# A Guide to the Herschel Science Archive User Interface

## herschel science archive

HOME | SEARCH | RESULTS | CATALOGUES | HSA USERS GUIDE | HERSCHEL DOCUMENTATION | PUBLICATIONS

### Welcome to the Herschel Science Archive

Herschel was the fourth cornerstone in ESA's Horizon 2000 science programme, designed to observe the 'cool' universe. It performed photometry and spectroscopy in the poorly explored 55-670 µm spectral range with a 3.5 m diameter Cassegrain telescope, providing unique observing capabilities and bridging the gap between earlier infrared space missions and groundbased facilities. Herschel successfully performed ~37000 science observations and ~6600 science calibration observations which are publicly available to the worldwide astronomical community through the Herschel Science Archive.

The Herschel Science Archive offers access to:

- science data products automatically generated by the data processing pipelines (at various - user selected - levels)
- interactively reduced data provided by the community (User Provided Data Products; UPDP) and by the mission experts in the Herschel ground segment (Highly Processed Data Products; HPDP)
- publications linked to the data
- preview images and connectivity to common astromonical tools over Virtual Observatory (VO) protocols

*Herschel's swan*

**SEARCH**
Query the Herschel Science Archive (HSA).

**HSA USERS GUIDE**
A comprehensive users guide to the HSA.

**CONTACT**
For questions, suggestions or problem reports, please contact the HSC Helpdesk at:
https://support.cosmos.esa.int/herschel/

**CONTENTS**
An overview description of the different type of data products contained in the HSA.

**NEWS**
What's new for the different HSA versions.

**FAQS**
Frequently Asked Questions about the HSA.

**HSC WEB SITE**
Visit the Herschel Science Centre web site for more information.

**HERSCHEL DOCUMENTATION**
Portal to the Herschel Explanatory Legacy Library.

**CATALOGUES**
Herschel Catalogues.

**Eva Verdugo, Herschel Science Centre, ESAC, ESA**

# Table of content

**Overview**

The Herschel Science Archive (HSA) offers access to all public Herschel data and offers a host of features:

- Access to all Herschel science data products at various (user selected) levels of processing.
- Access to interactively reduced data provided by the observers (User Provided Data Products; UPDPs).
- Access to interactively reduced data produced by ground segment experts (Highly Processed Data Products; HPDPs)
- Visualization of accurate footprints projected onto the Digitized Sky Survey (DSS) image of the field.
- Search on publications, providing links to the Herschel data used for these publications and to the corresponding publication registry in ADS, including the possibility to search for observations in the archive without known associated publications.
- Preview images and connectivity to common astronomical tools over VO protocols.
- Search on Herschel Catalogues through the VO Table Access Protocol (TAP)

Data retrieval and other features require that users have registered already with the Herschel system: http://www.cosmos.esa.int/web/herschel/registration.

HSA users are reminded that the typical size of Herschel data products corresponding to an average observation can be as large as several Gbytes (the interface does not report the total size of the download since the files are dynamically combined in the tar file to be retrieved). They are therefore recommended to optimise the datasets to retrieve. The HSA user interface offers several different methods to customise datasets to retrieve and download data. Please, use the most suitable for your needs.

# HSA Contents

The Herschel Science Archive contains products obtained by processing the observations data through an **automatic processing pipeline**. This corrects well a number of instrumental artifacts in an automatic fashion. The final products can, however, be significantly improved by processing them further, e.g. by means of the interactive analysis software tools available within HIPE (The Herschel Interactive Analysis Environment) or with the help of other external tools and/or user contributed software. The resulting products are called User Provided Data Products ( **UPDPs**) and Highly Processed Data Products ( **HPDPs**).

The HSA user Interface can be tuned to return all kinds of products per observation, or just the desired dataset from the Herschel Data Collection selection in the Search page.
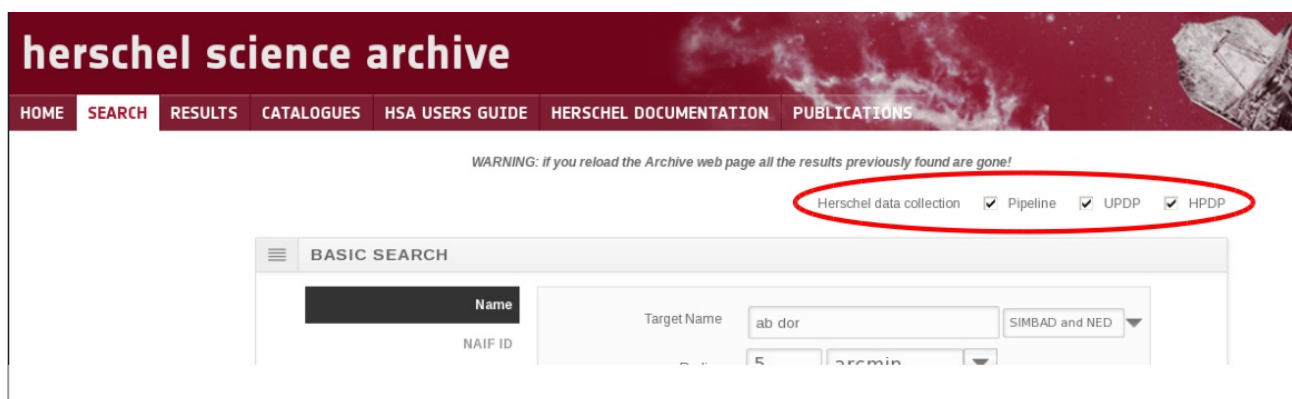


## Pipeline Products

All data obtained with Herschel were systematically pipeline-processed with the Herschel Data Processing system (Standard Processing Generation; SPG), at the Herschel Science Centre (HSC), to generate a range of scientific and auxiliary data products.

All raw and processed data were ingested in the Herschel Science Archive immediately after the pipeline processing was completed, typically 1-2 days after execution.

In addition all data were systematically re-processed with every improved version of the Pipeline up to the current and final SPG version 14.1 for HIFI and SPIRE and 14.2 for PACS.

**Observational products** are classified depending on the level of the processing from Level-0 (Raw telemetry data minimally formatted) to Level3 (super-combinations of observations like mosaics or stitched spectra) and are stored in the HSA together with the auxiliary products required directly or indirectly for the processing and analysis of the scientific data and the calibration products used in the processing of the raw data to produce astronomically calibrated products (see the Data Product Overview web page for more details).

**Postcards** (static representations in JPG-format of the final products) are also created by the pipeline to facilitate a quick look at the data and to identify which data need to be retrieved.

A **Quality Control Report** associated to every science observation is also provided. It gathers a summary of the information required to evaluate the technical quality of the executed observation and the products generated, and provide a global quality assessment: **PASSED** if the observation does not show any problem that makes it useless for science or **FAILED** when the observation is completely useless, mainly due to problems during its execution.
By default, **Failed** observations are discarded when making queries.
Note that, the Data Quality assessment was not repeated for every version of the pipeline. Hence, some reports refer to old versions (SPG version in the report) but the information given is still valid.

Calibration observations were not systematically inspected. Therefore a Quality Control Report is not provided for all of them.

A particular subset of the final science products generated by the pipeline can also be retrieved directly from the HSA User Interface as ***Standalone Browse Products***.

Further details of products and of the data processing can be found in:

- Product Definition Document ([PDF](#) and [HTML](#))
- Data Analysis Guide ([PDF](#) and [HTML](#))
- PACS Data Reduction Guide: Spectroscopy ([PDF](#) and [HTML](#))
- PACS Data Reduction Guide: Photometry ([PDF](#) and [HTML](#))
- PACS Products Explained ([PDF](#) and [HTML](#))
- SPIRE Data Reduction Guide ([PDF](#) and [HTML](#))
- SPIRE Products Explained ([PDF](#))
- HIFI Data Reduction Guide ([PDF](#) and [HTML](#))
- HIFI Products Explained ([PDF](#))

**User Provided Data Products (UPDP)**

The automatic processing pipeline corrects automatically most of the instrumental artifacts. The final products can, however, be significantly improved by processing them further, e.g. by means of the interactive analysis software tools available within [HIPE](#) or with the help of other external tools and/or user contributed software.

In particular, and as agreed at the time of submission of the Key Programme observing proposals, KP consortia were committed to deliver to the HSC the User Provided Data Products corresponding to the data obtained as part of the Science Demonstration Phase. In addition, on a best effort basis, they were also expected to continue contributing with additional User Provided Data Products corresponding to the results obtained as part of their core programmes as they become published in refereed literature.

A compilation of the currently available User Provided Data Products is given in [https://www.cosmos.esa.int/web/herschel/user-provided-data-products](https://www.cosmos.esa.int/web/herschel/user-provided-data-products).

**Highly Processed Data Products (HPDP)**

In addition to the User provided Data Products, the HSA also contains Highly Processed Data Products (HPDPs) generated by different experts in the Herschel Science Ground Segment. In a broader sense this definition includes catalogues and atlases.

A compilation of the currently available HPDPs is given in [https://www.cosmos.esa.int/web/herschel/highly-processed-data-products](https://www.cosmos.esa.int/web/herschel/highly-processed-data-products).

**Ancillary Data Products (ADPs)**

Ancillary Products are data (products, tables, plots, etc..) generated in the course of the different phases of the Herschel mission which are not necessarily linked to a particular observation in the Herschel Science Archive, but which contain valuable additional information like e.g. the planetary, asteroid and stellar models used by the various instruments for their calibration, PSFs, trend analysis plots, etc.

# How to search the HSA
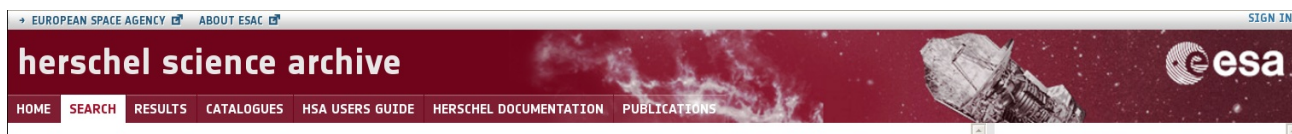


On the top of the interface there are three different tabs for searching:

- SEARCH which allows to search the HSA content using different filters
- CATALOGUES which allows to search the main Herschel Catalogues
- PUBLICATIONS which allows to search the publications based on Herschel observations
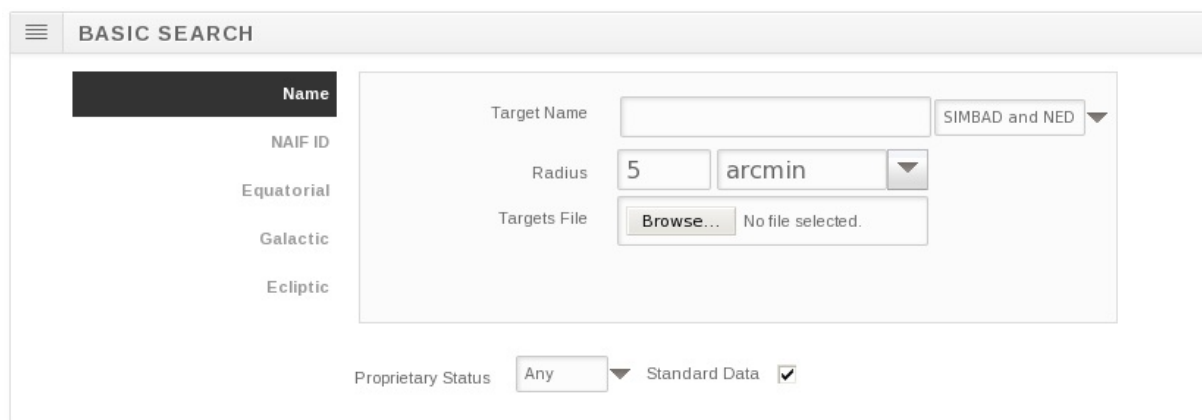
## The SEARCH tab

