



Updated heliophysics services in VESPA: science products, service design and capabilities

Baptiste CECCONI
Chloé AZRIA

Heliophysics services in VESPA

(Virtual European Solar and Planetary Access)

- Heliophysics:
Solar physics, interplanetary medium, planetary magnetospheres/plasma
- Remote sensing: images, spectra, dynamic spectra, events, cubes
In-situ: time-series, dynamic spectra, events
Modelled: images, spectra, time-series, dynamic spectra, events, cubes
- VESPA: TAP table compliant with EPNcore dictionary
- Updated services from ObsParis and Nançay

Updated Heliophysics Services

- BASS2000: daily solar images at various wavelengths
- HFC1AR/HFC1T3: heliophysics features: active regions, radio bursts
- NRH: Nançay Radio Heliograph: images, movies
- ORFEES: Nançay solar monitoring antenna: dynamic spectra
- NDA: Nançay Decameter Array: dynamic spectra (Sun and Jupiter)
- MASER (Voyager/PRA, Cassini/Kronos, Wind/Waves, Juno/Waves...): dynamic spectra, events, times-series (Sun and planets)
- EXPRES: modelled dynamic spectra (Jupiter)
- **Update:** Move to DaCHS 2.5, add datalink support, new metadata ingestion methods...

Updates and new features

- Using new metadata ingestion method.
- Using datalink when possible, to link with:
 - progenitor data
 - quicklooks (often several formats available)
 - documentation / metadata
 - data access API (see below)
- Data access API for time-series and dynamic spectra (data streaming):
 - HAPI (Heliophysics API: <https://github.com/hapi-server>)
 - Das2 (temporal resampling on the y: <http://das2.org>)
- Serve catalogues of spectral-temporal features (using TFCat format)
- Serve collections and datasets associated to recent publications (e.g., supplementary material)

