SODA-next, followed by some other thoughts about DAP(SIA) and DataLink implementation

F.Bonnarel and co-authors + DAL WG

CENTRE DE DONNÉES ASTRONOMIQUES DE STRASBOURG

Table of content

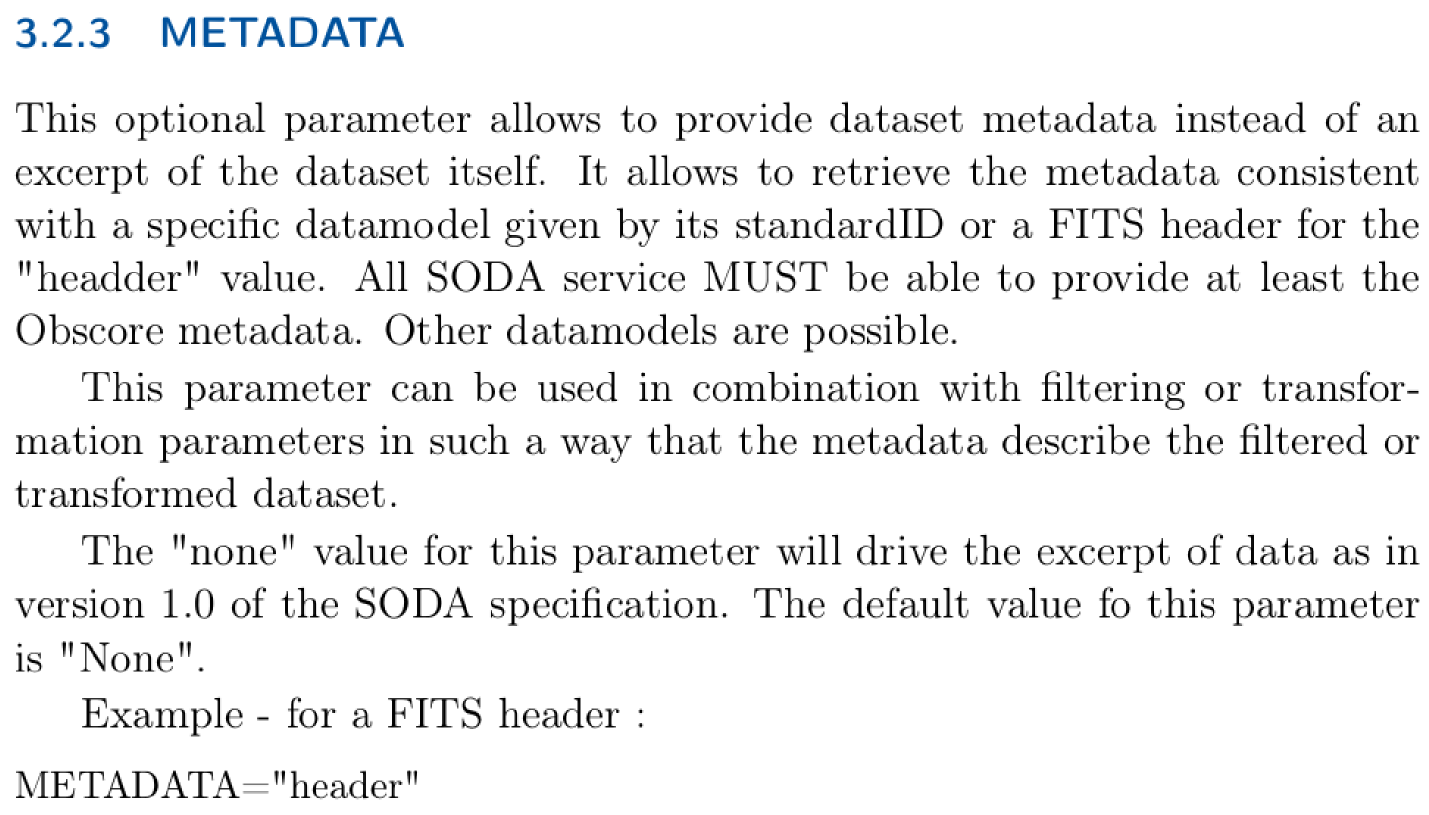
SODA-next : why ?

- What kind of server side operations for data access do we (want/need) to standardize ?
  - SODA-1.0 is only selecting data samples from a dataset.
  - On the other side UWS services may provide a lot of ad hoc functionalities but only the process managment is standardized
  - Emerging projects datasets are huge (eg SKA data : cube of petabytes).
    - Even a single spectral slice could be a couple of hundreds gigabytes or a terabyte
    - Simple SODA-1.0 extraction is limited , but
    - Even in « code to the data » context, SODA extraction may be useful
  - Drivers for extending SODA standardization
    - Delivering extended (standard) metadata
    - Advanced preview functionality
    - rebinning/resampling
    - Data product type transformation

Extending SODA standardization

- Delivering extended (standard) metadata
  - WCS (transform) : use case (A.Micol and ESO):
    - Extract sub cubes from a velocity cube requires WCS information
    - A SODA mode could deliver the WCS metadata
  - Other candidates : Dataset-Cube DM, CAOM, last step provenance ?
  - Prototype available at CADC

Delivering extended (standard) metadata



3.2.3 METADATA

This optional parameter allows to provide dataset metadata instead of an excerpt of the dataset itself. It allows to retrieve the metadata consistent with a specific datamodel given by its standardID or a FITS header for the "headder" value. All SODA service MUST be able to provide at least the Obscore metadata. Other datamodels are possible.

This parameter can be used in combination with filtering or transformation parameters in such a way that the metadata describe the filtered or transformed dataset.

The "none" value for this parameter will drive the excerpt of data as in version 1.0 of the SODA specification. The default value fo this parameter is "None".

