

# Apache Arrow Home

This page is the Apache Arrow developer wiki. If you are involved in building or maintaining the project, this is a good page to have bookmarked. If you are a prospective user of the project, check out user-facing library and API documentation linked to from <http://arrow.apache.org/>.

- [Developer Resources](#)
  - [Language-specific Development Resources](#)
- [Roadmap and Initiatives](#)
  - [Columnar Format](#)
    - [Columnar Computational Libraries](#)
  - [C++ Libraries](#)
  - [C GLib Libraries](#)
  - [Feather File Format](#)
  - [GPU Support](#)
  - [Go Libraries](#)
  - [Java Libraries](#)
  - [JavaScript Libraries](#)
  - [Julia Libraries](#)
  - [Machine Learning Framework Integrations](#)
  - [Parquet File Support](#)
  - [Plasma Shared Memory Store](#)
  - [Python Libraries](#)
  - [RPC System \(Arrow Flight\)](#)
  - [R Libraries](#)
  - [Ruby Libraries](#)
  - [Rust Libraries](#)
- [Release Planning](#)
  - [Top-level Releases](#)
  - [JavaScript Releases](#)
- [Future Work and Ideas](#)
  - [Developing an open standard for in-memory records](#)

## Developer Resources

- [Guide for Committers and Project Maintainers](#)
- [Guide for Contributors](#)
- [Release Management Guide](#)
- [Packaging and Task Automation Tools](#)
- [How to Verify Release Candidates](#)
- [Open Patches \(Pull Requests\)](#)
- [JIRA Health Dashboard](#)

## Language-specific Development Resources

- [Guide for C++ Developers](#)

## Roadmap and Initiatives

### Columnar Format

[Columnar Format JIRA Dashboard](#)

The "Arrow columnar format" is an open standard, language-independent binary in-memory format for columnar datasets. It can be used to create data frame libraries, build analytical query engines, and address many other use cases.

- [Columnar Format 1.0 Milestone](#)
- [Compression Support in Binary Protocol](#)

## Columnar Computational Libraries

### C++ Libraries

[C++ JIRA Dashboard](#)

**Projects:**

- [C++ CSV / Delimited File Reader Project](#)
- [C++ Arrow-optimized Database Clients](#)
- [C++ Analytic Functions](#)
- [HDFS Filesystem Support](#)

## C GLib Libraries

[C GLib JIRA Dashboard](#)

## Feather File Format

- [Feather File Format V2](#)

## GPU Support

[GPU JIRA Dashboard](#)

## Go Libraries

[Go JIRA Dashboard](#)

## Java Libraries

[Java JIRA Dashboard](#)

## JavaScript Libraries

[JavaScript JIRA Dashboard](#)

## Julia Libraries

We have been discussing involving the Julia community in Apache Arrow

- Discussion on ExpandingMan/Arrow.jl <https://github.com/ExpandingMan/Arrow.jl/issues/28>
- Feather implementation in pure Julia: <https://github.com/JuliaData/Feather.jl>

## Machine Learning Framework Integrations

Modern machine learning frameworks can leverage technologies we are developing in Apache Arrow, and vice versa.

- [PyTorch integrations](#)
- [TensorFlow integrations](#)

## Parquet File Support

- [C++ Parquet Development](#)
- [Python Parquet Development](#)

## Plasma Shared Memory Store

[Plasma JIRA Dashboard](#)

## Python Libraries

[Python JIRA Dashboard](#)

### Projects:

- [Python Filesystems and Filesystem API](#)
- [Python Parquet Format Support](#)

## RPC System (Arrow Flight)

- Jacques's [initial proposal as pull request](#)

- [GitHub issue for GRPC Protobuf Performance issues in Java](#)

## R Libraries

[R JIRA Dashboard](#)

## Ruby Libraries

[Ruby JIRA Dashboard](#)

## Rust Libraries

[Rust JIRA Dashboard](#)

---

## Release Planning

### Top-level Releases

- [Arrow 0.15.0 Release](#) (next)
- [Arrow 0.14.0 Release](#)
- [Arrow 0.13.0 Release](#) (DONE)
- [Arrow 0.12.1 Release](#) (DONE)
- [Arrow 0.12.0 Release](#) (DONE)
- [Arrow 0.11.0 Release](#) (DONE)
- [Arrow 0.10.0 Release](#) (DONE)

### JavaScript Releases

## Future Work and Ideas

Developing an open standard for in-memory records

We began to [discuss this on the mailing list](#).