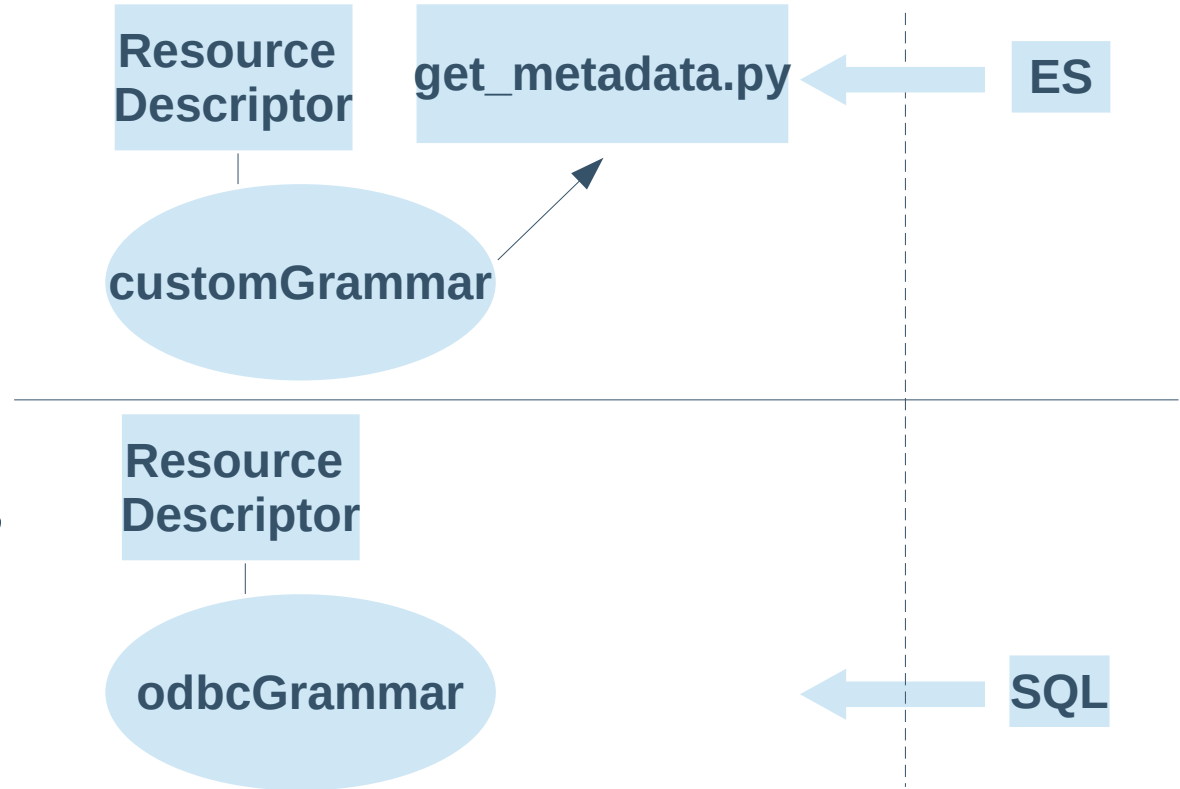
EPN-TAP Helio Services

Technical aspects

DaCHS

EPN-TAP standard

- NRH / ORFEES
 - CustomGrammar
 - Elasticsearch
- bass2000, hfc1ar, hfc1t3
 - odbcGrammar



NRH/ORFEES

CustomGrammar

Gathering metadata

- CustomGrammar calls the `get_metadata.py`
- Json containing query parameters
- `Elasticsearch.helpers.scan`
 - To iterate over query result

```
class RowIterator(CustomRowIterator):
    def _iterRows(self):
        with open(self.sourceToken, 'r') as f:
            rdsb_search_params = json.load(f)
            headers={'Content-type' : 'application/json'}

            es = Elasticsearch(rdsb_search_params['query_url'], verify_certs=False)

            results = elasticsearch.helpers.scan(es,
                index="nrh",
                query= rdsb_search_params['query_data'])

            for item in results:
                #print(item["_source"])
                md = my_metadata(item['_id'], item['_source'])

            yield md
```

[Extract of get_metadata.py](#)

NRH thumbnails / datalinks

VESPA Virtual European Solar and Planetary Access

Refine your search [ADQL Query](#) [Back To Services Results](#)

Main Parameters

Target Name

Target Class

Dataproduct Type

Instrument Host Name

Instrument Name

Processing level

Time

Location

Spectral

Illumination

Data Reference

Optional

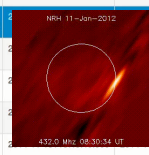
Other

Results in service NRH

NRH - Nancy Radio Heliograph Observation database
Service description to be provided
Credits:
Publisher: PADC/CDN

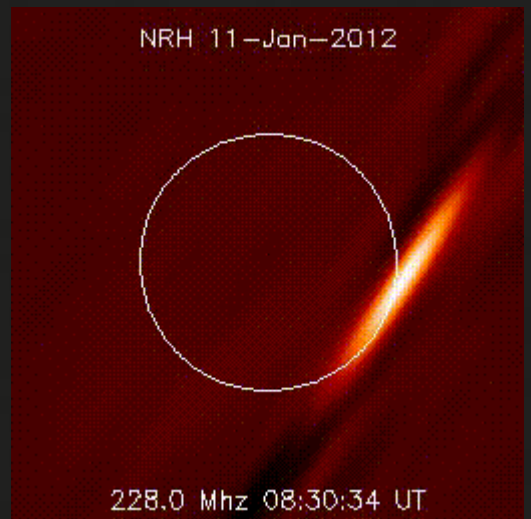
Column visibility Show all Hide all
Select All in current page Reset Selection

