

Advanced ways to find and retrieve data in the LTA

There are some useful ways to find and retrieve your data in the LTA that might not be immediately obvious. This page explains some of the more advanced options you have.

Queries

- You can use colons in numeric queries, to select ranges. This will for example give all observations and pipelines that have a SAS/Observation ID in the range from 432000 to 432190:

Observation Id	432000:432190
Observing or Pipeline Run Date	From 0000-00-00 00:00:00 To 0000-00-00 00:00:00
Project	any
Maximum Number of Rows	

In textual entries, wildcards can be used.

Target Name	3c19*
-------------	-------

- You can put a list of SAS/Observation IDs in the query:

Observation Id	146112,147775,151778
Observing Date	From 0000-00-00 00:00:00 To 0000-00-00 00:00:00

Viewing data

When you are looking at the results of a query you might see something like this:

Number Of Correlated DataProducts
0 / 488

This means that the observation is known in the LTA, it knows what data was produced, the produced data was not archived, but further processing happened on the raw data and the results of some of those pipelines were archived. If you click on the zero, you will see something like this:

#	<input type="checkbox"/>	DataProduct Identifier	SubArray Pointing Identifier	Subband	Stations	Observations	Pipeline	Derived DataProducts
1		7260485	293855	479	show	1		AveragingPipeline
2		7260483	293855	477	show	1		AveragingPipeline
3		7260488	293855	482	show	1	back to observation	AveragingPipeline
4		7260489	293855	483	show	1		AveragingPipeline
5		7260492	293855	486	show	1		AveragingPipeline
6		7260490	293855	484	show	1		AveragingPipeline
7	Can not be downloaded	7260493	293855	487	show	1		AveragingPipeline
8		7260486	293855	480	show	1		AveragingPipeline
9		7260487	293855	481	show	1	To pipeline	AveragingPipeline
10		7260482	293855	476	show	1		AveragingPipeline
11		7260491	293855	485	show	1		AveragingPipeline
12		7260484	293855	478	show	1		AveragingPipeline
13		7260436	293854	430	show	1		AveragingPipeline

This allows you to navigate from a pipeline back to the original observation, or from the observation to any pipelines that have run on the raw data.

Retrieving data

- You can retrieve data on the Observation and Pipeline level, you don't have to select all files individually.

#	<input type="checkbox"/>	Observation Id	Observing Mode	Antenna Set	Instrun Filter
1	<input checked="" type="checkbox"/>	146448	Interferometer	HBA Dual Inner	110-190
2	<input type="checkbox"/>	146447	Interferometer	HBA Dual Inner	110-190
3	<input checked="" type="checkbox"/>	146446	Interferometer	HBA Dual Inner	110-190
4	<input type="checkbox"/>	146445	Interferometer	HBA Dual Inner	110-190
5	<input checked="" type="checkbox"/>	146444	Interferometer	HBA Dual Inner	110-190
6	<input checked="" type="checkbox"/>	146443	Interferometer	HBA Dual Inner	110-190
7	<input type="checkbox"/>	146442	Interferometer	HBA Dual Inner	110-190
8	<input checked="" type="checkbox"/>	146441	Interferometer	HBA Dual Inner	110-190
9	<input checked="" type="checkbox"/>	146456	Interferometer	HBA Dual Inner	110-190
10	<input checked="" type="checkbox"/>	146455	Interferometer	HBA Dual Inner	110-190
11	<input type="checkbox"/>	146454	Interferometer	HBA Dual Inner	110-190
12	<input type="checkbox"/>	146453	Interferometer	HBA Dual Inner	110-190
13	<input type="checkbox"/>	146452	Interferometer	HBA Dual Inner	110-190

- If you have a query with more than 1000 results, you can open the multiple pages each in a separate tab/window.

Observation 1001 to 1100 (showing 100 of total 1156) ▾

edit columns | stage selected

first | previous | ... | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | next | last

nr Of SubArray	Start Time	Duration	Nr Stations	Nr Stations
----------------	------------	----------	-------------	-------------

- With the small triangle next to a list, you can fold or unfold the list to get a better overview.

