parquet-cpp dev roadmap

[Release 0.1 scope and timeline](#)
[Release 0.2 scope and timeline](#)
[Process and goals](#)
[Build and development tooling](#)
[Thrift metadata](#)
[Parquet file structure read](#)
[Parquet file structure write](#)
[Primitive column read / write](#)
[Decimal support](#)
[Nested data reassembly](#)
[Debugging tools](#)
[Schema analysis and resolution](#)
[IO](#)
[Multithreading and pipelining](#)
[Performance optimization miscellanea](#)
[File scanning / Predicate pushdown](#)
[Hardware / data structure matters](#)
[Integration testing](#)
[Enabling parquet-cpp to become a dependency of Apache Impala (incubating)](#)

Release 0.1 scope and timeline

Target date: 3/1/2016

The high level goal for the first release is a 90% functional (e.g. no Parquet 2.0 features) Parquet file reader.

- Documentation
  - Markdown document for users getting started
- Column reading / scanning business logic
  - **DONE** PARQUET-549: Unit testing for dictionary-encoded data
  - **DONE** PARQUET-545: DECIMAL logical type metadata support
  - **DONE** PARQUET-494: Implement PLAIN_DICTIONARY encoding
  - **DONE** PARQUET-502 (BUG): Scanner segfaults on FIXED_LEN_BYTE_ARRAY data
  - **DONE** PARQUET-526: Scanner test coverage / user API solidification
  - **DONE** PARQUET-538: Enhance ColumnReader tests
  - **DONE** PARQUET-499: Complete PLAIN encoding
  - **WONT FIX** PARQUET-524: More intuitive batch reads with batches spanning multiple data pages

- Bugs, robustness, and memory management
    - **DONE** PARQUET-537: Test that LocalFileSource cleans up after itself
    - **DONE** PARQUET-520: Zero-copy input streams
    - **DONE** PARQUET-516: More robust handling of LocalFileSource FS interactions
    - **DONE** PARQUET-533: Combine Read/Seek operations
    - **DONE** PARQUET-525: Handling of malformed files
- Code cleaning and library tooling
    - **DONE** PARQUET-482: Organize and document the public API for 3rd-party users
    - **DONE** PARQUET-519: Compiler warning scrubbing
    - **DONE** PARQUET-463: Add DCHECK macros for debug-build assertions
    - **DONE** PARQUET-518: unsigned integer scrubbing
    - **DONE** PARQUET-514: automated code coverage reports
    - **DONE** PARQUET-468: Use installed Thrift to generate C++ bindings
    - **DONE** PARQUET-471: Review Travis CI / sandbox build environment setup
    - **DONE** PARQUET-466: parquet.thrift management plan
- Deserialization
    - **DONE** PARQUET-531 (BUG): Unable to read gzipped file created by Spark
    - **DONE** PARQUET-457: Deserializing compressed data / dictionary pages
    - **DONE** PARQUET-523: Support/test reading column chunks with multiple data pages

Release 0.2 scope and timeline

Target date: 4/1/2016

The high level goal for the 0.2 release is to offer 90% functionally-complete file writer support.

- PARQUET-530: Add support for LZO compression with GPL workarounds

parquet-2.0 format support

- PARQUET-458: Implement DataPageV2 (This is not enabled by default in parquet-mr)
- Stretch goal: complete Parquet 2.0 value encodings

Start path towards predicate pushdown

- PARQUET-473: Develop external predicate pushdown API for column readers

Process and goals

The goal of this document is to organize JIRAs into high level buckets of functional components and build/development tools to help steer our efforts.

Testing falls into two buckets

- Unit tests

- Integration tests

The number of unit tests will (should) vastly exceed the number of integration tests. To facilitate unit testing, we must develop functional components in a way that is designed for testing [1].

Integration testing (especially between parquet-mr and parquet-cpp) will be a focus when the library is more mature.

