VOTable Metadata for Parquet

Mark Taylor (University of Bristol)

with input from: Gregory Dubois-Felsmann (IPAC/Caltech)
François-Xavier Pineau (CDS)
Brigitta Sipőcz (IPAC/Caltech)
Mario Jurić (U. Washington)
Apps WG

Applications WG

IVOA Interop

Valletta, Malta

16 November 2024

$Id: votparquet.tex,v 1.22 2024/11/14 00:30:49 mbt Exp $

History

This builds on previous work

- Gregory Dubois-Felsmann presentation Tucson 2023
- Brigitta Sipőcz Astropy PR #16375
- François-Xavier Pineau CDS HATS service implementation
- ... and other implementations

Apps WG Telecon 25 November 2024

- 22 participants from multiple institutions
- Good discussion
- Basic agreement on most points
- Refinement of some details of approach
- Enthusiasm and urgency for working with parquet

Basic Idea

Parquet

- ☺ Efficient (column-oriented, transparent compression)
- ☺ Industry standard, lots of tool support
- ☺ Good for storage/distribution/processing of very large tables, science-platform friendly
- ☹ Very limited standard metadata (columns just have names + types)

VOTable

- ☺ Rich metadata
- ☹ Standard within VO, but not used in industry
- ☹ Not good for huge tables

⇒ Combine the two

- Parquet data storage + VOTable metadata
- If you can already work with Parquet and VOTable, it is not particularly hard to implement (depending on the details...)

Parquet/VOTable Combination Options

VOTable-in-Parquet

- Prepare a `DATA`-less VOTable describing the table metadata
- Store VOTable XML in Parquet file-level Key-Value metadata
  - ▷ Implementations: Astropy PR #16375, CDS HATS, TOPCAT prototype, Firefly PR #1633
  - ▷ File is valid parquet: non-VOTable-aware consumers just ignore the extra metadata
  - ▷ Resembles "FITS-plus" convention long used by TOPCAT

VOTable-in-Avro-in-Parquet

- Prepare a `DATA`-less VOTable describing the table metadata
- Store VOTable XML in Avro schema, which is encoded and stored in Parquet file-level Key-Value metadata (`ARROW:schema`)
  - ▷ Implementations: Astropy PR #16375, CDS HATS, LSDB
  - ▷ File is valid parquet: non-VOTable-aware/non-Avro-aware consumers just ignore the extra metadata
  - ▷ Makes assumptions about Parquet parsing software (Avro-based)

Parquet-in-VOTable

- Write a VOTable document with normal metadata
- But `DATA` element has a `PARQUET` child referencing external file (not `TABLEDATA`, `BINARY` or `FITS`)
  - ▷ Implementations: Astropy 6.0
  - ▷ Resembles existing VOTable-FITS wrapping
  - ▷ File not valid VOTable; requires major update to VOTable standard
    (VOTable parsers would have to parse Parquet)

Parquet/VOTable Combination Options

VOTable-in-Parquet

- Prepare a `DATA`-less VOTable describing the table metadata
- Store VOTable XML in Parquet file-level Key-Value metadata
  - ▷ Implementations: Astropy PR #16375, CDS HATS, TOPCAT prototype, Firefly PR #1633
  - ▷ File is valid parquet: non-VOTable-aware consumers just ignore the extra metadata
  - ▷ Resembles "FITS-plus" convention long used by TOPCAT

← **Go forward with this one**

VOTable-in-Avro-in-Parquet

- Prepare a `DATA`-less VOTable describing the table metadata
- Store VOTable XML in Avro schema, which is encoded and stored in Parquet file-level Key-Value metadata (`ARROW:schema`)
  - ▷ Implementations: Astropy PR #16375, CDS HATS, LSDB
  - ▷ File is valid parquet: non-VOTable-aware/non-Avro-aware consumers just ignore the extra metadata
  - ▷ Makes assumptions about Parquet parsing software (Avro-based)

Parquet-in-VOTable

- Write a VOTable document with normal metadata
- But `DATA` element has a `PARQUET` child referencing external file (not `TABLEDATA`, `BINARY` or `FITS`)
  - ▷ Implementations: Astropy 6.0
  - ▷ Resembles existing VOTable-FITS wrapping
  - ▷ File not valid VOTable; requires major update to VOTable standard
    (VOTable parsers would have to parse Parquet)