There is dedicated help on usage and syntax provided for every one of these filters just by positioning the cursor on them and clicking.
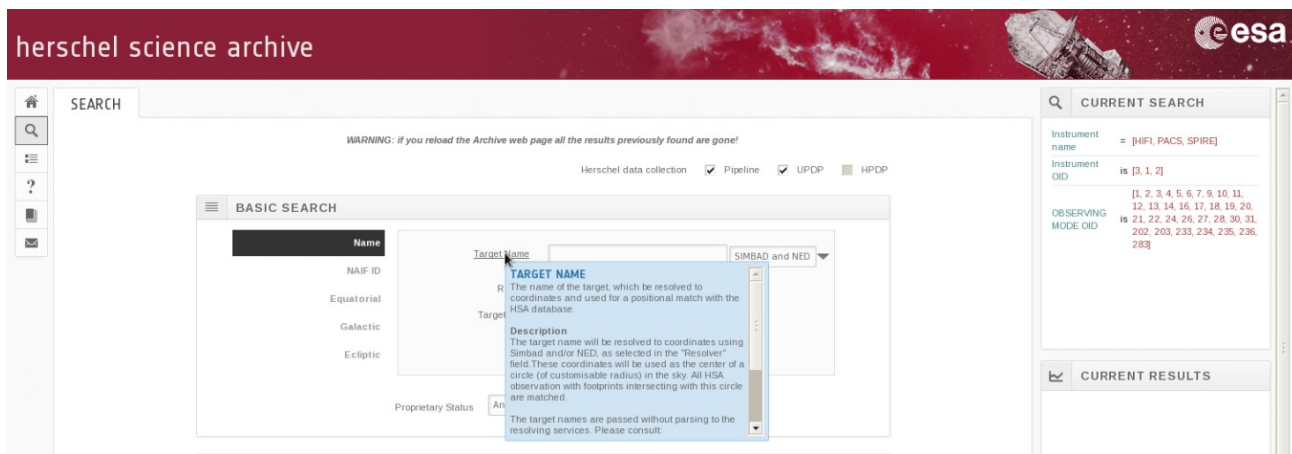


There are three main panels:

- **A Basic Search** by Target Name, NAIF ID or Coordinates (Equatorial, Galactic and Ecliptic). It is also possible to submit a list of targets, NAIF IDs or coordinates.



The coordinates entered (or the resolved target name) are used as the center of a circle (of 5 arcmin radius by default) in the sky. All HSA observations with footprints intersecting with this

circle are matches. For instance a search around 3h 45m, +68d 38m with a search radius of 5 arcmin will result in three observations which are big SPIRE/PACS Parallel maps that intersect with the coordinates given although the center of the maps are far away from the search point.
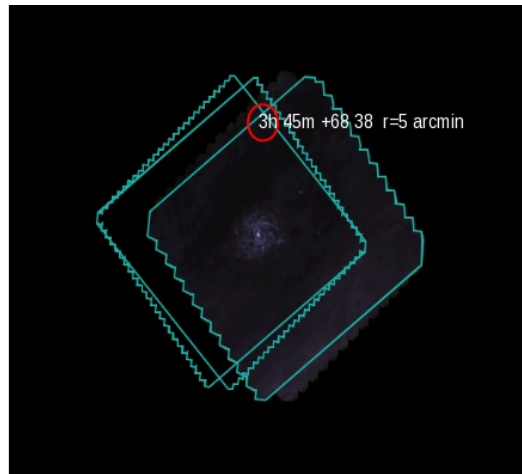


- **Observation Constraints** which are contained in different tabs filters per Observation ID, Instrument and refined queries by instrument settings, proposal information, object type and publications.



- **Product Selection**. The query can be restricted to those observations which are contained in a UPDP or an HPDP. Also, a set of Ancillary products can be retrieved directly from this panel.

All the filters are combined to perform a single query. By default, only observations taken in standard observing modes and that have not been declared *FAILED* by the Data Quality Control assessment, are returned by a query.



The list of filters entered to perform a query is displayed at the top right-side of the Search page, in the window called **CURRENT SEARCH**. Below this window, the **CURRENT RESULTS** gives a pre-query result with the number of observations found for which products generated by the pipeline can be retrieved, the number of observations which are included in UPDPs and HPDPs and the number of publications in which any of the observations resulting from the query are included.

For instance a query of target DZ Cha for all instruments and modes and without known publications, will look like:
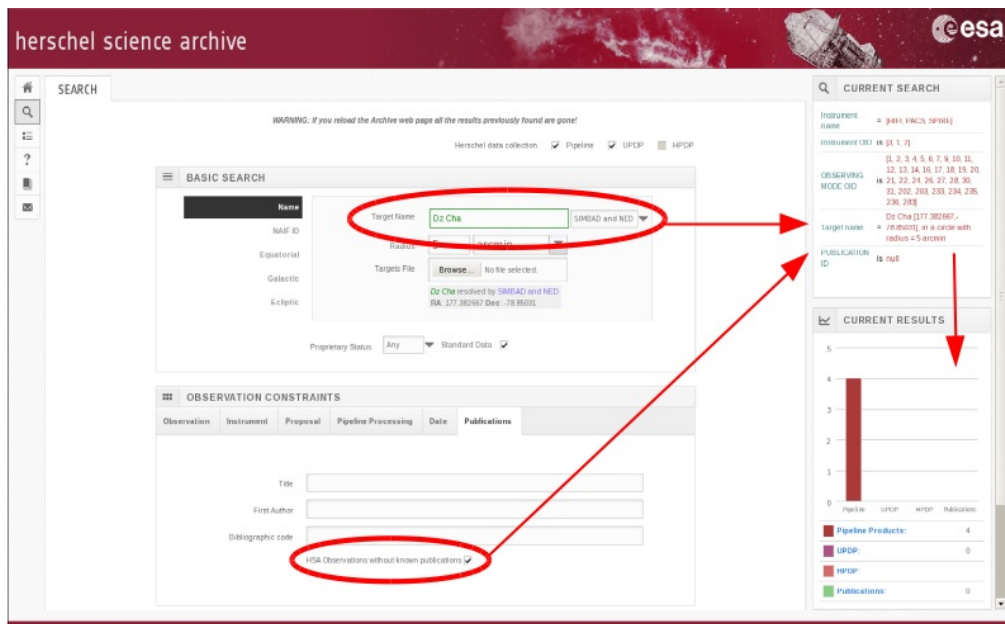
Which gives a pre-query result of 4 observations taken in standard observing modes, executed successfully and processed by the Pipeline for which there are no known publications and which are not included yet in any delivered/ingested UPDP or HPDP.

At the bottom of the interface, there are the two buttons to manage the query:

- **Search** will execute the query using the values entered in all the fields of the Search page.
- **Clear** will reset all values in the Search page to their default values.

We encourage you to *Clear* the Interface for every new query since hidden tabs could contain selections from previous queries that can contaminate the expected result.

**The CATALOGUES tab**

The User Interface also allows to query the contents of the main Herschel catalogues: The Photometric Catalogues and the Spectral Line Catalogues.
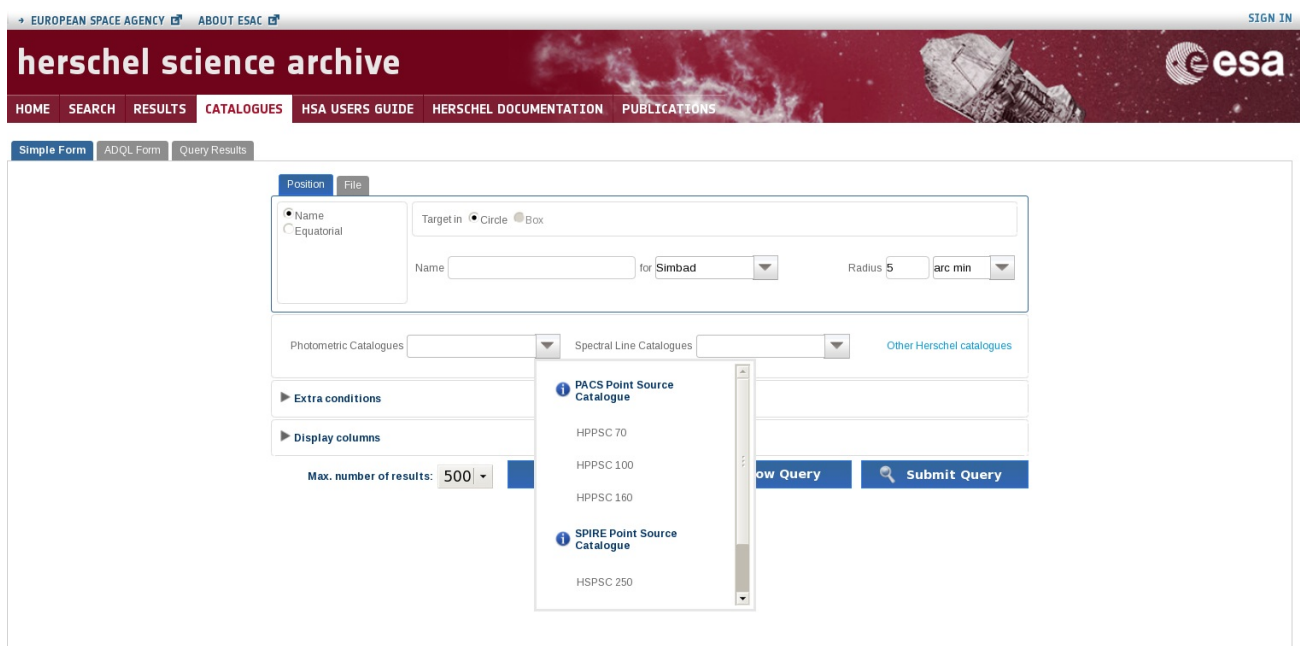
The selection of the catalogue can be combined with geometrical searches by target name or coordinates. Also after selection of one catalogue, extra conditions for the query can be added:



The columns displayed as output can be selected.



And the result can be saved as VOTable, CVS, FITS...

Users can also perform more complex queries through an ADQL Form:

For this, a number of examples are given under "Query examples" at the top right. More information on the syntax and usage of ADQL is given in the [Command Line Access](#) section of this Guide.

On the left, the HSA tables available for queries are displayed. Click on the name of the table and its parameters to see a description.

The result of an ADQL Form query can be saved, sent to any VO application through SAMP or displayed in the "QueryResults" tab.

**The PUBLICATIONS tab**

This functionality allows to make all kind of queries on the database of refereed publications linked to Herschel observations.

By default the full list of papers is shown:



Different filters (and combinations of them) for making queries are possible. For instance, the

column "Status" refers to the continuous activity carried out by the HSC which tries to link papers with Herschel observations: **Done** means that all the observations linked to that particular publication are already identified, **In progress** is the status assigned to those cases in which the list of observations to be linked to that paper is still being updated (not final) and **Pending** when the paper still needs to be scrutinized in order to identify which observations were actually used in that publication.