Build and development tooling

- Static and dynamic linking build options (for libparquet and its dependencies)
  - https://issues.apache.org/jira/browse/PARQUET-487
- Symbol visibility
  - https://issues.apache.org/jira/browse/PARQUET-489
- DONE Organize a public API so that “internal” files can be refactored and moved without causing API breakages
  - https://issues.apache.org/jira/browse/PARQUET-482
- DONE Start performance testing suite
  - https://issues.apache.org/jira/browse/PARQUET-512
- DONE clang-tidy https://issues.apache.org/jira/browse/PARQUET-477
- DONE Unsigned integer scrubbing
  - https://issues.apache.org/jira/browse/PARQUET-518
- DONE Compiler warning scrubbing
  - https://issues.apache.org/jira/browse/PARQUET-519
- DONE Add glog to build toolchain for DCHECK* macros and other development tools
  - https://issues.apache.org/jira/browse/PARQUET-463
  - Note that these must not leak to the public API
- DONE Create publicly-viewable code coverage of test suite
  - https://issues.apache.org/jira/browse/PARQUET-514
  - https://issues.apache.org/jira/browse/PARQUET-486
  - https://issues.apache.org/jira/browse/PARQUET-500
- DONE Thrift >= 0.9.2 build problems
  - https://issues.apache.org/jira/browse/PARQUET-470
- DONE Use common sandbox library toolchain in Travis CI and out-of-the-box local development
  - https://issues.apache.org/jira/browse/PARQUET-471
- DONE update cpplint https://issues.apache.org/jira/browse/PARQUET-496
- DONE include-what-you-use
  - https://issues.apache.org/jira/browse/PARQUET-522
- DONE Prevent build dependencies from being transitively included in public headers
  - Thrift https://issues.apache.org/jira/browse/PARQUET-446

[1] For a great reference on this subject, see:
http://www.amazon.com/Clean-Code-Handbook-Software-Craftsmanship/dp/0132350882

    - Boost https://issues.apache.org/jira/browse/PARQUET-277
- DONE: Convert to C++11 and make into a shared library that can be linked
    - https://issues.apache.org/jira/browse/PARQUET-416
- DONE Debug, FastDebug, and Release cmake build types to manage compiler flags
    - https://issues.apache.org/jira/browse/PARQUET-447
    - See, e.g. CMakeLists.txt in Apache Kudu
- DONE Add Zlib to build toolchain
    - https://issues.apache.org/jira/browse/PARQUET-456
- DONE Add option to skip building unit tests
    - https://issues.apache.org/jira/browse/PARQUET-448
- DONE Compiler warnings on OS X
    - https://issues.apache.org/jira/browse/PARQUET-455

Thrift metadata

- DONE Handling of syncing parquet.thrift with upstream apache/parquet-format
    - https://issues.apache.org/jira/browse/PARQUET-466
    - WM: I feel we should manually sync Parquet thrift for now
- DONE Auto-generate Thrift bindings for installed version of Thrift
    - https://issues.apache.org/jira/browse/PARQUET-468

Parquet file structure read

- DONE Malformed file failure modes
    - https://issues.apache.org/jira/browse/PARQUET-525
- DONE Engineering ParquetFileReader to be testable
  https://issues.apache.org/jira/browse/PARQUET-497

Parquet file structure write

- DONE Add a ParquetFileWriter public API
  https://issues.apache.org/jira/browse/PARQUET-436
- DONE Add a public API for building new RowGroups to writing to a file
    - https://issues.apache.org/jira/browse/PARQUET-452
- DONE Add a serialized ColumnChunk build class, as a decoupled companion to the RowGroupBuilder https://issues.apache.org/jira/browse/PARQUET-498

Primitive column read / write

- Failure modes for malformed data pages:
    - https://issues.apache.org/jira/browse/PARQUET-483
- Parquet 2.0 value encodings
    - DELTA_BINARY_PACKED: https://issues.apache.org/jira/browse/PARQUET-490