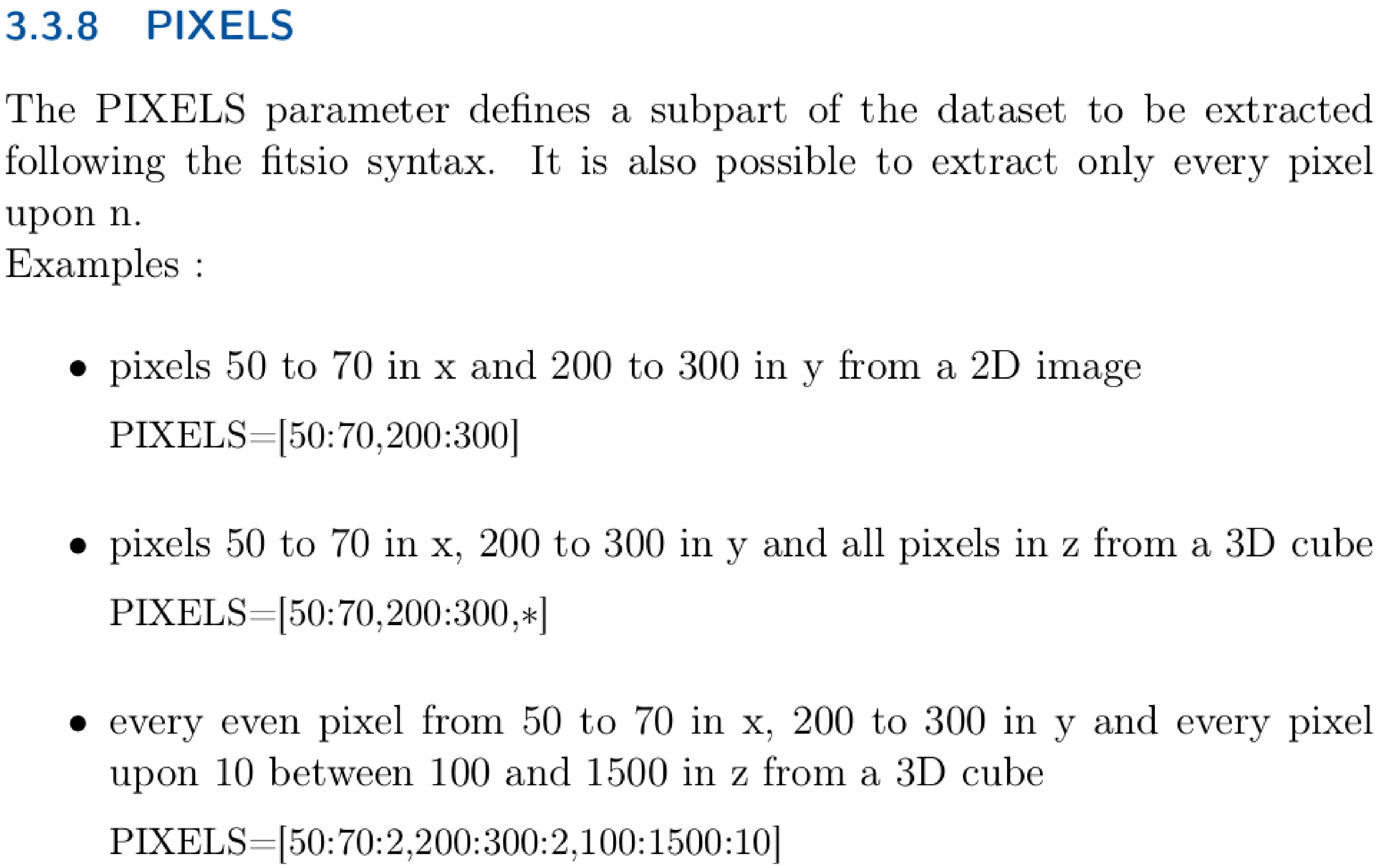
Example - for a FITS header :

METADATA="header"

Extending SODA standardization

- Advanced preview functionality
  - Extracting 1 upon n pixel
  - On the fly HiPS génération
    - 3D HiPS (on the fly tile)
    - 3D → 2D reduction.

Advanced preview functionality



3.3.8 PIXELS

The PIXELS parameter defines a subpart of the dataset to be extracted following the fitsio syntax. It is also possible to extract only every pixel upon n.
Examples :

- pixels 50 to 70 in x and 200 to 300 in y from a 2D image

  PIXELS=[50:70,200:300]

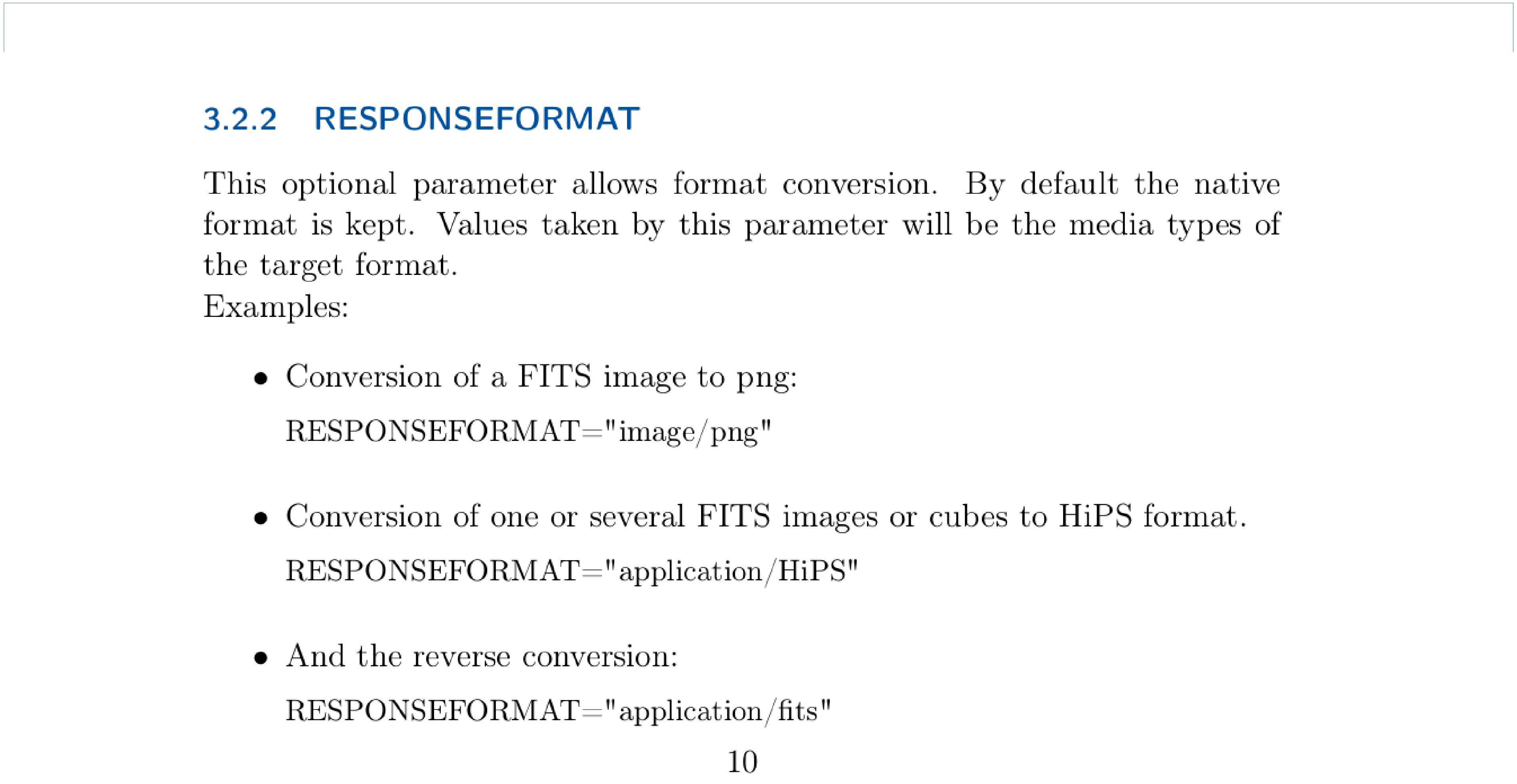
- pixels 50 to 70 in x, 200 to 300 in y and all pixels in z from a 3D cube

  PIXELS=[50:70,200:300,*]

- every even pixel from 50 to 70 in x, 200 to 300 in y and every pixel upon 10 between 100 and 1500 in z from a 3D cube

  PIXELS=[50:70:2,200:300:2,100:1500:10]

Advanced preview functionality



3.2.2 RESPONSEFORMAT

This optional parameter allows format conversion. By default the native format is kept. Values taken by this parameter will be the media types of the target format.
Examples:

- Conversion of a FITS image to png:

  RESPONSEFORMAT="image/png"

- Conversion of one or several FITS images or cubes to HiPS format.

