VOTable: Why, What and How?

Mark Taylor, Bristol University
m.b.taylor@bristol.ac.uk

Plan

Why was the VOTable format invented?

What do VOTables look like?

How do I use and manipulate them?

Why? Motivation for VOTable

There are many tabular formats out there:

- ASCII: CSV, TST, 'plain' ASCII
- RDBMS
- HDF5
- FITS: BINTABLE, TABLE

FITS is almost good enough . . . :

- can store scalar/array numeric/character types in columns
- can store large amounts of data
- efficient data access
- header cards can store table and column metadata

. . . but not quite:

- Metadata: FITS header cards not really up to the job in the VO-era
- Scalability: can't stream a FITS file in response to a query
- Data/Metadata Separability: might want these in different places

VOTable Design Principles

Rich metadata capabilities

- XML-based format

Scalability/Streamability

- Can stream table output (don't need row count up front)

Data/Metadata Separability

- Metadata document can (optionally) point to actual data elsewhere

FITS Compatibility

- can losslessly convert:
  - metadata + data: FITS $\longrightarrow$ VOTable
  - data only: FITS $\longleftrightarrow$ VOTable

Not VOTable Design Principles, Honestly

VOTable is XML

- buzzword-compliant
- can use existing XML technology for VOTable manipulation
  - XSLT styling
  - XML Schema validation
  - XML binding tools
  - . . .

You can look at VOTables in a text editor

- as long as they're TABLEDATA format
- as long as they're not too big

What do VOTables look like?

XML-based

Metadata part stored separately from the data part

- optionally, in a separate file

Format is described in the VOTable standard

- http://www.ivoa.net/Documents/latest/VOT.html

VOTable Metadata section

Information grouped hierarchically using `RESOURCE` elements

- can group tables together, nest sets of tables etc

Per-table metadata described in `PARAM`/`INFO` elements

- Typically curation metadata like table source, author, processing history, . . .

Per-column metadata described in `FIELD` elements

- data type
- name
- units
- description
- UCD
- "utype"

VOTable Data section

There are three (or five) ways of filling the `DATA` element

- `TABLEDATA`
  - Stored as a sequence of `TR` and `TD` elements (like HTML tables)
  - "Pure" XML
  - Somewhat human readable
  - Not very efficient
- `BINARY`
  - Store either in-line or reference to an external file/stream
  - Efficient to process and transmit
- `FITS`
  - Store either in-line or reference to an external file/stream
  - Efficient to process and transmit
  - Can describe existing FITS file with separate VOTable metadata

. . . nearly everybody uses `TABLEDATA`

Example VOTable (TABLEDATA format)

```
<?xml version="1.0"?>
<VOTABLE version="1.1">
  <RESOURCE>
    <TABLE>
      <DESCRIPTION>A few Messier objects</DESCRIPTION>
      <PARAM datatype="char" arraysize="*" ucd="meta.bib.author"
             name="Author" value="Mark Taylor"/>
      <FIELD name="Identifier" datatype="char" arraysize="*"
             ucd="meta.id;meta.main">
        <DESCRIPTION>Messier Identifier for the object</DESCRIPTION>
      </FIELD>
      <FIELD name="RA" datatype="double" unit="deg" ucd="pos.eq.ra;meta.main"/>
      <FIELD name="Dec" datatype="double" unit="deg" ucd="pos.eq.dec;meta.main"/>
      <DATA>
        <TABLEDATA>
          <TR><TD>M31</TD><TD> 10.502</TD><TD>41.266</TD></TR>
          <TR><TD>M57</TD><TD>283.253</TD><TD>33.033</TD></TR>
          <TR><TD>M82</TD><TD>148.753</TD><TD>69.683</TD></TR>
        </TABLEDATA>

      </DATA>
    </TABLE>
  </RESOURCE>
</VOTABLE>
```

Example VOTable (BINARY format)

```
<?xml version="1.0"?>
<VOTABLE version="1.1">
  <RESOURCE>
    <TABLE>
      <DESCRIPTION>A few Messier objects</DESCRIPTION>
      <PARAM datatype="char" arraysize="*" ucd="meta.bib.author"
             name="Author" value="Mark Taylor"/>
      <FIELD name="Identifier" datatype="char" arraysize="*"
             ucd="meta.id;meta.main">
        <DESCRIPTION>Messier Identifier for the object</DESCRIPTION>
      </FIELD>
      <FIELD name="RA" datatype="double" unit="deg" ucd="pos.eq.ra;meta.main"/>
      <FIELD name="Dec" datatype="double" unit="deg" ucd="pos.eq.dec;meta.main"/>
      <DATA>
        <BINARY>
          <STREAM encoding='base64'>
            AAAAA00zMUAlAQYk3S8bQESiDEm6XjUAAAADTTU3QHG0DEm6XjVAQIQ5WBBiTgAA
            AANNODJAYpgYk3S8akBRa7ZFocrB
          </STREAM>
        </BINARY>
      </DATA>
    </TABLE>
  </RESOURCE>
</VOTABLE>
```

