



Fig. 1



Fig. 2



Fig. 3

## 1. Solar System Data in the VO

(cf. Fig. 1)

Markus Demleitner  
*msdemlei@ari.uni-heidelberg.de*

(cf. Fig. 2)

- EPN-TAP: What is it?
- EPN-TAP: Usage scenarios
- Publishing solar system data with DaCHS

(cf. Fig. 3)

## 2. What's EPN-TAP?

Short answer: A scheme for useful publication of solar system data in the VO. Thanks to existing VO tech, all it needs is a "data model".

## 3. The EPN-TAP Model

Essentially, EPN-TAP defines a (large) set of columns of a database table, like:

- target\_class, target\_name
- time\_min, time\_max
- c<sub>n</sub>\_min, c<sub>n</sub>\_max
- spatial\_frame\_type
- instrument\_name, processing\_level
- access\_url, thumbnail\_url
- and so on, about 100 total<sup>1</sup>

## 4. That's Enough.

A table definition is enough because EPN-TAP re-uses VO tech:

- TAP – transport queries to the service and responses back
- ADQL – write queries in a SQL dialect
- Registry – discover services offering EPN-TAP services

(Plus a few more behind the curtain)

## 5. Interface

By assembling these building blocks, the VESPA project offers a UI to all EPN-TAP services<sup>2</sup> in several modes:

(cf. Fig. 4)

## 6. Access Alternatives

But since it's all VO tech, you can just as well use TOPCAT...  
...or any other VO client to access EPN-TAP services. Or...

(cf. Fig. 5)

<sup>1</sup> <https://voparis-confluence.obspm.fr/display/VES/EPN-TAP+V2.0+parameters>  
<sup>2</sup> <http://vespa.obspm.fr/planetary/data/epn/query/all/>

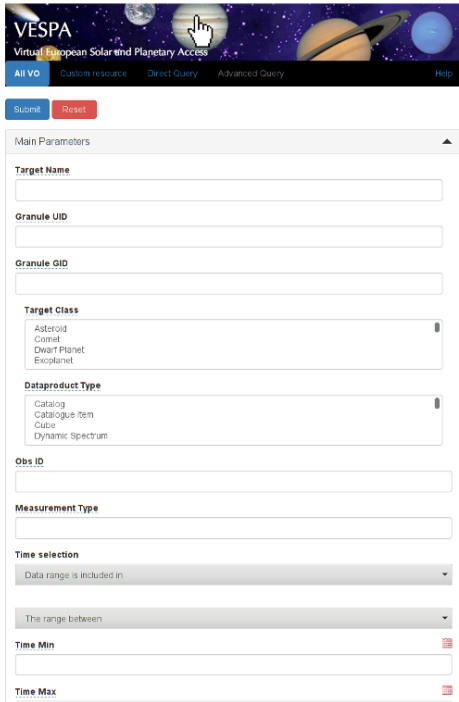


Fig. 4

## 7. Access Alternatives

And then there's easy programmatic access using `astropy` and `pyVO`. All-VO execution of a query as simple as:

```
import pyvo

glots_svc = pyvo.dal.TAPService("http://dc.g-vo.org/tap")
epn_services = glots_svc.run_sync(
    "SELECT accessurl, ivo_string_agg(table_name, '#') as tables"
    " FROM glots.services NATURAL JOIN glots.tables"
    " WHERE table_name LIKE '%epn_core'"
    " GROUP BY accessurl")

for svcrow in epn_services.table:
    service = pyvo.dal.TAPService(svcrow["accessurl"])
    for table_name in svcrow["tables"].split("#"):
        for row in service.run_sync(
            "SELECT * FROM %s WHERE target_name='Mars'%"table_name
            ).table:
            print(row)
```

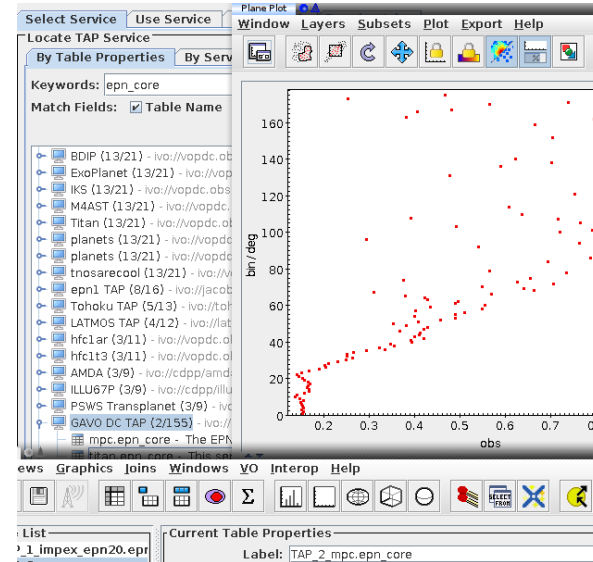


Fig. 5

## 8. Got Data?

Publishing your solar system data through EPN-TAP is easy!

*Essentially*, all you need to do is write mapping rules like

```
<rowmaker idmaps="*">
  <var key="exptime">float(@SR_ACQUIRE_OPTIONS[
    "EXPOSURE_DURATION"].split()[0])</var>

  <apply procDef="//epntap2#populate-2_0" name="fillepn">
    <bind key="access_format">"PDS"</bind>
    <bind key="c1min">@ra</bind> [...]
  ...
```

– these map data from file headers, literals, computed values, whatever, to EPN-TAP columns. More on this<sup>3</sup>.

## 9. Learn More

Go to <http://docs.g-vo.org>,  
click through to the DaCHS tutorial  
and the EPN-TAP chapter there.  
... and follow our blog: <https://blog.g-vo.org>

Thanks!

<sup>3</sup> <http://wiki.ivoa.net/internal/IV0A/InterOpMay2017-SSIG/dachsepn.pdf>