



European Space Astronomy Centre  
P.O. Box 78  
28691 Villanueva de la Cañada  
Madrid  
Spain  
T +34 91 8131 100  
F +34 91 8131 139  
[www.esa.int](http://www.esa.int)

# DOCUMENT

## HSA Archive Inter-Operability (AIO) System User's Manual

**Prepared by** MHS  
**Reference**  
**Issue** 1  
**Revision** 0  
**Date of Issue** 19/09/2012  
**Status** Draft  
**Document Type** UM  
**Distribution**



# APPROVAL

<b>Title</b>	
<b>Issue 1</b>	<b>Revision 0</b>
<b>Author</b>	<b>Date</b> 19/09/2012
<b>Approved by</b>	<b>Date</b>

# CHANGE LOG

<b>Reason for change</b>	<b>Issue</b>	<b>Revision</b>	<b>Date</b>

# CHANGE RECORD

<b>Issue 1</b>		<b>Revision 0</b>	
<b>Reason for change</b>	<b>Date</b>	<b>Pages</b>	<b>Paragraph(s)</b>



**Table of contents:**

<b>1 INTRODUCTION.....</b>	<b>4</b>
Capabilities: .....	4
<b>2 METADATA REQUESTS.....</b>	<b>6</b>
2.1 Metadata Request Parameters.....	6
2.2 Use of QUERY parameter .....	9
<b>3 DATA REQUESTS .....</b>	<b>11</b>
3.1 Product requests .....	11
3.2 Product postcard requests .....	14
3.3 Login requests .....	15
<b>4 LIST OF HSA ENTITIES AND ATTRIBUTES.....</b>	<b>17</b>



# 1 INTRODUCTION

Within the ESA's<sup>1</sup> Science Operations Department, the SAT<sup>2</sup> in ESAC<sup>3</sup>, Spain, is responsible of developing and maintaining ESA Scientific Archives. For the Herschel Science Archive (HSA), the standard way to access to this ESA archive is through a powerful Java interface (<http://archives.esac.esa.int/hsa/ui/hui.jnlp>).

To allow the access to external applications and expert or normal users that need to retrieve data directly from the archive bypassing the Java User Interface, an integrated Archive Inter Operability system (AIO) has been developed for HSA. This AIO system can be accessed using standard HTTP requests.

This subsystem shares the core system of the standard HSA User Interface. It is normally called using scripts, application code or command line tools.

Capabilities:

- **Observation Retrieval:** It allows the user to obtain one single observation file.
- **Product Retrieval:** It allows the user to obtain the products associated to a determinate observation.
- **Observation / Product Search:** It allows the user to discover observations or products that fulfill a certain search criteria.
- **Multiple Observation / Products Search / Retrieval:** By combining the two previous points, the HAIO allows the download of all the observations or all products that fulfill a certain search criteria, by using pagination internally.

This interface is ideally created to fulfill the necessity of retrieving data massively. Through it you can retrieve either one or all the observation or product files that match a given criteria (HCSS\_URN, OD\_NUMBER, HCSS\_CLASS\_TYPE, etc).

---

<sup>1</sup> European Space Agency (<http://www.esa.int>)

<sup>2</sup> Science Archives Team (<http://archives.esac.esa.int>)

<sup>3</sup> European Space Astronomy Centre (<http://www.esa.int/esaMI/ESAC/index.html>)



HAIO accepts two kinds of requests:

- Metadata request: To be used when we want to retrieve the metadata details of observation file(s)
- Data request: To be used when we want to retrieve an observation file

Ideally, this interface should be used by a script, application code or command line tools. First, we retrieve a list of observation details through a metadata request, and then we issue a data request for each one of them in order to retrieve the file(s).

## 2 METADATA REQUESTS

Metadata requests are used to retrieve the metadata details of one or many observations. They are HTTP requests that the user will have to build using several parameters. Once this request is sent, the HSA backend will return the metadata that matches the given criteria in the requested format.

It is important to highlight that the default return type for all metadata requests is the VOTable<sup>4</sup> format.

