

VOTable 1.3

Applications WG

Mark Taylor (Bristol)

IVOA Interop
São Paulo

23 October 2012

`$Id: votable.tex,v 1.26 2012/10/23 01:18:40 mbt Exp $`

Outline

VOTable 1.3

- Status
- Issues
 - ▷ Null values
 - ▷ MIME type
 - ▷ VOTable/STC
 - ▷ SKOS Concepts
- Next steps

Status

Agreed at Urbana-Champaign (May 2012):

- Changes:
 - ▷ Improve Null handling:
 - Adjust TABLEDATA encoding definition
 - New BINARY2 encoding definition
 - ▷ Extend MIME type
 - Add optional parameter describing encoding type
 - ▷ Keep other changes to a minimum
- Schedule:
 - ▷ Oct 2012: Stable draft
 - ▷ May 2013: Implementations + RFC

Current status:

- First Draft [WD-VOTable-1.3-20121015](#) (in [Volute](#) repository)
- Prototype implementation of null handling in STIL (not released)

Nulls: TABLEDATA

Empty `<TD/>` element changes

- VOTable 1.2:
 - ▷ **Integer columns:** illegal
 - ▷ **Floating point columns:** meant NaN
 - ▷ **Variable-length array columns:** meant zero-length array

Nulls: TABLEDATA

Empty `<TD/>` element changes

- VOTable 1.2:
 - ▷ **Integer columns:** illegal
 - ▷ **Floating point columns:** meant NaN
 - ▷ **Variable-length array columns:** meant zero-length array
- VOTable 1.3:
 - ▷ Always means NULL

Nulls: BINARY2

- **BINARY** data encoding (VOTable 1.2):

... <FIELD type="short"><VALUES null="99"></FIELD> ...

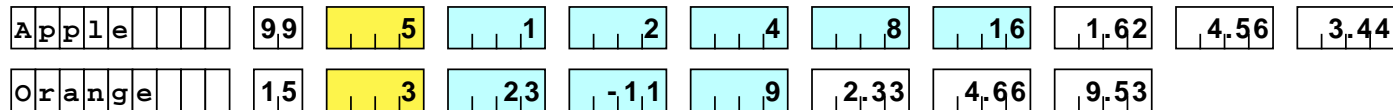
A	p	p	e				99	5	1	2	4	8	16	1.62	4.56	3.44
O	r	a	n	g	e		15	3	23	-11	9	2.33	4.66	9.53		

- Fixed length
- Length of variable length data
- Variable length data

Nulls: BINARY2

- **BINARY** data encoding (VOTable 1.2):

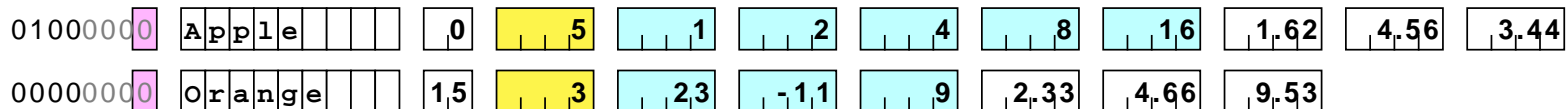
... <FIELD type="short"><VALUES null="99"></FIELD> ...



- Fixed length
- Length of variable length data
- Variable length data

- **BINARY2** data encoding (new in VOTable 1.3):

... <FIELD type="short"/> ...



- Fixed length
- Length of variable length data
- Variable length data
- Null flags (bits)

— **VALUES null** attribute no longer required for null cells

Nulls: Subtleties

- `<VALUES null="-32768">` no longer required to indicate null *cells*:
 - TABLEDATA has `<TD/>`
 - BINARY2 has nullness flags
 - FITS makes its own arrangements
 - BINARY is obsoleted by BINARY2⇒ Deprecate/eliminate `VALUES null`?
- Current status in WD:
 - `VALUES null` *must* be respected where present (even with TABLEDATA/BINARY2)
 - `VALUES null` can be used to indicate null *elements* of *array-valued* cells
 - ▷ No other way to do this (new TABLEDATA/BINARY2 mechanisms can't handle it)
 - ▷ This was probably(?) the intention in earlier versions of VOTable
 - ▷ This is how it works in FITS

Question: should we instead eliminate `VALUES null` altogether?

- 😊 Tidier, easier to implement
- ☹ No null elements of non-null arrays (⇒ *VOTable less expressive than FITS*)
 - Null integers impossible in BINARY → remove BINARY in 1.3?