HOME  SEARCH  RESULTS  CATALOGUES  HSA USERS GUIDE  HERSCHEL DOCUMENTATION  **PUBLICATIONS**

Search

Filters

BIB code [          ]     Authors [          ]     Search

Title [          ]     Abstract [          ]     Clear

Publication date between (>=) [          ] and (<=) [          ]

Citations [>=] [     ]    Journal [     ]    Status [     ]
                                            ✓
                                            Pending
                                            In progress
                                            Done

Science Area [Any ▼]    Science Catego [     ▼]

| ▲ BIB Code | Title | Authors | Journal | Date | Citations | #OBS | Status |
|---|---|---|---|---|---|---|---|
| 2018MNRAS.473.2493M | A census of radio-selected AGNs on the COSMOS field and of their FIR properties | Magliocchetti, M., Popesso, P., Brusa, M., Salvato, M. | Monthly Notices of the Royal Astronomical Society, Volume 473, Issue 2, p.2493-2505 | 2018-01-01 | 1 | 49 | DONE |
| 2017A&A...608A.144Y | Molecular gas in the Herschel-selected strongly lensed submillimeter galaxies at z 2-4 as probed by multi-J CO lines | Yang, C., Omont, A., Beelen, A., Gao, Y., van der Werf, P., Gavazzi, R., Zhang, Z.-Y., Ivison, R., Lehnert, M., Liu, D., Oteo, I., González-Alfonso, E., Dannerbauer, H., Cox, P., Krips, M., Neri, R., Riechers, D., Baker, A. J., Michałowski, M. J., Cooray, A., Smail, I. | Astronomy & Astrophysics, Volume 608, id.A144, 41 pp. | 2017-12-01 | 3 | 27 | DONE |
| 2017A&A...608A..21S | Correlation of gas dynamics and dust in the evolved filament G82.65-02.00 | Saajasto, M., Juvela, M., Dobashi, K., Shimoikura, T., Ristorcelli, I., Montillaud, J., Marshall, D. J., Malinen, J., Pelkonen, V.-M., Fehér, O., Rivera-Ingraham, A., Toth, L. V., Montier, L., Bernard, J.-Ph., Onishi, T. | Astronomy & Astrophysics, Volume 608, id.A21, 22 pp. | 2017-12-01 | 1 | 3 | DONE |
| | | Favre, C., López-Sepulcre, A., Ceccarelli, | | | | | |

|◄ ◄ 1-15 of 2,099 ► ►|

In the results table, column "BIB Code" is a link to ADS and "#OBS" gives the number of observations linked to this paper. Clicking on this number, the list of observations is given like in a "Search" query of the archive.

## The Result of a Query

The result of a query is always a list of observations which matches the conditions given by the parameters used for searching the Herschel Archive. These parameters are also visible by placing the cursor on the *RESULT* tab to allow users to distinguish which Result page belongs to a given query.
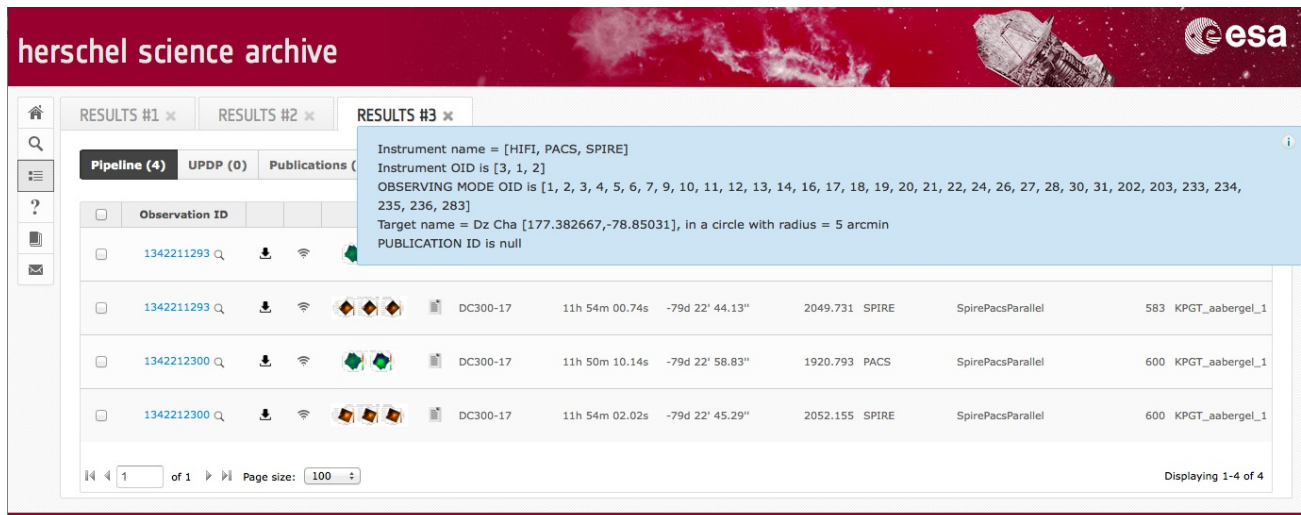


The Interface provides 4 different result tabs per query:

- **Pipeline** gives the list of observations matching the query and information related to every observation and the associated Pipeline products:

  - Observation's details 🔍: Pops up the 'Result Details' panel, which provides details of the observation, instrument settings and publications in which the observation was used. It also displays a postcard illustrating the final pipeline products and the footprint of the observation projected onto a Digital Sky Survey colour image (DSS2).
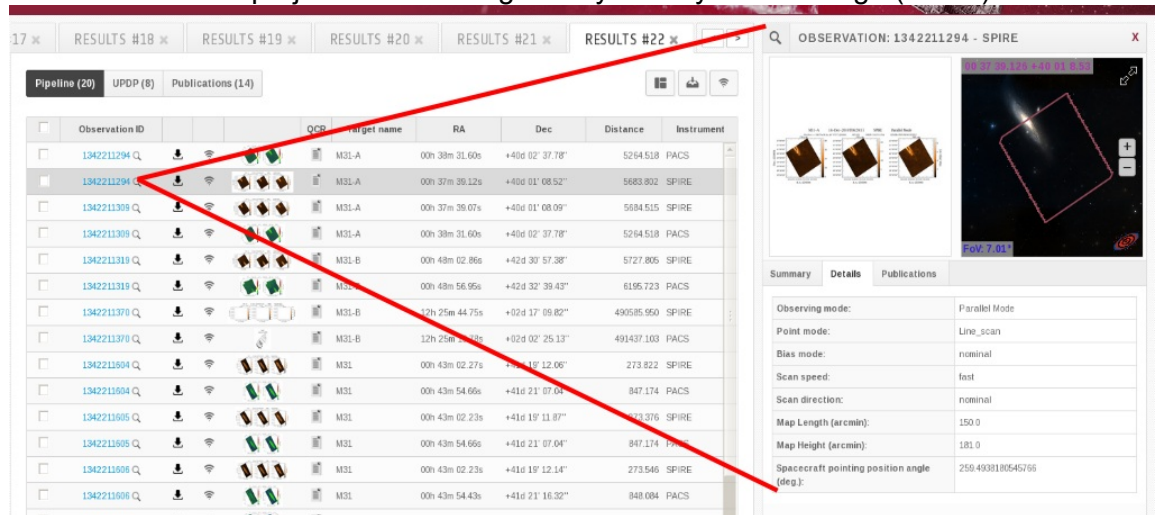


  - An icon to pop-up the Postcard (jpeg file) which illustrates the final pipeline products. Retrieval and visualization of several pipeline products is possible from the postcard.
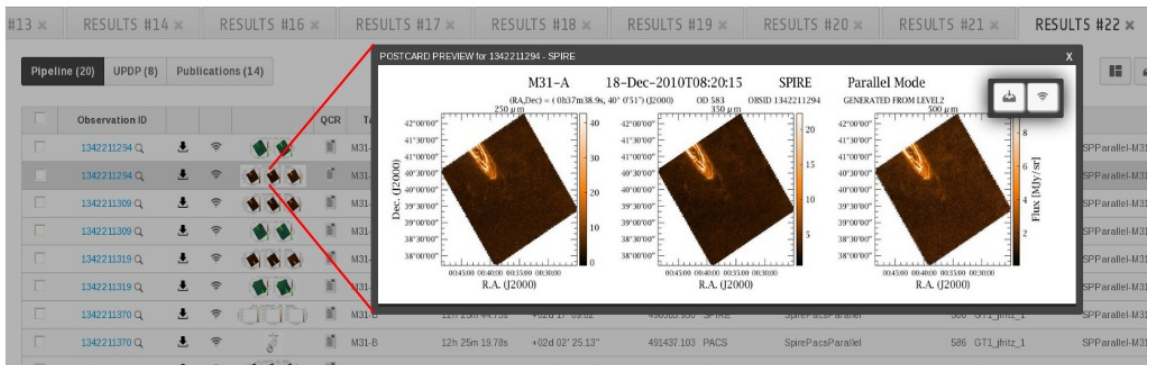
- The Quality Control Summary Report  gathers a summary of the information required to evaluate the technical quality of the executed observation and the products generated. More details on these reports are provided in this guide in the section HSA Contents under "Pipeline Products"
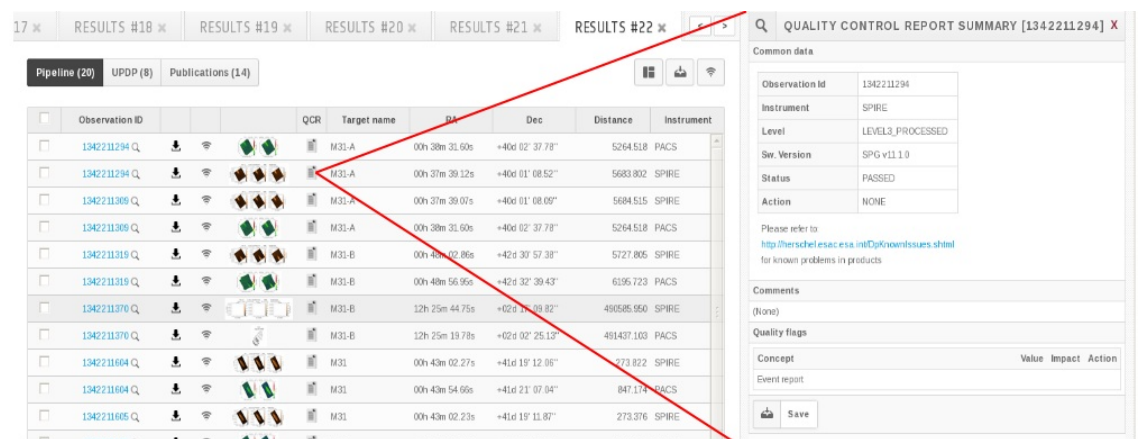


- Number of Publications which gives the total number of different publications in the database which made use of the data contained in that observation. Clicking on the number displayed under this column, the list of associated publications is displayed using the "Publications tab" functionality.



- Different columns with information on the target, the observation, the instrument and the pipeline

- **UPDP** gives the list of observations matching the query which are contained in one or more User Provided datasets. It provides information on the UPDP (through the UPDP Keyword

column) which is also illustrated when available through a Postcard per observation.