Parquet/VOTable Combination Options

VOTable-in-Parquet

- Prepare a `DATA`-less VOTable describing the table metadata
- Store VOTable XML in Parquet file-level Key-Value metadata
  - ▷ Implementations: Astropy PR #16375, CDS HATS, TOPCAT prototype, Firefly PR #1633
  - ▷ File is valid parquet: non-VOTable-aware consumers just ignore the extra metadata
  - ▷ Resembles "FITS-plus" convention long used by TOPCAT

← **Go forward with this one**

VOTable-in-Avro-in-Parquet

- Prepare a `DATA`-less VOTable describing the table metadata
- Store VOTable XML in Avro schema, which is encoded and stored in Parquet file-level Key-Value metadata (`ARROW:schema`)
  - ▷ Implementations: Astropy PR #16375, CDS HATS, LSDB
  - ▷ File is valid parquet: non-VOTable-aware/non-Avro-aware consumers just ignore the extra metadata
  - ▷ Makes assumptions about Parquet parsing software (Avro-based)

Parquet-in-VOTable

- Write a VOTable document with normal metadata
- But `DATA` element has a `PARQUET` child referencing external file (not `TABLEDATA`, `BINARY` or `FITS`)
  - ▷ Implementations: Astropy 6.0
  - ▷ Resembles existing VOTable-FITS wrapping
  - ▷ File not valid VOTable; requires major update to VOTable standard
    (VOTable parsers would have to parse Parquet)

