NRC·CNRC

VOSI-1.2 prototype Implementation Status

Arati Kakadiya
Canadian Astronomy Data Centre

IVOA InterOp Nov 2026

Context and Overview

- Previous Session
  - Initial OpenAPI assembly for VOSI 1.2 and TAP 1.2 presented previously
- Progress:
  - Advanced in WD-VOSI-1.2 work by fully implementing and validating the OpenAPI specifications in our youcat TAP service
- Today's Focus
  - Live features running on the youcat TAP service.
    - Async Table Operations via UWS
    - Prototype Query-plan inspection feature
    - Granular Table Permissions via HTTP Headers
    - Demonstration of all these features

VOSI Table Operations (/table-update)

- UWS Pattern implementations
  - POST /table-update: Create an asynchronous table update job.
  - GET /table-update/{jobID}: Get a uws job description
  - POST /table-update/{jobID}/phase: change a uws job phase(execute/abort)
  - GET /table-update: List async table update jobs with filter(phase,after,last)
  - GET /table-update/{JobID}/phase: Get phase of a uws job

Index Creation Parameters & Capabilities

- Query Parameters on POST /table-update:
  - TABLE={name}: Target table name.
  - INDEX={columns}: Comma-separated column list (order matters for composite indexes).
  - INDEX_TYPE={enum}: List of Special qualifiers (e.g. unique, long-lat, x-y).
- Supported Index Types:
  - Regular Index: Default behavior when INDEX_TYPE is omitted.
  - Unique: Single or multi-column unique indexes.
  - Spatial Index: long-lat for 2-column spherical coordinates.
- Implementation Guardrails & Design Choices:
  - Cartesian indexes not yet supported
  - INDEX_TYPE structured as a list for future flexibility
  - Constraints: Blocked invalid combinations (eg. nesting/combining unique and long-lat together are rejected)

Metadata Directions

- Current Metadata Behavior:
  - indexed Flag Status: We do not update tap_schema.columns.indexed upon index creation.
  - Why? A single boolean flag on individual columns is misleading for multi-column or spatial indexes (e.g., marking ra and dec individually does not accurately represent a joint spatial index).
  - Next Steps: Patrick Dowler’s talk will cover this topic in more detail.

Prototype Query Plan Inspection Feature

- Why we built it
  - Needed a reliable way to prove that database query planners were actively using newly created user indices.
- How it works
  - Add `query-plan=true` to standard synchronous TAP queries endpoint.
  - Returns raw database execution plan in application/yaml format.
- Implementation Status
  - Deployed on youcat service for internal testing and verification. (Not yet advertised in public OpenAPI spec)

VOSI Table Permissions via HTTP Headers

- Header based design choice: Permissions are managed directly on /tables/{name} via HTTP request and response headers rather than XML/JSON body payloads.
- Benefit: Allows clean, atomic permission updates independent of table content or data uploads.
- Supported Headers:
  - x-vosi-auth-read: `true|false|'null'`
  - x-vosi-group-read: `<GMS group>|'null'`
  - x-vosi-group-write: `<GMS group>|'null'`
  - x-vosi-owner: `<user ID>`

Demonstration

1. Fetch query plan(before index creation)
2. Create Index on the ra and dec fields
3. Fetch query plan(after index creation)
4. Fetch permission headers

1. Query plan Sequential Scan

Query: `SELECT top 10 * FROM aratikakadiya_schema.ps_gaia_subset WHERE INTERSECTS(POINT('ICRS GEOCENTER', raps, decps), CIRCLE('ICRS GEOCENTER', 240.45, 28.6203, 0.0167)) = 1`

```
~/Desktop/demo ❯ curl -E ~/.ssl/cadcproxy.pem --location -d "LANG=ADQL&query-plan=true&QUERY=SELECT top 10 * FROM aratikakadiya_schema.ps_gaia_subset WHERE INTE
RSECTS(POINT('ICRS GEOCENTER', raps, decps), CIRCLE('ICRS GEOCENTER', 240.45, 28.6203, 0.0167)) = 1" https://ws-uv.canfar.net/youcat/sync
- Plan:
    Node Type: "Limit"
    Parallel Aware: false
    Async Capable: false
    Startup Cost: 0.00
    Total Cost: 298.60
    Plan Rows: 10
    Plan Width: 32
    Plans:
      - Node Type: "Seq Scan"
        Parent Relationship: "Outer"
        Parallel Aware: false
        Async Capable: false
        Relation Name: "ps_gaia_subset"
        Alias: "ps_gaia_subset"
        Startup Cost: 0.00
        Total Cost: 2986.00
        Plan Rows: 100
        Plan Width: 32
        Filter: "((spoint(radians(raps), radians(decps)))::scircle && '<(4.19664418642037 , 0.499518467908533) , 0.000291469985083053>'::scircle)"
```

