



*International
Virtual
Observatory
Alliance*

Parquet in the VO

Version 1.0

IVOA Note 2025-01-14

Working Group

Applications

This version

<https://www.ivoa.net/documents/Notes/voparquet/20250114>

Latest version

<https://www.ivoa.net/documents/Notes/voparquet>

Previous versions

This is the first public release

Author(s)

Mark Taylor, François-Xavier Pineau, Gregory Dubois-Felsmann,
Brigitta Sipőcz

Editor(s)

Mark Taylor

Version Control

Revision 7916569, 2025-01-14 14:44:31 +0000

Abstract

Parquet is a file format for record-based data, with widespread industry tool support. It is being adopted by several astronomy projects for bulk storage and distribution of large tabular data products. This Note discusses best practice for use of parquet within the VO, and in particular defines the VOParquet convention which uses VOTable to attach rich astronomical metadata to otherwise metadata-poor parquet files.

Status of this document

This is an IVOA Note expressing suggestions from and opinions of the authors. It is intended to share best practices, possible approaches, or other perspectives on interoperability with the Virtual Observatory. It should not be referenced or otherwise interpreted as a standard specification.

A list of current IVOA Recommendations and other technical documents can be found in the IVOA document repository¹.

Contents

1	Introduction	2
1.1	Scope	4
2	VOParquet Convention	4
2.1	Serialization Approach	4
2.2	Serialization Format	5
2.3	Data/Metadata Mismatches	6
3	Alternative Approaches	7
A	Changes from Previous Versions	7
	References	7

Conformance-related definitions

The words “MUST”, “SHALL”, “SHOULD”, “MAY”, “RECOMMENDED”, and “OPTIONAL” (in upper or lower case) used in this document are to be interpreted as described in IETF standard RFC2119 (Bradner, 1997).

The *Virtual Observatory (VO)* is a general term for a collection of federated resources that can be used to conduct astronomical research, education, and outreach. The *International Virtual Observatory Alliance (IVOA)* is a global collaboration of separately funded projects to develop standards and infrastructure that enable VO applications.

1 Introduction

The Apache Parquet file format² is a column-oriented storage format for record-based data first developed in 2013. It offers per-column compression,

¹<https://www.ivoa.net/documents/>

²<https://parquet.apache.org/docs/>

dictionary encoding, and a kind of column value indexing. A number of data processing environments optimised for parallel computing make use of these features to enable fast processing of large or very large tables. At the time of writing (early 2025), several astronomy projects including Rubin, Gaia and SPHEREx are using or planning to use parquet for storage, processing and distribution of large-scale astronomical data products. I/O libraries are available in various languages including Python, Java, C++ and Rust, and these have been leveraged by astronomer-facing software such as Astropy, CDS services and TOPCAT to facilitate use of parquet data in astronomy.

While offering efficient data storage however, the standard semantic metadata provided by Parquet files is quite rudimentary. Apart from a name and datatype for each column, there is only a list of untyped key-value pairs per table and per column-chunk, with no standard semantics for the keys. For scientific usability, better semantic metadata is desirable and even necessary, especially in view of the complexity of the data represented; astronomy tables can easily contain hundreds of columns. A minimum requirement is column attributes such as units, descriptions, UCDs and precisions; in many cases additional information relating to coordinate systems, service descriptors or processing flags may also be required.

The VOTable format (Ochsenbein and Taylor et al., 2019) has been developed within the VO since its inception to hold exactly the kind of metadata required here. Combining the virtues of VOTable and Parquet therefore can supply a format which delivers storage efficiency alongside rich astronomical metadata.

This Note addresses the question of how to effect that combination. In particular it defines a convention named VOParquet that stores VOTable metadata in parquet files in a way which will be interoperable between data producers and consumers from different projects. Although usage may be refined in future as the result of developing requirements and implementation experience, the intention (at least, the hope) is that the prescriptions here will remain valid as a backwardly compatible baseline for some while, so that future iterations of parquet I/O software in the VO will remain compatible with files written according to this Note. For this reason, we restrict ourselves here to the minimum rules that will enable interoperability, and avoid imposing requirements on points of detail for which the best way forward is not obvious.