- Like the UPDP tab, the **HPDP** tab gives the list of observations matching the query which are contained in one or more expert reduced dataset and gives similar information about it.



- **Publications**. This tab gives the total list of papers in which any of the observations resulting from the query is included and a link to the corresponding publication registry in the ADS. Also, the number of observations included in every paper is given.



The top icon (  ) can be used to Show/Hide columns in all the Result tabs.

The size of the Results page can be selected from 50 to 200 results per page (default is 50). By clicking on the left/right arrows available at the bottom of the page the previous/next page of results will be displayed. Alternatively a page number can be entered manually.

# How to retrieve/visualise Herschel Data

## Visualisation of Herschel Data

The Herschel Science Archive makes use of the SAMP Web Profile (Javascript library) to interoperate with other astronomy tools via SAMP (Simple Application Messaging Protocol). The icon 🛜 is used to send the observation(s) references to another application through SAMP:

- The pipeline products can be sent (without retrieval) to the Herschel Interactive Processing Environment (HIPE) for their visualization and analysis. Details are given in Section 1.4 of the Data Analysis Guide

- The *Standalone Browse Products* can be sent directly to VO (Virtual Observatory) tools like Aladin, DS9, VOSpec, CASSIS...(see the SAMP connectivity section) provided they are already opened in the user machine.
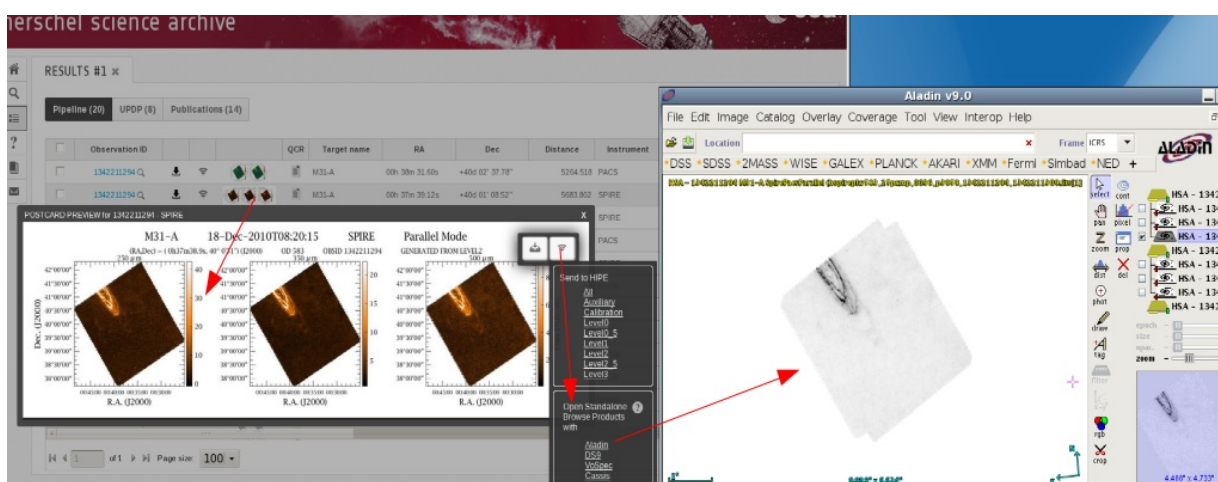


This allows visualization and inspection of the Herschel data before the actual download.

## Download Herschel Data

Herschel Data can be retrieved observation by observation through the download button ( ⬇ ) close to the Observation ID. Alternatively several/all observations can be marked and the data retrieved in one go using the top icon 📥 :

- **Direct Product Download**(⬇) allows retrieval:

  - From the Pipeline result tab: <u>All</u> products generated by the pipeline, or one/several (multiple) levels of processing. Photometric Level-3 products are generally large mosaics that may occupy several GBs and refer to a group of observations. Hence, the possibility of retrieving <u>All products except Level-3</u> is given. *Standalone Browse Products* can be retrieved from here too. In all cases, if one selects a product that does not exist a pop-up window will appear with the message: "No products found".
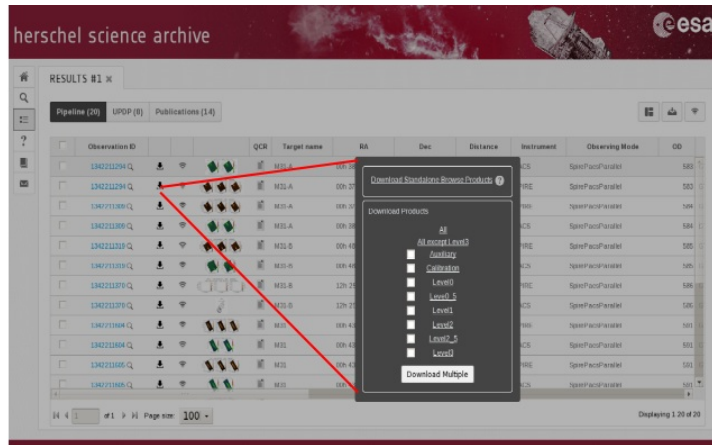
- From the UPDP/HPDP result tabs: Only the data files and the postcard included in the UPDP/HPDP dataset, which are associated to the chosen observation(s).
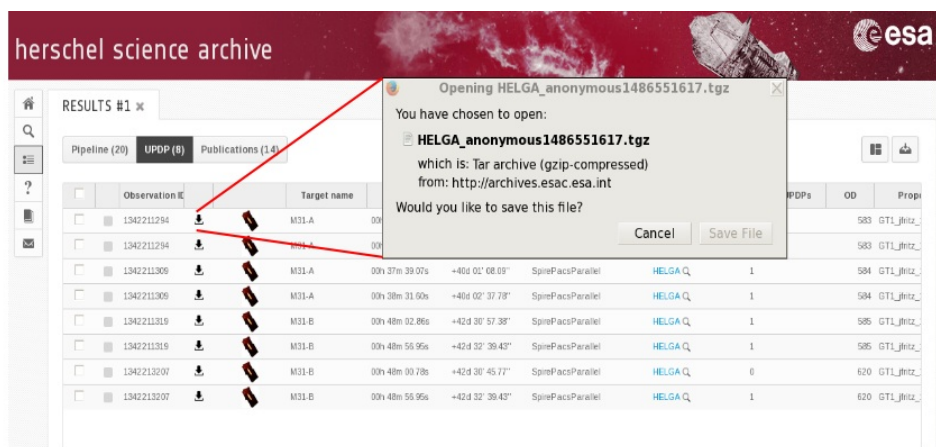


- **Save Results/Download Products from selected observations** (⬆) allows you to save the result table (in CSV or VOTable format), or the products for the selected observations, both for pipeline products and for UPDP/HPDPs.

It is also possible to download the *Standalone Browse Products*, or send them to VO tools through SAMP, as explained before from the **Postcard** generated by the pipeline.



The full content of an UPDP/HPDP can be retrieved from:

- **The Search page**: The tabs for UPDP and HPDP display the list of datasets available and for every dataset the ⓘ icon gives an overview (title, authors and abstract) and the possibility to download the following items:
    - Documentation to download the release note
    - Catalogue to download the catalogues
    - Data to download cubes, images and spectra
    - Postcard to download the postcards
    - FULL UPDP/HPDP to download the full content of the dataset

- **The Results page**: The UPDP/HPDP Keyword column pops-up a window with the same details as the ⓘ icon in the Search page, which gives the overview of the dataset and allows the different retrieval options explained above.



**Ancillary Data Products (ADP)** can only be retrieved from the dedicated tab in the Search page as a complete set, or a group of tar files. Since Ancillary Data Products are not associated to any particular observation no links to these products exist from the Results page.

# Retrieving a tar file containing Pipeline data

## Description of the directory structure

The data retrieved from the Herschel Science Archive (HSA) as a tar file will unpack into a directory with the request identifier (an arbitrary number) as the name, inside which the data are organised in a tree-like structure of directories:

- The **auxiliary** directory contains all Herschel non-science spacecraft data required directly or indirectly for the processing and analysis of the scientific data.
- The **calibration** directory contains the uplink and downlink calibration products.
- A directory with the observation identifier (**obsid**) as the name, which contains the **science data** distributed in sub-directories called level0/0.5/1/2/2.5/3. These levels hold data at different stages of processing: Level 0 are raw, Level 0.5 and 1 are partially processed, Level 2 are of science quality, Level 2.5 are combinations of Level 2 products, and Level 3 are super-combinations of observations (like mosaics or stiched spectra; see the Data Product Overview web page for more details).

A generic unpacked HSA *tar* file will look like:

```
>request_Id/

    auxiliary/

            h<obsid>auxcontext_<timestamp>.fits.gz

            AcmsTelemetryProduct/        OrbitEphemeris/

            EventsLogProduct/     OrbitEventsProduct/     Pointing/

            Siam/              HorizonsProduct/         SremCalProduct/

            Housekeeping/       SremRawProduct/        MissingTm/

            TeleCommandHistory/              MissionTimeLine/

            TimeCorr/          OOL/              UplinkProduct/

    calibration/

            h<instrument>calibrationcontext_<timestamp>.fits.gz

            Downlink (for HIFI)/           Uplink (for HIFI)/

            photometer (for PACS)/         spectrometer (for PACS)/

            Phot (for SPIRE)/           Spec (for SPIRE)/

    <obs_id>/

            h<instrument><obsid>obs_<timestamp>.fits.gz

            browseImageProduct/     browseProduct/      logObsContext/

            quality/              qualitySummary/

            level0/     level0_5/     level1/     level2/     level2_5/     level3/

    <obs_id>-herschel.ia.obs.ObservationContext-XXXXXX.xml
```

This latter XML file (that you can ignore in most cases) is included for administrative purposes. It is used only when loading the observation into the Herschel Interactive Processing Environment ( HIPE).

Many of the FITS files provided do not contain science data, but rather references to other products. These are the so-called context files.

The top context file (h<instrument><obsid>obs_<timestamp>.fits.gz) inmediately under the <obs_id> directory contains the main keywords related to the observation and the instrument setup in the first extension header.

In most cases, you will be interested in the FITS files that contain the science data, which are located in the deepest levels of the <obs_id> directory. Hence, you will find the final science products resulting from the pipeline processing in the sub-directories under the level2 and level2_5 directories.

The general structure can be described as:

```
> directory/

        <context_file>.fits.gz

     sub_directory/

             <context_file>.fits.gz

        sub_subdirectory1/

                    <science_data>.fits.gz

        sub_subdirectory2/

                    <science_data>.fits.gz
```

A particular subset of the final science products generated by the pipeline can also be retrieved directly from the HSA User Interface as *Standalone Browse Products* , either from the retrieval options or by clicking on the observation postcard.