granule_uid	dataproduct_type	target_name	time_min (d)	time_max (d)	access_url	datalink
zXnyA30BB1GYkXdcvHcl	image	Sun	2004-05-31T08:18:37.999	2004-05-31T09:33:43.000	https://rsdb.obs-nan...	SEN
zXnVAH0BB1GYkXdcYhlo	image	Sun	2012-01-11T08:29:34.000	2012-01-11T11:21:04.999	https://rsdb.obs-nan...	SEN
zXnUA0BB1GYkXdcjD1m	image	Sun	2014-09-02T14:59:06.000	2014-09-02T15:20:59.000	https://rsdb.obs-nan...	SEN
ZXnsAX0BB1GYkXdcWEEE	image	Sun	2014-07-27T14:19:16.000	2014-07-27T14:19:16.000	https://rsdb.obs-nan...	SEN
zXnOBH0BB1GYkXdcNZ4U	image	Sun	2014-07-06T14:16:50.000	2014-07-06T14:16:50.000	https://rsdb.obs-nan...	SEN
zXnoAX0BB1GYkXdcseEBN	image	Sun	2014-07-13T15:04:58.999	2014-07-13T15:04:58.999	https://rsdb.obs-nan...	SEN
ZXnkAX0BB1GYkXdcdzzy	image	Sun	2014-07-29T08:29:08.999	2014-07-29T11:20:41.000	https://rsdb.obs-nan...	SEN
zXnJA30BB1GYkXdcPgyg	image	Sun	1998-03-03T08:18:15.999	1998-03-03T15:47:58.000	https://rsdb.obs-nan...	SEN
zXnhBX0BB1GYkXdcWMTY	image	Sun	2010-02-17T08:34:33.999	2010-02-17T11:26:06.000	https://rsdb.obs-nan...	SEN
zXnhAX0BB1GYkXdcj@Q	image	Sun	2014-10-18T08:06:23.999	2014-10-18T10:57:56.000	https://rsdb.obs-nan...	SEN
zXnFB40BB1GYkXdcv@k4u	image	Sun	2009-09-11T11:18:44.000	2009-09-11T14:09:48.000	https://rsdb.obs-nan...	SEN



SEN

https://rsdb.obs-nancy.fr/QL/Nrh/gif/nrh11012012_2280.gif



VESPA Virtual European Solar and Planetary Access

Refine your search

Main Parameters

Target Name

Target Class

Dataproduct Type

Instrument Host Name

Instrument Name

Processing level

Datalink

GIF preview at frequency = 1509 MHz

GIF preview at frequency = 1509 MHz

GIF preview at frequency = 2280 MHz

GIF preview at frequency = 4080 MHz

GIF preview at frequency = 4320 MHz

Close Submit

NRH - Nancy Radio Heliograph Observation database
Service description to be provided
Credits:
Publisher: PADC/CDN

Column visibility Show all Hide all
Select All in current page Reset Selection

granule_uid	dataproduct_type	target_name	time_min (d)	time_max (d)	access_url	datalink
zXnyA30BB1GYkXdcvHcl	image	Sun	2004-05-31T08:18:37.999	2004-05-31T09:33:43.000	https://rsdb.obs-nan...	SEN
zXnVAH0BB1GYkXdcYhlo	image	Sun	2012-01-11T08:29:34.000	2012-01-11T11:21:04.999	https://rsdb.obs-nan...	SEN
zXnUA0BB1GYkXdcjD1m	image	Sun	2014-09-02T14:59:06.000	2014-09-02T15:20:59.000	https://rsdb.obs-nan...	SEN

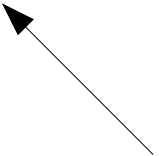
Bass2000, hfc1ar, hfc1t3 odbcGrammar

Contains connection chain



```
<sources pattern="data/driver.txt">
```

```
<odbcGrammar query="SELECT * FROM  
hfc1.view_sp_hqi JOIN hfc1.sunspots ON  
hfc1.view_sp_hqi.ID_SUNSPOT=hfc1.sunspots.ID_S  
UNSPOT LIMIT 100">
```




SQL query

Hfc1ar : s_region active regions

- From Chaincode to s_region

- Chaincode : contour of the active region
 - Each number in the chain defines the location of next pixel
- Coordinate conversions :
 - Pixels , original record referential
 - chaincode
 - Helioprojective : center of the sun, angular radius
 - Pairs of coordinates (x,y)
 - Carrington : spherical coordinates in an absolute referential
 - Pairs of coordinates (lon,lat)



```
sunpy.net.helio.chaincode
```

```
astropy.coordinates.SkyCoord  
transform_to  
sunpy.coordinates frames
```

s_region

- Transform (lon,lat) pairs into s_region polygons
- Only simple polygons allowed
 - shapely.is_simple function
 - Simple polygons
 - « Polygon lon1 lat1 lon2 lat2 ... »
 - Multi polygons
 - Shapely.make_simple function returns a list of shapes
 - Several polygons : regroup in one, small circle to make the union
 - Other shapes : (eg : line, multipolygon) solutions to be found

