



Fig. 1



Fig. 2

1. Identifiers Validation

(cf. Fig. 1)

Markus Demleitner
 msdemlei@ari.uni-heidelberg.de

(cf. Fig. 2)

- Online validator for IVOIDs
- Ad-hoc API
- PubDIDs

2. Scope/Implementation

Identifiers 2.0 says

- IVOIDs are RFC 3986-compliant URIs
- having a scheme of `ivo://`
- without complicated mess in authority and path
- the non-local part of which resolves in the Registry.

Result: currently six groups of tests.

Source code¹ (weak dependency on DaCHS, strong dependency on python rfc3986) available.

¹ <http://svn.ari.uni-heidelberg.de/svn/gavo/hdinputs/ivoidval/bin>

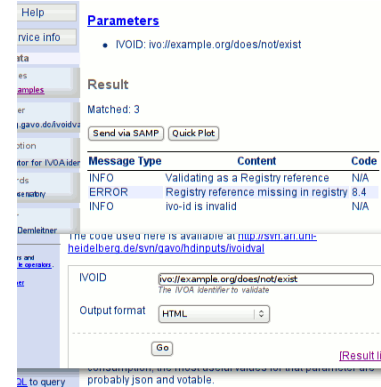


Fig. 3

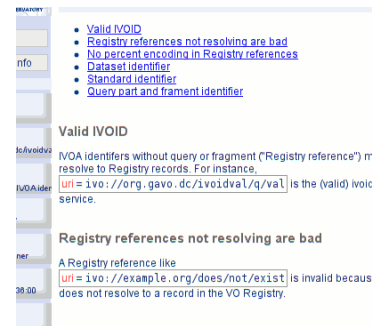


Fig. 4

3. Web Interface

<http://dc.g-vo.org/validate-identifier>

(cf. Fig. 3)

Note: It's a table.

Isn't everything a table?

4. Examples

It's got VOSI examples, too:

(cf. Fig. 4)

5. Ad-hoc API

VOSI-style:

```
http://dc.g-vo.org/ivoidval/q/val/api?MAXREC=0
$ curl -F uri="ivo://org.gavo.dc/?feros/data/f89411.vot" \
-F RESPONSEFORMAT=json http://dc.g-vo.org/ivoidval/q/val/api
{"data": [
  [{"INFO", "Validating as a full ivoid", null},
  [{"INFO", "ivo-id is valid", null}],
  "contains": "table"},
  "columns": [
    {"description": "Type of the message; an identifier is invalid if...",
      "name": "msg_type", "datatype": "char", "ucd": "", "arraysize": "*"},
      "id": "msg_type", "unit": ""},
    ...
  ]
}
```

Yes, I'm totally in favour of a canonical json mapping for VOTables.

6. A little extra: PubDIDs

These are dataset identifiers looking like:

```
ivo://org.gavo.dc/data?feros/data/f89411.vot
```

At <http://dc.g-vo.org/glopidir>, there's a tool letting data providers see if some common resolution strategies succeed.

Absent strong rules for PubDID, this is not really a validator. It's in the same "did I get it right?" league, though.

And it's got examples, too!

7. What's left?

IVOA identifiers are fairly plain.

But there's still many ways to get them wrong.

And bad ones have been found in the wild.

If in doubt, use a validator.