**File Naming Convention**

The Herschel product naming convention for exported FITS files takes a format which depends on the type of product. The generic format is as follows, although not all of the items are present in all filenames.

h<product/instrument><subinst><obsid/od>_<bbid>_<level><type>_<slice>_<timestamp>

where

- h stands for Herschel
- <product/instrument>: is the product type such as aux for auxiliary products or the instrument name either hifi, pacs or spire (note that all letters in the filename are lowercase).
- <subinst>: This is only relevant for instrument data. Depending on the instrument and on the type of product, it stands for the subinstrument used, the detector, polarisation mode etc...
- <obsid/od>: The observation identifier given in decimal format, or the Herschel operational day
- <bbid>: Some observational products are split into logical units such as building blocks. When this occurs the bbid is given in hexadecimal format.
- <level>: Two digits corresponding to the level of the product. Level 0 products are represented by 00, Level 0.5 by 05, Level 1 by 10, Level 2 by 20, Level 2.5 by 25 and Level 3 by 30.
- <type>: This indicates the type of product as given in the meta keyword TYPE.
- <slice>: When data from an observation needs to be split further than by building block, or in a way unrelated to building block, the number of the "slice" is given here. If 100 or fewer products result from the split, then two digits (nn) represent the slices (in time order). For more than 100 slices, three digits (nnn) are used.

These items are organised in a way intended to produce a logical ordering of the filenames when listed in a directory.

All product names also contain at the end a 13 digit number which is a <timestamp> that the system generates when the FITS product is created.

The specific format per product is given in Section 2.10 of the Product Definition Document.


**Structure of the FITS files**

All Herschel FITS files are compatible with the majority of standard FITS readers. They are composed of several extensions:

- The first extension (0) contains always only a header with general information about the observation and the specific product.
- The following extensions (one or more) contain the data (images, spectra, coverage, uncertainties...). For instance, a FITS file from a SimpleImage has at least three image extensions called image, error and coverage. A SpectralSimpleCube has two three-dimensional datasets, image and coverage, and one table dataset ImageIndex, with two columns relating each cube layer to its wavelength. SpectralSimpleCube objects in Level 2.5 HIFI products are made of three extensions, called image, weight and flag.
- The last four (History extensions) are common to all Herschel products and contain the history of the data processing (scripts, tasks and parameters used).

A typical Herschel SpectralSimpleCube FITS file header looks like:

| XTENSION | EXTNAME | EXTVER | EXTLEVEL | BITPIX | GCOUNT | PCOUNT | NAXIS | NAXIS |
|---|---|---|---|---|---|---|---|---|
| 0 | | | | 32 | 0 | 0 | 0 | |
| 1 | IMAGE | image | | -64 | 1 | 0 | 3 | 39x39x29 |
| 2 | IMAGE | coverage | | -64 | 1 | 0 | 3 | 39x39x29 |
| 3 | BINTABLE | ImageIndex | | 8 | 1 | 0 | 2 | 12x29 |
| 4 | IMAGE | History | | 32 | 1 | 0 | 0 | |
| 5 | BINTABLE | HistoryScript | | 8 | 1 | 0 | 2 | 80x7 |
| 6 | BINTABLE | HistoryTasks | | 8 | 1 | 0 | 2 | 35x1 |
| 7 | BINTABLE | HistoryParameters | | 8 | 1 | 0 | 2 | 103x12 |

More detailed information on the structure of Herschel FITS files is provided in Section 1.15.4 of the Data Analysis Guide.

## SAMP connectivity

The Herschel Science Archive makes use of the SAMP Web Profile (Javascript library) to interoperate with other astronomy tools, via the SAMP protocol.

SAMP (see the IVOA Standard and additional material) is a messaging protocol that enables astronomy software tools to interoperate and communicate.

SAMP Web Profile allows web applications to interoperate and communicate with SAMP-capable desktop-based tools such as Aladin, Topcat or VOSpec. Standalone Browse Products (cubes, images and spectra) can be directly sent to these VO tools (without being locally retrieved) making use of the SAMP protocol, for visualization and analysis.

Launch one or more of the following software tools to interoperate with the Herschel Science Archive User Interface:

(http://aladin.u-strasbg.fr/java/download/aladin.jnlp): Interactive software sky atlas which allows users to visualise digitised astronomical images and superimpose entries from astronomical catalogues or databases. (Recommended for PACS and SPIRE Photometric observations)

(http://esavo.esac.esa.int/webstart/VOSpec.jnlp): A multi-wavelength spectral analysis tool with access to spectra, theoretical models and atomic and molecular line databases registered in the VO. (Recommended for PACS Spectroscopic observations)

(http://www.stsci.edu/institute/software_hardware/specview/download): Specview, like VOSpec is a multi-wavelength spectral analysis tool with access to spectra, theoretical models and atomic and molecular line databases. It also allows users to read multi-extension FITS files, which is recommended for SPIRE Pointing Spectroscopic observations.

(http://cassis.irap.omp.eu/?page=installation): A different multi-wavelength spectral analysis tool which also allows access to spectra, theoretical models and atomic and molecular line databases, recommended for HIFI 1D spectra.

(http://ds9.si.edu/site/Download.html): An astronomical imaging and data visualization application. It supports FITS images and binary tables, multiple frame buffers, region manipulation and many scale algorithms and colormaps. (Recommended for cubes, as found in HIFI Mapping and SPIRE Raster observations)

(http://www.cosmos.esa.int/web/herschel/hipe-download): The Herschel Interactive Processing Environment, developed specifically for working with Herschel data of any kind, including retrieving data products, interactive analysis, plotting of data and data manipulation.

In order to communicate between applications, SAMP makes use of a broker service for routing SAMP Messages (Hub application). This service is automatically provided by the above VO tools, with the exception of DS9.

If an SAMP Hub service is not running on your system, the HSA web interface will provide a prompt in order to start one. Then you must allow the interface to connect to this service and start your chosen VO tool, which will then be automatically connected.

# Data Access

Herschel TAP ([Table Access Protocol](#)) has been registered in the EuroVO registry and it is therefore available to any Virtual Observatory tool designed to access it, such as [TOPCAT](#).

## TOPCAT Access

The access to any registered TAP is located in in the main tool bar TOPCAT application under "VO" > "Table Access Protocol (TAP) Query".
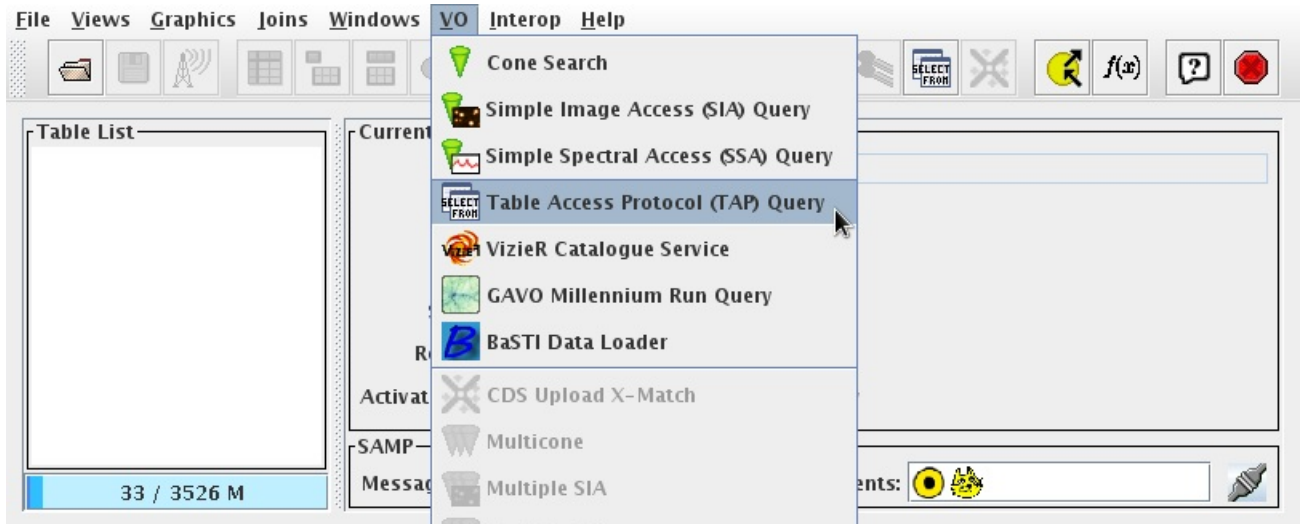


A new window allows the user to enter any keywords to to refine the search for a specific TAP. To search for Herschel Archive TAP, input "Herschel" in the "Locate TAP Service" tab and select the corresponding ESA VO TAP right at the bottom of the list (*ivo://esavo/hsa/tap*).
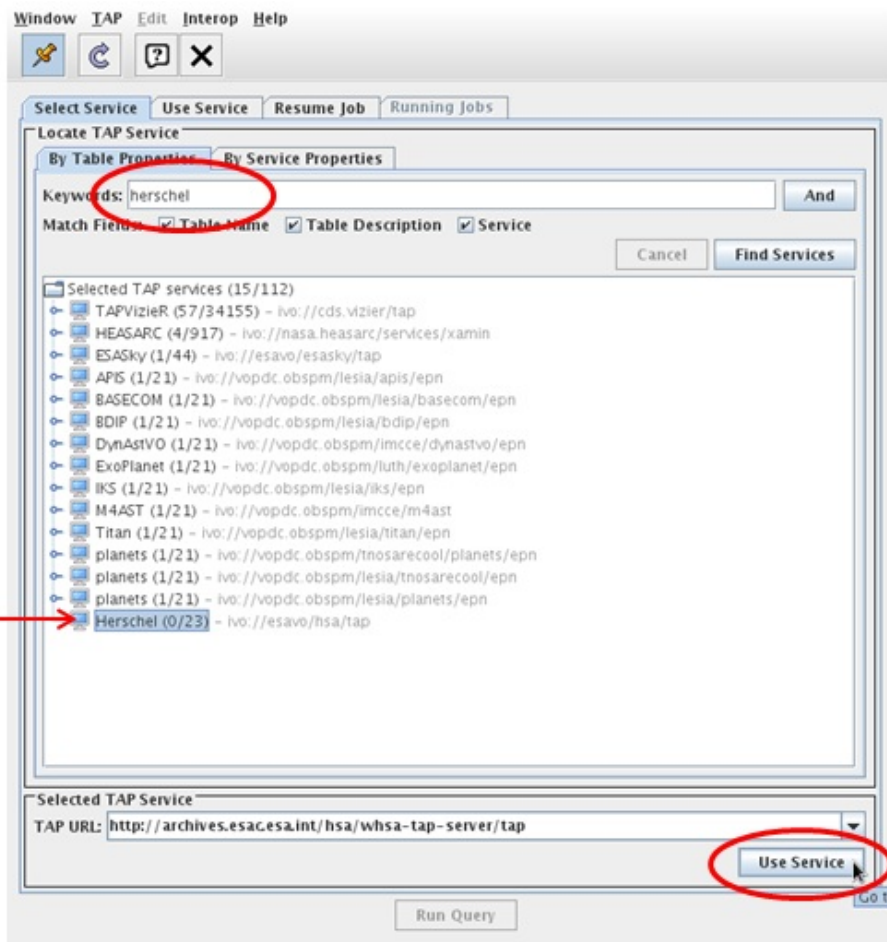
Once the service is selected, the user can navigate through the table tree to learn about the structure of the tables and the information provided. On the "ADQL Text" box at the bottom the user can query and retrieve the results using the ADQL ( Astronomical Data Query Language) as it can be seen in the image below.
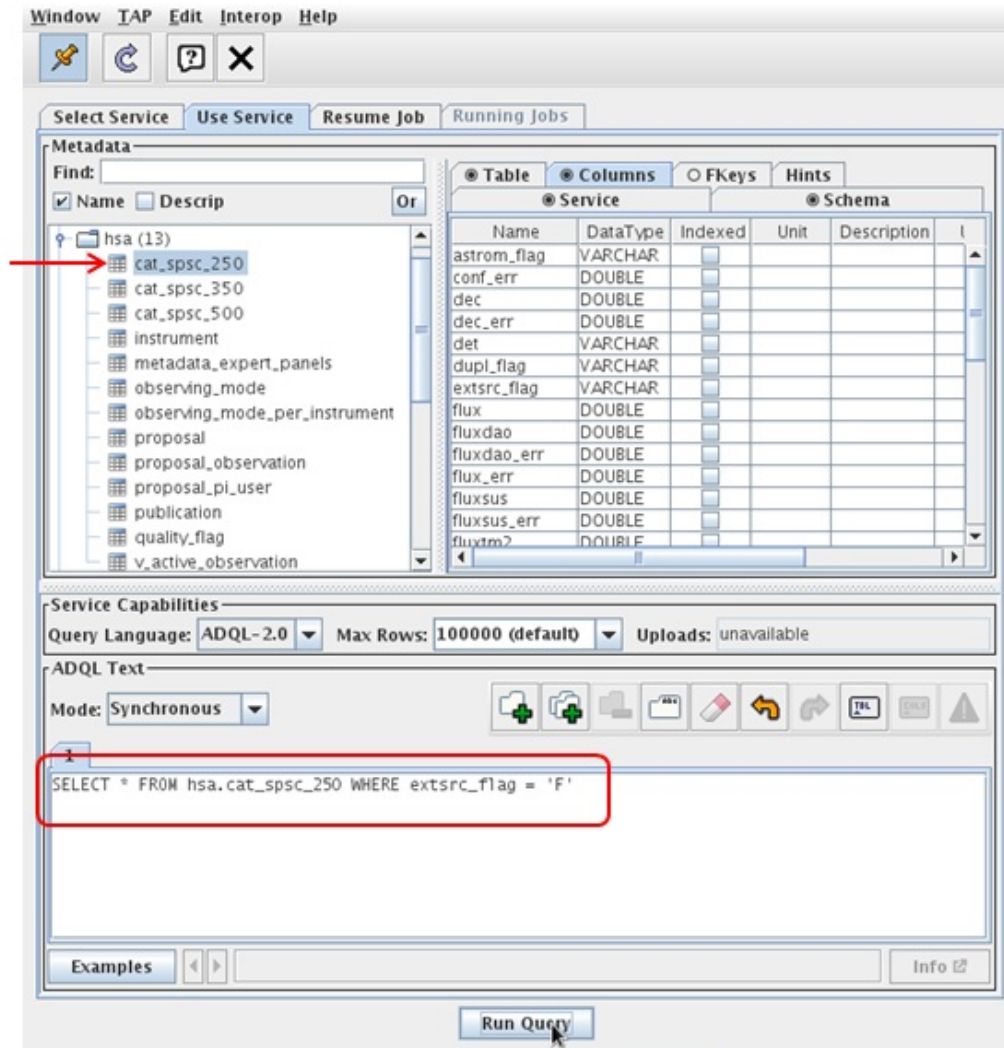
The table with the results from the query will finally be loaded into TOPCAT main context window.

