

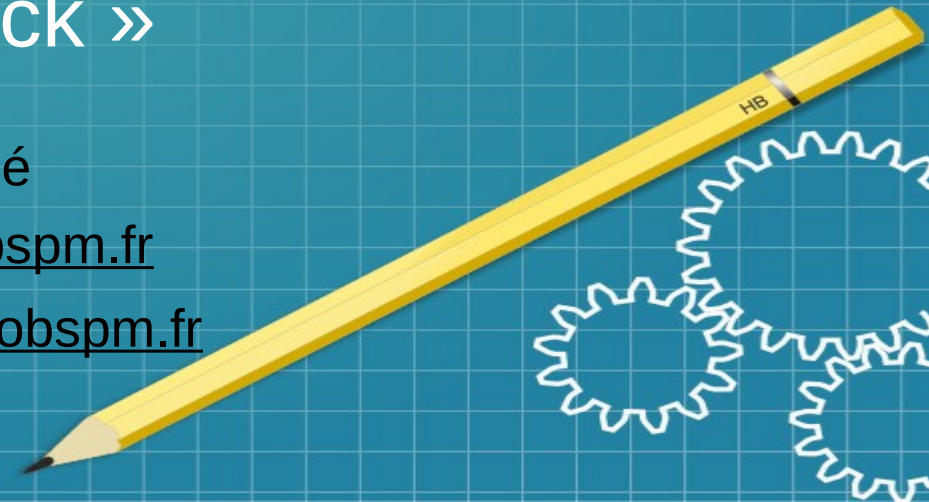
## Tutorial Session :

# Setting-up a VESPA service « Starter pack »

Azria Chloé

[chloe.azria@obspm.fr](mailto:chloe.azria@obspm.fr)

[support.epntap@obspm.fr](mailto:support.epntap@obspm.fr)



# Setting-up an EPN-TAP service

- VO server (mainly DaCHS)
  - Virtual Machine
  - Docker Container
- Build a Resource Descriptor (RD) – xml-like
  - Map the source information to the `e pn_core` schema
    - Mandatory / optionnal columns
    - Columns added manually
    - Post-treatment using python scripts
- Import the service
- Register the service

# VO Server

## Virtual Machine / Docker

- Debian
- Install DaCHS + postgres + Apache 2
- DaCHS Configuration (gavo.rc, userconfig.rd, defaultmeta.txt)

(for production)

<https://voparis-wiki.obspm.fr/display/VES/EPN-TAP+Server+Installation+for+VESPA+Data+Provider+Tutorial>

# Create the service

Where is the information ?

- CSV : map and post-treated in the Resource Descriptor

```
column A, column B, column C
```

```
valueA1, valueB1 , valueB1
```

```
valueA2, valueB2 , valueB2
```

The easiest, create it with your favourite language

- From a SQL database :
  - Create your connection chain to access the database
  - Map and post-treat it in the Resource Descriptor
- Other : Fits, MongoDB ...



# The Resource Descriptor

Generate a template with instructions with the command :

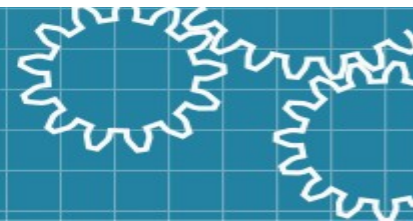
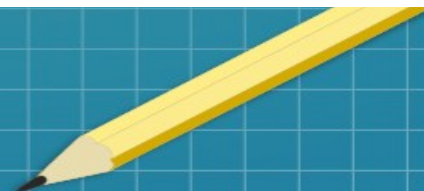
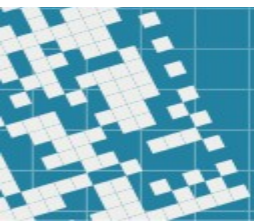
*'dachs start epntap'*