Maybe some use? But no VO standard changes currently proposed ←

Reference: Parquet File Format

Parquet defines file-level KeyValue pairs

- This is where VOTable can go

https://parquet.apache.org/docs/file-format/metadata/
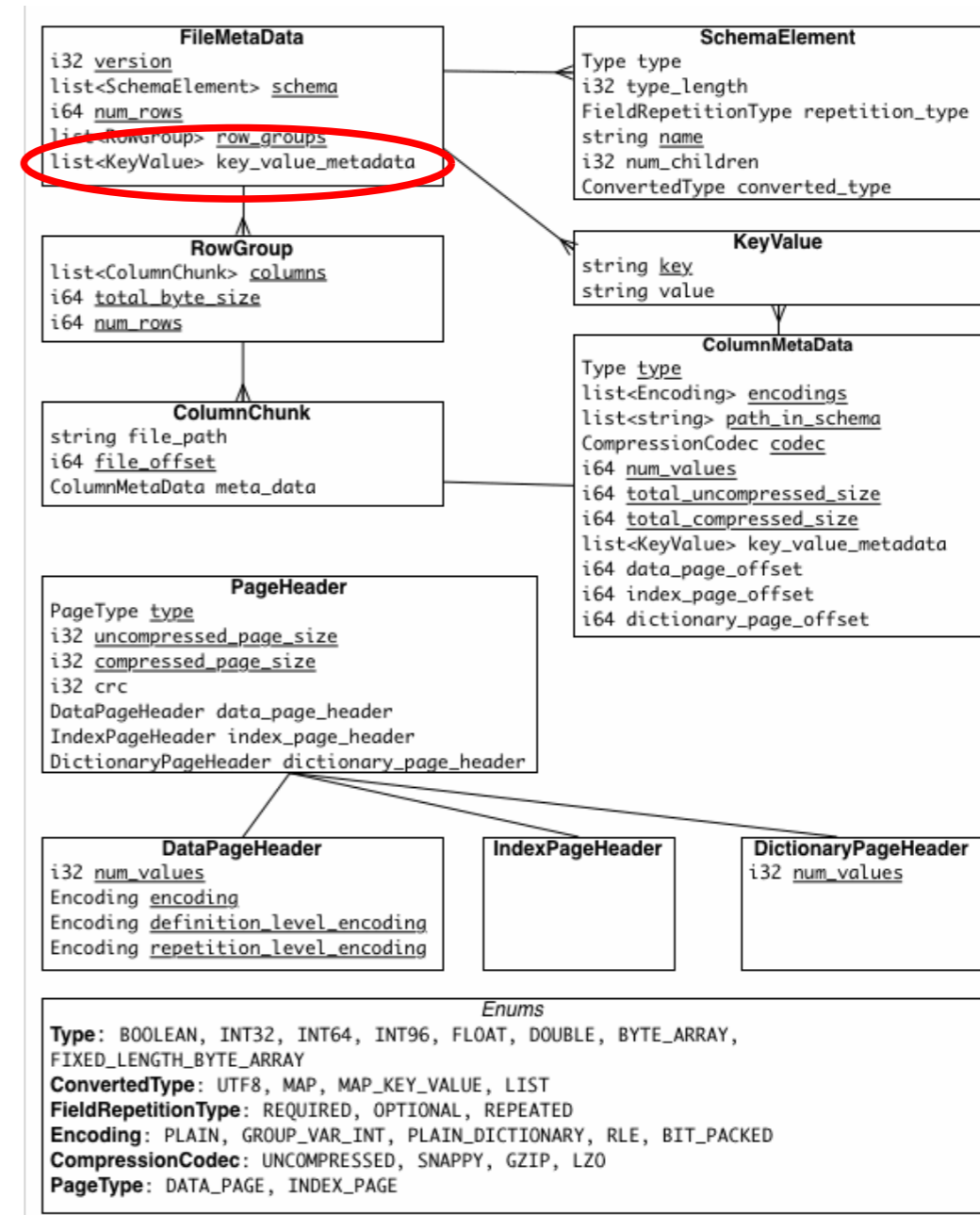
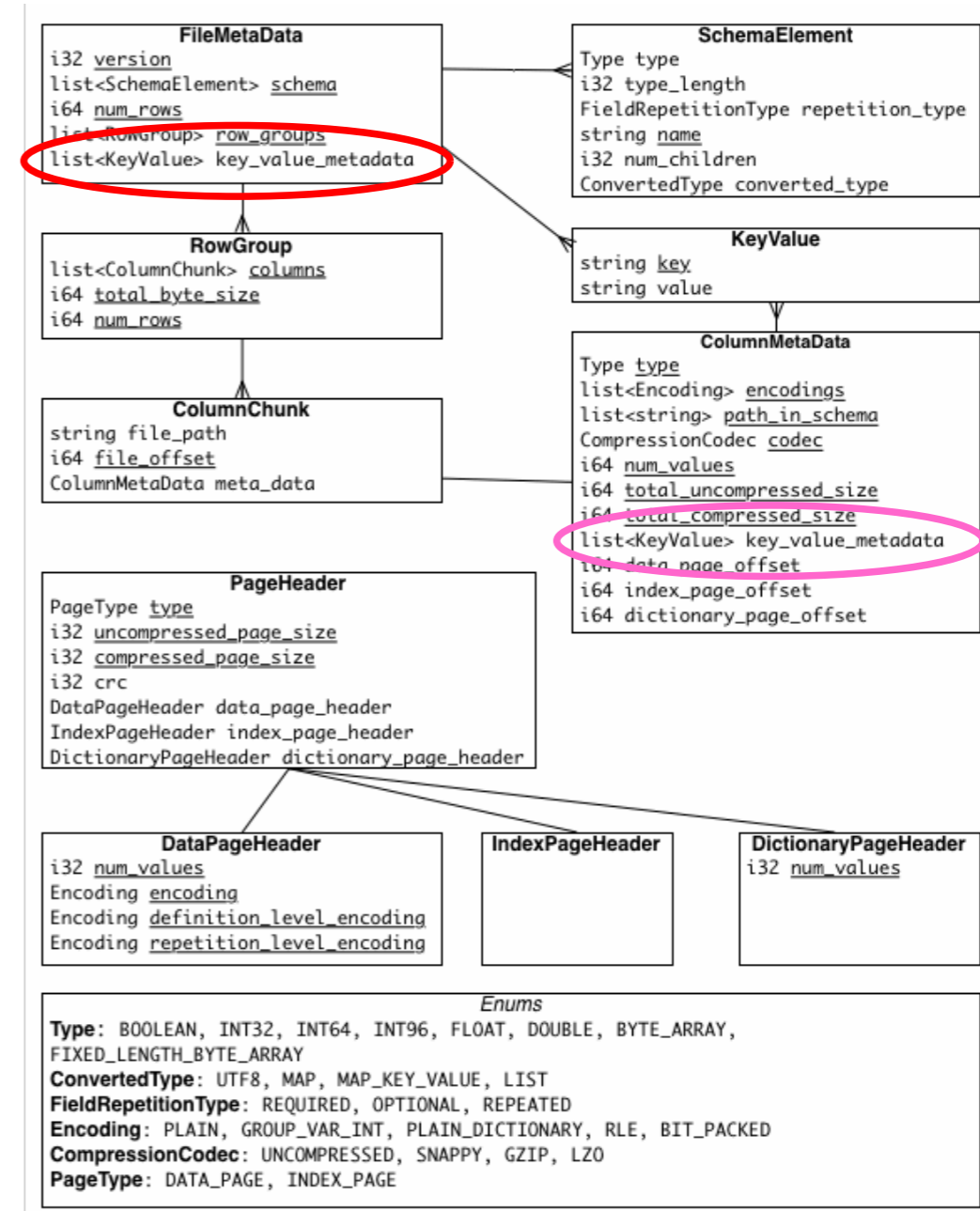# Reference: Parquet File Format