Result: Sequential Scan (No Index Utilized)

Query plan Sequential Scan output

- Plan:
    Node Type: "Limit"
    Parallel Aware: false
    Async Capable: false
    Startup Cost: 0.00
    Total Cost: 298.60
    Plan Rows: 10
    Plan Width: 32
    Plans:
      - Node Type: "Seq Scan"
        Parent Relationship: "Outer"
        Parallel Aware: false
        Async Capable: false
        Relation Name: "ps_gaia_subset"
        Alias: "ps_gaia_subset"
        Startup Cost: 0.00
        Total Cost: 2986.00
        Plan Rows: 100
        Plan Width: 32
        Filter: "((spoint(radians(raps), radians(decps)))::scircle && '<(4.19664418642037 , 0.499518467908533) ,
0.000291469985083053>'::scircle)"

2. Create Index on the ra and dec fields

Step 1: Create UWS Job to create Index

```
curl -E ~/.ssl/cadcproxy.pem
    -d "TABLE=aratikakadiya_schema.ps_gaia_subset&
        INDEX=raps,decps&
        INDEX_TYPE=long-lat"
    https://ws-uv.canfar.net/youcat/table-update
```

```
~/Desktop/demo ❯ curl -E ~/.ssl/cadcproxy.pem -d "table=aratikakadiya_schema.ps_gaia_subset&index=raps,decps&index_type=long-lat" https://ws-uv.canfar.net/youcat/table-update
```

(Tip: Using `-v` will print the HTTP `Location` header containing the new `<jobID>` URL directly).

2. Create Index on the ra and dec fields

Step 2: Execute Job and get job details

```
[~/Desktop/demo ❯ curl -E ~/.ssl/cadcproxy.pem -d "PHASE=run" https://ws-uv.canfar.net/youcat/table-update/iallsnwnw51evui2/phase
[~/Desktop/demo ❯ curl -E ~/.ssl/cadcproxy.pem https://ws-uv.canfar.net/youcat/table-update/iallsnwnw51evui2
<?xml version="1.0" encoding="UTF-8"?>
<uws:job xmlns:uws="http://www.ivoa.net/xml/UWS/v1.0" xmlns:xlink="http://www.w3.org/1999/xlink" version="1.1">
  <uws:jobId>iallsnwnw51evui2</uws:jobId>
  <uws:runId />
  <uws:ownerId>aratikakadiya</uws:ownerId>
  <uws:phase>COMPLETED</uws:phase>
  <uws:quote>2026-11-01T22:04:51.011Z</uws:quote>
  <uws:creationTime>2026-10-31T22:04:51.011Z</uws:creationTime>
  <uws:startTime>2026-10-31T22:09:01.213Z</uws:startTime>
  <uws:endTime>2026-10-31T22:09:01.708Z</uws:endTime>
  <uws:executionDuration>14400</uws:executionDuration>
  <uws:destruction>2026-11-07T22:04:51.011Z</uws:destruction>
  <uws:parameters>
    <uws:parameter id="index">raps,decps</uws:parameter>
    <uws:parameter id="index_type">long-lat</uws:parameter>
    <uws:parameter id="table">aratikakadiya_schema.ps_gaia_subset</uws:parameter>
  </uws:parameters>
  <uws:results />
</uws:job>
```

3. Query Plan Index Scan (Query with Small Limit)

```
  ~/Desktop/demo ❯ curl -E ~/.ssl/cadcproxy.pem --location -d "LANG=ADQL&query-plan=true&QUERY=SELECT top 10 * FROM aratikakadiya_schema.ps_gaia_subset WHERE
INTERSECTS(POINT('ICRS GEOCENTER', raps, decps), CIRCLE('ICRS GEOCENTER', 240.45, 28.6203, 0.0167)) = 1" https://ws-uv.canfar.net/youcat/sync
- Plan:
    Node Type: "Limit"
    Parallel Aware: false
    Async Capable: false
    Startup Cost: 0.28
    Total Cost: 10.90
    Plan Rows: 10
    Plan Width: 32
    Plans:
      - Node Type: "Index Scan"
        Parent Relationship: "Outer"
        Parallel Aware: false
        Async Capable: false
        Scan Direction: "NoMovement"
        Index Name: "i_aratikakadiya_schema_ps_gaia_subset_raps_decps"
        Relation Name: "ps_gaia_subset"
        Alias: "ps_gaia_subset"
        Startup Cost: 0.28
        Total Cost: 106.53
        Plan Rows: 100
        Plan Width: 32
        Index Cond: "((spoint(radians(raps), radians(decps)))::scircle && '<(4.19664418642037 , 0.499518467908533) , 0.000291469985083053>'::scircle)"
```

Result: Index Scan utilized (efficient for highly selective small result sets).