Show 10 entries

Column visibility: Show all Hide all

Select All in current page Reset Selection

granule_uid	dataprodu	target_name	time_min (d)	time_max (d)	access_url	s_region
ar_20220419_035533_808_2610	catalogue_item	Sun	2022-04-19T03:55:32.999	2022-04-19T03:55:32.999	ftp://fpbass2000.ob...	
ar_20220419_035533_3362_1165	catalogue_item	Sun	2022-04-19T03:55:32.999	2022-04-19T03:55:32.999	ftp://fpbass2000.ob...	Polygon UNKNOWNFrame 248.64
ar_20220419_035533_3182_2569	catalogue_item	Sun	2022-04-19T03:55:32.999	2022-04-19T03:55:32.999	ftp://fpbass2000.ob...	Polygon UNKNOWNFrame 201.18
ar_20220419_035533_3088_1756	catalogue_item	Sun	2022-04-19T03:55:32.999	2022-04-19T03:55:32.999	ftp://fpbass2000.ob...	Polygon UNKNOWNFrame 190.70
ar_20220419_035533_2718_2600	catalogue_item	Sun	2022-04-19T03:55:32.999	2022-04-19T03:55:32.999	ftp://fpbass2000.ob...	Polygon UNKNOWNFrame 178.24
ar_20220419_035533_2356_1474	catalogue_item	Sun	2022-04-19T03:55:32.999	2022-04-19T03:55:32.999	ftp://fpbass2000.ob...	Polygon UNKNOWNFrame 164.96
ar_20220419_035533_2273_2721	catalogue_item	Sun	2022-04-19T03:55:32.999	2022-04-19T03:55:32.999	ftp://fpbass2000.ob...	Polygon UNKNOWNFrame 162.65
ar_20220419_035533_1695_1480	catalogue_item	Sun	2022-04-19T03:55:32.999	2022-04-19T03:55:32.999	ftp://fpbass2000.ob...	Polygon UNKNOWNFrame 135.86
ar_20220419_035533_1404_1233	catalogue_item	Sun	2022-04-19T03:55:32.999	2022-04-19T03:55:32.999	ftp://fpbass2000.ob...	Polygon UNKNOWNFrame 124.02
ar_20220419_035533_1298_1432	catalogue_item	Sun	2022-04-19T03:55:32.999	2022-04-19T03:55:32.999	ftp://fpbass2000.ob...	Polygon UNKNOWNFrame 118.83