  RESPONSEFORMAT="application/HiPS"

- And the reverse conversion:

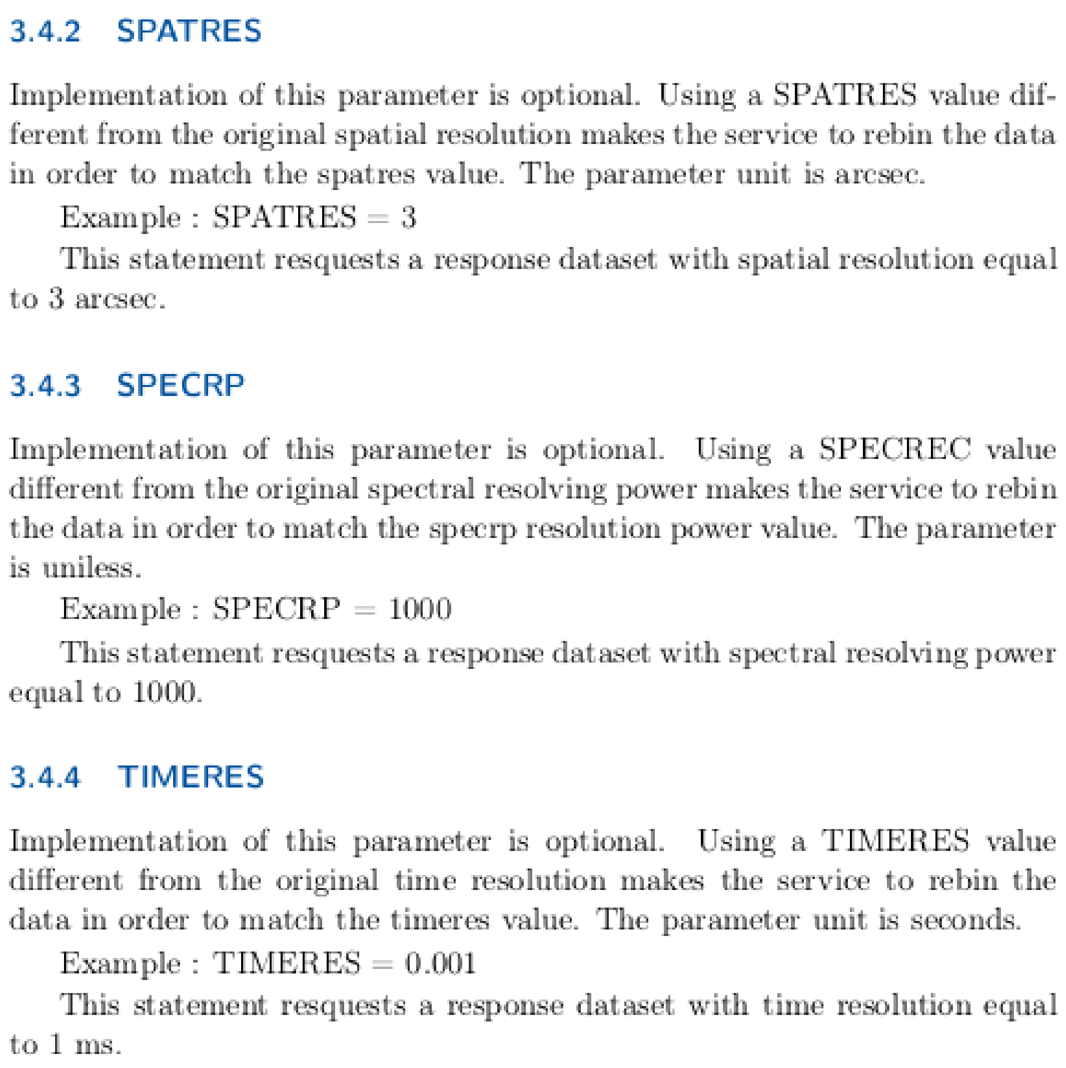
  RESPONSEFORMAT="application/fits"

10

Extending SODA standardization

- Rebinning/resampling
  - required for many reasons
    - (dataset comparisons, get rid of odd (HPX) projections, preview generation, etc.)
    - Adding PROJ, SPECRP, SPATRES, etc.

Rebinning/resampling



3.4.2 SPATRES

Implementation of this parameter is optional. Using a SPATRES value different from the original spatial resolution makes the service to rebin the data in order to match the spatres value. The parameter unit is arcsec.

Example : SPATRES = 3

This statement resquests a response dataset with spatial resolution equal to 3 arcsec.

3.4.3 SPECRP

Implementation of this parameter is optional. Using a SPECREC value different from the original spectral resolving power makes the service to rebin the data in order to match the specrp resolution power value. The parameter is unitless.

Example : SPECRP = 1000

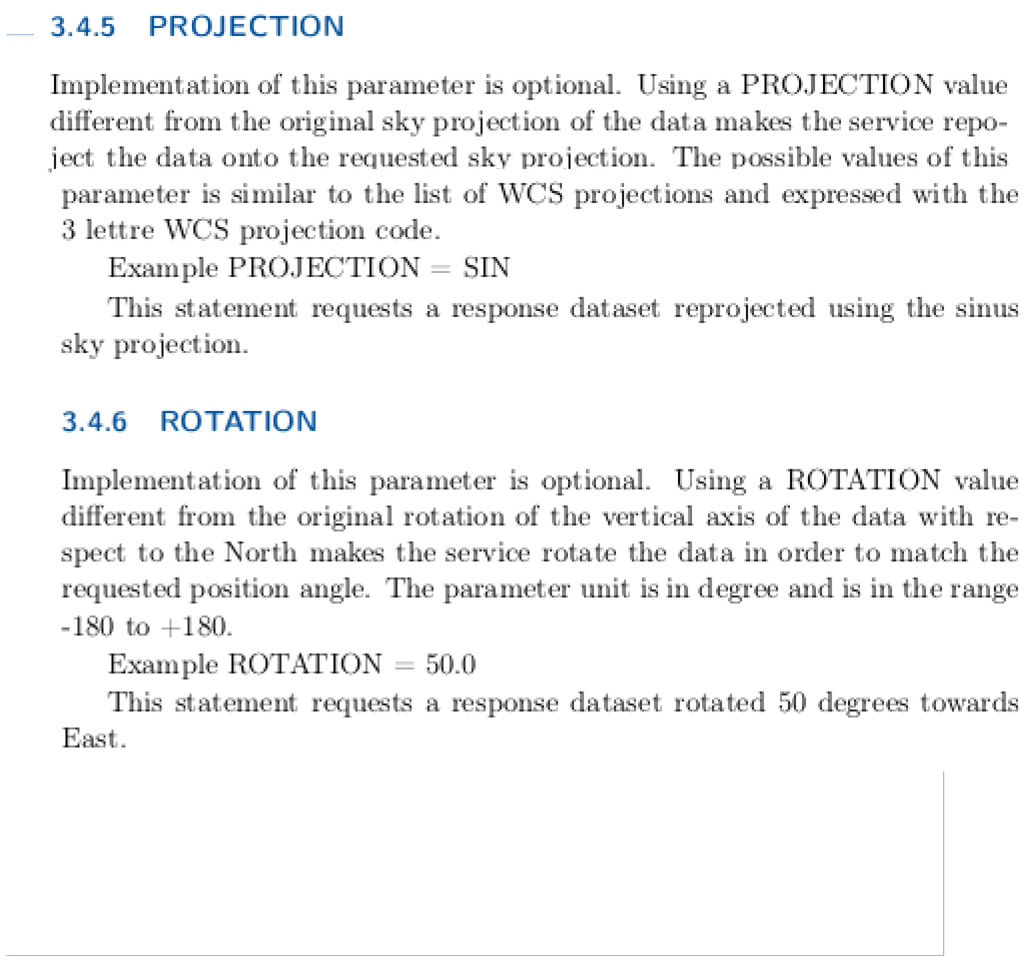
This statement resquests a response dataset with spectral resolving power equal to 1000.

