NAVO NASA Astronomical Virtual Observatories

HEASARC • IRSA
NED • MAST

# NAVO Archives: Adventures in ObsTAP & DataLink

May 2024

Presenter: Anastasia Laity (Caltech-IPAC/IRSA)

Contributors: Tom Donaldson (MAST), Tess Jaffe (HEASARC)

# Overview

The NASA Astronomical Virtual Observatories (**NAVO**) program coordinates the efforts of NASA astronomy archives in providing ***comprehensive*** and ***consistent*** access to NASA's astronomical data through ***standardized interfaces***. NAVO comprises:

- the Mikulski Archive at Space Telescope (MAST)
- the High Energy Astrophysics Science Archive Research Center (HEASARC)
- the NASA/IPAC Infrared Science Archive (IRSA)
- the NASA Extragalactic Database (NED)

**Today's talk** - grew out of monthly technical working group discussions**:**

- Status update on NAVO ObsTAP/DataLink services
- Lessons learned + ***questions*** for the community

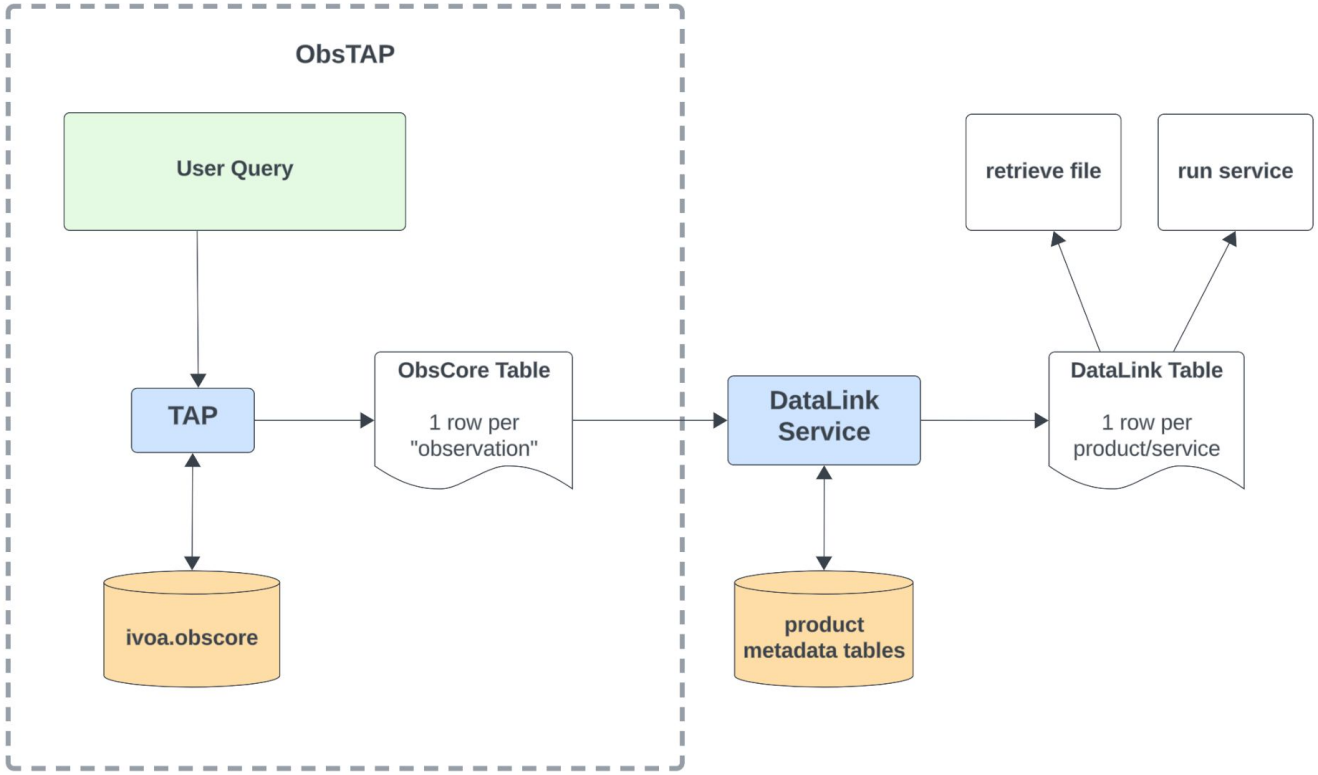
# Supporting cross-archive science and data discovery