## Direct Product Access using TAP

## 1. Product Access

Herschel TAP ([Table Access Protocol](#)) provides a product access at
`http://archives.esac.esa.int/whsa-tap-server/data` entry point.

The system will create a tar file automatically (compressed if it is requested) if the number of files to be retrieved are more than one file.

The requested file (or tar file) can be compressed if it is requested (`compress=true` parameter).

A single compressed file can be uncompressed if it is requested (`uncompress=true` parameter).

The following sections explain how to retrieve data using [curl](#) tool.

## 2. Observation Product Access

### 2.1. Product retrieval

By observation identifier, instrument name and level

```
curl "http://archives.esac.esa.int/hsa/whsa-tap-server/data?
  retrieval_type=OBSERVATION&observation_id=1342195355&
  instrument_name=PACS&product_level=LEVEL0" > retrieval.tar
```

Compressed:

```
curl "http://archives.esac.esa.int/hsa/whsa-tap-server/data?
  retrieval_type=OBSERVATION&observation_id=1342195355&
  instrument_name=PACS&product_level=LEVEL0&compress=true"
  > retrieval.tar.gz
```

### 2.2. Postcard retrieval

```
curl "http://archives.esac.esa.int/hsa/whsa-tap-server/data?
  retrieval_type=POSTCARD&observation_id=1342195355&
  instrument_name=PACS" > postcard.jpg
```

### 2.3. Ingestion request XML file retrieval

```
curl "http://archives.esac.esa.int/hsa/whsa-tap-server/data?
  retrieval_type=REQUESTFILE_XML&requestfile_xml=1342177457-
  herschel.ia.obs.ObservationContext-795122.xml" > request.xml
```

## 3. Standalone Browse Products

### 3.1. Readme file retrieval

By observation identifier, instrument name and level

```
curl "http://archives.esac.esa.int/hsa/whsa-tap-server/data?
  retrieval_type=STANDALONE&sabp_readme=true" > readme.pdf
```

### 3.2. Products retrieval

(It is possible to add some filters to provide the required products for different services: SIAP SSAP VO_APP)

By observation identifier, instrument name

```
curl "http://archives.esac.esa.int/hsa/whsa-tap-server/data?
  retrieval_type=STANDALONE&observation_id=1342195355&
  instrument_name=PACS" > sabp.tar
```

Only metadata:

```
curl "http://archives.esac.esa.int/hsa/whsa-tap-server/data?
  retrieval_type=STANDALONE&observation_id=1342195355&
  instrument_name=PACS&only_metadata=true" > sabp.tar
```

Using filter by SIAP:

```
curl "http://archives.esac.esa.int/hsa/whsa-tap-server/data?
  retrieval_type=STANDALONE&observation_id=1342195355&
  instrument_name=PACS&filter=SIAP" > sabp.tar
```

## 4. User Provided Data Product Access

### 4.1. Full retrieval

By keyword:

```
curl "http://archives.esac.esa.int/hsa/whsa-tap-server/data?
  retrieval_type=UPDP&updp_type=UPDP_FULL&keyword=GOODS" > updp.tar
```

By keyword and observation:

```
curl "http://archives.esac.esa.int/hsa/whsa-tap-server/data?
  retrieval_type=UPDP&updp_type=UPDP_FULL&updp_full_qualified=
  PEP_PACS#1342195355#PACS" > updp.tar
```

### 4.2. Postcard

Single postcard:

```
curl "http://archives.esac.esa.int/hsa/whsa-tap-server/data?
  retrieval_type=UPDP&updp_type=UPDP_POSTCARDS&keyword=GOODS
  &postcard_single=true"> postcard
```

All postcards:

```
curl "http://archives.esac.esa.int/hsa/whsa-tap-server/data?
  retrieval_type=UPDP&updp_type=UPDP_POSTCARDS&keyword=GOODS"
  > postcard
```

### 4.3. Data

```
curl "http://archives.esac.esa.int/hsa/whsa-tap-server/data?
  retrieval_type=UPDP&updp_type=UPDP_DATA&keyword=GOODS" > data.tar
```

### 4.4. Release notes

```
curl "http://archives.esac.esa.int/hsa/whsa-tap-server/data?
  retrieval_type=UPDP&updp_type=UPDP_RELEASE_NOTES&keyword=GOODS"
  > release_notes
```

### 4.5. Catalogues

```
curl "http://archives.esac.esa.int/hsa/whsa-tap-server/data?
  retrieval_type=UPDP&updp_type=UPDP_CATALOGUE&keyword=GOODS"
  > catalogs.tar
```

## 5. Highly Processed Data Product Access

### 5.1. Full retrieval

By keyword:

```
curl "http://archives.esac.esa.int/hsa/whsa-tap-server/data?
  retrieval_type=HPDP&hpdp_type=HPDP_FULL&keyword=GOODS" > hpdp.tar
```

By keyword and observation:

```
curl "http://archives.esac.esa.int/hsa/whsa-tap-server/data?
  retrieval_type=HPDP&hpdp_type=HPDP_FULL&hpdp_full_qualified=
  PEP_PACS#1342195355#PACS" > hpdp.tar
```

### 5.2. Postcard

Single postcard:

```
curl "http://archives.esac.esa.int/hsa/whsa-tap-server/data?
  retrieval_type=HPDP&hpdp_type=HPDP_POSTCARDS&keyword=GOODS&
  postcard_single=true"> postcard
```

All postcards:

```
curl "http://archives.esac.esa.int/hsa/whsa-tap-server/data?
```

```
retrieval_type=HPDP&hpdp_type=HPDP_POSTCARDS&keyword=GOODS"
  > postcard
```

## 5.3. Data

```
curl "http://archives.esac.esa.int/hsa/whsa-tap-server/data?
  retrieval_type=HPDP&hpdp_type=HPDP_DATA&keyword=GOODS"
   > data.tar
```

## 5.4. Release notes

```
curl "http://archives.esac.esa.int/hsa/whsa-tap-server/data?
  retrieval_type=HPDP&hpdp_type=HPDP_RELEASE_NOTES&keyword=GOODS"
  > release_notes
```

## 5.5. Catalogues

```
curl "http://archives.esac.esa.int/hsa/whsa-tap-server/data?
  retrieval_type=HPDP&hpdp_type=HPDP_CATALOGUE&keyword=GOODS"
   > catalogs.tar
```

# 6. HTTP parameters

## 6.1 Generic

| Parameter | Value | Comments |
|---|---|---|
| retrieval_type | <ul><li>OBSERVATION</li><li>PRODUCT</li><li>POSTCARD</li><li>POSTCARDFITS</li><li>REQUESTFILE_XML</li><li>STANDALONE</li><li>UPDP</li><li>HPDP</li></ul> | Mandatory |
| retrieval_access | <ul><li>DIRECT (default value): returns the file(s) directly</li><li>IDS: returns products identifiers only</li></ul> | The output format can be specified using 'retrieval_access_ids_format' parameter |
| retrieval_access_ids_format | <ul><li>xml (default value): returns the identifiers as XML</li><li>json: returns the identifiers as JSON</li></ul> | |
| compress | <ul><li>true</li><li>false</li></ul> | |
| uncompress | <ul><li>true</li><li>false</li></ul> | Can be used for single files only (I.e. it will not work for a tar |

| | | file) |
|---|---|---|

## 6.2 OBSERVATION Retrieval type

| Parameter | Value | Comments |
|---|---|---|
| observation_id | Observation identifier | |
| observation_oid | Observation internal identifier (unique) | This is the database identifier |
| instrument_name | • PACS<br>• SPIRE<br>• HIFI | |
| instrument_oid | • 1<br>• 2<br>• 3 | This is the database identifier |
| product_level | • ALL<br>• AUXILIARY<br>• CALIBRATION<br>• LEVEL0<br>• LEVEL0_5<br>• LEVEL1<br>• LEVEL2<br>• LEVEL2_5<br>• LEVEL3<br>• ALL-LEVEL3 | You may specify several levels separated by comma ','.<br><br>Eg.<br><br>`product_level=LEVEL0,LEVEL1` |

## 6.3 PRODUCT Retrieval type

| Parameter | Value | Comments |
|---|---|---|
| hcss_urn | Observation identifier | |

## 6.4 POSTCARD Retrieval type

See OBSERVATION Retrieval type for observation parameters.