3.4.4 TIMERES

Implementation of this parameter is optional. Using a TIMERES value different from the original time resolution makes the service to rebin the data in order to match the timeres value. The parameter unit is seconds.

Example : TIMERES = 0.001

This statement resquests a response dataset with time resolution equal to 1 ms.



3.4.5 PROJECTION

Implementation of this parameter is optional. Using a PROJECTION value different from the original sky projection of the data makes the service reproject the data onto the requested sky projection. The possible values of this parameter is similar to the list of WCS projections and expressed with the 3 lettre WCS projection code.

Example PROJECTION = SIN

This statement requests a response dataset reprojected using the sinus sky projection.

3.4.6 ROTATION

Implementation of this parameter is optional. Using a ROTATION value different from the original rotation of the vertical axis of the data with respect to the North makes the service rotate the data in order to match the requested position angle. The parameter unit is in degree and is in the range -180 to +180.

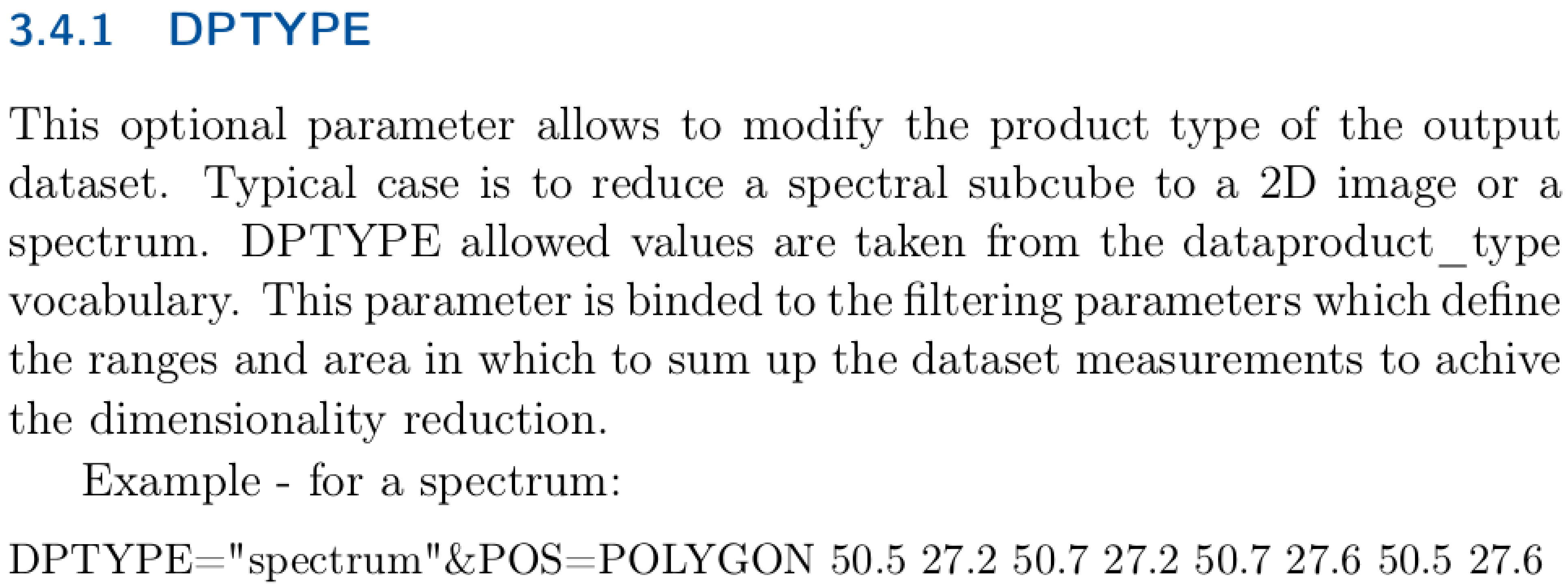
Example ROTATION = 50.0

This statement requests a response dataset rotated 50 degrees towards East.

Extending SODA standardization

- Data product type transformation
  - Choose between image, spectrum and cube
  - CASDA provides two different SODA services (spectrum and cube extraction)
  - Proposal to use DPTYPE to drive this

Data product type transformation



3.4.1 DPTYPE

This optional parameter allows to modify the product type of the output dataset. Typical case is to reduce a spectral subcube to a 2D image or a spectrum. DPTYPE allowed values are taken from the dataproduct_type vocabulary. This parameter is binded to the filtering parameters which define the ranges and area in which to sum up the dataset measurements to achive the dimensionality reduction.

Example - for a spectrum:

DPTYPE="spectrum"&POS=POLYGON 50.5 27.2 50.7 27.2 50.7 27.6 50.5 27.6

How it could work on a couple of use cases

- Preliminary statement :
    - Filtering and transform parameters force the ObsCore characterization of the response
- Extract a spectrum from a cube around a position
    - ID=ivo://sample.org/dataset1&DPTYPE=Spectrum&CIRCLE=5.0 26.0 0.4
- Degrade time_res of a TIMESERIES at 1 minute to improve SNR
    - ID=ivo://sample.org/ts1&DPTYPE=timeseries&TIMERES=60
- Extract 2D HiPS on the fly from a cube
    - ID=ivo://sample.org/cube1&DPTYPE=image&RESPONSEFORMAT=application/HiPS&POLYGON=*a1 a2 b1 b2 c1 c2 d1 d2*&SPATRES=5

SODA
Other changes

- MOC parameter
  - – I get a response from a service which provides a MOC.
  - – I want to retrieve a cutout matching this MOC
- Multiple input parameters : choic of combinations issue
  - Allowed combinations described in Json-LD ? See INAF
- Coordinate system change ? See INAF

