Blank Values in VOTable

Mark Taylor (Bristol)

IVOA TCG Meeting
IUCAA Pune

16 October 2011

$Id: votable-nulls.tex,v 1.8 2011/10/12 17:33:16 mbt Exp $

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)
- Discussions
  - DAL list July 2011
  - (DAL list September 2011 — a bit peripheral)
  - VOTableIssues wiki page — mostly Tom McGlynn

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 special value (all encodings):

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

    - Empty `<TD/>` not permitted! *(but often seen)*
  - Arrays (including `char[]`/`unicodeChar[]` ≈ strings):
    - empty array? all elements null?
- Summary:
  - No null/NaN/empty array distinction
  - Need to do work (choose out of band value) to write integer blanks
- 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

Options:

- Permit empty TD elements for integers? (`<TD/>`)
  - ▷ Solves streaming problem, for TABLEDATA only
- Add `null` attribute to TD element? (`<TD null="true">`)
  - ▷ Solves null/NaN distinction, for TABLEDATA only
- Add special column with bitmasks for each column?
  (`<FIELD name="__NULLCOLS__" datatype="bit" arraysize="ncol"/>`)
  - ▷ Solves streaming and null/NaN, for all encodings
- All of the above (bitmasks for BINARY/FITS, `null` attribute for TABLEDATA)?
- 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??
- How much do different kinds of compatibility matter?

Procedure

- Options for changing permitted VOTable usage:
  - Update VOTable standard
    - VOTable WG is dormant
    - Revive it?
    - TCG handles update?
  - Sanction illegal usages?
    - Issue a Note?
    - Turn a blind eye?
- Rôle of TCG:
  - VOTable widely used in IVOA, cuts across WGs, suitable for TCG discussion ...
  - ... but discussions (especially leading to changes) should be open to wider IVOA

If you ask me ...

Sanction use of empty <TD/> elements

- Results:
  - Solves streaming problem (for TABLEDATA encoding)
  - Solves problem of unavailable byte/short magic values (for TABLEDATA encoding)
  - Does not solve null/NaN distinction
  - Abandons TABLEDATA $\leftrightarrow$ BINARY $\leftrightarrow$ FITS equivalence
- Compatibility
  - Semantics is clear
  - Many VOTable producers already do it
  - Most VOTable consumers already understand it
  - Is effectively in unofficial use already
- Effects:
  - Software of some conscientious VOTable producers more simple/efficient
  - Consciences of some conscientious VOTable producers eased
  - No change to software of sloppy producers and cautious consumers
  - No benefit for producers of BINARY/FITS encoded VOTables
  - VOTable standard needs update

... but not everyone might agree.