



Fig. 1



Fig. 2

1. Standards for Special IVORNs

(cf. Fig. 1)

Markus Demleitner
msdemlei@ari.uni-heidelberg.de

(cf. Fig. 2)

In the VO, we have several “magic” IVORNs:

- Dataset identifiers
- “Standard identifiers” in protocols, etc.

Since Waikoloa the need for clarifying both came up.

2. Dataset Identifiers

Natural habitat:

- SSA (publisher DID, creator DID)
- Obscore (publisher DID)

Why standardise them?

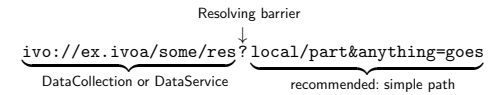
- General confusion
- Clarify they must resolve
- Historical # use potentially problematic – Norman was kind enough to write his objections up in the IVOA Note “URI Fragments in IVOA Specifications”
- Build for advanced functionality – see below

3. Proposed DID form

To ensure that:

- DID is resolvable
- The registry isn’t burdened with lots of datasets (its job are data collections)
- We keep IVORN fragment semantics within common URI customs

it’s almost inevitable to choose:



4. Accessing Datasets with DIDs?

The IVORN’s resource part (“the part before the ?”) can be resolved to a resource record.

SHOULD this

- have a datalink capability and/or
- have relationships to one or more services

allowing access to the dataset?

And if so, what sort of relationship?

I’ve tried to poll sentiments on this and actually received one reply, stating there should be a relation to some service (not necessarily datalink, but then we’d have to give rules how to actually access a dataset based on a PubDID using a service).

5. StandardIds

Natural habitat:

- `vr:capability/@standardId` – this is our standard way to locate services!
- `tr:dataModel/@ivo-id`

Why standardise them?

- Various forms around; for a given standard, it’s unclear which form applies, and disagreement reigns
- Versioning of these must be uniform
- Use StandardsRegExt’s facilities

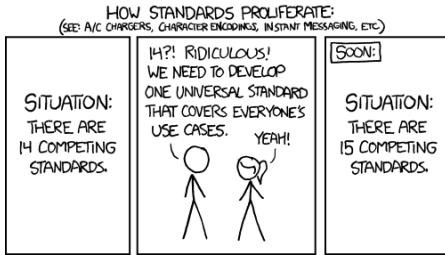


Fig. 3

6. Fun with URI forms

Here's some standardId forms found in the wild:

- Without version, full resource: `ivo://ivoa.net/std/registry`
- With version, full resource: `ivo://...eu/std/fullquery/v1.0`
- Without version, fragment: `ivo://ivoa.net/std/vosi#tables`
- With version, fragment: `ivo://org.../std/myspace/v1.0#myspace`
- With version in fragment: None **yet**

(cf. Fig. 3)

7. What to reference?

Considerations:

- StandardsRegExt lets you have versions within the resource record
- Metadata for minor versions of standards hopefully doesn't change (much)
- What about major versions?
- For a client to identify services it can operate, the major version should suffice
- For a client to find out where it can use extended features, the minor version is required
- StandardsRegExt doesn't have elements for major versions

8. standardId Number 15

I weighed up things. My result:

$$\underbrace{\text{ivo}://\text{ivoa.net}/\text{std}/\text{DataLink}}_{(\text{Service})?Standard} \# \overbrace{\text{links} - 1.0}^{\text{standard key}}$$

Note that the fragment *never* points to an `endorsedVersion` – these are not persistent. This also lets a single standard have multiple `standardIds` (think VOSI).

Consequences:

- Standards records will have to enumerate their endpoint versions
- Clients may have to query for partial identifiers (LIKE '...links-1.%')

9. Implications for Standards

I am now fairly convinced all this needs to go into a section of "IVOA Identifiers".

Can I have a mandate for starting this?

While I'm touching the doc: Does anyone still want the XML Format? The schema in Appendix A?