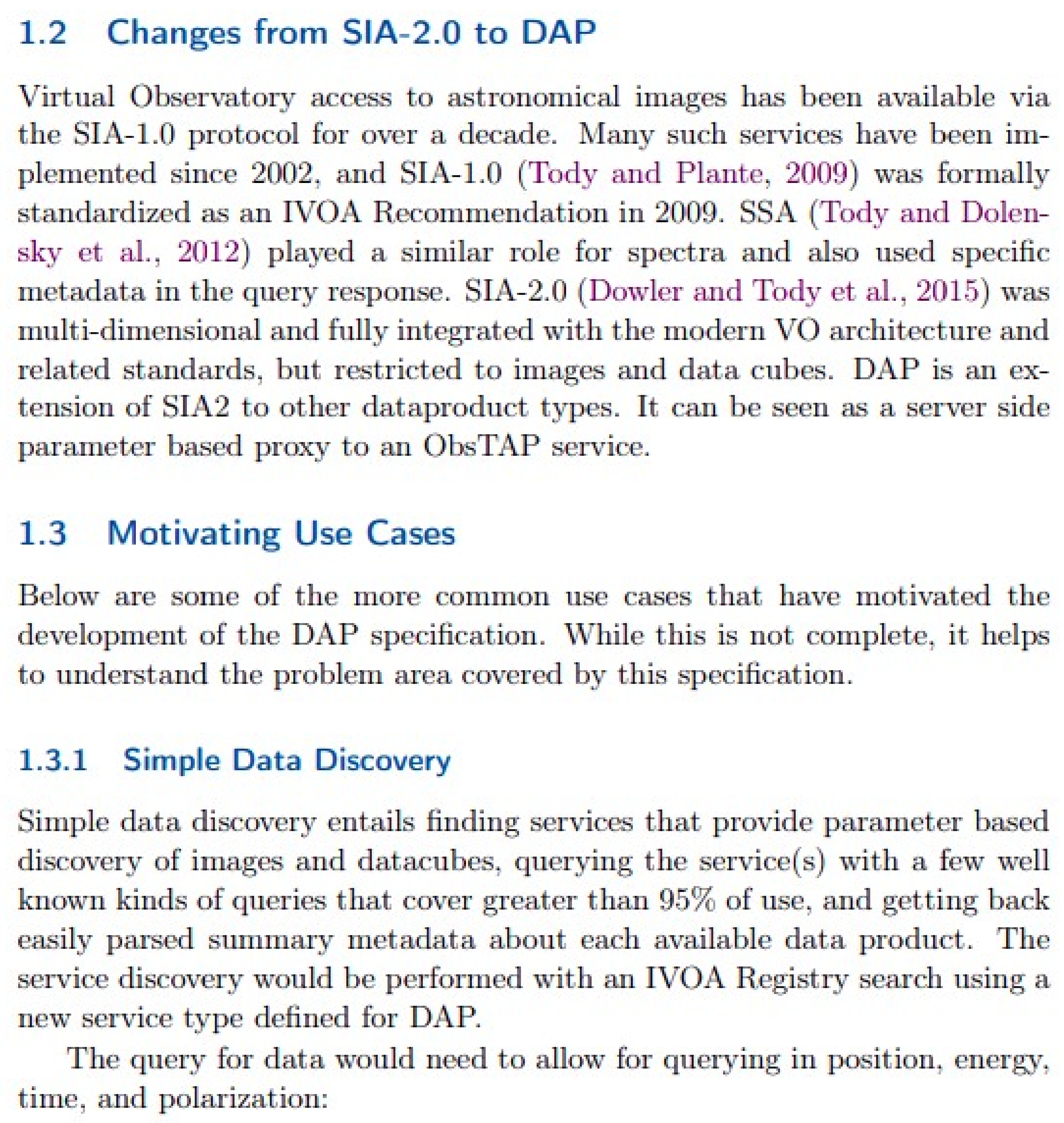
SODA documents

- Github : https://github.com/ivoa-std/SODA: see PR and Issues.
- Pre-draft :
  https://wiki.ivoa.net/internal/IVOA/SODA-1_0-Next/SODA-20240315.pdf
- Twiki : https://wiki.ivoa.net/twiki/bin/view/IVOA/SODA-1_0-Next
- One BIG pull request which will change for a while

SIA2 extending scope towards Data acces protocol

- The idea is we need parameter based protocols for Spectra and TimeSeries
- For spectra it could be something like an SSA2 consistent with ObsCore
- For TimeSeries it could be something like a TimeSeries discovery service
- Basically can be seen as a server-side paramater based front-end to ObsTAP
- Also useful to discover all sky datasets (HiPS or whatever) and catalogs

SIA2 extending scope towards Data acces protocol



1.2 Changes from SIA-2.0 to DAP

Virtual Observatory access to astronomical images has been available via the SIA-1.0 protocol for over a decade. Many such services have been implemented since 2002, and SIA-1.0 (Tody and Plante, 2009) was formally standardized as an IVOA Recommendation in 2009. SSA (Tody and Dolensky et al., 2012) played a similar role for spectra and also used specific metadata in the query response. SIA-2.0 (Dowler and Tody et al., 2015) was multi-dimensional and fully integrated with the modern VO architecture and related standards, but restricted to images and data cubes. DAP is an extension of SIA2 to other dataproduct types. It can be seen as a server side parameter based proxy to an ObsTAP service.

1.3 Motivating Use Cases

Below are some of the more common use cases that have motivated the development of the DAP specification. While this is not complete, it helps to understand the problem area covered by this specification.

1.3.1 Simple Data Discovery

Simple data discovery entails finding services that provide parameter based discovery of images and datacubes, querying the service(s) with a few well known kinds of queries that cover greater than 95% of use, and getting back easily parsed summary metadata about each available data product. The service discovery would be performed with an IVOA Registry search using a new service type defined for DAP.