### 2.1 Metadata Request Parameters

All metadata requests have the following HTTP prefix:

<http://archives.esac.esa.int/hsa/aio/jsp/metadata.jsp?>

After the HTTP prefix, the following parameters are used to build the request:

<b>Parameter Name(s)</b>	<b>Description</b>	<b>Mandatory?</b>	<b>Possible Values</b>
<b>SELECTED_FIELDS</b>	Single Entity ("Entity") or list of Attributes requested ("Entity"."Attribute" list separated by commas)	YES	Any HSA Entity or any combination of HAS attributes (See List of HSA entities and attributes)
<b>RETURN_TYPE</b>	Output format of the response. The default response format is VOTable. Other formats are JSON or CSV ( in next releases )	NOT	VOTABLE (default), CSV,JSON
<b>RESOURCE_CLASS</b>	Single Entity which defines the granularity level of the metadata query response	YES	Any HSA Entity (See List of HSA entities and attributes)
<b>PAGE_SIZE &amp; SIZE</b>	Number of records per page & page number	NOT, but highly recommended	Any integer value
<b>QUERY</b>	Complex restrictions of the request	NOT	Any Jython valid query expression (see below)
<b>ORDER_BY</b>	Criteria to sort metadata results	NOT	Any HSA attribute (See List of HSA entities and attributes)
<b>LAST_TRAC_VERSION</b>	'TRUE' means that the results must contain only latest versions ( grouped by HCSS_TRACK_ID and	NOT	TRUE

<sup>4</sup> International Virtual Observatory standard for the interchange of data (<http://www.ivoa.net/Documents/VOTable>)



	HCSS_TRACK_VERSION )		
<b>LIMIT</b>	If it is specified and the number of results are greater than the specified value, a VOTable containing an error is returned.	NOT	Integer value less than 25000 (HSA Hard Limit).
<b>SHOW_PROPRIETARY</b>	'TRUE' means that each VOTable record will contain a field that specifies proprietary rights access.	NOT	TRUE
<b>EXPIRED</b>	'TRUE' means that the results must contain only public observations whereas 'FALSE' means that results must contain only proprietary observations.	NOT	[ TRUE   FALSE ]

Table 1: Metadata request parameters

In addition to all these parameter names, simple attribute restrictions can be added as "Entity"."Attribute"="value" to the metadata request (for example INSTRUMENT.NAME=PACS). Once again, the possible values for this "Entity" and "Attribute" fields need to be compliant with the List of SSA entities and attributes.

Please note that for metadata requests:

- Dates are expressed in DD/MM/YYYY format and contained inside apostrophes (')
- HTTP prefix and parameters are separated by a question mark (?)
- Requests are build using parameter=value sentences separated by an ampersand (&)

Let's see the use of all the above through some examples:

- Get the greatest HCSS\_URN\_VERSION of an HCSS\_CLASS\_TYPE in VOTable format:

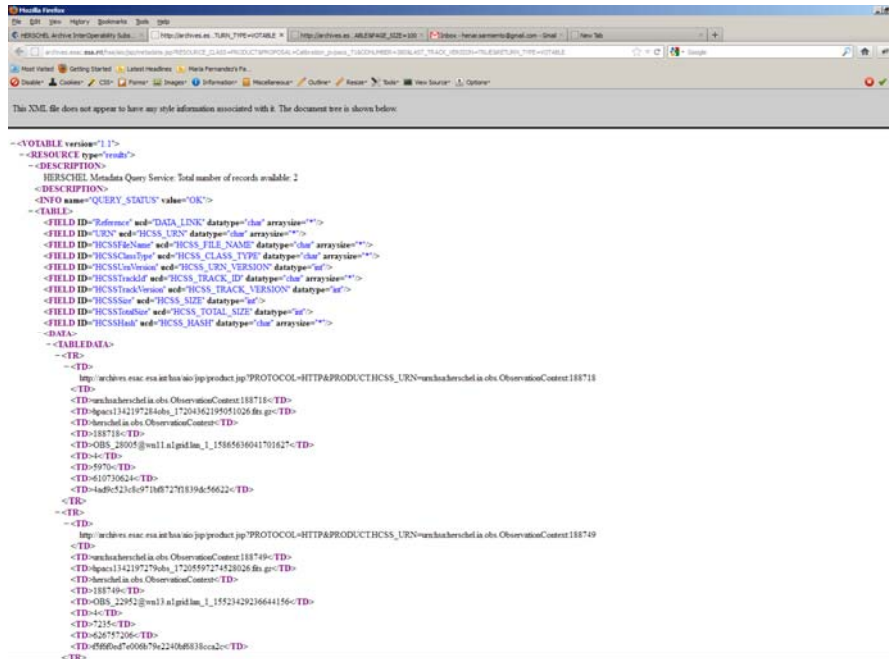
```
http://archives.esac.esa.int/hsa/aio/jsp/metadata.jsp?RESOURCE_CLASS=HCSS_CLASS_TYPE&HCSS_CLASS_TYPE=herschel.ia.dataset.Product&LAST_TRACK_VERSION=TRUE&RETURN_TYPE=VOTABLE&PAGE_SIZE=100
```