Expanding the datasets available through ObsTAP and harmonizing how they appear at different NASA archives will:

- Improve the user experience for cross-archive use cases, including multi-wavelength and multi-messenger astronomy
- Ensure seamless integration and accessibility.

NAVO goal: a user searching for data on some region in the sky at some wavelength in some time frame can submit the same query to all archives and get consistent and understandable results.

# ObsTAP + DataLink




## What is ObsTAP?

- Use TAP
- Query **ivoa.obscore** (a database table or view)
- Result: **ObsCore Table**

## What is DataLink?

- **DataLink Service** (“links service”): input an ID
- Returns a **DataLink Table**
- Each row is a URL or a **service descriptor** reference

# Development Status: May 2024

| | ObsTAP | DataLink Service |
|---|---|---|
| **HEASARC** | **In development:**<br>• ivoa.obscore built explicitly from existing (pre-VO) HEASARC standard 'master' tables<br>• (no plans to use CAOM)<br>• Test endpoint (hidden) coming soon | **In production since 2019**. Enhancements planned:<br>• Flatten current hierarchical links model<br>• Implement ObsCore `obs_publisher_did` as unique identifier (currently a bespoke and not repeatable identifier is used)<br>• Cloud addressing to be added along with service descriptors (TBD) |
| **IRSA** | **In development:**<br>• ivoa.obscore: view of of CAOM tables; originally one row per data product but final version to reference DataLink | **In production:**<br>• Internal DataLink services driving some UIs<br><br>**In development:**<br>• General DataLink service (using CAOM metadata) to be integrated into ObsTAP and SxA |
| **MAST** | **In production:**<br>• ivoa.obscore: view of of CAOM tables; one row per data product<br><br>**In development:**<br>• ivoa.obscore: new CAOM view; multiple products per row (requires DataLink) | **In development:**<br>• Annotate ObsTAP results to show DataLink info for new obscore view.<br>• DataLink service to support queries based on new ObsTAP results |

NAVO NASA Astronomical Virtual Observatories
HEASARC • IRSA
NED • MAST

# Defining an “observation” / ObsCore row

Each row in ObsCore table can represent:

- A single file/service (Fig A)
- A set of related files/services grouped together using DataLink (Fig B)

Fig A: MAST

```
mast_single_obs_results = mast_caom_tap.service.run_sync(obscore_single_obs_query)
mast_single_obs_results.to_table()["obs_id", "obs_collection", "calib_level", "access_url"].show_in_notebook()
```

*Table length=3*

| idx | obs_id | obs_collection | calib_level | access_url |
|---|---|---|---|---|
| 0 | u29r4d02t | HST | 3 | https://mast.stsci.edu/portal/Download/file?uri=mast:HST/product/u29r4d02t_drw.fits |
| 1 | u29r4d02t | HST | 3 | https://mast.stsci.edu/portal/Download/file?uri=mast:HST/product/u29r4d02t_drw.jpg |
| 2 | u29r4d02t | HST | 3 | https://mast.stsci.edu/portal/Download/file?uri=mast:HST/product/u29r4d02t_drw_thumb.jpg |

Set of related data in DataLink table should have common values for all the ObsCore columns:

- Instrument, facility
- Data Product Type (image/spectrum/cube)
- Calibration level
- Spatial, energy, and time properties

