lockstep with the rest of the arrow language implementation)

Proposed Process

- Every other Monday, a maintainer would
 - Propose a minor (e.g. 4.1.0 to 4.2.0) or patch (e.g 4.1.0 to 4.1.1) release, depending on changes to the repo in the previous 2 weeks.
 - Create a git tag / branch in the arrow-rs repo at the proposed sha, updating the version numbers on that tag/branch appropriately.

¹ For example, DataFusion ([link](#)), and IOx ([link](#))

- A proposal for release would be sent to the dev@arrow.apache.org mailing list, with a link to the git tag, for a vote
- If approved, a new version from that tag would be released to crates.io later in the week
- During the 2 weeks leading up to the main quarterly Arrow release, arrow-rs would instead prepare a Rust release with the corresponding major release version.

FAQ

Q: Would arrow-rs have maintenance branches?

A: We would not plan for maintenance branches at this time, all development work would continue directly on master.

Q: How would we handle breaking changes to the arrow crate?

A: We would wait with such breaking changes until right before each quarterly major release. During the minor releases period we'd advise the PR contributors to put their changes behind feature flags or add alternative experimental APIs instead of changing the existing interface - enabling the co-dependence on such PRs and enabling the experiments to a wider audience. Ideally only changes to the defaults would happen right before the quarterly major release - avoiding the extra load on the maintainers and lowering the risk caused by the unusually high volume of code merged.

Q: Would we always release a new major version of arrow-rs even if there were no breaking API changes?

A: Yes. We would initially do so to signal the Rust library's alignment to the rest of the Arrow implementations. We can revisit this policy in the future after we have some experience with a faster release cadence.

Mailing List Discussion Cross Link:

<https://lists.apache.org/thread.html/r6b9baf59e3cd1a91905b5f802057026dfa627f00507638b605a3ff1b%40%3Cdev.arrow.apache.org%3E>