- All observations of operational day '380' and proposal 'Calibration\_pvpacs\_71' in VOTable format:

```
http://archives.esac.esa.int/hsa/aio/jsp/metadata.jsp?RESOURCE_CLASS=OBSERVATION&OD_NUMBER=380&PROPOSAL=Calibration_pvpacs_71&RETURN_TYPE=VOTABLE&PAGE_SIZE=100
```

- All products of operational day '380' and proposal 'Calibration\_pvpacs\_71' in VOTable format:

[http://archives.esac.esa.int/hsa/aio/jsp/metadata.jsp?RESOURCE\\_CLASS=PRODUCT&PROPOSAL=Calibration\\_pvpacs\\_71&ODNUMBER=380&LAST\\_TRACK\\_VERSION=TRUE&RETURN\\_TYPE=VOTABLE](http://archives.esac.esa.int/hsa/aio/jsp/metadata.jsp?RESOURCE_CLASS=PRODUCT&PROPOSAL=Calibration_pvpacs_71&ODNUMBER=380&LAST_TRACK_VERSION=TRUE&RETURN_TYPE=VOTABLE)



```

<?xml version="1.1"?>
<!--RESOURCE TYPE-->
<!--DESCRIPTION-->
<!--TABLE-->
<!--TABLE DATA-->
</TABLE DATA>
</TABLE-->
</VOTABLE-->

```

**Figure 1: Returned VOTable file**

- Get first 1000 public observations in VOTable format:

[http://archives.esac.esa.int/hsa/aio/jsp/metadata.jsp?RESOURCE\\_CLASS=OBSERVATION&EXPIRED=TRUE&PAGE\\_SIZE=1000&PAGE=1&RETURN\\_TYPE=VOTABLE](http://archives.esac.esa.int/hsa/aio/jsp/metadata.jsp?RESOURCE_CLASS=OBSERVATION&EXPIRED=TRUE&PAGE_SIZE=1000&PAGE=1&RETURN_TYPE=VOTABLE)

- Get observations 1001 to 2000 which proprietary status is 'private' in VOTable format:

[http://archives.esac.esa.int/hsa/aio/jsp/metadata.jsp?RESOURCE\\_CLASS=OBSERVATION&EXPIRED=FALSE&PAGE\\_SIZE=1000&PAGE=2&RETURN\\_TYPE=VOTABLE](http://archives.esac.esa.int/hsa/aio/jsp/metadata.jsp?RESOURCE_CLASS=OBSERVATION&EXPIRED=FALSE&PAGE_SIZE=1000&PAGE=2&RETURN_TYPE=VOTABLE)

When requesting in VOTABLE format, e.g: (SELECTED\_FIELDS=OBSERVATION), additional information will be provided for each result:

- A "Data Link" to retrieve the file
- A "Postcard Link" to retrieve its postcard (the default one if it does not exist)
- A "Details Link" to retrieve more precise information about this observation



## 2.2 Use of QUERY parameter

In the metadata request, the Keyword=value paradigm creates simple queries using the AND argument as logical operator. If a user wants to create a complex request, by using >, <, >=, <=, like (including SQL wildcards), and, or... operators he/she will have to use the QUERY parameter. The rest of the keyword parameters will be seen as other AND conditions.

