



Fig. 1



Fig. 2

1. What's new with RegTAP?

(cf. Fig. 1)

Markus Demleitner
msdemlei@ari.uni-heidelberg.de

(cf. Fig. 2)

- Standard Status
- Implementation Status
- Open questions

2. Standard Status

RegTAP is in RFC since 2013-12-03.

One major piece of feedback.

There's now a validation suite, accessible from the RFC page [Question: Where should this be hosted?]

A minor set of open questions will be discussed below.

3. Implementation Status

Server side

- Both GAVO and ESAVO implementations pass the validation suite
- VODance-based implementation still on an older draft
- VAO Registry: Theresa?

Client side

- <http://dc.g-vo.org/WIRR> is now published, if a bit rough. Also: uses `array_agg` cheat
- Mark Taylor is working on a prototype for a RegTAP interface for TOPCAT

4. Open Questions

Word of wisdom: If you write a standard, write a validation suite. It'll help you find ambiguities and gaps.

- `security_method` in `rr.interface` is still missing (there's just one record in the VO using it so far, and that's for a service that's dead, but the use cases appear fairly well-defined) → add `security_standard_id` column
- Require `unicodeChar` VOTable output for some strings (e.g., authors)? Or utf-8 encoded char? Or keep saying nothing? → opinions? Also, watch discussion over in VOTable.
- Should we deprecate `intf_role="std"` on grounds it doesn't work? And just codify current practices? (which boil down to: use `vs:ParamHTTP`-typed interfaces of a capability with a `standardId`)
- Should we require UDFs `string_agg` or even `array_agg`? → Implementors, please see how much effort that is for you.

5. Whitespace Normalisation?

Should we require ingestors to just indiscriminately strip whitespace? Even normalise it within strings?

Or: honor `xs:whiteSpace` (preserve, replace, collapse) – not used in VO registry so far. Plus:

- `xs:string` – all whitespace legal: e.g., `vstd:EndorsedVersion`, base for some enumerations like `vs:Waveband`
- `xs:normalizedString` – no `cr`, `lr`, `tab`: not used
- `xs:token` – normalised whitespace: most "free" strings in the VO

But of course, all this is theory. What an ingestor receives depends on whether the parser knows the schema, how it is configured in case the schema is ignored, as well as potentially the general mood of the machine. In short: it's a mess, and in the interest of defensive spec writing, I'd say we should mandate stripping and normalisation for all strings to simulate schema-aware parsing.

6. NULLs. Empty Strings. Oh My.

Unfortunately, we need to tell apart `''` and `NULL`.

The VOTable transport of TAP usually doesn't preserve `NULL` vs. `''`.

In the database, however, `IS NULL` is very different from `=''`.

I see two options:

1. All empty strings become `NULLs`
2. empty elements/attributes → `''`; missing e/a → `NULL`

People have so far gone for (2). (1) is a bit simpler, though. . .

7. Go forth and . . . Implement!

(clients, especially)