Showing 1 to 10 of 1,153,779 entries 3 rows selected

Data Selection - Metadata Selection - All Data - All Metadata -

Download
Send Table

SAMP

VESPA



Aladin Beta

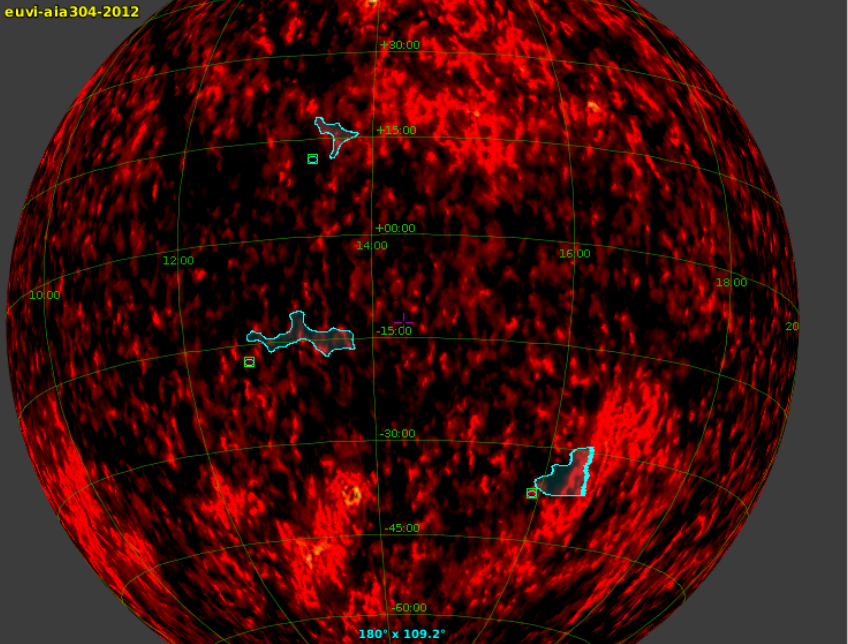
s_region

Aladin v11.0 *** BETA VERSION (based on v11.024) ***

Catalogue Graphique Couverture Outil Vue Interop Aide

Commande: 15:53:18.35954 -37:13:53.5977 Référentiel: ICRS Projection: Aitoff

Sun evui-aia304-2012



Warning: You are probably using an incompatible spatial reference (planets vs sky). This incompatibility is ignored in this beta release (test phase).

c2max	c3min	c3max	s_region	c1 resol min	c1 resol max	c2 resol m
..-29.182031031..			FoV			
..-10.708683328..			FoV			
..18.535449782..			FoV			

Chercher

257.403000 -77.399

14:18:03.22 -12:43:51.8

180° x 109.2°

hfc1ar: update

Heavy database

odbcGrammar with **update** : new **<makeQuery>** element (DaCHS 2.5.5)

```
<table id="epn_core" onDisk="True" adql="True" primary="id_ar" dupePolicy="dropOld" >
<data id="import" updating="True">
  <sources pattern="data/driver.txt"/>
  <odbcGrammar>
    <makeQuery>
      <code>
        try:
          with base.getTableConn() as conn:
            print(next(conn.query("SELECT MAX(time_min) FROM \schema.epn_core"))[0])

            localMax = next(conn.query("SELECT MAX(time_min) FROM \schema.epn_core"))[0]
            fragment = " WHERE (jdint + jdfraction) >= {}".format(escapeSQL(localMax))
            fragment=str(fragment)
            print(fragment)

          except base.DBError as msg:
            base.ui.notifyWarning(f"{msg} while harvesting: full re-harvest")
            fragment = ""

        return f"SELECT * FROM hfc1.view_ar_hqi JOIN hfc1.activeregions USING (id_ar) {fragment}"

      </code>
    </makeQuery>
  </odbcGrammar>
</data>
</table>
```

Build a fragment of the odbc query

Access the postgres database of the service – previously imported

**Return the final odbc query
Default : query without fragment**

Daily cron with service import



Updated heliophysics services in VESPA: science products, service design and capabilities

support.epntap@obspm.fr

Baptiste CECCONI
Chloé AZRIA



Updated heliophysics services in VESPA: science products, service design and capabilities

Baptiste CECCONI
Chloé AZRIA

Heliophysics services in VESPA

(Virtual European Solar and Planetary Access)

- Heliophysics:
Solar physics, interplanetary medium, planetary magnetospheres/plasma
- Remote sensing: images, spectra, dynamic spectra, events, cubes
In-situ: time-series, dynamic spectra, events
Modelled: images, spectra, time-series, dynamic spectra, events, cubes
- VESPA: TAP table compliant with EPNcore dictionary
- Updated services from ObsParis and Nançay

