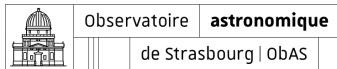


ADQL Validation

Grégory Mantelet¹

¹CDS (Centre de Données astronomiques de Strasbourg)

2nd November 2021



□ Table of Contents

1. Context
2. Lyonetia
3. And now?
4. ADQL queries
5. Validator
6. Next steps

□ 1. Context

- ADQL-2.1's next step: **RFC**
- **But...**
 - ... still no way to validate this new language (minor) version
 - ... to ensure it allows fulfilling the desired use-cases
- **Problem:** validation not possible with the grammar as provided in 2.0 and 2.1
 - no existing parser for the exact type of BNF used for ADQL

□ 2. Lyonetia

- **GitHub repository** initiated few years ago by Dave Morris:
<https://github.com/ivoa/lyonetia>
- 2 main goals:
 1. **To parse the BNF** and use it to validate ADQL
 - **Failed** due to the flavour of BNF not supported by tested parsers
 2. **To collect ADQL queries** to validate or expected to fail
 - 1 directory for each service (only 2 services: **GAVO** and the **Royal Observatory of Edinburgh (ROE)**)
 - ADQL queries to validate
 - generally a script to test the queries on the ADQL service and report the result

□ 3. And now?

- **Proposition:**
 - Keep collecting ADQL queries
 - Stop validating against online ADQL services
 - ... use instead a standalone off-line parser: [VOLLT/ADQL-Lib](#)

4. ADQL queries

Keep the same format already used in Lyonetia - XML:

```
1 <queries>
2
3   <contact>
4     <name>Markus Demleitner</name>
5     <url>mailto:msdemlei@ari.uni-heidelberg.de</url>
6   </contact>
7
8   <publisher>
9     <name>Heidelberg GAVO Data Center</name>
10    <url>http://dc.g-vo.org</url>
11  </publisher>
12
13  <description>ADQL geometry test queries</description>
14
15  <query uuid="ccd99070-4508-11e6-b60c-9d2c33f9b7a2">
16    <description>Simple geometry predicates</description>
17    <adql valid="true" version="adql-2.1">select x from y where Point(NULL, 2, 3)=x</adql>
18  </query>
19
20  ...
21
22 </queries>
```

□ 5. Validator

- Simple Java program running VOLLT/ADQL-Lib
 - A single JAR file
 - Can be started using Gradle while in development
- Can validate...
 - ... a single or multiple XML documents
 - ... a directory containing XML documents
- Reports possible in:
 - Text
 - Markdown (*for nice visualization in GitHub/CI*)
 - *anything else useful on demand*

5. Textual report

Starting validation...

```
=====
Origin       : File (/home/mantelet/Seafile/workspace/adql-
validation/sample/geometry.xml)
Publisher    : Heidelberg GAVO Data Center (http://dc.g-vo.org)
Contact      : Markus Demleitner (msdemlei@ari.uni-heidelberg.de)
Description  : ADQL geometry test queries
Available tests: 33
=====
```

1/33 - ccd99070-4508-11e6-b60c-9d2c33f9b7a2 - FAILED

```
Due to: a parsing error: Encountered "NULL". Was expecting one of: "(" "+" "-" "AVG"
"MAX" "MIN" "SUM" "COUNT" "\" <SCIENTIFIC_NUMBER> <UNSIGNED_FLOAT> <UNSIGNED_INTEGER>
"\" <REGULAR_IDENTIFIER_CANDIDATE> <REGULAR_IDENTIFIER_CANDIDATE>
(HINT: "NULL" is a reserved ADQL word in v2.1. To use it as a column/table/schema
name/alias, write it between double quotes.)
Executed in 4.998616ms
```

```
-----
Description: Simple geometry predicates
```

```
Query       : select x from y where Point(NULL, 2, 3)=x
Expected    : success
=====
```

2/33 - d2ed137e-4508-11e6-b60c-9d2c33f9b7a2 - FAILED

[...]

```
=====
Validation completed!
=====
```

```
Summary : 22/33 successful tests (66%)
```

```
Duration: total=0.10508757s, average=3.184471ms
=====
```




5. Markdown report - 1

ADQL Validation report

About the validation set

Key	Value
Origin	File (/home/mantelet/Seafile/workspace/adql-validation/sample/geometry.xml)
Publisher	Heidelberg_GAVO Data Center
Contact	Markus Demleitner
Description	ADQL geometry test queries
Available tests	33

Summary

GLOBAL RESULT	11 FAILED
Successful tests	22 (66%)
Total duration	<u>0.100464866s</u>
Average test duration	3.044389ms
Execution date (UTC)	Oct 21, 2021 2:17:29 PM

Validation tests

Listing

#	UUID	Status
1	ccd99070-4508-11e6-b60c-9d2c33f9b7a2	FAILED
2	d2ed137e-4508-11e6-b60c-9d2c33f9b7a2	FAILED

5. Markdown report - 2

35

[/70b0db-400z-11e0-8c9d-819z880z9z44z](#)

[UK](#)

1

- **UUID:** ccd99070-4508-11e6-b60c-9d2c33f9b7a2

- **Description:**

Simple geometry predicates

- **Query:**

```
select x from y where Point(NULL, 2, 3)=x
```

- **ADQL version:** v2.1

- **Expected:** success

- **Test:**

- **Duration:** 4.363201ms

- **Status:** FAILED

- **Error:**

Parsing error: Encountered "NULL". Was expecting one of: "(" "+" "-" "AVG" "MAX" "MIN" "SUM" "COUNT" "" <SCIENTIFIC_NUMBER> <UNSIGNED_FLOAT> <UNSIGNED_INTEGER> "" "" <REGULAR_IDENTIFIER_CANDIDATE> <REGULAR_IDENTIFIER_CANDIDATE> (HINT: "NULL" is a reserved ADQL word in v2.1. To use it as a column/table/schema name/alias, write it between double quotes.)

2

□ 6. Next steps

- ☑ Choose a standalone parser => *VOLLT/ADQLLib*
- **Write a validator** reading these ADQL queries sets and reporting validation results => *almost done*
- **Re-organize the Lyonetia repository**
- Collect ADQL queries => **need volunteers**
- Set up GitHub CI to automatically run the validator (*reports + badge in README.md*)
- Start the RFC of ADQL-2.1