Query Plan Index Scan output


```
- Plan:
    Node Type: "Limit"
    Parallel Aware: false
    Async Capable: false
    Startup Cost: 0.28
    Total Cost: 10.90
    Plan Rows: 10
    Plan Width: 32
    Plans:
      - Node Type: "Index Scan"
        Parent Relationship: "Outer"
        Parallel Aware: false
        Async Capable: false
        Scan Direction: "NoMovement"
        Index Name: "i_aratikakadiya_schema_ps_gaia_subset_raps_decps"
        Relation Name: "ps_gaia_subset"
        Alias: "ps_gaia_subset"
        Startup Cost: 0.28
        Total Cost: 106.53
        Plan Rows: 100
        Plan Width: 32
        Index Cond: "((spoint(radians(raps), radians(decps)))::scircle && '<(4.19664418642037 , 0.499518467908533) ,
0.000291469985083053>'::scircle)"
```

3. Query Plan Bitmap Scan (Query with No Limit)

```
~/Desktop/demo ❯ curl -E ~/.ssl/cadcproxy.pem --location -d "LANG=ADQL&query-plan=true&QUERY=SELECT * FROM aratikakadiya_schema.ps_gaia_subset WHERE
INTERSECTS(POINT('ICRS GEOCENTER', raps, decps), CIRCLE('ICRS GEOCENTER', 240.45, 28.6203, 0.0167)) = 1" https://ws-uv.canfar.net/youcat/sync
- Plan:
    Node Type: "Bitmap Heap Scan"
    Parallel Aware: false
    Async Capable: false
    Relation Name: "ps_gaia_subset"
    Alias: "ps_gaia_subset"
    Startup Cost: 2.15
    Total Cost: 104.45
    Plan Rows: 100
    Plan Width: 32
    Recheck Cond: "((spoint(radians(raps), radians(decps)))::scircle && '<(4.19664418642037 , 0.499518467908533) , 0.000291469985083053>'::scircle)"
    Plans:
      - Node Type: "Bitmap Index Scan"
        Parent Relationship: "Outer"
        Parallel Aware: false
        Async Capable: false
        Index Name: "i_aratikakadiya_schema_ps_gaia_subset_raps_decps"
        Startup Cost: 0.00
        Total Cost: 2.13
        Plan Rows: 100
        Plan Width: 0
        Index Cond: "((spoint(radians(raps), radians(decps)))::scircle && '<(4.19664418642037 , 0.499518467908533) , 0.000291469985083053>'::scircle)
"
```

Result: Bitmap Index Scan chosen by query planner to optimize larger scan volumes.

Query Plan Bitmap Scan output


```
- Plan:
    Node Type: "Bitmap Heap Scan"
    Parallel Aware: false
    Async Capable: false
    Relation Name: "ps_gaia_subset"
    Alias: "ps_gaia_subset"
    Startup Cost: 2.15
    Total Cost: 104.45
    Plan Rows: 100
    Plan Width: 32
    Recheck Cond: "((spoint(radians(raps), radians(decps)))::scircle && '<(4.19664418642037 , 0.499518467908533) ,
0.000291469985083053>'::scircle)"
    Plans:
      - Node Type: "Bitmap Index Scan"
        Parent Relationship: "Outer"
        Parallel Aware: false
        Async Capable: false
        Index Name: "i_aratikakadiya_schema_ps_gaia_subset_raps_decps"
        Startup Cost: 0.00
        Total Cost: 2.13
        Plan Rows: 100
        Plan Width: 0
        Index Cond: "((spoint(radians(raps), radians(decps)))::scircle && '<(4.19664418642037 , 0.499518467908533) ,
0.000291469985083053>'::scircle)"
```

4. Fetch permission headers

Get Permission Headers

```
~/Desktop/demo ❯ curl -E ~/.ssl/cadcproxy.pem \
--head \
https://ws-uv.canfar.net/youcat/tables/aratikakadiya_schema.ps_gaia_subset
HTTP/1.1 200
server: OpenCADC/cadc-rest
x-vo-authenticated: aratikakadiya
x-vosi-owner: aratikakadiya
x-vosi-auth-read: true
date: Thu, 05 Nov 2026 12:21:55 GMT
```

Questions?