Updated Heliophysics Services

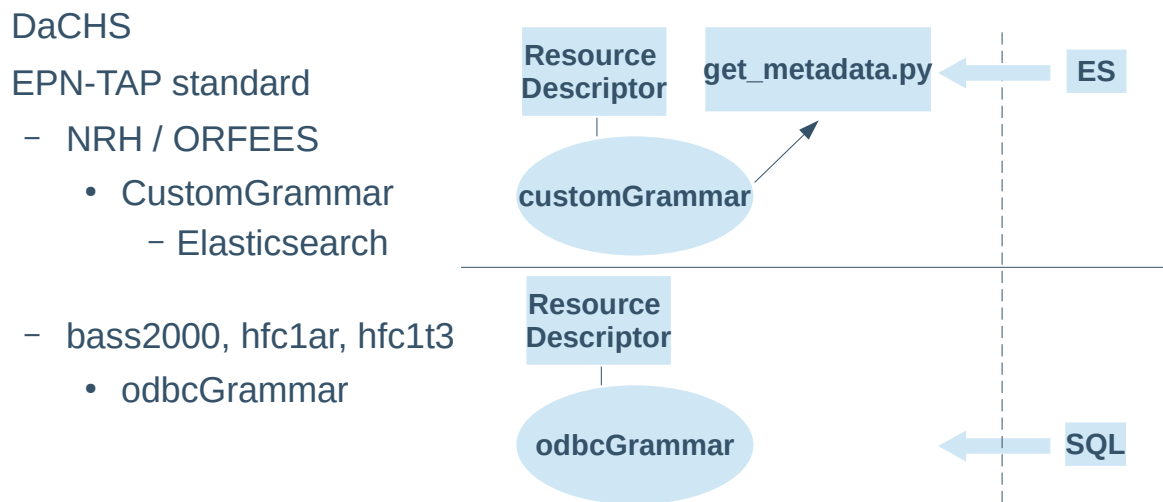
- BASS2000: daily solar images at various wavelengths
- HFC1AR/HFC1T3: heliophysics features: active regions, radio bursts
- NRH: Nançay Radio Heliograph: images, movies
- ORFEES: Nançay solar monitoring antenna: dynamic spectra
- NDA: Nançay Decameter Array: dynamic spectra (Sun and Jupiter)
- MASER (Voyager/PRA, Cassini/Kronos, Wind/Waves, Juno/Waves...): dynamic spectra, events, times-series (Sun and planets)
- ExPRES: modelled dynamic spectra (Jupiter)

- **Update:** Move to DaCHS 2.5, add datalink support, new metadata ingestion methods...

Updates and new features

- Using new metadata ingestion method.
- Using datalink when possible, to link with:
 - progenitor data
 - quicklooks (often several formats available)
 - documentation / metadata
 - data access API (see below)
- Data access API for time-series and dynamic spectra (data streaming):
 - HAPI (Heliophysics API: <https://github.com/hapi-server>)
 - Das2 (temporal resampling on the y: <http://das2.org>)
- Serve catalogues of spectral-temporal features (using TFCat format)
- Serve collections and datasets associated to recent publications (e.g., supplementary material)

EPN-TAP Helio Services Technical aspects



5

I will talk about technical aspects of other helophysics services we have updated.

These services are set up with DaCHS and following EPN-TAP standards.

The services NRH and ORFEES, provides data from nancay radio observations. It is deployed using the customgrammar method of dachs. Which means the resource descriptor calls an external python script. This python script interrogates the elasticsearch database to extract metadata and fill the table.

The services bass2000, hfc1 active regions and hfc type 3 uses odbc grammar method to fill the epn_core table from SQL databases.

NRH/ORFEES CustomGrammar

Gathering metadata

- CustomGrammar calls the `get_metadata.py`
- Json containing query parameters
- `Elasticsearch.helpers.scan`
 - To iterate over query result

```
class RowIterator(CustomRowIterator):
    def _iterRows(self):
        with open(self.sourceToken, 'r') as f:
            rsdb_search_params = json.load(f)
            headers={'Content-type': 'application/json'}

            es = Elasticsearch(rsdb_search_params['query_url'], verify_certs=False)

            results = elasticsearch.helpers.scan(es,
                index="nrh",
                query= rsdb_search_params['query_data'])

            for item in results:
                #print(item["_source"])
                md = my_metadata(item['_id'], item['_source'])

            yield md
```

[Extract of get_metadata.py](#)

6

The python script here called by the customgrammar method of dachs must contain the iterow method. It uses the files given in the sources element of the resource descriptor to make the import. Here the only file needed is a json file containing the query applied to the elasticsearch database.

As we want to obtain all the results matching this query and not just a sample like elasticsearch provides natively, we use elasticsearch helpers scan that iterates over query results.

Then, the function `my_metadata` written somewhere else in the python script makes a mapping.

**NRH
thumbnails /
datalinks**

https://rsdb.obs-nancay.fr/QL/Nrh/gif/nrh11012012_2280.gif

NRH 11-Jan-2012
228.0 Mhz 08:30:34 UT

Here you can see the quicklooks of the Nancay radioHeliograph service which are gifs video-plots. The thumbnail is chosen as the higher-frequency available video-plot. In datalinks we provide a list of alternative quicklooks at different frequencies. (excluding some frequencies where there is too much noise). This is done by making a post query to the database in the datalink element to obtain links to the gifs with the matching unique_id.