The value of this query field should be URL encoded<sup>5</sup>.

Inside the value of the QUERY parameter, we allow any Jython<sup>6</sup> valid query expression which makes use of SSA attributes (see List of SSA entities and Attributes).

Jython query operations are slightly different to the SQL ones, as stated in this table:

<i>QUERY Operators</i>	<i>Type</i>	<i>Precedence</i>	<i>Symbol</i>
<b>PARENTHESIS</b>	Grouping	1	()
<b>EQUALS</b>	Relational	2	==
<b>NOT EQUALS</b>	Relational	2	!=
<b>GREATHER THAN</b>	Relational	2	>
<b>LOWER THAN</b>	Relational	2	<
<b>GREATER OR EQUALS THAN</b>	Relational	2	>=
<b>LOWER OR EQUALS THAN</b>	Relational	2	<=
<b>LIKE:</b> '%' => <b>MULTIPLE CHARACTERS</b> '_' => <b>ONE CHARACTER</b>	Relational	2	LIKE
<b>AND</b>	Logical	3	AND
<b>OR</b>	Logical	4	OR

*Table 2: QUERY parameter operators*

<sup>5</sup> URL encoding reference ([http://www.w3schools.com/tags/ref\\_urlencode.asp](http://www.w3schools.com/tags/ref_urlencode.asp))

<sup>6</sup> <http://www.jython.org/>



Let's see the use of all the above through some examples:

- **First 100 observations that fulfill the criteria:**

```
QUERY=( INSTRUMENT=='HIFI' AND
INSTRUMENT_OBS_MODE=='HifiSScanModeDBS' )
```

Equivalent URL:

```
http://archives.esac.esa.int/hsa/aio/jsp/metadata.jsp?RESOURCE_CLASS=OBSERVATIO
N&QUERY=%28INSTRUMENT==%27HIFI%27%20AND%20INSTRUMENT_OBS_MOD
E==%27HifiSScanModeDBS%27%29&PAGE_SIZE=100&PAGE=1&RETURN_TYPE=VOT
ABLE
```

- **Observations 101 to 200 that fulfill the criteria:**

```
QUERY=INSTRUMENT=='PACS' AND ( INSTRUMENT_OBS_MODE=='PacsPhoto' OR
INSTRUMENT_OBS_MODE=='PacsRangeSpec' )
```

Equivalent URL:

```
http://archives.esac.esa.int/hsa/aio/jsp/metadata.jsp?RESOURCE_CLASS=OBSERVATIO
N&SELECTED_FIELDS=OBSERVATION&QUERY=%28INSTRUMENT==%27PACS%27%
20AND%20%28INSTRUMENT_OBS_MODE==%27PacsPhoto%27%20OR%20INSTRUM
ENT_OBS_MODE==%27PacsRangeSpec%27%29%29&PAGE_SIZE=100&PAGE=2
```

- **Get observations that fulfil the criteria:**

```
QUERY=( AOR=='PACS-SDPhotSetup_na_nStd_orbitproWait_na_0114' AND
PROPOSAL =='Calibration_pvpacs_71'
```

Equivalent URL:

```
http://archives.esac.esa.int/hsa/aio/jsp/metadata.jsp?RESOURCE_CLASS=OBSERVATIO
N&QUERY=(AOR=='PACS-
SDPhotSetup_na_nStd_orbitproWait_na_0114'%20AND%20PROPOSAL=='Calibration_p
vpacs_71')&RETURN_TYPE=VOTABLE
```



### 3 DATA REQUESTS

Data requests are used to retrieve files. These files can be science files (fits files, Ascii files, ...) or postcards (jpg files). They are HTTP requests that the user will have to build using several parameters. Once this request is sent, the HSA backend will return one HTML link to download one specific file.