The normative part of this VOParquet convention is quite short and can be found in Section 2 and especially Subsection 2.2. The other sections provide context and discussion.

1.1 Scope

The topic of this Note suggests other discussions, including best practice for sharding large datasets among multiple parquet files, policy for choice of compression algorithms within parquet, suitability of parquet for archival storage, and the application of similar ideas to enhance other metadata-poor file formats using VOTable.

This Note avoids those questions in the interest of achieving rapid consensus on the question of combining VOTable and parquet. Several projects will be generating large parquet collections in the near future, so that early agreement on the basics of the format is required to achieve interoperability between a number of datasets too large to be rewritten at a later date.

Future work may build on the current document and on implementation experience to produce a revised Note or a Recommendation-track document that extends the current proposal or addresses some of these wider questions.

2 VOParquet Convention

2.1 Serialization Approach

The parquet and VOTable file formats both provide serialization of tabular data, along with some degree of file- and column-level metadata. Given an abstract input table with rich metadata, the basic prescription for writing a VOParquet file is:

1. serialize the input table to VOTable but without including the data part, thus producing an XML document containing table metadata only
2. serialize the input table to parquet in the usual way, but
3. include the data-less VOTable document in the file-level metadata of the parquet file

The parquet file generated in the final step is the VOParquet output. When reading such a file:

1. read the parquet data in the usual way
2. search for a VOTable in the file-level metadata
3. if one is present, parse it and use the table- and column-level metadata it contains to decorate the data read from parquet

The serialised table is therefore a perfectly legal parquet file, which can be read by any parquet I/O software. But VOParquet-aware software can use the attached dataless VOTable to recover the rich metadata associated with the original table.

2.2 Serialization Format

The VOTable metadata document stored in the parquet metadata must contain a *TABLE* element describing the parquet data table. This looks exactly like a normal *TABLE* element except that it has no *DATA* child; such data-less tables are permitted by the VOTable schema. In particular it must contain *FIELD* elements describing the columns of the parquet data, and it may contain other elements such as *PARAM*, *COOSYS* etc providing additional table-level metadata.

This *DATA*-less *TABLE* must be the first *TABLE* element in the VOTable document. Other *TABLE*s, for instance providing auxiliary data or metadata, may appear in the VOTable document, but are not used to describe the parquet data directly. The VOTable document must be a schema-valid and legal VOTable instance. No particular VOTable version is mandated by this convention.

An example VOTable metadata document describing a 3-column table might look like this³:

```
<VOTABLE version="1.4" xmlns="http://www.ivoa.net/xml/VOTable/v1.3">
  <RESOURCE>
    <TABLE name="MessierObjects">
      <DESCRIPTION>Nebulae and clusters</DESCRIPTION>
      <PARAM name="author" datatype="char" arraysizes="*"
        value="Charles Messier"/>
      <FIELD datatype="long" name="ID">
        <DESCRIPTION>Source identifier</DESCRIPTION>
      </FIELD>
      <FIELD datatype="double" name="RA" ucd="pos.eq.ra" unit="deg">
        <DESCRIPTION>ICRS Right Ascension</DESCRIPTION>
      </FIELD>
      <FIELD datatype="double" name="DEC" ucd="pos.eq.dec" unit="deg">
        <DESCRIPTION>ICRS Declination</DESCRIPTION>
      </FIELD>
      <!-- Metadata-only TABLE - no DATA element -->
    </TABLE>
  </RESOURCE>
</VOTABLE>
```

This dataless VOTable document is stored in the `key_value_metadata` list of the `FileMetaData` structure in the parquet footer. That list is defined by the parquet file format⁴ to contain an unstructured collection of string-string key-value pairs, and is available for applications to populate with arbitrary metadata. The VOParquet convention requires the following key-value pairs to be present:

IVOA.VOTable-Parquet.version

The version of this convention. Must be "1.0" at this version.