From `/var/gavo/inputs/{Your shema name}`

# Tutorial : CSV example « planets »



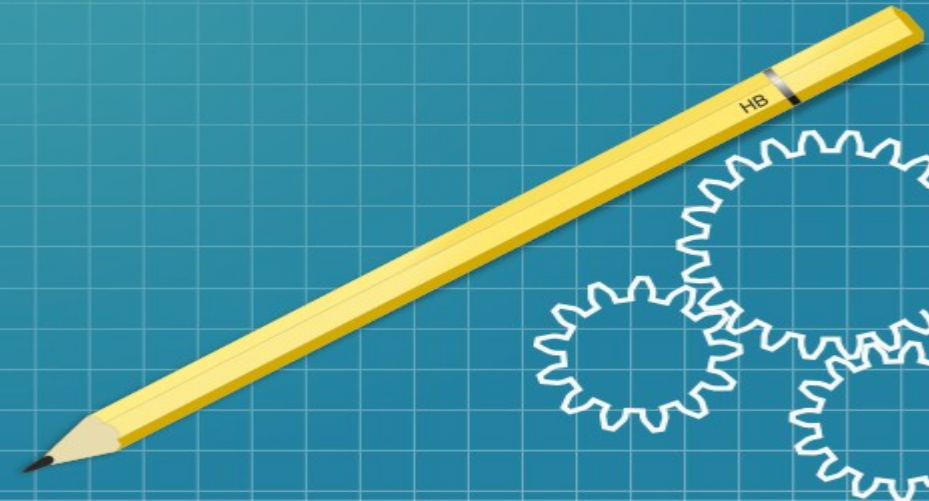
name	mean radius (km)	mean radius uncertainty (km)	equatorial radius (km)	equatorial radius uncertainty (km)	polar radius (km)	polar radius uncertainty (km)	rms deviation (km)	elevation max (km)	elevation min (km)	mass (kg)	distance to primary (km)	sidereal rotation period (h)
Mercury	2439.7	1.0	2439.7	1.0	2439.7	1.0	1	4.6	2.5	3.3014E23	57909227.	1407.504
Venus	6051.8	1.0	6051.8	1.0	6051.8	1.0	1	11	2	4.86732E24	108209475.	-5832.432
Earth	6371.00	0.01	6378.14	0.01	6356.75	0.01	3.57	8.85	11.52	5.97219E24	149598262.	23.93447232
Mars	3389.5	0.2	3396.19	0.1	3376.	0.1	3.0	22.64	7.55	6.41693E23	227943824.	24.624
Jupiter	69911.	6	71492.	4	66854.	10	62.1	31	102	1.89813E27	778340821.	9.92496
Saturn	58232.	6	60268.	4	54364.	10	102.9	8	205	5.68319E26	1426666422.	10.656
Uranus	25362.	7	25559.	4	24973.	20	16.8	28	0	8.68103E25	2870658186.	-17.232
Neptune	24622.	19	24764.	15	24341.	30	8	14	0	1.0241E26	4498396441.	16.104



# Tutorial : CSV example « planets »



- Define Granules – Epn-Core parameters
- CSV file
- Resource Descriptor



# Tutorial : CSV example « planets »

- `docker exec -ti dachs bash`
- `cd /var/gavo/inputs`
- `git clone`  
`https://voparis-gitlab.obspm.fr/workshop-2021-material/planets.git`

Masses2.csv

CSV file

q.rd

Resource Descriptor



# Tutorial : CSV example « planets »

## *Resource Descriptor*

```
<resource schema="...">
  <meta .../>
  ...
  <meta .../>

  <table ...>
    <mixin .../>

    <column .../>
    ...
    <column .../>
  </table>

  <data id="import">
    <sources .../>

    <csvGrammar> <rowfilter procDef="//products#define"> <bind name="table">"\schema.epn_core"</bind> </rowfilter> </csvGrammar>

    <make table="epn_core">
      <rowmaker idmaps="*">
        <var key="...">...</var>
        ...
        <var key="...">...</var>

        <apply procDef="//epntap2#populate-2_0" name="fillepn">
          <bind name="...">@...</bind>
          ...
          <bind name="...">@...</bind>
        </apply>
      </rowmaker>
    </make>
  </data>
</resource>
```

# Tutorial : CSV example « planets »

- Meta tags
  - subject

## Meta tags

The first part is a set of meta tags with different attributes which defines global characteristics of the table. Meta tags aim to describe the service in the registry.

```
<resource schema="planets">
  <meta name="title">Characteristics of Planets (demo)</meta>
  <meta name="description" format="plain">
    Main characteristics of planets. Data are included in the table, therefore most relevant parameters are non-standard in EPN-TAP
  </meta>
  <meta name="creationDate">2015-08-16T09:42:00Z</meta>
  <meta name="subject">solar-system-astronomy</meta>
  <meta name="subject">planetary-science</meta>
  <meta name="subject">solar-system-planets</meta>
  <meta name="subject">periodic-orbit</meta>
  <meta name="copyright">LESIA-Obs Paris</meta>
  <meta name="creator.name">Stephane Erard</meta>
  <meta name="publisher">Paris Astronomical Data Centre - LESIA</meta>
  <meta name="contact.name">Stephane Erard</meta>
  <meta name="contact.email">vo.paris@obspm.fr</meta>
  <meta name="contact.address">Observatoire de Paris VOPDC, bat. Perrault, 77 av. Denfert Rochereau, 75014 Paris, FRANCE</meta>
  <meta name="source">2000asqu.book....C</meta>
  <meta name="contentLevel">General</meta>
  <meta name="contentLevel">University</meta>
  <meta name="contentLevel">Research</meta>
  <meta name="contentLevel">Amateur</meta>
</resource>
```

Most of the attributes are easy to understand, see [this page](#) for detailed explanations and more meta elements.