Example VOTable (FITS format)

```
<?xml version="1.0"?>
<VOTABLE version="1.1">
  <RESOURCE>
    <TABLE>
      <DESCRIPTION>A few Messier objects</DESCRIPTION>
      <PARAM datatype="char" arraysize="*" ucd="meta.bib.author"
             name="Author" value="Mark Taylor"/>
      <FIELD name="Identifier" datatype="char" arraysize="*"
             ucd="meta.id;meta.main">
        <DESCRIPTION>Messier Identifier for the object</DESCRIPTION>
      </FIELD>
      <FIELD name="RA" datatype="double" unit="deg" ucd="pos.eq.ra;meta.main"/>
      <FIELD name="Dec" datatype="double" unit="deg" ucd="pos.eq.dec;meta.main"/>
      <DATA>
        <FITS>
          <STREAM href="tdata-1.fits"/>
        </FITS>



      </DATA>
    </TABLE>
  </RESOURCE>
</VOTABLE>
```

How to Produce VOTables

Various possibilities for writing them:

- Use print statements
- VOTable output libraries
- Convert from an existing table in a known format

But note:

- it's easy to write a broken/non-conforming VOTable
- it's not easy to tell if you've got a broken VOTable by looking
- some libraries/applications don't read BINARY/FITS format

Use STILTS `votlint`!!!

How to Read VOTables

Write your own parser?

- not all that hard for simple TABLEDATA tables
- but difficult to cover all cases
- FITS and BINARY parsing takes some work

Use a pre-existing VOTable parsing library in your own software

- STIL, JAVOT, SAVOT, ...
- note not all cope with all VOTable variants (STIL does)

Convert from VOTable to some format you're happier with

- STILTS `tcopy`

Use an existing analyis package that understands VOTable

- TOPCAT (user-friendly GUI table analysis)
- STILTS (command-line table analysis/manipulation)
- VOPlot (VOTable plotting applet/application)

STIL (Starlink Tables Infrastructure Library)

Public Java library for (VO)Table I/O

Features:

- Reads/writes all VOTable variants (TABLEDATA, BINARY, FITS)
- Efficient (streams data if at all possible)
- Understands other table formats as well as VOTable
- Pure Java

http://www.starlink.ac.uk/stil/

Other VOTable parsing libraries are available

STILTS (STIL Tool Set)

Provides table manipulation functions from the command line

- or in a script
- or on a server

Features:

- Easy to install (one or two files, pure Java)
- Provides a lot of the features of TOPCAT from the command line
- Highly scalable and efficient

Syntax

```
stilts <command-name> <command-args>
```

Capabilities:

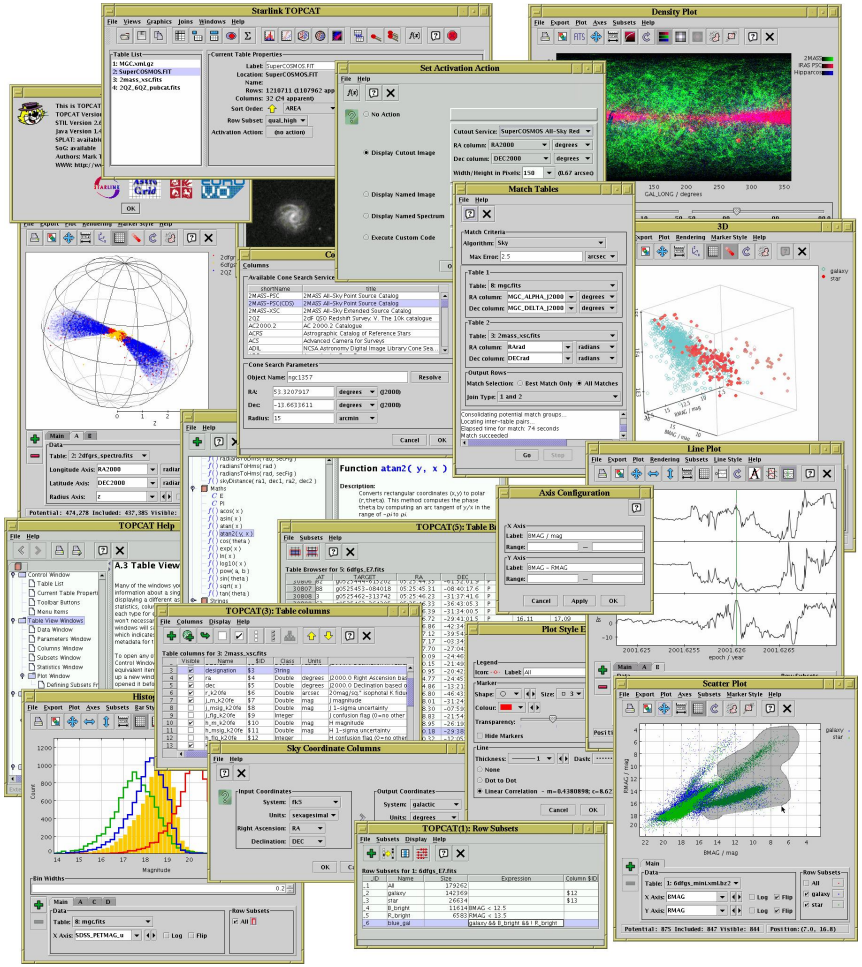
- conversion between different formats
- conversion between different VOTable encodings (TABLEDATA/FITS/BINARY)
- VOTable validation/error checking
- cross matching
- all manner of processing (sort, row select, column add/remove . . .)

http://www.starlink.ac.uk/stilts/

TOPCAT

Powerful GUI tool for (VO)Table manipulation

Does most of the same things as STILTS, plus flexible interactive graphics



STILTS Example: VOTable validation

```
% stilts votlint in=some-table.xml

INFO (1.4): No arraysize for character, FIELD implies single character
ERROR (1.7): Element "TABLE" does not allow "DESCRIPTION" here.
WARNING (1.11): Characters after first in char scalar ignored (missing arraysize?
WARNING (1.15): Wrong number of TDs in row (expecting 3 found 4)
ERROR (1.18): Row count (1) not equal to nrows attribute (2)
```

Checks misspelt elements, missing attributes, inconsistent data, . . .

Much more rigorous than validating against the schema/DTD

Many (most?) VOTables in the wild are broken in some way

Please use this command if you are producing VOTables!

- and fix them if they show up errors

Also useful to check suspect files if you're consuming VOTables

- and inform the publisher if they show up errors

STILTS Example: Table format conversion

VOTable → FITS:

```
stilts tcopy ifmt=votable ofmt=fits in=table.xml out=table.fits
```

FITS → VOTable:

```
stilts tcopy ifmt=fits ofmt=votable in=table.fits out=table.xml
```

Also convert between other formats:

- CSV
- IPAC
- TST
- ASCII
- . . .

Easy!

Obviously, conversion can't always be perfect

- e.g. converting from VOTable to comma-separated values will lose metadata

STILTS Example: Import/export from RDBMS

Export data from MySQL to VOTable:

```
stilts -classpath /usr/local/jars/mysql-connector-java.jar \
       -Djdbc.drivers=com.mysql.jdbc.Driver \
       tcopy \
       in="jdbc:mysql://localhost/db1#SELECT * FROM xcat WHERE mag<24" \
       ofmt=votable out=result.xml
```

Import data from FITS file to PostgreSQL:

```
stilts -classpath /usr/local/jars/pg73jdbc3.jar \
       -Djdbc.drivers=org.postgresql.Driver \
       tpipe \
       in=photogals.fits ifmt=fits \
       omode=tosql \
       protocol=postgresql host=localhost database=ASTRO1 newtable=PHOTOGALS
       user=mbt
```

STILTS Example: Table manipulation

Find objects in a table with the largest area:

```
stilts tpipe in=2dfgrs_ngp.vot \
            cmd='keepcols "SEQNUM AREA ECCENT"' \
            cmd='sort -down AREA' \
            cmd='head 20'
```

Calculate galactic from equatorial coordinates:

```
stilts tpipe cmd='addskycoords -inunit sex fk5 gal RA2000 DE2000 GAL_LONG GAL_LAT'
            in=query-result.xml ifmt=votable
            out=query-result2.xml ofmt=votable
```

. . . and many, many more