Parquet defines file-level KeyValue pairs

- This is where VOTable can go

Parquet also defines column-level KeyValue pairs

- ... but we're not going to use these
- we need more than units+UCDs here
  (DataLink, COOSYS, error flags, ...)
- there are also problems in some libraries
  with accessing these fields

FileMetaData
i32 version
list<SchemaElement> schema
i64 num_rows
list<RowGroup> row_groups
list<KeyValue> key_value_metadata

SchemaElement
Type type
i32 type_length
FieldRepetitionType repetition_type
string name
i32 num_children
ConvertedType converted_type

RowGroup
list<ColumnChunk> columns
i64 total_byte_size
i64 num_rows

KeyValue
string key
string value

ColumnChunk
string file_path
i64 file_offset
ColumnMetaData meta_data

ColumnMetaData
Type type
list<Encoding> encodings
list<string> path_in_schema
CompressionCodec codec
i64 num_values
i64 total_uncompressed_size
i64 total_compressed_size
list<KeyValue> key_value_metadata
i64 data_page_offset
i64 index_page_offset
i64 dictionary_page_offset

PageHeader
PageType type
i32 uncompressed_page_size
i32 compressed_page_size
i32 crc
DataPageHeader data_page_header
IndexPageHeader index_page_header
DictionaryPageHeader dictionary_page_header

DataPageHeader
i32 num_values
Encoding encoding
Encoding definition_level_encoding
Encoding repetition_level_encoding

IndexPageHeader

DictionaryPageHeader
i32 num_values

Enums
Type: BOOLEAN, INT32, INT64, INT96, FLOAT, DOUBLE, BYTE_ARRAY,
FIXED_LENGTH_BYTE_ARRAY
ConvertedType: UTF8, MAP, MAP_KEY_VALUE, LIST
FieldRepetitionType: REQUIRED, OPTIONAL, REPEATED
Encoding: PLAIN, GROUP_VAR_INT, PLAIN_DICTIONARY, RLE, BIT_PACKED
CompressionCodec: UNCOMPRESSED, SNAPPY, GZIP, LZO
PageType: DATA_PAGE, INDEX_PAGE

https://parquet.apache.org/docs/file-format/metadata/

VOTable-in-Parquet Details

From Gregory Dubois-Felsmann's Tucson 2023 presentation (p. 8):

> The file-level metadata of the Parquet file MUST include these key-value pairs:
>
> IVOA.VOTable-Parquet.version = 1.0
> IVOA.VOTable-Parquet.type = "Parquet-local-XML"
> IVOA.VOTable-Parquet.encoding = (encoding for the XML content of the following)
> IVOA.VOTable-Parquet.content = (character data in a format valid for an XML file,
>   comprising a single <VOTABLE> element and its hierarchical content)
>
> The <VOTABLE> must contain exactly one <TABLE> with its <DATA> content specified as follows:
>
> <DATA><PARQUET type="Parquet-local-XML"/></DATA>

- There are some deviations from this in existing implementations

- Adjustments under discussion:

  ▷ `<PARQUET>` child of `<DATA>` element is problematic: not legal VOTable
    - Suggestion: omit `<DATA>` element altogether?
    - But want to accommodate multiple `<TABLE>`s in attached VOTable document
    - Possible solution: designate parquet data table as `RESOURCE[@type='results']/TABLE`? — see DALI 1.1 sec 4.4

  ▷ `encoding` key (designating XML encoding) not necessary/useful?
    - Additional reader implementation burden for questionable gain
    - Let's just mandate UTF-8

  ▷ `type` key not necessary/useful?
    - Not clear what clients are supposed to do with this
    - Remove?