Bass2000, hfc1ar, hfc1t3 odbcGrammar

Contains connection chain

```
<sources pattern="data/driver.txt">  
<odbcGrammar query="SELECT * FROM  
hfc1.view_sp_hqi JOIN hfc1.sunspots ON  
hfc1.view_sp_hqi.ID_SUNSPOT=hfc1.sunspots.ID_S  
UNSPOT LIMIT 100">
```

SQL query

8

For the services bass 2000, hfc1 active regions and hfc1 type 3 that uses odbcgrammar. In the resource descriptor, the sources links to the driver (the remote database connection information)

And the SQL source database is interrogated with a query into the odbcgrammar element.


Hfc1ar : s_region active regions


- From Chaincode to s_region

- Chaincode : contour of the active region
 - Each number in the chain defines the location of next pixel
- Coordinate conversions :
 - Pixels , original record referential
 - chaincode

- Helioprojective : center of the sun, angular radius
 - Pairs of coordinates (x,y)

- Carrington : spherical coordinates in an absolute referential
 - Pairs of coordinates (lon,lat)

 `sunpy.net.helio.chaincode`

 `astropy.coordinates.SkyCoord`
`transform_to`
`sunpy.coordinates.frames`

One of the difficulties of this update was to add s_regions on the active regions service.

The information provided initially was chaincodes of the contour. A chaincode is a string of numbers where each number represents a shift to the next pixel.

So we had to make the conversion between the original chaincode in the original record pixel referential to helioprojective pairs of x,y positions. The helioprojective referential is a projected referential centered on the sun and the x and y axis increment are the angular radius. We do that using `sunpy.net.helio.chaincode` and the informations given in the database.

Then, the next step is to convert these projected coordinates on carrington heliographic coordinates, which is a spherical referential centered on the center of the sun that does not depends on the observator location. The `astropy.coordinates.SkyCoord` allows to create an instance of a position in a given referential defined thanks to `sunpy.coordinates.frames` so we can use the `transform_to` method to obtain the pairs of coordinates in carrington.

s_region

- Transform (lon,lat) pairs into s_region polygons
- Only simple polygons allowed
 - shapely.is_simple function
 - Simple polygons
 - « Polygon lon1 lat1 lon2 lat2 ... »
 - Multi polygons
 - Shapely.make_simple function returns a list of shapes
 - Several polygons : regroup in one, small circle to make the union
 - Other shapes : (eg : line, multipolygon) solutions to be found

Another difficulty of the s_region build is that DaCHS only handle simple polygons. So, the contour must not cross itself.

For the already-simple polygons, we basically write the s_region with the pairs of longitude and latitude.

For the other, we try to use the make_simple function that returns a list of shapely shapes. If everything happens like we want, it returns a list of polygons and we make the union using a small circle that makes the original polygons a simple one.

There still some cases where the make_simple does not return a list of polygons, but some other types of shapes.

In this case, we do not yet have a solution.

This case happens mostly when the active regions is located at limbs on the observation

HFC1AR - HelioPhysics Feature Catalog active regions
Solar active regions extracted from HelioPhysics Feature Catalog

Credits:
Contact: Jean Aboudarham
Contributing: Xavier Bonin, Christian Renski, PAIC
Publisher: Paris Astronomical Data Centre - LPDA

Show 10 entries
Column visibility Show all Hide all
Select All in current page Reset Selection

