Astronomical Data Analysis Software and Systems XV **P.3**
ASP Conference Series, Vol. XXX, 2005
C. Gabriel, C. Arviset, D. Ponz and E. Solano, eds.

STILTS - A Package for Command-Line Processing of Tabular Data

Mark B. Taylor

H H Wills Physics Laboratory, Tyndall Avenue, Bristol University, Bristol, UK

Abstract. STILTS, the STIL Tool Set, is a set of non-interactive tools for manipulation of tables such as astronomical object catalogues. It can read and write data in many formats, including VOTable, FITS, relational databases and ASCII. Facilities provided include table format conversion, row selection and sorting, column creation and rearrangement, coordinate conversion, metadata manipulation and display, flexible cross-matching, per-row and statistical calculations and VOTable validation. STILTS is based on the Starlink Tables Infrastructure Library, which also underlies the interactive table-analysis tool TOPCAT, and can be considered its non-interactive counterpart, providing many of the same features in a form which is suitable for headless, batch or scripted environments. Uses include data manipulation from the desktop or as part of server-based workflows or query operations. The package is portable (Java), open source, fully documented, efficient and scalable; in particular it is designed for use with large, and for many purposes arbitrarily large, tables.

1. Introduction

Tabular data, especially in the form of object catalogues, are common in astronomy, and much of the data and processing in the virtual observatory in particular is dedicated to tables in various forms. The IVOA-endorsed VOTable standard is important in this context as defining a metadata-rich transport and storage format for tables; however it has not and will not displace FITS files on the one hand, and relational databases on the other, not to mention a host of ad-hoc ASCII-based formats, as a repository for catalogues and other tables.

STILTS[1] is a new package of command-line tools for performing versatile and powerful manipulations on tables in a number of formats including those mentioned above. It provides some tools for generic (format-independent) table processing, and some specific to VOTables.

The package is based on the Starlink Tables Infrastructure Library (STIL[2]) a table I/O and processing library whose first public release was early in 2004. Among the features of STIL are fully compliant VOTable parsing (it was the

[1] `http://www.starlink.ac.uk/stilts/`

[2] `http://www.starlink.ac.uk/stil/`

first parser to handle BINARY and FITS as well as TABLEDATA-format VOTables), a streaming model which supports efficient processing of arbitrary length tables in limited memory and a pluggable data handling architecture which is easily adapted for reading or writing new table formats. STILTS benefits from these features.

STIL was originally developed as the table handling engine for TOPCAT[3], which is an interactive graphical viewing and analysis package for tabular data. TOPCAT is becoming widely used both within and without the context of the Virtual Observatory, and is good for interactive investigation of the properties of one or more tables, providing facilities for data visualisation, row selection, column and statistical calculations, column rearrangements, data and metadata manipulation, sorting, cross-matching, joining and so on (Taylor 2005). However it is not scriptable or otherwise controllable except from within a GUI, and so STILTS has been developed primarily as a non-graphical counterpart to TOPCAT. Usage scenarios in which such non-GUI tools are required include performing the same operation on multiple tables without user intervention, and processing on a headless server, either as part of a user-specified batch-type workflow or to produce data for delivery to the client of a web service.

STILTS, like STIL and TOPCAT, is written in pure Java and available under the GNU General Public License.

2. Generic Table Processing Tools

STILTS provides four commands for generic table processing. Each of these conceptually takes one or more tables as input and produces a table as output. The input tables can be in any of the supported formats. The table generated as output can either be written to a stream or to disk in one of the supported formats, or can be operated on in some other way, such as having statistics calculated on it or sending it for display in TOPCAT.

The input/output tables can be operated on using 'filter'-like operations such as sorting, various kinds of row selection and sampling, column calculations, metadata manipulation and so on. These filters operate analogously to the filters in a Unix pipeline, except that what is flowing from one to the next is a stream of table data and metadata rather than a sequence of bytes. The way it is implemented means that the table at each stage is 'virtual', so that usually data not required for the output are never actually obtained from the input. This means that operation can be very efficient, and in particular much better than writing the results of each intermediate processing stages to disk.

The individual commands are described in the following subsections.

2.1. tpipe

`tpipe` performs general purpose table-processing pipeline operations. Processing steps ('filters'), which can be combined freely, include coordinate conversion, row selection, data sampling, metadata display and editing, column calculations,

[3] `http://www.starlink.ac.uk/topcat/`

row sorting and blank value substitution amongst others. A powerful but intuitive expression language (based on Java) is provided for specifying algebraic expressions.