Folded entries

Observation 1 to 100 (showing 100 of total 1156) ▲

Averaging Pipeline 1 to 100 (showing 100 of total 4060) ▲

Calibration Pipeline (total 30) ▲

Imaging Pipeline (total 0) ▲

UnspecifiedProcess 1 to 100 (showing 100 of total 125) ▲

Unfolded entries

Calibration Pipeline (Total 30) -

48 columns - skip column

#	Pipeline Name	Pipeline Version	Process Identifier	Observation ID	Start Time	Duration (s)	End Time	Strategy Name	Strategy Description	Frequency Integration	Time Integration	Flag Rate Correction	Systemic Offset	Scintillation	Number Of Instrument Points	Number Of Generated DataPoints	Source Reference
1	PIPE_CALIBRATION_V1	1.0.0	100001	100001	2023-01-01 10:00:00	0.5	2023-01-01 10:00:30	Preprocessing Pipeline (see desc)	Preprocessing with default	10	5	0	0	40	data		
2	PIPE_CALIBRATION_V1	1.0.0	100002	100002	2023-01-01 10:00:30	0.5	2023-01-01 10:01:00	Preprocessing Pipeline (see desc)	Preprocessing with default	10	5	0	0	40	data		
3	PIPE_CALIBRATION_V1	1.0.0	100003	100003	2023-01-01 10:01:00	0.5	2023-01-01 10:01:30	Preprocessing Pipeline (see desc)	Preprocessing with default	10	5	0	0	40	data		
4	PIPE_CALIBRATION_V1	1.0.0	100004	100004	2023-01-01 10:01:30	0.5	2023-01-01 10:02:00	Preprocessing Pipeline (see desc)	Preprocessing with default	10	5	0	0	40	data		
5	PIPE_CALIBRATION_V1	1.0.0	100005	100005	2023-01-01 10:02:00	0.5	2023-01-01 10:02:30	Preprocessing Pipeline (see desc)	Preprocessing with default	10	5	0	0	40	data		
6	PIPE_CALIBRATION_V1	1.0.0	100006	100006	2023-01-01 10:02:30	0.5	2023-01-01 10:03:00	Preprocessing Pipeline (see desc)	Preprocessing with default	10	5	0	0	40	data		
7	PIPE_CALIBRATION_V1	1.0.0	100007	100007	2023-01-01 10:03:00	0.5	2023-01-01 10:03:30	Preprocessing Pipeline (see desc)	Preprocessing with default	10	5	0	0	40	data		
8	PIPE_CALIBRATION_V1	1.0.0	100008	100008	2023-01-01 10:03:30	0.5	2023-01-01 10:04:00	Preprocessing Pipeline (see desc)	Preprocessing with default	10	5	0	0	40	data		
9	PIPE_CALIBRATION_V1	1.0.0	100009	100009	2023-01-01 10:04:00	0.5	2023-01-01 10:04:30	Preprocessing Pipeline (see desc)	Preprocessing with default	10	5	0	0	40	data		
10	PIPE_CALIBRATION_V1	1.0.0	100010	100010	2023-01-01 10:04:30	0.5	2023-01-01 10:05:00	Preprocessing Pipeline (see desc)	Preprocessing with default	10	5	0	0	40	data		
11	PIPE_CALIBRATION_V1	1.0.0	100011	100011	2023-01-01 10:05:00	0.5	2023-01-01 10:05:30	Preprocessing Pipeline (see desc)	Preprocessing with default	10	5	0	0	40	data		
12	PIPE_CALIBRATION_V1	1.0.0	100012	100012	2023-01-01 10:05:30	0.5	2023-01-01 10:06:00	Preprocessing Pipeline (see desc)	Preprocessing with default	10	5	0	0	40	data		
13	PIPE_CALIBRATION_V1	1.0.0	100013	100013	2023-01-01 10:06:00	0.5	2023-01-01 10:06:30	Preprocessing Pipeline (see desc)	Preprocessing with default	10	5	0	0	40	data		
14	PIPE_CALIBRATION_V1	1.0.0	100014	100014	2023-01-01 10:06:30	0.5	2023-01-01 10:07:00	Preprocessing Pipeline (see desc)	Preprocessing with default	10	5	0	0	40	data		
15	PIPE_CALIBRATION_V1	1.0.0	100015	100015	2023-01-01 10:07:00	0.5	2023-01-01 10:07:30	Preprocessing Pipeline (see desc)	Preprocessing with default	10	5	0	0	40	data		
16	PIPE_CALIBRATION_V1	1.0.0	100016	100016	2023-01-01 10:07:30	0.5	2023-01-01 10:08:00	Preprocessing Pipeline (see desc)	Preprocessing with default	10	5	0	0	40	data		
17	PIPE_CALIBRATION_V1	1.0.0	100017	100017	2023-01-01 10:08:00	0.5	2023-01-01 10:08:30	Preprocessing Pipeline (see desc)	Preprocessing with default	10	5	0	0	40	data		
18	PIPE_CALIBRATION_V1	1.0.0	100018	100018	2023-01-01 10:08:30	0.5	2023-01-01 10:09:00	Preprocessing Pipeline (see desc)	Preprocessing with default	10	5	0	0	40	data		
19	PIPE_CALIBRATION_V1	1.0.0	100019	100019	2023-01-01 10:09:00	0.5	2023-01-01 10:09:30	Preprocessing Pipeline (see desc)	Preprocessing with default	10	5	0	0	40	data		
20	PIPE_CALIBRATION_V1	1.0.0	100020	100020	2023-01-01 10:09:30	0.5	2023-01-01 10:10:00	Preprocessing Pipeline (see desc)	Preprocessing with default	10	5	0	0	40	data		
21	PIPE_CALIBRATION_V1	1.0.0	100021	100021	2023-01-01 10:10:00	0.5	2023-01-01 10:10:30	Preprocessing Pipeline (see desc)	Preprocessing with default	10	5	0	0	40	data		
22	PIPE_CALIBRATION_V1	1.0.0	100022	100022	2023-01-01 10:10:30	0.5	2023-01-01 10:11:00	Preprocessing Pipeline (see desc)	Preprocessing with default	10	5	0	0	40	data		
23	PIPE_CALIBRATION_V1	1.0.0	100023	100023	2023-01-01 10:11:00	0.5	2023-01-01 10:11:30	Preprocessing Pipeline (see desc)	Preprocessing with default	10	5	0	0	40	data		
24	PIPE_CALIBRATION_V1	1.0.0	100024	100024	2023-01-01 10:11:30	0.5	2023-01-01 10:12:00	Preprocessing Pipeline (see desc)	Preprocessing with default	10	5	0	0	40	data		
25	PIPE_CALIBRATION_V1	1.0.0	100025	100025	2023-01-01 10:12:00	0.5	2023-01-01 10:12:30	Preprocessing Pipeline (see desc)	Preprocessing with default	10	5	0	0	40	data		
26	PIPE_CALIBRATION_V1	1.0.0	100026	100026	2023-01-01 10:12:30	0.5	2023-01-01 10:13:00	Preprocessing Pipeline (see desc)	Preprocessing with default	10	5	0	0	40	data		
27	PIPE_CALIBRATION_V1	1.0.0	100027	100027	2023-01-01 10:13:00	0.5	2023-01-01 10:13:30	Preprocessing Pipeline (see desc)	Preprocessing with default	10	5	0	0	40	data		
28	PIPE_CALIBRATION_V1	1.0.0	100028	100028	2023-01-01 10:13:30	0.5	2023-01-01 10:14:00	Preprocessing Pipeline (see desc)	Preprocessing with default	10	5	0	0	40	data		
29	PIPE_CALIBRATION_V1	1.0.0	100029	100029	2023-01-01 10:14:00	0.5	2023-01-01 10:14:30	Preprocessing Pipeline (see desc)	Preprocessing with default	10	5	0	0	40	data		
30	PIPE_CALIBRATION_V1	1.0.0	100030	100030	2023-01-01 10:14:30	0.5	2023-01-01 10:15:00	Preprocessing Pipeline (see desc)	Preprocessing with default	10	5	0	0	40	data		