BINARY2: Subtleties

- BINARY2 is better than BINARY?

- More expressive (null arrays possible, no forbidden values)
- Easier to produce (no need to choose null magic values up front)
- Slightly more bytes (typically $\lesssim 5\%$)

⇒ Eliminate/deprecate BINARY in VOTable 1.3?

- Current status in WD:

- Both BINARY and BINARY2 are permitted
 - ▷ Avoids confusion in parsers/documents that may confuse 1.2 and 1.3

Question: should we instead deprecate the old BINARY encoding?

- Disallow/deprecate BINARY encoding in VOTable 1.3 in favour of BINARY2?
- Redefine BINARY in VOTable 1.3 to have null flags, never use name BINARY2?
 - ☹ complicates MIME type paraterization (*q.v.*)

MIME Type

- VOTable 1.2 MIME Types

`text/xml`

`application/x-votable+xml`

MIME Type

- VOTable 1.2 MIME Types

`text/xml`

`application/x-votable+xml`

- VOTable 1.3: Add new **encoding** parameter for application type:

`text/xml`

`application/x-votable+xml; encoding=BINARY2` (*or other encodings*)

MIME Type

- VOTable 1.2 MIME Types

`text/xml`

`application/x-votable+xml`

- VOTable 1.3: Add new **encoding** parameter for application type:

`text/xml`

`application/x-votable+xml; encoding=BINARY2` (*or other encodings*)

- Encoding parameter:

- ▷ May take values `TABLEDATA`, `FITS`, `BINARY`, `BINARY2`

- ▷ Optional

- ▷ Not case-sensitive

- ▷ Must be absent if more than one encoding used in the same document

- ▷ Applies to `application/x-votable+xml` but not `text/xml`

- ▷ Allows content negotiation so older clients can avoid `BINARY2` VOTables

- Clarifications added:

- ▷ Parameter [Unicode] `charset` also permitted (as per RFC 3023)

- ▷ `text/xml` requires CRLF linebreaks (as per RFC 2046)

Question: Is name **encoding** OK?

- Term used in text for `TABLEDATA`, `BINARY` etc is “serialization”

- but TAPRegExt and STIL already reference **encoding** (in anticipation of standardisation)

STC References

- Example changed to match Note *Referencing STC in VOTable*.

- Before:

```
<GROUP utype="stc:AstroCoords" ID="J2000">
  <PARAM name="cooframe" datatype="char" arraysize="*" ucd="pos.frame"
    utype="stc:AstroCoords.coord_system_id" value="UTC-ICRS-TOP0"/>
  <FIELDref ref="col1"/>
  <FIELDref ref="col2"/>
</GROUP>
<FIELD name="RA" ID="col1" ucd="pos.eq.ra;meta.main"
  datatype="float" width="6" precision="2" unit="deg"
  ref="J2000" utype="stc:AstroCoords.Position2D.Value2.C1"/>
<FIELD name="Dec" ID="col2" ucd="pos.eq.dec;meta.main"
  datatype="float" width="6" precision="2" unit="deg"
  ref="J2000" utype="stc:AstroCoords.Position2D.Value2.C2"/>
```

- After:

```
<GROUP utype="stc:CatalogEntryLocation">
  <PARAM name="href" datatype="char" arraysize="*"
    utype="stc:AstroCoordSystem.href" value="ivo://STCLib/CoordSys#UTC-ICRS-TOP0"/>
  <PARAM name="URI" datatype="char" arraysize="*"
    utype="stc:DataModel.URI" value="http://www.ivoa.net/xml/STC/stc-v1.30.xsd"/>
  <FIELDref ref="col1" utype="stc:AstroCoords.Position2D.Value2.C1"/>
  <FIELDref ref="col2" utype="stc:AstroCoords.Position2D.Value2.C2"/>
</GROUP>
<FIELD name="RA" ID="col1" ucd="pos.eq.ra;meta.main"
  datatype="float" width="6" precision="2" unit="deg"/>
<FIELD name="Dec" ID="col2" ucd="pos.eq.dec;meta.main"
  datatype="float" width="6" precision="2" unit="deg"/>
```

SKOS Concepts

Requirement: Associate a SKOS concept with a table column

- Requirement from Theory WG, additional input from Semantics
- Associate other elements too (e.g. table **PARAM** as well as **FIELD**)?
- Several possibilities considered:
 - ▷ Use existing **LINK** element
 - ▷ Use new **META** element
 - ▷ Use new microformat attributes (**property** and **resource**) on **FIELDS**