Also:

| Parameter | Value | Comments |
|---|---|---|
| postcard_single | • true<br>• false (default value) | 'true' to retrieve one single postcard (main one) |

## 6.5 POSTCARDFITS Retrieval type

| Parameter | Value | Comments |
|---|---|---|
| product_oid | Internal product identifier (unique) | |

## 6.6 REQUESTFILE_XML Retrieval type

| Parameter | Value | Comments |
|---|---|---|
| requestfile_xml | Ingestion xml file name | |

## 6.7 STANDALONE Retrieval type

| Parameter | Value | Comments |
|---|---|---|
| sabp_readme | • true<br>• false (default value) | If 'true', only Readme file is returned. |
| filter | • SSAP<br>• SIAP<br>• VO_APP | Value to apply specific file filters. |
| metadata | • true<br>• false (default value) | If 'true' only files metadata are returned. |

## 6.8 UPDP Retrieval type

| Parameter | Value | Comments |
|---|---|---|
| updp_type | • UPDP_DATA: files<br>• UPDP_POSTCARDS: postcard files<br>• UPDP_FULL: complete set of files<br>• UPDP_RELEASE_NOTES: only release notes<br>• UPDP_CATALOGUE: only catalogues | |
| request_oid | request internal identifier | This parameter is used if 'updp_full_qualified' parameter is not present<br>this parameter cannot be present if 'keyword' is used |
| keyword | Keyword | This parameter is used if 'updp_full_qualified' and 'request_oid' parameters are not present |
| updp_full_qualified | REQOID#OBSID#INSTRUMENT | If this parameter is found, 'request_oid' parameter is not used. |

## 6.9 HPDP Retrieval type

| Parameter | Value | Comments |
|---|---|---|
| hpdp_type | • HPDP_DATA: files<br>• HPDP_POSTCARDS: postcard files<br>• HPDP_FULL: complete set of files<br>• HPDP_RELEASE_NOTES: | |

| | | |
|---|---|---|
| | only release notes<br>• HPDP_CATALOGUE: only catalogues | |
| request_oid | request internal identifier | This parameter is used if 'hpdp_full_qualified' parameter is not present<br>this parameter cannot be present if 'keyword' is used |
| keyword | Keyword | This parameter is used if 'hpdp_full_qualified' and 'request_oid' parameters are not present |
| hpdp_full_qualified | REQOID#OBSID#INSTRUMENT | If this parameter is found, 'request_oid' parameter is not used. |

## Command Line Access

The entry point is a TAP ([Table Access Protocol](#)) server.

You may use HTTP protocol to execute TAP requests at **http://archives.esac.esa.int/hsa/whsa-tap-server/tap**.

TAP provides two operation modes: Synchronous and Asynchronous:

- Synchronous: the response to the request will be generated as soon as the request received by the server.
- Asynchronous: the server will start a job that will execute the request. The first response to the request is the required information (a link) to obtain the job status. Once the job is finished, the results can be retrieved.

Our TAP server provides two access mode: public and authenticated:

- Public: this is the standard TAP access. A user can execute ADQL queries and upload tables to be used in a query 'on-the-fly' (these tables will be removed once the query is executed). The results are available to any other user and they will remain in the server for a limited space of time.
- Authenticated: some functionalities are restricted to authenticated users only The results are saved in a private user space and they will remain in the server for ever (they can be removed by the user).
    - ADQL queries and results are saved in a user private area.
    - Persintance of uploaded tables: a user can upload a table in a private space.

You can find more about ADQL at:

- [GAVO ADQL reference card](#)
- [VizierR: ADQL cheat sheet](#)

Here you can find some examples about how to interact with a TAP server (we are using the [curl](#) tool):

## 1. Non authenticated access

### 1.1. Getting all public tables

```
curl "http://archives.esac.esa.int/hsa/whsa-tap-server/tap/tables"
```

### 1.2. Synchronous query

```
curl "http://archives.esac.esa.int/hsa/whsa-tap-server/tap/sync?
  REQUEST=doQuery&LANG=ADQL&FORMAT=votable&QUERY=SELECT+TOP+5+*
  +FROM+hsa.v_active_observation"
```

The retrieved results is a VO table by default (see '3.2. Synchronous Queries' section parameters to specify a different output format). The results can be saved in a file and inspected using any analysis tool like TOPCAT, for instance.

### 1.3. Synchronous query on an 'on-the-fly' uploaded table

```
curl --form UPLOAD="table_c,param:table1" --form
  table1=@test_ra_dec.vot --form LANG=ADQL --form REQUEST=
  doQuery --form QUERY="select top 5 * from tap_upload.table_c"
  http://archives.esac.esa.int/hsa/whsa-tap-server/tap/sync
```

Where `'test_ra_dec.vot'` is a file that contains the VOTable to be uploaded (in order to be used by the query)

The retrieved results is a VO table by default (see '3.2. Synchronous Queries' section parameters to specify a different output format). The results can be saved in a file and inspected using any analysis tool like TOPCAT, for instance.

**1.4. Asynchronous query**

```
curl -i -X POST --data
  "PHASE=run&LANG=ADQL&LANG=ADQL&REQUEST=doQuery&
  QUERY=select+top+5+*+from+hsa.v_active_observation"
  "http://archives.esac.esa.int/hsa/whsa-tap-server/tap/async"
```

Note that there is the possibility to use the optional parameters "JOBNAME" to assign a name to the job and "JOBDESCRIPTION" to add a description:

```
curl -i -X POST --data
      "PHASE=run&LANG=ADQL&JOBNAME=optionalJobName&JOB
      DESCRIPTION=optionalDescription&LANG=ADQL&REQUEST=doQuery
      &QUERY=select+top+5+*+from+hsa.v_active_observation"
      "http://archives.esac.esa.int/hsa/whsa-tap-server/tap/async"
```

The response will contain the URL of the job running at server side (see `Location` header):

```
HTTP/1.1 303 See Other
Date: Mon, 30 Jun 2014 14:44:39 GMT
Server: Apache-Coyote/1.1
Location: http://archives.esac.esa.int/hsa/whsa-tap-server/tap/async/
          1404139480755A
Content-Type: application/x-www-form-urlencoded
Connection: close
Transfer-Encoding: chunked
```

To obtain the status of the running job:

```
curl "http://archives.esac.esa.int/hsa/whsa-tap-server/tap/async/
  1404139480755A"
```

The status response is something like:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```xml
<uws:job xmlns:uws="http://www.ivoa.net/xml/UWS/v1.0" xmlns:
xlink="http://www.w3.org/1999/xlink" xmlns:xs="http://www.w3.
org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        <uws:jobId><![CDATA[1404139480755A]]></uws:jobId>
        <uws:runId xsi:nil="true" />
        <uws:ownerId><![CDATA[anonymous]]></uws:ownerId>
        <uws:phase>COMPLETED</uws:phase>
        <uws:quote xsi:nil="true" />
        <uws:startTime>2014-06-30T16:44:40.766+0200</uws:startTime>
        <uws:endTime>2014-06-30T16:44:40.830+0200</uws:endTime>
        <uws:executionDuration>0</uws:executionDuration>
        <uws:destruction>2014-07-07T16:44:40.754+0200
        </uws:destruction>
        <uws:parameters>
                <uws:parameter id="maxRec"><![CDATA[100000]]>
                </uws:parameter>
                <uws:parameter id="query"><![CDATA[select top 5 * from
                hsa.v_active_observation]]></uws:parameter>
                <uws:parameter id="request"><![CDATA[doQuery]]>
                </uws:parameter>
                <uws:parameter id="format"><![CDATA[votable]]>
                </uws:parameter>
                <uws:parameter id="keepAuthenticatedUserJobs">
                <![CDATA[true]]></uws:parameter>
                <uws:parameter id="lang"><![CDATA[ADQL]]>
                </uws:parameter>
                <uws:parameter id="version"><![CDATA[1.0]]>
                </uws:parameter>
        </uws:parameters>
        <uws:results>
                <uws:result id="result" xlink:type="simple" xlink:href=
                "http%3A%2F%2Farchives.esac.esa.int%2Fhsa%2Fwhsa-tap-
                server%2Ftap%2Fasync%2F1404139480755A%2Fresults%2Fresult"
                mime="application/x-votable+xml" size="8863" rows="6"/>
        </uws:results>
        <uws:errorSummary xsi:nil="true" />
```

To obtain the results of the job (once the job is finished):

```
curl "http://archives.esac.esa.int/hsa/whsa-tap-server/tap/async/
  1404139480755A/results/result"
```

The retrieved results is a VO table by default (see '3.3. Asynchronous Queries' section parameters to specify a different output format). The results can be saved in a file and inspected using any analysis tool like TOPCAT, for instance.

**1.5. Python script to execute an asynchronous query and wait until the results are available**

```python
#ASYNCHRONOUS REQUEST

import httplib
import urllib
#import http.client in Python 3
```

```python
#import urllib.parse in Python 3
import time
from xml.dom.minidom import parseString

host = "archives.esac.esa.int"
port = 80
pathinfo = "/hsa/whsa-tap-server/tap/async"


#-----------------------------------
#Create job

params = urllib.urlencode({\
        "REQUEST": "doQuery", \
        "LANG":    "ADQL", \
        "FORMAT":  "votable", \
        "PHASE":   "RUN", \
        "JOBNAME":  "Any name (optional)", \
        "JOBDESCRIPTION":  "Any description (optional)", \
        "QUERY":    "SELECT DISTANCE(POINT('ICRS',ra,dec), \
        POINT('ICRS',266.41683,-29.00781)) AS dist, * FROM \
        hsa.v_active_observation \
        WHERE 1=CONTAINS(POINT('ICRS',ra,dec), \
        CIRCLE('ICRS',266.41683,-29.00781, 0.08333333)) \
        ORDER BY dist ASC" \
        })

headers = {\
        "Content-type": "application/x-www-form-urlencoded", \
        "Accept":       "text/plain" \
        }

connection = httplib.HTTPConnection(host, port)
connection.request("POST",pathinfo,params,headers)

#Status
response = connection.getresponse()
print "Status: " +str(response.status), "Reason: " + str(response.reason)

#Server job location (URL)
location = response.getheader("location")
print "Location: " + location

#Jobid
jobid = location[location.rfind('/')+1:]
print "Job id: " + jobid

connection.close()

#-----------------------------------
#Check job status, wait until finished

while True:
        connection = httplib.HTTPConnection(host, port)
        connection.request("GET",pathinfo+"/"+jobid)
        response = connection.getresponse()
        data = response.read()
```

```
        #XML response: parse it to obtain the current status
        dom = parseString(data)
        phaseElement = dom.getElementsByTagName('uws:phase')[0]
        phaseValueElement = phaseElement.firstChild
        phase = phaseValueElement.toxml()
        print "Status: " + phase
        #Check finished
        if phase == 'COMPLETED': break
        #wait and repeat
        time.sleep(0.2)

#print "Data:"
#print data

connection.close()


#--------------------------------------
#Get results
connection = httplib.HTTPConnection(host, port)
connection.request("GET",pathinfo+"/"+jobid+"/results/result")
response = connection.getresponse()
data = response.read()
outputFileName = "example3_votable_output.vot"
outputFile = open(outputFileName, "w")
outputFile.write(data)
outputFile.close()
connection.close()
print "Data saved in: " + outputFileName
```

The saved file is a VO table (by default, see '3.3. Asynchronous Queries' section parameters to specify a different output format). The file can be inspected using any analysis tool like TOPCAT, for instance.

## 2. Authenticated access

### 2.1. Login