Imaging Pipeline (Total 8) -

UnspecifiedProcess 5 to 100 (skipping 100 of total 120) -

48 columns - skip column

Rep / process / 1 /

DBView

There is a server that gives the option to run your own queries on the database

<http://lofar-dbview.target.rug.nl/>

A useful query might be this one, that gives you all files for a certain Obs Id (SAS VIC tree ID).

```
SELECT fo.URI, dp."dataProductType", dp."dataProductIdentifier",
       dp."processIdentifier"
FROM AWOPER."DataProduct+" dp,
      AWOPER.FileObject fo,
      AWOPER."Process+" pr
WHERE dp."processIdentifier" = pr."processIdentifier"
      AND pr."observationId" = '123456'
      AND fo.data object = dp."object id"
```

```
AND dp."isValid" > 0
```

In this '123456' should be replaced with the Obs Id of an Observation/Pipeline you're looking for.

AstroWise Python Interface

There is also a python interface to the LTA. With this you can also script some advanced queries. To have this working, you first need to install the [LTA client](#) in your machine. Once you have installed the client, set up your user name and password. These are the same as for MoM. Remember that this is just a different user interface to the LTA catalog: you will need the same credentials as for the web interface.

In your home directory, add the following to the file <your home directory>/.awe/Environment.cfg

```
database_user : <your username>
database_password : <your password>
```

Finally, if you are using your own laptop, your hostname cause issuing an error, if it does not contain a full domain. In this case, check your /etc/hosts file. You should find a line that looks like this

```
127.0.0.1 localhost
```

Change that line into this

```
127.0.0.1 localhost <your_host_name>
```

Now, you may still see some warnings, but the following script will print out all observations on a certain patch of the sky:

```
# python code
from pprint import pprint
from common.database.Context import context
from awlofar.main.aweimports import Observation, Pointing, SubArrayPointing
result = {}
for project in sorted(context.get_projects()) :
    print "Project %(project)s" % vars()
    ok = context.set_project(project)
    # do your query
    obs_ids = set()
    query = (Pointing.rightAscension > 95) & \
            (Pointing.rightAscension < 105) & \
            (Pointing.declination > 20) & \
            (Pointing.declination < 30)
    print "Total Pointings %d" % len(query)
    for pointing in query :
        print "Pointing found RA %f DEC %f" % (pointing.rightAscension,
        pointing.declination)
        query_subarr = SubArrayPointing.pointing == pointing
        for subarr in query_subarr:
```

```
        query_obs = Observation.subArrayPointings.contains(subarr)
        for obs in query_obs :
            obs_ids.add(obs.observationId)
    result[project] = sorted(list(obs_ids))
    print result[project]

pprint(result)
```

From:

<https://www.astron.nl/lofarwiki/> - **LOFAR Wiki**

Permanent link:

https://www.astron.nl/lofarwiki/doku.php?id=public:lta_tricks&rev=1457258982

Last update: **2016-03-06 10:09**

