Blank Values in VOTable

Mark Taylor (Bristol)

IVOA Interop Meeting
IUCAA Pune

20 October 2011

`$Id: votable-nulls2.tex,v 1.8 2011/10/20 04:38:43 mbt Exp $`

Summary

There is a *subtle* issue with “blank” (null, NaN) values in VOTable

Note: NULL $\neq$ 0 is not in doubt!

Should we do something about it?

- . . . if so, what? . . .
- . . . and how?

History

- Issue not new
  - Has been in VOTable since (before) v1.1 (2004)
- Debate stimulated recently via TAP:
  - RDBMS/VOTable correspondance more closely scrutinised
  - Requirement to stream data from a service (SQL query)
  - Some people looking more closely at standards
    (rather than just hacking something that seems to work)
- Previous discussions
  - DAL list July 2011
  - (DAL list September 2011 — a bit peripheral)
  - VOTableIssues wiki page — mostly Tom McGlynn
- Pune discussions
  - Discussed in TCG meeting (Sunday 16 Oct)
  - Agreed to schedule splinter session for wider participation

VOTable DATA Encoding Refresher

- VOTable has three alternative data encoding mechanisms:
  - TABLEDATA *(widely used)*:
    ```
    <DATA>
      <TABLEDATA>
        <TR> <TD>M51</TD> <TD>202.43</TD> <TD>47.22</TD> </TR>
        <TR> <TD>M97</TD> <TD>168.63</TD> <TD>55.03</TD> </TR>
      </TABLEDATA>
    </DATA>
    ```
  - BINARY *(not much used)*:
    ```
    <DATA>
      <BINARY>
        <STREAM encoding="base64">
           TTUxAAAAAAAAAEBpTcKPXCj2QEecKPXCj1xNOTcAAAAAAAAAQGUUKPXCj1xAS4PX
           Cj1wpA==
        </STREAM>
      </BINARY>
    </DATA>
    ```
  - FITS *(hardly ever used?)*:
    ```
    <DATA>
      <FITS>
        <STREAM href="fcat-2.fits"/>
      </FITS>
    </DATA>
    ```
- These encode exactly the same data

VOTable Rules

Representation of “blank” values in VOTable columns:

- Varies by column data type:
  - Float scalars (`float`, `double`):
    - BINARY/FITS encoding: IEEE NaN bit pattern
    - TABLEDATA encoding: `<TD/>` or `<TD>NaN</TD>`
  - Integer scalars (`unsignedByte`, `short`, `int`, `long`):
    - nominated “magic” value (all encodings):

      ```
      <FIELD datatype="short" name="COUNT">
        <VALUES null="-32768"/>
      </FIELD>
      ```

    - Empty `<TD/>` not permitted! *(but often seen)*
  - Arrays (including `char[]` ≈ strings), complex, bit:
    - . . . are more complicated, but less important
- Summary:
  - No null/NaN/empty array distinction
  - Need to do work (choose non-data value) to write integer blanks
- Design motivation/benefits:
  - TABLEDATA ↔ BINARY ↔ FITS encoding transformations are lossless
  - All makes sense if you think in FORTRAN or FITS BINTABLE!

Problems

Consequences of VOTable encoding rules:

- Null is not distinguished from NaN/empty string/empty array

  either: omits fundamental element from value space *(RDBMS view)*

  or: chooses different model for numeric data than RDBMS *(FORTRAN view)*

- Choosing a magic value for integer columns can be problematic:
  - ▷ May need to examine all values in column to find an unused one

    → prevents streaming (magic value must be declared up front)
  - ▷ For shorter types (`unsignedByte`, `short`) there may be no unused values

Possible Workarounds

Suggestions to solve some or all of the problems:

- Permit empty TD elements for integers? (`<TD/>` = NULL)
- Change semantics of empty TD elements for floats (`<TD/>` = NULL not NaN)
- Add `null` attribute to TD element? (`<TD null="true">`)
- Add special column with bitmasks for each column?
- Some combination of these?
- Something else?
- Nothing?

Considerations:

- Which, if any, of these problems need to be solved?
- What is VOTable for? *(Delivering data to user code? DB⟶DB communication?)*
- Do we need to retain lossless TABLEDATA ↔ BINARY conversion?
- Do we need to retain BINARY/FITS encodings??
- Is backward compatibility important? Required?
  (VOTable 1.2 parser making sense of VOTable 1.3(?) document)

Option A: Empty TD Elements for Integers

Allow `<TD/>` in integer columns to represent NULL

- Change:
  - ▷ Currently empty TD for integer column is illegal
- Effect:
  - ▷ Solves streaming problem, for TABLEDATA encoding only
  - ▷ Solves problem of unavailable byte/short magic values, for TABLEDATA encoding only
- Compatibility:
  - ▷ Semantics is clear
  - ▷ Many VOTable producers already do it
  - ▷ Most VOTable consumers already understand it
  - ▷ Is effectively in unofficial use already

Option B: Empty TD Elements for Floats

Declare `<TD/>` in float columns to represent NULL

- Change:
  - Currently empty TD for float column means NaN
- Effect:
  - Solves NULL/NaN problem, for TABLEDATA encoding only
- Compatibility:
  - Backwardly incompatible semantic change:
    - Cells previously interpreted as NaN are now interpreted as NULL
    - ... but NaN and NULL are somewhat conflated in existing model anyway

Option C: TD “null” Attribute

Mark null values with `<TD null="true">...</TD>`

- Change:
  - Existing TD element has no attributes.
- Effect:
  - ▷ Solves all problems (NULL/NaN, streaming, unavailable magic values) for TABLEDATA encoding only
- Compatibility:
  - ▷ Parsers unaware of new attribute will either fail or ignore it

Option D: Bitmask

Mark NULLness of each cell using special non-data bitmask column:

```
<FIELD name="__NULLCOLS__" datatype="bit" arraysize="ncol"/>
```

- Change:
  - ▷ Currently, no non-data columns in VOTable
- Effect:
  - ▷ Solves all problems (NULL/NaN, streaming, unavailable magic values) for all encodings
  - ▷ Increases size of table
  - ▷ Table processing somewhat complicated
- Compatibility:
  - ▷ Unaware parsers may present/propagate bitmask column as data

Discussion

Questions

0: Who cares?

1: Which, if any, of the null issues need fixing?

a. Streaming is difficult for integers

b. NULL values sometimes impossible for short integers

c. No NULL/NaN distinction for floats

d. NULL arrays, NULL values in arrays, NULL bitmasks, NULL complex values problematic

2: What about the TABLEDATA/BINARY/FITS encodings?

a. Do the null fixes need to work for all of these encodings?

b. Do we need to retain all of the encodings?

c. Do we need to retain lossless convertability between all the encodings?

3: What fix(es)?

a. Permit empty TD elements for integers?

b. Change semantics of empty TD elements for floats?

c. Add null attribute to TD element?

d. Add new bitmask column to mark nulls? (For all or only BINARY(/FITS)?)

Procedure

- People not here should get a chance to contribute
- Options for changing permitted VOTable usage:
  - Update VOTable standard
    - ▷ VOTable WG is dormant
    - ▷ Revive VOTable WG?
    - ▷ Make a special TCG-sanctioned update?
  - Sanction illegal usages?
    - ▷ Issue a Note?
    - ▷ Turn a blind eye? (if applicable)

. . . probably TBD by TCG

While We're Here. . .

Are there other pressing issues with VOTable that need fixes?

- MIME type capable of specifying encoding? (TAPRegExt)
- . . . others?