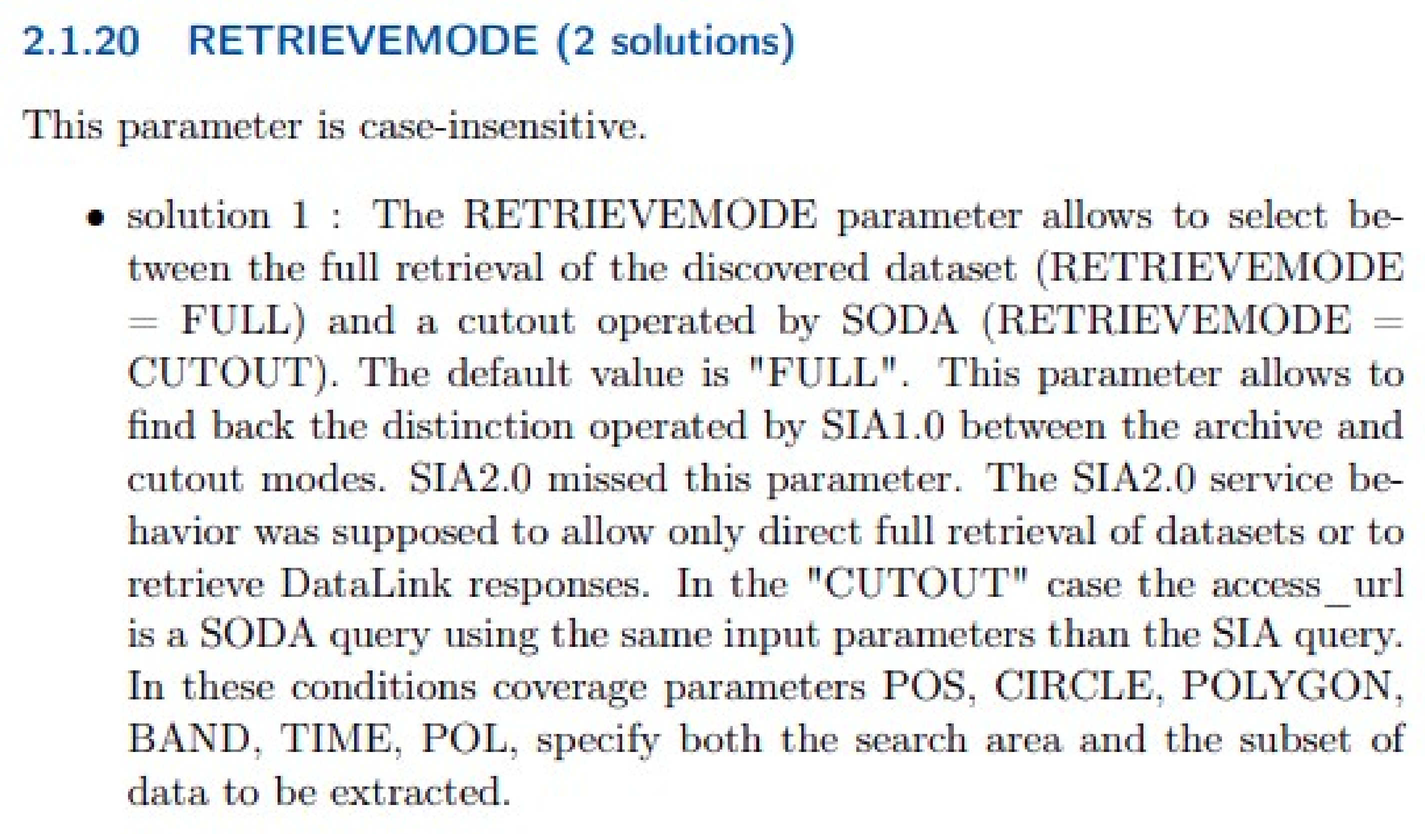
The query for data would need to allow for querying in position, energy, time, and polarization:

DAP to SsO Data Access

- From DAP (ex SIA2) : how do I discover cutouts and SODA-transformed data ?
- Standard way is multistep = ObsCore --(→ DataLink --)→

  SODA ServiceDescriptor menu → parameter choice → access to data
- Something more direct ? (« à la » SIA1?) → 2 solutions, but the nice thing is that DAP/SIA and SODA have the same parameters.
  - Either Include the SODA URL based on DAP parameters in the access_url
  - Or the client use the SODA service descriptor in the response to directly generate the query with DAP input parameters

DAP to SsO Data Access

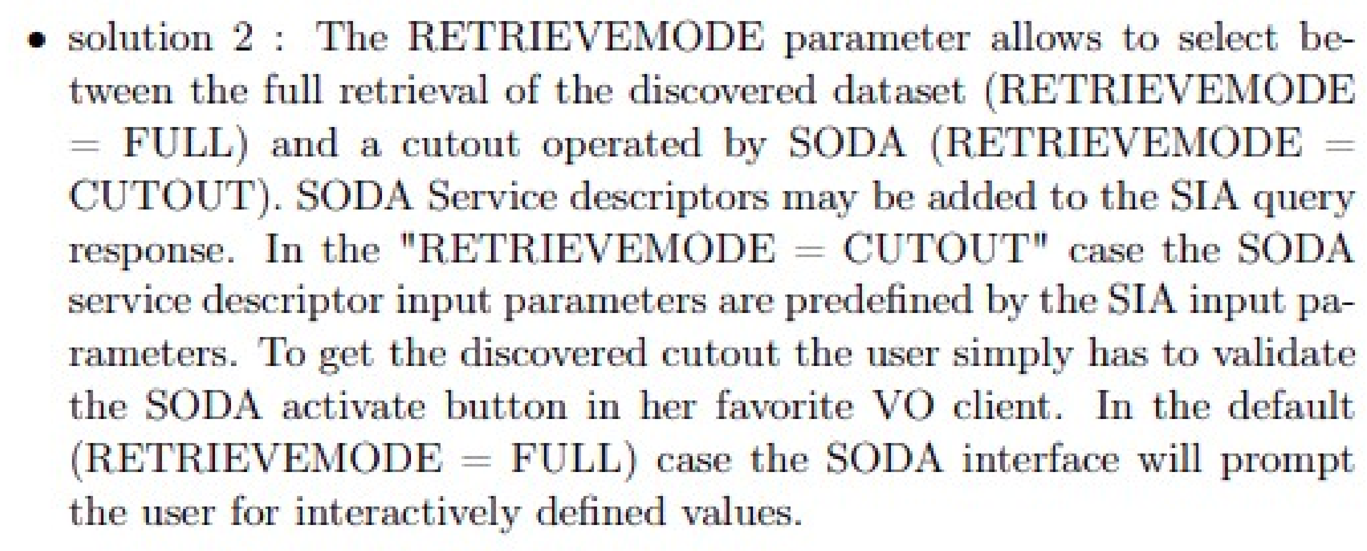


2.1.20 RETRIEVEMODE (2 solutions)

This parameter is case-insensitive.

- solution 1 : The RETRIEVEMODE parameter allows to select between the full retrieval of the discovered dataset (RETRIEVEMODE = FULL) and a cutout operated by SODA (RETRIEVEMODE = CUTOUT). The default value is "FULL". This parameter allows to find back the distinction operated by SIA1.0 between the archive and cutout modes. SIA2.0 missed this parameter. The SIA2.0 service behavior was supposed to allow only direct full retrieval of datasets or to retrieve DataLink responses. In the "CUTOUT" case the access_url is a SODA query using the same input parameters than the SIA query. In these conditions coverage parameters POS, CIRCLE, POLYGON, BAND, TIME, POL, specify both the search area and the subset of data to be extracted.