Here is an example which performs bad value substitution (999→NULL) in some of the columns, selects only those rows in a given range of proper motions and with a given value in one column, adds a new column calculated from two existing ones, and sorts the result according to redshift.

```
stilts tpipe in=data.cat.gz ifmt=ascii out=data.fits \
   cmd='badval 999 *MAG' \
   cmd='select OBJ_CLASS==3 && PMRA*PMRA+PMDEC*PMDEC<0.5' \
   cmd='addcol I_V IMAG-VMAG' \
   cmd='sort -down Z_ABS'
```

2.2. tcopy

`tcopy` is used to convert between table formats. This doesn't actually require any specialist processing beyond the normal I/O facilities of STILTS, and in fact `tcopy` is just a simplified form of `tpipe`, provided for convenience. Supported I/O formats include FITS, VOTable, relational databases (via SQL queries), plain ASCII, and comma-separated values. Input can be from a disk file, a URL or a stream, and data compression (gzip, bip2, Unix compress) is detected and handled automatically.

An example invocation might look like this:

```
stilts tcopy ifmt=votable ofmt=fits in=cat.vot.gz out=cat.fit
```

2.3. tmatch2

`tmatch2` is a crossmatching tool for finding rows which match between two tables (single-table and multi-table crossmatchers will be introduced in the future). The algorithm is generic and extensible and can perform approximate or exact matches in various different physical or notional parameter spaces of arbitrary dimensionality. The most common case for astronomers is matching on sky position with a global or per-row maximum angular separation, but there are other possibilities for the matching criteria, for instance proximity in two- or three-dimensional Cartesian space, or requiring proximity in flux value as well as sky position. Either all matches or only the best match can be retained, and the form of the joined table to be output is configurable. Calculation time scales as approximately $O(N \ln N)$, where N is the sum of number of rows in both tables, making it suitable for medium-sized data sets (e.g. a sky match of two 10^5-row tables takes of the order of one minute on a current laptop).

The following locates all matches within 3 arcsec and 0.5 blue magnitudes, returning only matched pairs:

```
stilts tmatch2 in1=mgc.fits in2=6dfgs.xml out=matched.fits \
   join=1and2 find=all matcher='sky+1d' params='3 0.5' \
   values1='ra dec bmag' values2='RA2000 DE2000 B_MAG'
```

2.4. tcat

`tcat` simply joins multiple input tables top to bottom, giving one output table with a row count equal to the sum of the input table row counts. The input column types have to be compatible with each other; if they're not you can introduce preprocessing filters in the command to make them so.

This example modifies two input tables so that they contain only identifier and galactic longitude and latitude columns before concatenating them to give a three-column output:

```
stilts tcat in1=t1.fits in2=t2.fits out=t.fits \
   cmd1='addskycoords fk5 galactic RA DEC GLON GLAT' \
   cmd1='keepcols "ID GLON GLAT"' \
   cmd2='keepcols "index gal_long gal_lat"'
```

3. VOTable-Specific Tools

Although the generic table tools listed in the previous section can operate on VOTables as well as other formats, some of the detailed structure permitted in VOTable documents may be lost as a consequence of interpreting them in terms of STIL's abstract model of what a table is (some per-table metadata, some per-column metadata, and a sequence of rows). The two tools listed below deal with VOTable document structures directly.

3.1. votcopy

The VOTable standard defines three encodings in which the data content of each table element can be represented: TABLEDATA, BINARY and FITS. Much existing VOTable software is able to produce/consume only the first of these, although being text-based it is less efficient than the others for transmission or processing. `votcopy` can take a VOTable document and convert the data encodings freely between these formats, leaving the rest of the XML structure untouched.

3.2. votlint

It is easy to make mistakes when writing VOTable documents and often hard to spot them except by seeing that software behaves unexpectedly; many, perhaps most, VOTable documents at large exhibit errors of varying degrees of seriousness. `votlint` is a VOTable validator designed to be used by authors and users of VOTable-producing software to identify such errors and thereby to ensure that the documents they publish conform to the relevant standards. The tool simply examines a VOTable document and emits warnings about erroneous or questionable usages, which it's then up to the user to address. It provides much more rigorous tests than simply validating against a DTD or schema can offer.

Acknowledgments. The bulk of the work for STILTS was performed under the now-terminated Starlink Project. Development and support is continuing, funded by the UK Particle Physics and Astronomy Research Council.

References

Taylor, M. B. 2005, will be given at a later time, [FM-3]