SKOS Option 1: LINK

Use existing **LINK** element with `content-role="type"` (*Sébastien*)

```
<LINK content-role="type"
      href="http://purl.org/astrology/vocab/PhysicalQuantities/Distance"/>
```

- Details:
 - ▷ Works because SKOS concepts are syntactically URLs
 - ▷ More general than SKOS; any “type-like” relationship can be so labelled
 - ▷ Possible with existing **LINK** element — no VOTable change *necessary*
 - ▷ But semantics for `content-role="type"` must be defined somewhere
- Optional extras:
 - ▷ Define semantics of `content-role="type"` in VOTable doc
 - ▷ Re-write semantics of `content-role="doc"` in VOTable doc
 - ▷ Redefine `content-role` attribute as `anyURI` not `NMTOKEN` for namespaced roles
 - ▷ More detailed explanation of intended use of **LINK** element
- Issues:
 - ▷ Abuse of **LINK**? SKOS URLs not necessarily dereferenceable (not link *to* anywhere)
(*Norman: there is no distinction between URLs as names and addresses*)
 - ▷ Can accomodate non-SKOS type-like relationships, but only if syntactically URLs
 - ▷ Interferes with existing (questionable?) **LINK** usage within Aladin

SKOS Option 2: META

Introduce new element **META** (*Mark*)

```
<META name="skos-type"  
      value="http://purl.org/astro/vocab/PhysicalQuantities/Distance"/>
```

- Details:

- ▶ Just represents a key/value pair
- ▶ Position: anywhere a **LINK** can go
- ▶ Various possibilities for form of **name** attribute value:
 - “**type**” (generic)
 - “**http://example.org/Metadata/provenance**” (specific, namespaced)
 - ... anything else you like
- ▶ Names to be defined by clients (other standards, user communities), not in VOTable
- ▶ Rather (but not completely) general: could address future requests for new attributes etc
- ▶ Follows HTML usage (though syntax slightly different):
 - **LINK** references external resource *content*, **META** gives inline value

- Issues:

- ▶ Additional change to VOTable
(we said we weren't going to do that, but no backward compatibility issues)
- ▶ Only usable in VOTable 1.3+

SKOS Option 3: Enhanced META

As **META** but compatible with RDFa syntax (*Norman*)

- Syntax:

- ▷ Content model ANY (i.e. can include plain text, HTML, other things)
- ▷ Attributes `rel`, `content`, `resource`, `content-type` (+ others from RDFa?)

- Examples:

- ▷ `<META rel="rdf:type" resource="http://purl.org/astro/vocab/PhysicalQuantities/Distance"/>`
- ▷ `<META rel="http://example.ivoa.org/has-utype">foo:bar.baz</META>`
- ▷ `<META rel="http://example.ivoa.org/has-utype" content="foo:bar.baz"/>`
- ▷ `<META rel="doap:developer">A N Other</META>`
- ▷ `<META rel="doap:description" content-type="text/plain">`
`This data was produced by the Foo Project.</META>`
- ▷ `<META rel="doap:description" content-type="text/html">`
`<p>This data was produced by the <blink>Foo</blink> project</p></META>`

- Advantages (*additional to those of META*):

- ▷ Flexibility
- ▷ RDFa-aware tools can extract information

- Issues (*additional to those of META*):

- ▷ Flexibility?

SKOS Decision

Question: Should we:

- allow use of **LINK** element with no changes to standard?
- allow use of **LINK** with more semantic description of **content-role** in VOTable document?
- add simple **META** element?
- add RDFa-enhanced **META** element?
- use microformats?

VOTable Document

VOTable 1.3 WD document

- Document now resides in Volute <http://code.google.com/p/volute/>
- Apart from as required by changes above and build mechanics tried to leave text alone
- See all changes for yourself (`svn diff -r1828 projects/votable/votable.tex`)

Other Changes

Any other changes at VOTable 1.3?

hoping for the answer "No"

Next Steps

Plan:

- Schedule: Implementations + validator + RFC by May 2013
- Implementations:
 - ▷ STIL (→ STILTS, TOPCAT):
 - VOTable 1.3 input capability and output option
 - Empty TD = null TABLEDATA encoding I/O for v1.3
 - BINARY2 encoding I/O option for v1.3
 - META element?
 - ▷ ... others?
- Validator:
 - ▷ STILTS: `votlint` VOTable 1.3 validation
- Document:
 - ▷ Revised WD?
 - by January if required
 - ▷ PR + call RFC
 - before or after Heidelberg, depending on implementation experience