Fig B: CADC (w DataLink)

```
cadc_single_obs_results = cadc_caom_tap.service.run_sync(obscore_single_obs_query)
```

```
cadc_single_obs_results.to_table()["obs_id", "obs_collection", "calib_level", "access_url"].show_in_notebook()
```

*Table length=2*

| idx | obs_id | obs_collection | calib_level | access_url |
|---|---|---|---|---|
| 0 | u29r4d02t | HST | 2 | https://ws.cadc-ccda.hia-iha.nrc-cnrc.gc.ca/caom2ops/datalink?ID=ivo%3A%2F%2Fcadc.nrc.ca%2Fmirror%2FHST%3Fu29r4d02t%2Fu29r4d02t-CALIBRATED |
| 1 | u29r4d02t | HST | 1 | https://ws.cadc-ccda.hia-iha.nrc-cnrc.gc.ca/caom2ops/datalink?ID=ivo%3A%2F%2Fcadc.nrc.ca%2Fmirror%2FHST%3Fu29r4d02t%2Fu29r4d02t-RAW_STANDARD |

***What about an “observation” that includes multiple data product types?***

***Or derived observations (i.e. mosaics) with multiple facilities, instruments?***

# Defining an "observation" / ObsCore row

IRSA and MAST use **CAOM** data model - **plan to follow CADC example:**

`access_url` : URL of DataLink query
`obs_publisher_did` : DataLink service input ID:

| obs_publisher_did | obs_collection | target_name | access_url | access_format |
|---|---|---|---|---|
| char | char | char | char | char |
| ivo://cadc.nrc.ca/CFHT?2955256/2955256g | CFHT | TOI1696 | https://ws.cadc-ccda.hia-iha.nrc-cnrc.gc.ca/caom2ops/datalink?ID=ivo%3A%2F%2Fcadc.nrc.ca%2FCFHT%3F2955256%2F2955256g | application/x-votable+xml;content=datalink |
| ivo://cadc.nrc.ca/CFHT?2955333/2955333o | CFHT | Engineering | https://ws.cadc-ccda.hia-iha.nrc-cnrc.gc.ca/caom2ops/datalink?ID=ivo%3A%2F%2Fcadc.nrc.ca%2FCFHT%3F2955333%2F2955333o | application/x-votable+xml;content=datalink |
| ivo://cadc.nrc.ca/CFHT?2959508/2959508o | CFHT | M43 | https://ws.cadc-ccda.hia-iha.nrc-cnrc.gc.ca/caom2ops/datalink?ID=ivo%3A%2F%2Fcadc.nrc.ca%2FCFHT%3F2959508%2F2959508o | application/x-votable+xml;content=datalink |

- The work of figuring out how to group artifacts happened when we converted metadata to CAOM
- ObsCore row = caom.plane row: **set** of artifacts / services with matching values of ObsCore columns
- DataLink table row = caom.artifact row

| ID | access_url | semantics | description | content_type |
|---|---|---|---|---|
| char | char | char | char | char |
| ivo://cadc.nrc.ca/CFHT?2955333/2955333o | https://ws.cadc-ccda.hia-iha.nrc-cnrc.gc.ca/raven/files/cadc:CFHT/295 | #preview | download cadc:CFHT/2955333o_preview_zoom_1024.jpg | image/jpeg |
| ivo://cadc.nrc.ca/CFHT?2955333/2955333o | https://ws.cadc-ccda.hia-iha.nrc-cnrc.gc.ca/raven/files/cadc:CFHT/295 | #this | download cadc:CFHT/2955333o.fits.fz | application/fits |
| ivo://cadc.nrc.ca/CFHT?2955333/2955333o | https://ws.cadc-ccda.hia-iha.nrc-cnrc.gc.ca/raven/files/cadc:CFHT/295 | #preview | download cadc:CFHT/2955333o_preview_1024.jpg | image/jpeg |
| ivo://cadc.nrc.ca/CFHT?2955333/2955333o | https://ws.cadc-ccda.hia-iha.nrc-cnrc.gc.ca/raven/files/cadc:CFHT/295 | #thumbnail | download cadc:CFHT/2955333o_preview_256.jpg | image/jpeg |

