

Improving Validator resources



Tamara Civera Lorenzo

Scientific Database Engineer (CEFCA)

Vice-chair IG Operations

IVOA November 2023

IVOA Validators list

https://wiki.ivoa.net/twiki/bin/view/IVOA/IvoaValidatorsSummary

IVOA

Log In or Register

IVOA.net
 Wiki Home
 WebChanges
 WebTopicList
 WebStatistics

Twiki Meta & Help
 IVOA
 Know
 Main
 Sandbox
 TWiki

Twiki intro
 Twiki tutorial
 User registration
 Notify me

Working Groups
 Applications
 Data Access Layer
 Data Model
 Grid & Web Services
 Registry
 Semantics

Interest Groups
 Data Curation
 Education
 Knowledge Discovery
 Operations
 Radio Astronomy
 Solar System
 Theory
 Time Domain

Committees
 Sids&Procs

www.ivoa.net
 Documents
 Events
 Members
 XML Schema

Jump Search

TWiki > IVOA Web > IvoaOps > IvoaValidatorsSummary (2023-10-03, TamaraCivera) [Edit](#) [Attach](#)

IVOA Validation Capabilities

This page provides a comprehensive summary of the validation capabilities associated with published IVOA standards. Validators include any software which takes an entity purported to be compliant with one or more IVOA standards and tests this compliance. The entity may a string, a file, a library, or an on-line service. Generally validators may produce false positives, entities which seem to pass the validator but which may still be non-compliant. Only limited tests are done and some semantic aspects of a protocol may not be amenable to automated testing. However false negatives, where failure is indicated even though the entity is compliant, should be rare and should be reported to the party responsible for the validator.

The page is given as a table with entries corresponding to each of the entries in the IVOA standards page. IVOA members who develop validators or who know of validation capabilities should update this table as appropriate. In cases where multiple validators exist for a protocol, separate entries are given for each.

Group	Standard	Versions	Status	Discussion	Type
App					
	SAMP	1.3	Yes	The JSAMP library/toolkit includes a fairly comprehensive set of validation tests (hubtester) for a SAMP Hub. In principle this can test both Standard and Web Profiles, though the web client is faked (it's not done from a browser). No SAMP client validator is included - such a thing wouldn't be able to do much, but could be a useful addition.	Standalone
	VOTable	1.4,1.3,1.2,1.10,1.0	Yes	STILTS STILTS votlint provides comprehensive validation for VOTable 1.0, 1.1, 1.2, 1.3, 1.4, includes schema/DTD validation as well as much more stringent tests. Can be used standalone, but API has hooks for programmatic incorporation into other validators that need to validate VOTables (taplint does this). You can also validate against the VOTable schema (or for v1.0 the DTD) provided in the standard.	Standalone and library, External XML validation
				Astropy In any environment with Astropy installed, there is a <code>votlint</code> command. The same output report can be	



IVOA Validators list

- Validator developers:
 - Is there any new validator that is not in the list?
 - Write us (Steve/Tamara or the Opps list) to add it to the page.
- For developers/users:
 - Do you find this page useful and/or easy to use?
 - Do you think there is any missing information?

Validators useful features

- Must be easy to use.
- Must be well documented:
 - Documentation about how use them.
 - Documentation about the results codes and format.
- Could be used to check private or authenticated VO services (desirable).
- Could be easily added in VO services development workflow (automated tests) (desirable).

IVOA validators in automated tests

- Testing is an essential part of software development, especially to guarantee quality.
- VO services development and deployment must include IVOA validators in development testing process to check their compliance.
- It is necessary that IVOA validators could be easily added in these automated tests and in this development workflow.

IVOA validators in automated tests

Necessary features:

- IVOA validators must be accessible by machines:
 - Easy way to call the service and indicate the parameters.
- IVOA validators results:
 - Format understandable by machines: xml, json,...
 - Maybe defining a standard format for all validators (desirable).
- Allow testing services in development environments (desirable):
 - Services not available outside the internal network.
 - Solutions: standalone validators, validators libraries,...

Let's start the discussion...

Suggestions?
Comments?