DAP to SsO Data Access



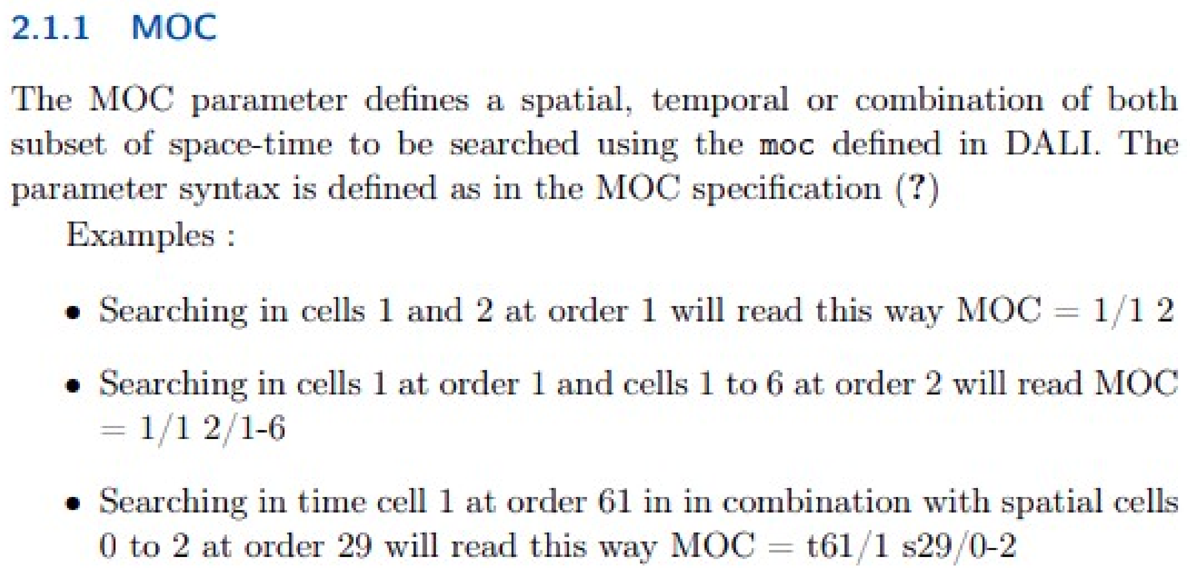
- solution 2 : The RETRIEVEMODE parameter allows to select between the full retrieval of the discovered dataset (RETRIEVEMODE = FULL) and a cutout operated by SODA (RETRIEVEMODE = CUTOUT). SODA Service descriptors may be added to the SIA query response. In the "RETRIEVEMODE = CUTOUT" case the SODA service descriptor input parameters are predefined by the SIA input parameters. To get the discovered cutout the user simply has to validate the SODA activate button in her favorite VO client. In the default (RETRIEVEMODE = FULL) case the SODA interface will prompt the user for interactively defined values.

DAP-SIAP2
Other changes

- Self description of service for list of supported STRING parameters
  - COLLECTION, FACILITY, INSTRUMENT, DPTYPES
  - Root URL query gives list of supported
- Include MOC parameter
  - I get a response from a service which provides a MOC.
  - I want to query available datasets in this MOC

DAP-SIAP2

Other changes



2.1.1 MOC

The MOC parameter defines a spatial, temporal or combination of both subset of space-time to be searched using the `moc` defined in DALI. The parameter syntax is defined as in the MOC specification (**?**)

Examples :

- Searching in cells 1 and 2 at order 1 will read this way MOC = 1/1 2
- Searching in cells 1 at order 1 and cells 1 to 6 at order 2 will read MOC = 1/1 2/1-6
- Searching in time cell 1 at order 61 in in combination with spatial cells 0 to 2 at order 29 will read this way MOC = t61/1 s29/0-2

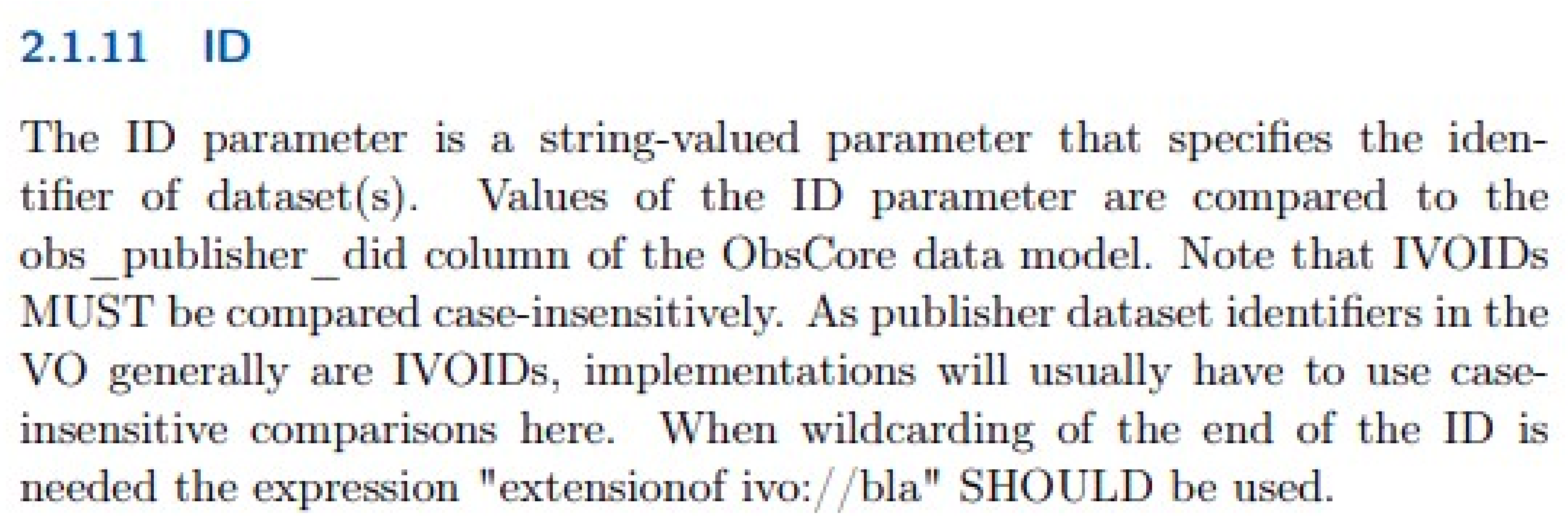
DAP-SIAP2
Other changes

- Wildcarding and case sensitivity
  - Case sensitiveness for FACILITY, COLLECTION, INSTRUMENT, TARGET – not for others
  - Wildcards could be useful for ivoa_ids

    → encoding issues.

    → A keyword is proposed instead
- Releasedate parameter

DAP-SIAP2
Other changes



2.1.11 ID

The ID parameter is a string-valued parameter that specifies the identifier of dataset(s). Values of the ID parameter are compared to the obs_publisher_did column of the ObsCore data model. Note that IVOIDs MUST be compared case-insensitively. As publisher dataset identifiers in the VO generally are IVOIDs, implementations will usually have to use case-insensitive comparisons here. When wildcarding of the end of the ID is needed the expression "extensionof ivo://bla" SHOULD be used.