***Note: no individual files in this ObsCore view. Is ObsTAP expected to support searching by individual file properties? (ie "return only FITS files")?***

# Defining an “observation” / ObsCore row

- HEASARC: not built on CAOM but our own bespoke “master” table standard.
- Planning to replace access_urls with DataLinks and to “flatten” current DataLink hierarchy to be more similar to other archives
- Tentative plan:
  - One row for accessing the complete package (browsable directory, download scripts, tarballs); may include multiple data product types
  - Additional rows for each individual product (to support ObsTAP searches on data product types)
  -

```
[3]: obstap = vo.dal.TAPService("https://heasarc.gsfc.nasa.gov/xamin_test/vo/tap")
     result = obstap.search("select * from ivoa.obscore where facility_name like 'XMM%' and obs_id = '0904310601' ")
     result.to_table()['dataproduct_type','dataproduct_subtype','obs_title','obs_collection','obs_id','access_url','access_format']
```

[3]: *le length=34*

| taproduct_type | dataproduct_subtype | obs_title | obs_collection | obs_id | access_url | access_format |
|---|---|---|---|---|---|---|
| object | object | object | object | object | object | object |
| | directory | Complete XMM Observation | XMM | 0904310601 | https://heasarc.gsfc.nasa.gov/FTP/xmm/data/rev0/0904310601/ | |
| | directory | XMM Pipeline Products | XMM | 0904310601 | https://heasarc.gsfc.nasa.gov/FTP/xmm/data/rev0/0904310601/PPS/ | |
| image | | XMM Full EPIC Image | XMM/EPIC | 0904310601 | https://heasarc.gsfc.nasa.gov/FTP/xmm/data/rev0/0904310601/PPS/P0904310601EPX000OIMAGE8000.FTZ | fits |
| image | | XMM MOS1 Medium Filter Image | XMM/EPIC | 0904310601 | https://heasarc.gsfc.nasa.gov/FTP/xmm/data/rev0/0904310601/PPS/P0904310601M1S001IMAGE_2000.FTZ | fits |

Test URL is accessible, not advertised, though we haven’t yet made the above changes yet so now looks like this.

***Question: does this approach make sense in ObsTAP?***

# What to use for DataLink input ID?

For calling DataLink service from ObsTAP/S*x*A results :

- NAVO planning to use the ObsCore column `obs_publisher_did`
  - Next question: how do we populate `obs_publisher_did`?
- Something like: `ivo://`**`archiveID`**`/`**`obs_collection`**`/`**`unique_string`**
- HEASARC challenge: non-persistent/repeatable IDs.
- IRSA challenge: original CAOM conversion resulted in non-unique values of `obs_publisher_did`