³The apparent mismatch between version and namespace is intended; see VOTable 1.4 section 3 for details.

⁴<https://github.com/apache/parquet-format>

IVOA.VOTable-Parquet.content

The content of the data-less VOTable document described above, encoded using UTF-8. An XML declaration (“<?xml ... ?>”) may optionally precede the content, but if present it must not declare a non-UTF-8 encoding.

2.3 Data/Metadata Mismatches

In the most straightforward case, there will be a one-to-one mapping of columns in the parquet data table and the VOTable metadata table, where the N^{th} *FIELD* of the VOTable metadata table provides an accurate description of the N^{th} top-level column (N^{th} *SchemaElement* child of the root *SchemaElement*) of the parquet data table, in particular declaring VOTable *datatype* and *arraysize* attributes that correspond to the physical/logical type of that parquet column. It is expected that for many or most tables in the VO this will be the case.

However, the data models of VOTable and Parquet do not exactly match, and it is possible to store columns in a parquet file that cannot accurately be described by VOTable metadata. Columns containing scalars and 1-d arrays of booleans, unsigned 8-bit integers, signed 16/32/64-bit integers and strings map straightforwardly to VOTable, but other types such as unsigned 64-bit integers and structured data do not.

In the case of such mismatches, it is recommended that for each top-level parquet column, VOParquet writers should write exactly one VOTable *FIELD* element that makes a best effort to describe the data. If a parquet column has a type that does not resemble any VOTable datatype, for instance structured data, it is recommended to describe it as a VOTable string, e.g. *datatype*="char" *arraysize*="*". It is not permitted to omit the *datatype* attribute altogether, since that would result in an illegal VOTable document. Other approaches are possible, for instance describing the primitive sub-elements of a structured parquet top-level column using multiple VOTable *FIELDS*, but these are not currently recommended.

Readers must treat the parquet data and datatypes as authoritative, and may make use of as much VOTable column metadata as is feasible, for instance using units and UCDs where present but discarding VOTable datatypes that are incompatible with the parquet data columns. Ignoring the metadata table *datatype* and *arraysize* attributes on *FIELD* elements altogether in favour of column types acquired from the parquet parsing would be one way for readers to handle this. If the number of VOTable columns cannot be reconciled with the parquet data, the reader must discard the VOTable column metadata. The reader is free to discard some or all of the VOTable metadata in case of any other difficulty in reconciling the VOTable metadata and parquet data, or for any other reason.

Future evolutions of this convention may refine or adjust this advice.

3 Alternative Approaches

Other approaches have been suggested for association of rich metadata with parquet files. These include variations on the idea of using a VOTable document stored *outside* of the described parquet serialization, and use of the existing per-table and per-column-chunk key-value lists defined by the parquet format for direct (non-VOTable) storage of metadata items.

This Note neither requires nor forbids use of these alternative approaches either alongside or instead of the VOParquet convention described here, nor does it prescribe how to reconcile the redundancy and possible conflicts that may arise from storage of metadata in more than one place. It does encourage use of the VOParquet format where interoperability within the VO is a goal, but parquet producers and consumers may wish to consider these other options as well.

A Changes from Previous Versions

No previous versions yet.

References

- Bradner, S. (1997), ‘Key words for use in RFCs to indicate requirement levels’, RFC 2119. <http://www.ietf.org/rfc/rfc2119.txt>.
- Ochsenbein, F., Taylor, M., Donaldson, T., Williams, R., Davenhall, C., Demleitner, M., Durand, D., Fernique, P., Giarretta, D., Hanisch, R., McGlynn, T., Szalay, A. and Wicenec, A. (2019), ‘VOTable Format Definition Version 1.4’, IVOA Recommendation 21 October 2019. doi:10.5479/ADS/bib/2019ivoa.spec.1021O, <https://ui.adsabs.harvard.edu/abs/2019ivoa.spec.1021O>.