DAP documents

- Github : https://github.com/ivoa-std/DAP : see unique PR and Issues.
- See also : https://github.com/ivoa-std/SIA for previous discussions
- Pre-draft :
  https://wiki.ivoa.net/internal/IVOA/SIAP-2_0-Next/DAP.pdf
- Twiki : https://wiki.ivoa.net/twiki/bin/view/IVOA/SIAP-2_0-Next
- One BIG pull request which will change for a while

DataLink

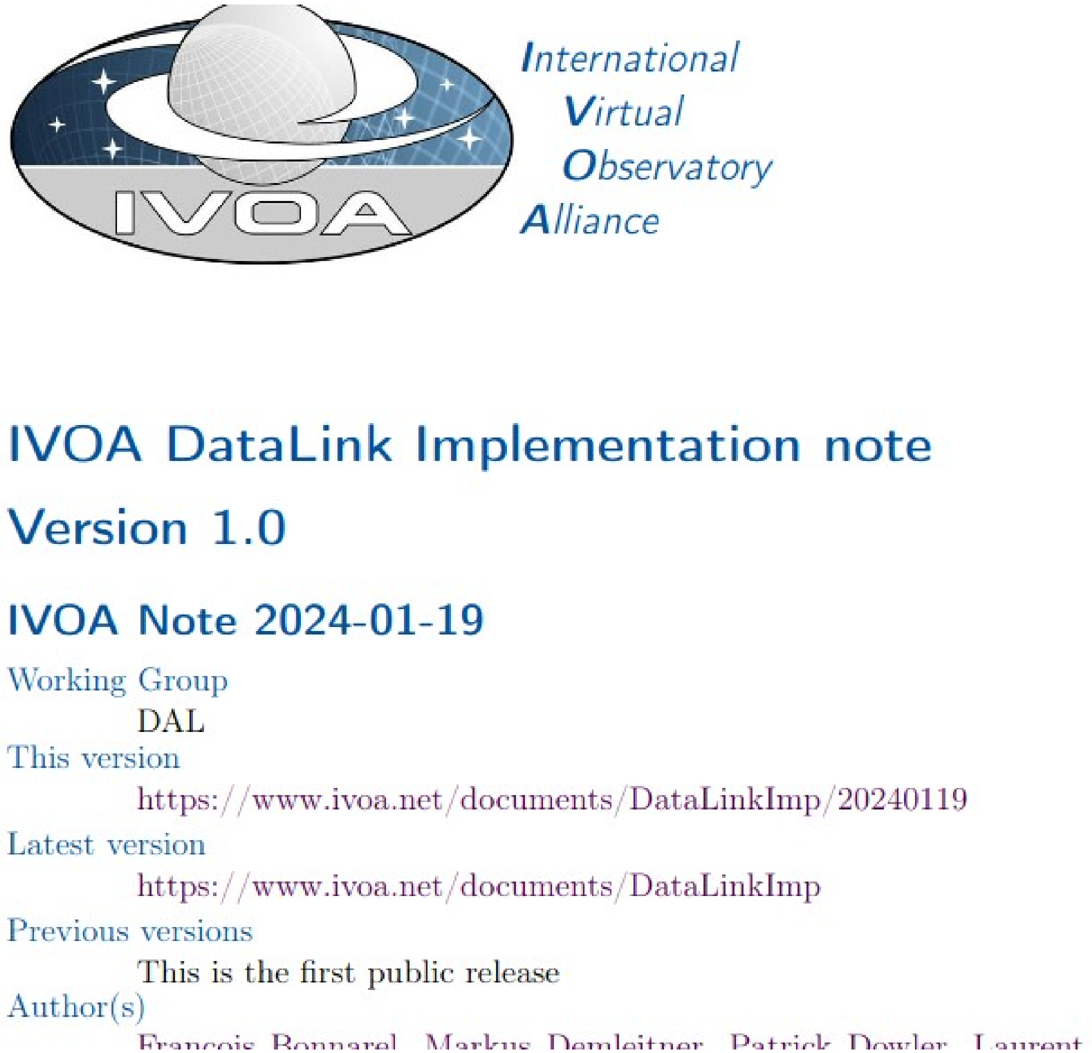
- Version 1.1 since december 2023
- Unsolved issues :

  service descriptors (https://github.com/ivoa-std/DataLink)

  - Can we template the access URL in the service descriptor (variability for anything different than « PARAM=Value » case)
  - Should this be solved and pushed to VOTable ?
  - How do we solve the optional/mandatory issue in the service input parameter description ?
  - Can we refer to PARAM instead of FIELDS in the service descriptor → erratum ?

DataLink implementation note

https://github.com/ivoa/DataLinkRecImplNote → push to an IVOA note



International Virtual Observatory Alliance

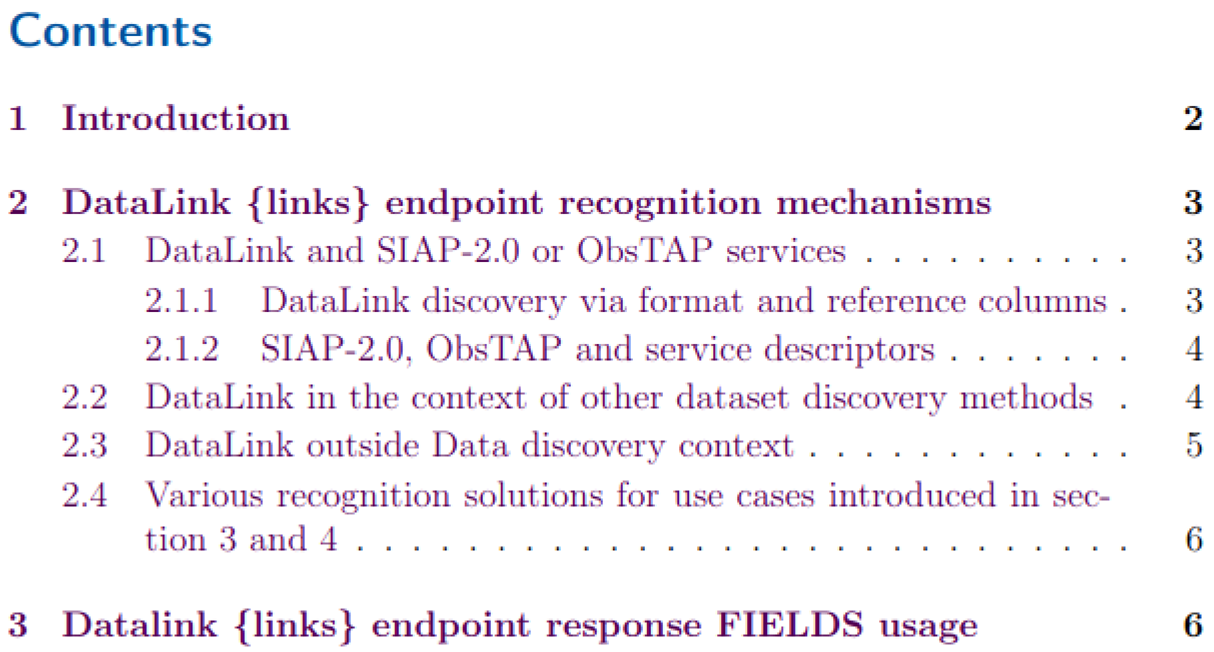
IVOA DataLink Implementation note

Version 1.0

IVOA Note 2024-01-19

Working Group
DAL
This version
https://www.ivoa.net/documents/DataLinkImp/20240119
Latest version
https://www.ivoa.net/documents/DataLinkImp
Previous versions
This is the first public release
Author(s)
Francois Bonnarel, Markus Demleitner, Patrick Dowler, Laurent

DataLink implementation note



Contents