granule_id	diaproduct_type	target_name	time_min (d)	time_max (d)	access_url	s_region
ar_20220419_030533_808_2610	catalogue_item	Sun	2022-04-19T03:55:32.999	2022-04-19T03:55:32.999	ftp://fpbase2000.ob...	Polygon UNKNOWNFrame 146.04
ar_20220419_030533_3182_2890	catalogue_item	Sun	2022-04-19T03:55:32.999	2022-04-19T03:55:32.999	ftp://fpbase2000.ob...	Polygon UNKNOWNFrame 201.19
ar_20220419_030533_3686_1750	catalogue_item	Sun	2022-04-19T03:55:32.999	2022-04-19T03:55:32.999	ftp://fpbase2000.ob...	Polygon UNKNOWNFrame 166.70
ar_20220419_030533_2718_2600	catalogue_item	Sun	2022-04-19T03:55:32.999	2022-04-19T03:55:32.999	ftp://fpbase2000.ob...	Polygon UNKNOWNFrame 178.24
ar_20220419_030533_2256_1474	catalogue_item	Sun	2022-04-19T03:55:32.999	2022-04-19T03:55:32.999	ftp://fpbase2000.ob...	Polygon UNKNOWNFrame 164.58
ar_20220419_030533_2273_2721	catalogue_item	Sun	2022-04-19T03:55:32.999	2022-04-19T03:55:32.999	ftp://fpbase2000.ob...	Polygon UNKNOWNFrame 162.65
ar_20220419_030533_1695_1480	catalogue_item	Sun	2022-04-19T03:55:32.999	2022-04-19T03:55:32.999	ftp://fpbase2000.ob...	Polygon UNKNOWNFrame 135.66
ar_20220419_030533_1404_1233	catalogue_item	Sun	2022-04-19T03:55:32.999	2022-04-19T03:55:32.999	ftp://fpbase2000.ob...	Polygon UNKNOWNFrame 124.02
ar_20220419_030533_1298_1432	catalogue_item	Sun	2022-04-19T03:55:32.999	2022-04-19T03:55:32.999	ftp://fpbase2000.ob...	Polygon UNKNOWNFrame 118.83

Show 1 to 10 of 1,103,779 entries - 3 rows selected

Data Selection - Metadata Selection - All Data - All Metadata

Download
Send Table

SAMP

VESPA

↓

Aladin Beta

s_region

Aladin v11.0 *** BETA VERSION (based on v11.024) ***

Catalogue Graphique Couverture Outil Vue Interop Aide

Commande: 15:53:18.35954 -37:13:53.5977

DSS | PanSTARRS | SDSS | 2MASS | GALEX | Gaia | SImbad | NED

Sun euvi-aiia304:2012

hfc1ar: update

Heavy database

odbcGrammar with **update** : new `<makeQuery>` element (DaCHS 2.5.5)

```
<table id="epn_core" onDisk="True" adql="True" primary="id_ar" dupePolicy="dropOld" >
</table>
<data id="import" updating="True">
  <sources pattern="data/driver.txt"/>
  <odbcGrammar>
    <makeQuery>
      <code>
try:
  with base.getTableConn() as conn:
    print(next(conn.query("SELECT MAX(time_min) FROM \schema.epn_core"))[0])
    localMax = next(conn.query("SELECT MAX(time_min) FROM \schema.epn_core"))[0]
    fragment = " WHERE (jdint + jdfrac) >= {}".format(escapeSQL(localMax))
    fragment=str(fragment)
    print(fragment)
except base.DBError as msg:
  base.ui.notifyWarning(f"{msg} while harvesting: full re-harvest")
  fragment = ""

return f"SELECT * FROM hfc1.view_ar_hqi JOIN hfc1.activeregions USING (id_ar) {fragment}"
      </code>
    </makeQuery>
  </odbcGrammar>
</data>
```

Access the postgres database of the service – previously imported

Build a fragment of the odbc query

Return the final odbc query
Default : query without fragment

Daily cron with service import

12

The active region database is very heavy and updating every day, the added s_regions had made the problem even worse.

Dachs basically re-import the whole service each time we want to make an update, but in our case it takes several days for one import.

So, recently from dachs 2.5.5 (so the beta version) the new makequery element has been made available. So this element allows to adapt the SQL query with python by accessing to the already-imported service.

In the resource descriptor, the data updating must be set to true and the dupe policy in the table must be set as dropold which means that if the primary key is duplicated, it will keep the new row.

So here, we access the local postgres database, and we use a monotonously increasing parameter (here the date, time_min). And the query is made with the fragment here in selecting the part of the



Updated heliophysics services in VESPA: science products, service design and capabilities

support.epntap@obspm.fr

Baptiste CECCONI
Chloé AZRIA



Laboratoire d'Études Spatiales et d'Instrumentation en Astrophysique



Thank you for your attention,

Feel free to ask questions or send e mails to
support epntap