Meta attribute "subject" is defined several times by different keywords defining data from UAT (Unified Astronomy Thesaurus). At least one of them must refer to a global topic listed in [this page](#). In the context of VESPA, the 3 appropriate global topics are : "Exoplanet astronomy", "Solar physics" and "Solar system astronomy". The attribute "source" refers to the resource-related paper. Here, "contentLevel" takes the four values "General", "University", "Research", "Amateur" but it could take only some of these.

# Tutorial : CSV example « planets »

- Table definition
  - epn\_core : collection of columns
    - Mandatory
    - Optional
  - Defined by the data provider

```
<table id="epn_core" onDisk="true" adql="True">  
  
  <mixin spatial_frame_type="none"  
    optional_columns= "time_scale publisher bib_reference" >//epntap2#table-2_0</mixin>
```

```
<column name="distance_to_primary" type="double precision"  
  tablehead="Distance_to_primary" unit="km"  
  description="Extra: Mean heliocentric distance (semi-major axis)"  
  ucd="pos.distance;stat.min"  
  verbLevel="2"/>  
<column name= ... />  
  ...  
<column name= ... />  
  
</table>
```

# Tutorial : CSV example « planets »

- Data ingestion

- Path to the Sources
- Grammar
- Mapping – associate columns previously defined on the RD to columns from the source (CSV, database, etc)
- Populate

```
<data id="import">
  <sources>Masses2.csv</sources>
  <csvGrammar>
    <rowfilter procDef="//products#define">
      <bind name="table">"\schema.epn_core"</bind>
    </rowfilter>
  </csvGrammar>
```

```
<apply procDef="//epntap2#populate-2_0" name="fillepn">
  <bind name="granule_gid">@granule_gid</bind>
  ...
  <bind name="instrument_name">" "</bind>
</apply>
```

```
<make table="epn_core">
  <rowmaker idmaps="*">
    <var key="obs_id" source="obs_id" />
    ...
    <map key="measurement_type">"phys.mass#phys.size.radius"</map>
```

# Tutorial : CSV example « planets »

Install and check tables

- *dachs val q.rd*
- *dachs imp q.rd*
- *dachs serve reload*

<http://localhost:8080>

[http://localhost:8080/\\_\\_system\\_\\_/dc\\_tables/list](http://localhost:8080/__system__/dc_tables/list)

```
SELECT * FROM planets.epn_core
```

# Tutorial : CSV example « planets »

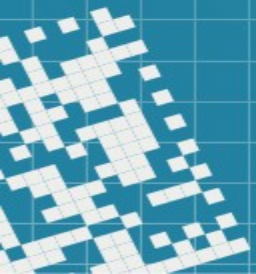
<https://voparis-wiki.obspm.fr/display/VES/Setting-up+an+EPN-TAP+service%3A+Tutorial+for+Beginners>

# From a SQL database :



## odbcGrammar

- python3-pyodbc must be installed on the server
  - *cd*
  - *git clone*  
*[https://voparis-gitlab.obspm.fr/workshop-2021-material/pyodbc\\_installation.git](https://voparis-gitlab.obspm.fr/workshop-2021-material/pyodbc_installation.git)*
  - *cd pyodbc\_installation*
  - *./install\_pyodbc.sh*



# Resource descriptor from SQL

*Clone and import a service from an SQL  
database*

→ `cd /var/gavo/inputs`

→ `git clone`

`https://voparis-gitlab.obspm.fr/vespa/dachs/services/padc/voparis-tap-helio/hfc1ar.git`

→ `dachs imp hfc1ar/q`



# Resource descriptor from SQL



Contains the 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">
```



Your SQL Query

# From a SQL database

- Using a Chaincode to access the database (instead of the CSV)

ex : Helio services

<https://voparis-gitlab.obspm.fr/vespa/dachs/services/padc/voparis-tap-helio>

- MySQL (BASS2000): Driver={MySQL ODBC 8.0 Unicode Driver};Server=145.238.155.92;Database=bass2000prod;User=guest;Password=guest;Option=3;
- PostgreSQL (HFC1AR): Driver={PostgreSQL Unicode};Server=bdd-lesia.obspm.fr;Port=5432;Database=hfcdB;Uid=guest;Pwd=guest;

# Resource Descriptors

- Other examples of features on RD (no matters the grammar) :

→ `<data id="import" updating="True">`

→ `<rowfilter procDef="//products#define">`

`<bind key="table">"\schema.epn_core"</bind>`

`<code> ... post-treatment with python code ... </code>`

`</rowfilter>`

→ Make functions, import modules, etc

• See hfc1ar service

[support.epntap@obspm.fr](mailto:support.epntap@obspm.fr)