    - DELTA_LENGTH_BYTE_ARRAY https://issues.apache.org/jira/browse/PARQUET-491
    - DELTA_BYTE_ARRAY https://issues.apache.org/jira/browse/PARQUET-492
- Support DataPageV2
    - https://issues.apache.org/jira/browse/PARQUET-458
- DONE Test that SerializedPageReader can handle compressed data pages
    - https://issues.apache.org/jira/browse/PARQUET-457
- WONT FIX Batch reading should keep reading data pages until batch size reached (or EOS)
    - https://issues.apache.org/jira/browse/PARQUET-524
- DONE Scanner test coverage
    - https://issues.apache.org/jira/browse/PARQUET-526
- DONE Boolean batch read API
    - Currently most likely broken because of bad behavior of bool*
- DONE Dictionary encoding
    - Encoding::RLE_DICTIONARY
        - Implement encoder: https://issues.apache.org/jira/browse/PARQUET-493
    - Encoding::PLAIN_DICTIONARY
        - Implement encoder and decoder https://issues.apache.org/jira/browse/PARQUET-494
- DONE Plain encoding (Encoding::PLAIN)
    - Complete encoder implementation, add tests for both encoding and decoding https://issues.apache.org/jira/browse/PARQUET-499
- DONE Test multiple data pages
    - https://issues.apache.org/jira/browse/PARQUET-523
- DONE Review and improve public ColumnReader API
    - https://issues.apache.org/jira/browse/PARQUET-461
- DONE Unit test fixtures and tooling: https://issues.apache.org/jira/browse/PARQUET-485
- DONE OutputStream API for encoders
- DONE Verify / fix / test boolean encoding/decoding https://issues.apache.org/jira/browse/PARQUET-454
- DONE Decode repetition levels (if the schema requires them): https://issues.apache.org/jira/browse/PARQUET-169
- DONE Add a LevelDecoder and LevelEncoder abstraction that can batch-encode or batch-decode levels in either RLE or BIT_PACKED format
    - https://issues.apache.org/jira/browse/PARQUET-462
    - https://issues.apache.org/jira/browse/PARQUET-467
- DONE Synchronize RLE encoder/decoder code and tests with Impala https://issues.apache.org/jira/browse/PARQUET-438

Decimal support

Decimals are stored in a FIXED_LEN_BYTE_ARRAY.

Nested data reassembly

- Nested structure reassembly for columnar data structures
    - https://issues.apache.org/jira/browse/PARQUET-478
    - https://issues.apache.org/jira/browse/PARQUET-445

Debugging tools

- Dump primitive column data https://issues.apache.org/jira/browse/PARQUET-476
- JSON output
    - https://issues.apache.org/jira/browse/PARQUET-481

Schema analysis and resolution

- Handling different forms of array encodings https://issues.apache.org/jira/browse/PARQUET-441
- Map schema encodings: https://issues.apache.org/jira/browse/PARQUET-443
- Implement a public API to build logical schemas: https://issues.apache.org/jira/browse/PARQUET-444
- Parse text schemas like parquet-mr
    - https://issues.apache.org/jira/browse/PARQUET-510
- DONE Converting vector of SchemaElement into a logical schema (necessary to know what will be in a primitive column's data pages): https://issues.apache.org/jira/browse/PARQUET-442

IO

- Memory management improvement, generally
- DONE Local File IO
    - https://issues.apache.org/jira/browse/PARQUET-516
- DONE Zero-copy input stream APIs
    - https://issues.apache.org/jira/browse/PARQUET-520

Multithreading and pipelining

- Make InputStream threadsafe
    - https://issues.apache.org/jira/browse/PARQUET-474
- Address DataPage buffer ownership semantics in a pipeline context

Performance optimization miscellanea

- Push down vectorization into RleEncoder and RleDecoder

- Vectorizing BitReader / BitWriter internally
- Dictionary encoding / decoding performance benchmarks
- Hash algorithms for ByteArray and FixedLenByteArray

File scanning / Predicate pushdown

- Start thinking about a basic predicate pushdown API
  https://issues.apache.org/jira/browse/PARQUET-473

Hardware / data structure matters

- SSE / SIMD build options
  - https://issues.apache.org/jira/browse/PARQUET-488
- Struct-packing options for Int96 and friends
  - https://issues.apache.org/jira/browse/PARQUET-464
- Aligned allocations
  - https://issues.apache.org/jira/browse/PARQUET-517

Integration testing

- Some prior work: https://github.com/Parquet/parquet-compatibility

Enabling parquet-cpp to become a dependency of Apache Impala (incubating)

This should be possible, but the scope of code sharing will be somewhat limited because of the coupling between the scan and write paths in the Impala runtime.

- Repetition level encoding / decoding algorithms
- Identifying code overlap
- Enumerating differences in code paths