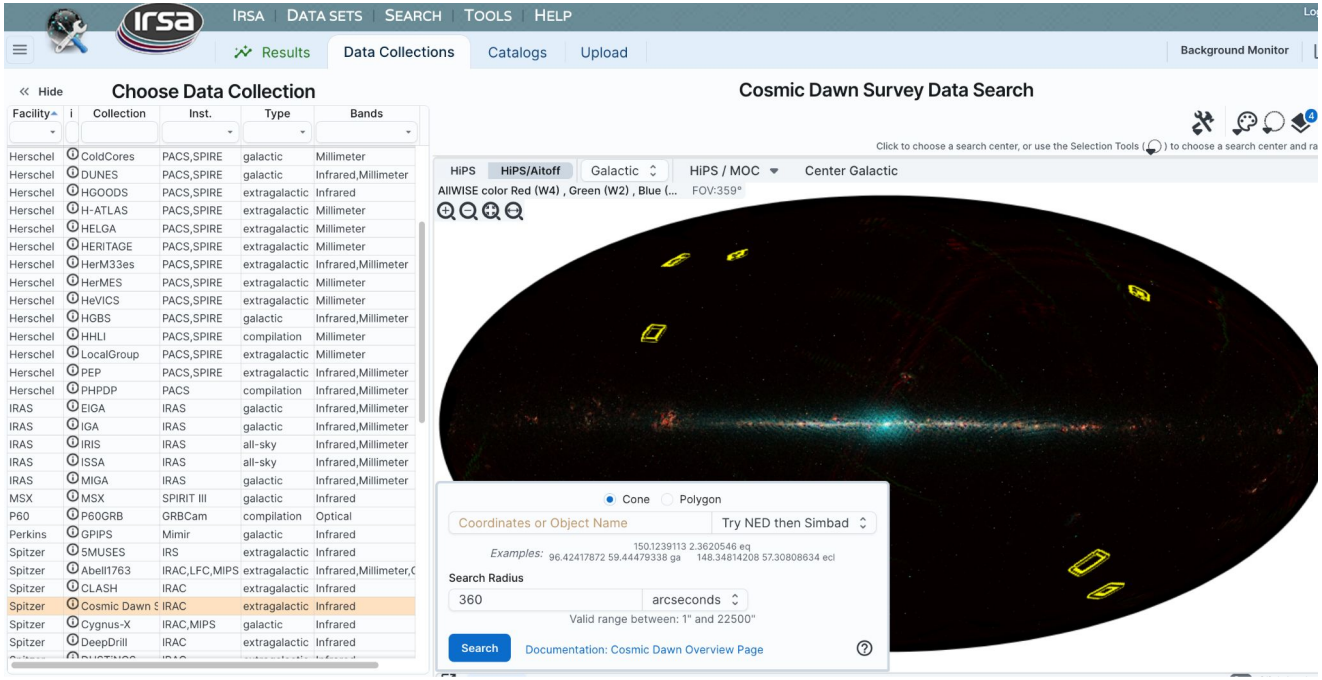
***Lesson learned: think AHEAD about the data flow between services.***

***Question: what are others using for DataLink ID and/or obs_publisher_did?***

# Multiple DataLink Services?

Not just for ObsTAP/SxA:

- Service descriptor in TAP results - use object ID to find all related files or services
- IRSA: using internal DataLink service to drive new Data Collection Explorer (for contributed datasets)




***Question: Support multiple ID types and behaviors in a single DataLink service? Or separate DataLink services/endpoints?***

# Fun with Databases

- Different archive DBMSes: PostGRES, SQLServer, Oracle
  - Different TAP support for spatial geometry queries

***Question: Do we need a way to communicate which types of spatial queries are supported?***

- Scaling / performance
  - ObsTAP queries can easily cause a full table scan
  - Indexing helps to a point - drawbacks to overuse
  - We can (should!) flag which columns are indexed in `TAP_SCHEMA.columns` - but can‘t force users to include them in queries.



# Contact Us


- IRSA - Anastasia Laity (anastasia.laity@caltech.edu)
- HEASARC - Tess Jaffe (tess.jaffe@nasa.gov)
- MAST - Tom Donaldson (tdonaldson@stsci.edu)


Contributors:

Antara Basu-Zych (HEASARC)
Kathryn Bello (MAST)
Tim Burke (IRSA)
Christine Chang (IRSA)
Tessa Dower (MAST)
Ben Falk (MAST)
Joshua Fraustro (MAST)
Meredith Gibb (HEASARC)
Justin Howell (IRSA)
Joyce Kim (IRSA)
Brian McLean (MAST)
David Rodriguez (MAST)
Judith Silverman (IRSA)
Scott Terek (IRSA)
Angela Zhang (IRSA)
Sarah Weissman (MAST)