```
curl -k -c cookies.txt -X POST -d username=USERNAME -d
  password=PASSWORD -L "https://archives.esac.esa.int/hsa/whsa-tap-server,
```

### 2.2. Logout

```
curl -k -b cookies.txt -X POST -d -L
  "https://archives.esac.esa.int/hsa/whsa-tap-server/logout"
```

### 2.3. Getting public and user tables

```
curl -k -b cookies.txt -X POST -L
  "https://archives.esac.esa.int/hsa/whsa-tap-server/tap/tables"
```

## 2.4. Asynchronous query

```
curl -k -b cookies.txt -i -X POST --data
  "PHASE=run&LANG=ADQL&REQUEST=doQuery&QUERY=select+top+5+*
  +from+hsa.v_active_observation" "https://archives.esac.esa.int/hsa/whsa
  tap/async"
```

ℹ Note that there is the possibility to use the optional parameters "JOBNAME" to assign a name to the job and "JOBDESCRIPTION" to add a description:

```
curl -k -b cookies.txt -i -X POST --data
     "PHASE=run&LANG=ADQL&JOBNAME=optionalJobName&JOB
     DESCRIPTION=optionalDescription&REQUEST=doQuery&
     QUERY=select+top+5+*+from+hsa.v_active_observation"
     "https://archives.esac.esa.int/hsa/whsa-tap-server/tap/async"
```

The response will contain the URL of the job running at server side:

```
HTTP/1.1 303 See Other
Date: Mon, 30 Jun 2014 15:02:00 GMT
Server: Apache-Coyote/1.1
Location: http://archives.esac.esa.int/hsa/whsa-tap-server/tap/async/
          1404140520859A
Content-Type: application/x-www-form-urlencoded
Connection: close
Transfer-Encoding: chunked
```

To obtain the status of the running job:

```
curl -k -b cookies.txt
  "https://archives.esac.esa.int/hsa/whsa-tap-server/tap/async/
  1404140520859A"
```

The status response is something like:

```
<?xml version="1.0" encoding="UTF-8"?>
<uws:job xmlns:uws="http://www.ivoa.net/xml/UWS/v1.0" xmlns:xlink=
"http://www.w3.org/1999/xlink" xmlns:xs="http://www.w3.org/2001/XMLSchema
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <uws:jobId><![CDATA[1404141177261A]]></uws:jobId>
        <uws:runId xsi:nil="true" />
        <uws:ownerId><![CDATA[USERNAME]]></uws:ownerId>
        <uws:phase>COMPLETED</uws:phase>
        <uws:quote xsi:nil="true" />
        <uws:startTime>2014-06-30T17:12:57.273+0200</uws:startTime>
        <uws:endTime>2014-06-30T17:12:57.864+0200</uws:endTime>
        <uws:executionDuration>0</uws:executionDuration>
        <uws:destruction>2014-07-07T17:23:31.362+0200
        </uws:destruction>
        <uws:parameters>
```

```
                    <uws:parameter id="jobdescription"><![CDATA[]]>
                    </uws:parameter>
                    <uws:parameter id="jobname"><![CDATA[]]>
                    </uws:parameter>
                    <uws:parameter id="session"><![CDATA[1404141103551]]>
                    </uws:parameter>
                    <uws:parameter id="maxRec"><![CDATA[100000]]>
                    </uws:parameter>
                    <uws:parameter id="query"><![CDATA[SELECT DISTANCE
                    (POINT('ICRS',ra,dec), POINT('ICRS',266.41683,-29.00781))
FROM hsa.v_active_observation
WHERE 1=CONTAINS(POINT('ICRS',ra,dec),CIRCLE('ICRS',266.41683,-29.00781,
ORDER BY dist ASC]]></uws:parameter>
                    <uws:parameter id="request"><![CDATA[doQuery]]>
                    </uws:parameter>
                    <uws:parameter id="keepAuthenticatedUserJobs">
                    <![CDATA[true]]></uws:parameter>
                    <uws:parameter id="format"><![CDATA[votable]]>
                    </uws:parameter>
                    <uws:parameter id="lang"><![CDATA[ADQL]]>
                    <uws:parameter>
                    <uws:parameter id="version"><![CDATA[1.0]]>
                    </uws:parameter>
        </uws:parameters>
        <uws:results>
                    <uws:result id="result" xlink:type="simple" xlink:href=
                    "http%3A%2F%2Farchives.esac.esa.int%2Fhsa%2Fwhsa-tap-
                    server%2Ftap%2Fasync%2F1404141177261A%2Fresults%2Fresult"
                    mime="application/x-votable+xml"
                    size="2468741" rows="4734" />
        </uws:results>
        <uws:errorSummary xsi:nil="true" />
```

To obtain the results of the job (once the job is finished):

```
curl -k -b cookies.txt
  "https://archives.esac.esa.int/hsa/whsa-tap-server/tap/async/
  1404140520859A/results/result"
```

The retrieved results is a VO table by default (see '3.3. Asynchronous Queries' section parameters to specify a different output format). The results can be saved in a file and inspected using any analysis tool like TOPCAT, for instance.

### 2.7. Listing jobs

Jobs can be listed using the following request:

```
curl -k -b cookies.txt
  "https://archives.esac.esa.int/hsa/whsa-tap-server/tap/jobs/list?
  offset=index&limit=jobs_number&order=order"
```

Where list specifies the jobs list (e.g. 'sync' or 'async'). It is mandatory.

index specifies the number of jobs to skip before beginning to return the first job. By default, it is '0'.

limit specifies the number of jobs to be returned. No limit by default is set.

order specifies the order of the results.

Example

```
curl -k -b cookies.txt
  "https://archives.esac.esa.int/hsa/whsa-tap-server/tap/jobs/async?
  offset=0&limit=20&order=CREATION_TIME:DESC"
```

**2.8. Deleting jobs**

Jobs can be deleted using their identifiers:

```
curl -k -b cookies.txt -X POST --data
  "JOB_IDS=job_id1,job_id2..." "https://archives.esac.esa.int/hsa/
  whsa-tap-server/tap/deletejobs"
```

## 3. Interface

See the following specifications:

- [TAP (Table Access Protocol)](#)
- [UWS (Universal Worker Service)](#)
- [ADQL (Astronomical Data Query Language)](#)

**3.1. TAP resources**

```
http://archives.esac.esa.int/hsa/whsa-tap-server/tap/
```

| Tables | `http://archives.esac.esa.int/hsa/whsa-tap-server/tap/tables` | |
|---|---|---|
| Synchronous access | `http://archives.esac.esa.int/hsa/whsa-tap-server/tap/sync` | |
| Asynchronous access | `http://archives.esac.esa.int/hsa/whsa-tap-server/tap/async` | |
| Service availability | `http://archives.esac.esa.int/hsa/whsa-tap-server/tap/availability` | |
| Events capability | `http://archives.esac.esa.int/hsa/whsa-tap-server/tap/event` | TAP+ |
| Notifications capability | `http://archives.esac.esa.int/hsa/whsa-tap-server/tap/notification` | TAP+ |
| Jobs listing capability | `http://archives.esac.esa.int/hsa/whsa-tap-server/tap/jobs` | TAP+ |
| Jobs removal capability | `http://archives.esac.esa.int/hsa/whsa-tap-server/tap/deletejobs` | TAP+ |

**3.2. Synchronous Queries**

| Parameter | Value | Comments |
|---|---|---|
| REQUEST | doQuery | Requests to execute the provided query |
| LANG | ADQL | Query language |
| | | |

| FORMAT | - votable<br>- votable_plain<br>- csv<br>- json | Results output format |
|---|---|---|
| QUERY | ADQL query | query |

### 3.3. Asynchronous Queries

| Parameter | Value | Comments |
|---|---|---|
| Same parameters as defined in *3.2 Synchronous Queries* and | | |
| PHASE | run | Query job initial phase |

The response header will contain the location of the job.

### 3.4. Query on an 'on-the-fly' uploaded table

Use a multipart/form-data (see IETF RFC 2388) HTTP POST

| Parameter | Value | Comments |
|---|---|---|
| Same parameters as defined in *3.2 Synchronous Queries* and | | |
| UPLOAD | query_table,<br>param:parameter_table_name | query_table: the name of the table used in the query<br>parameter_table_name: HTTP parameter name that points to the table to be uploaded |
| parameter_table_name | file | file name that contains the table to be uploaded |

For instance, in the following request:

```
curl --form UPLOAD="table_c,param:table1" --form
  table1=@test_ra_dec.vot --form LANG=ADQL --form REQUEST=doQuery --form
  QUERY="select top 5 * from tap_upload.table_c" http://archives.esac.esa
  whsa-tap-server/tap/sync
```

`table_c` is the name of the table used in the query: `QUERY="select top 5 * from tap_upload.table_c"`,
`table1` is the name of the HTTP parameter that provides the file: `table1=@test_ra_dec.vot`
and `test_ra_dec.vot` is the file name that contains the table to be uploaded.

### 3.5. TAP+ login

| Parameter | Value | Comments |
|---|---|---|
| username | user_name | User name |
| password | user_password | User password |

The response header will contain the session identifier.

### 3.6. TAP+ logout

| Parameter | Value | Comments |
|---|---|---|
| session identifier | session identifier | Session identifier provided by a login request<br>Must be added to the HTTP header |

### 3.7. TAP+ tables capabilities

In addition to the standar 'tables' TAP capability, the following parameters can be used too:

| Parameter | Value | Comments |
|---|---|---|
| tables | comma separated full qualified table names | A lis of the specified tables will be returned |
| schemas | comma separated schema names | A list of the specified schemas will be returned |
| only_tables | TRUE / FALSE (default: FALSE) | TRUE: no columns information will be returned |
| only_schemas | TRUE / FALSE (default: FALSE) | TRUE: no tables nor columns information will be returned |

These parameters are handled based on the following priorities

| Priority | Parameter | Comments |
|---|---|---|
| 1. | tables != null | No more checks are performed (share_info and share_accessible are handled if present) |
| 2. | tables == null (default) | More checks are performed |
| 2.1. | only_schemas = TRUE | No more checks are performed |
| 2.2. | only_schemas = FALSE (default) | The following checks are performed |
| 2.2.1. | schema_names != null | The following parameters are applied to the specified schemas only |
| 2.2.2 | only_tables = TRUE | No columns data are generated |
| 2.2.3 | only_tables = FALSE (default) | Columns data are generated |

### 3.8. TAP+ events capabilities

Tables creation/removal/sharing actions generate events. Events are grouped by types and each type has a 'last modification time' associated. Those events and times can be retrieved using:

| Parameter | Value | Comments |
|---|---|---|
| id | event type identifier | Optional parameter. If not provided, all events are returned. If provided, only the requested type is returned. |

The current event types are:

| Value | Comments |
|---|---|
| 100 | Job created |
| 101 | Job updated |
| 102 | Job removed |
| 210 | Shared item created |
| 211 | Shared item updated |
| 212 | Shared item removed |
| 220 | Shared group created |
| 221 | Shared group updated |
| 222 | Shared group removed |
| 230 | Shared user created |
| 231 | Shared user updated |
| 232 | Shared user removed |
| 300 | Log in |
| 301 | Log out |
| | |