STIL Prototype Implementation

```
% java -jar topcat-extra_metaparq.jar -stilts tpipe in=:skysim:3 cmd='keepcols "ra dec"' ofmt='parquet(votmeta=true)' out=sky.parquet
% java -classpath /mbt/github/parquet-mr/parquet-cli/target/parquet-cli-1.13.1.jar:`echo /mbt/github/parquet-mr/parquet-cli/target/dependency/*.jar | tr ' ' :`
       org.apache.parquet.cli.Main meta sky.parquet
 File path:  sky.parquet
 Created by: parquet-mr version 1.13.1 (build db4183109d5b734ec5930d870cdae161e408ddba)
 Properties:
    IVOA.VOTable-Parquet.content: <?xml version='1.0'?>
<VOTABLE version="1.4" xmlns="http://www.ivoa.net/xml/VOTable/v1.3">
<RESOURCE type="results">
<TABLE name="SimulatedSky-3" nrows="3">
<DESCRIPTION>Simulated star catalogue, based on Gaia EDR3</DESCRIPTION>
<PARAM datatype="int" name="HealpixLevel" value="5"><DESCRIPTION>Healpix level of aggregated statistics</DESCRIPTION></PARAM>
<FIELD datatype="float" name="ra" ucd="pos.eq.ra" unit="deg"><DESCRIPTION>ICRS Right Ascension</DESCRIPTION></FIELD>
<FIELD datatype="float" name="dec" ucd="pos.eq.dec" unit="deg"><DESCRIPTION>ICRS Declination</DESCRIPTION></FIELD>
<!-- Dummy VOTable - no DATA element -->
</TABLE>
</RESOURCE>
</VOTABLE>
    IVOA.VOTable-Parquet.version: 1.0
                           name: SimulatedSky-3
              writer.model.name: STIL
 Schema:
message table {
  optional float ra;
  optional float dec;
}


 Row group 0: count: 3  23.33 B records  start: 4  total(compressed): 70 B total(uncompressed):70 B
--------------------------------------------------------------------------------
      type       encodings count     avg size   nulls    min / max
ra    FLOAT      _   _       3        11.67 B     0        "104.37142" / "286.89078"
dec   FLOAT      _   _       3        11.67 B     0        "-64.85177" / "29.448921"
```

Wider/Related Questions

- What about parquet files that cannot be described by VOTable?
  - Columns containing structured data types, unsupported primitive types like unsigned integers, ...

- What if VOTable metadata is inconsistent with Parquet data/metadata?

- Can VOTable metadata outside of Parquet file also be used?
  - Could be useful at top level for describing a large collection of parquet files
  - May be updated more often/more easily than internal key-value pairs
  - In case of conflict which takes precedence?

- Should/could this be tackled as a more general question?
  - How can you use VOTable to attach rich metadata to a metadata-poor format?
  - STIL/TOPCAT has used a very similar approach with FITS for years ("FITS-plus" format)
  - We may encounter other popular formats in the future with a similar requirement

- Provide recognition of parquet as a common VO `RESPONSEFORMAT`? (DALI 1.1 sec 3.4.3)
  - Parquet does have a MIME type `application/vnd.apache.parquet`, but a "`parquet`" alias might be nice

- Compression formats
  - Parquet names multiple compression formats; not all parquet I/O libraries may support them all
  - Do we want to advise/restrict/recommend which ones to use?

- Should Parquet-in-VOTable be endorsed/standardised?
  - Would require major updates to VOTable standard and implementations

Current Status

Urgency

- Many projects are writing parquet now or want to very soon
- They want to incorporate rich metadata

Implementations

- There are several implementations roughly matching VOTable-in-Parquet
- Some of them interoperate, but there are differences of detail
- The convention is mostly implicit in code or hard to find

Action required

- Formalise the proposal in a visible document

Roadmap (1)

For immediate action:

- Agree we want to do VOTable-in-Parquet
  - ▷ Any dissenters?

- Nail down VOTable-in-Parquet details
  - ▷ What metadata keys to require/allow
  - ▷ How to serialize data-less VOTable

- Produce an **Endorsed Note** standardising this approach
  - ▷ Authors:
    - ○ Mark T, Gregory D-F, Brigitta S, others?
  - ▷ Timescale:
    - ○ Soon! Completed text by end of 2024?
  - ▷ Scope:
    - ○ Allow parquet producers/consumers to exchange metadata-rich parquet files interoperably
    - ○ Future revisions may refine/extend, but should remain compatible with this base
    - ○ Say something about Parquet/VOTable metadata model mismatch
    - ○ Say something about Parquet/VOTable metadata inconsistency resolution
    - ○ Recommend DALI `RESPONSEFORMAT` value for parquet output?
    - ○ Don't attempt to solve all related problems

- Make sure we have compliant, interoperable implementations
  - ▷ We're quite close already
  - ▷ Assemble a gallery of example files

Roadmap (2)

Longer term maybe:

- Produce REC-track document within Apps WG
    - ▷ Consolidate and/or extend content of Endorsed Note
    - ▷ Incorporate implementation experience gained
    - ▷ Address some of the more open issues