All the data requests have the following HTTP prefix:

<http://archives.esac.esa.int/hsa/aio/jsp/product.jsp?>

There are three types of data requests:

- Product requests
- Product postcard requests
- Latest available postcard requests

The kind of request you want to use is defined by the parameters RESOURCE\_CLASS and/or RETRIEVAL\_TYPE. Depending on its value, your request will be processed differently and different combination of parameters will be expected.

Please note that for data requests:

- HTTP prefix and parameters are separated by a question mark (?)
- Requests are build using parameter=value sentences separated by an ampersand (&)

#### 3.1 Product requests

One data request will provide, as response, one link to download one of the following options:

- **HSA Tarball:**

This object is a tarball file which contains the selected observation (or product level, auxiliary products, etc). The structure of this file is specified by the HSA-HCSS ICD.

- **Virtual Tar:**

This object is a TAR file that contains the specified URNs requested in an HAIO product query. The structure of this file is specified by the HSA-HCSS ICD.

<b>Parameter Name(s)</b>	<b>Description</b>	<b>Mandatory?</b>	<b>Possible Values</b>
<b>OBSERVATION_ID</b>	Unique identifier of the observation we want to retrieve	YES ( When RESOURCE_CLASS=OBSERVATION )	Any integer value
<b>URN</b>	Several URNs can be provided. If a single URN is provided, the requested product is served through the HTTP connection. If more that one URN is provided, a Virtual TAR is generated and served through the HTTP connection.	YES ( When RESOURCE_CLASS=PRODUCT )	Any value of Product HCSS_URN
<b>PROTOCOL</b>	If this parameter does not appear, an intermediate html page will be displayed in a human readable way.	NOT	HTTP
<b>METADATA</b>	If this parameter does not appear, the already ingested product is returned. If this parameter appears and its value is 'true', a FITS file containing product metadata only is returned.	NOT	TRUE
<b>REQUESTFILE_XML</b>	If this parameter is present, <b>urn</b> and <b>metadata</b> parameters are discarded.	NOT	XML ingestion request file name.
<b>OBSID</b>	Product Metadata attribute	NOT	Any integer value
<b>PRODUCT_LEVEL</b>	Used with <b>OBSID</b> or <b>OBSERVATION_ID</b> parameter.  If <b>PRODUCT_LEVEL</b> is ALL, an HSA Tarball wit the complete observation is generated and served through HTTP connection.  If <b>PRODUCT_LEVEL</b> is equal to any of Auxiliary, Calibration, etc, An HSA Tarball with the selected products, is generated and served	NOT	The following is a list with the possible values: <ul style="list-style-type: none"> <li>• All</li> <li>• Auxiliary</li> <li>• Calibration</li> <li>• Level0</li> <li>• Level0_5</li> <li>• Level1</li> <li>• Level2</li> <li>• Level2_5</li> </ul>

	through HTTP connection.		
<b>COMPRESSION</b>	If this parameter is present, a Tar compressed in 'tar.gz' format is generated and served through HTTP the connection	NOT	TARGZ

*Table 3: Product request parameters*

### Examples:

- File of observation with identifier '1342231052':

```
http://archives.esac.esa.int/hsa/aio/jsp/product.jsp?PROTOCOL=HTTP&OBSERVATION_ID=1342231052
```

- Retrieve a product which URN is 'urn:hsa:herschel.spire.ia.dataset.NhkTimeline:1045695':

```
http://archives.esac.esa.int/hsa/aio/jsp/product.jsp?PROTOCOL=HTTP&URN=urn:hsa:herschel.spire.ia.dataset.NhkTimeline:1045695&PROTOCOL=HTTP
```

- Download a FITS file containing only product metadata.

```
http://archives.esac.esa.int/hsa/aio/jsp/product.jsp?PROTOCOL=HTTP&URN=urn:hsa:herschel.spire.ia.dataset.NhkTimeline:1045695&METADATA=TRUE&PROTOCOL=HTTP
```

- Retrieve a product which metadata field obsid is '1342231052':

```
http://archives.esac.esa.int/hsa/aio/jsp/product.jsp?PROTOCOL=HTTP&OBSID=1342231052
```

- Download all Auxiliary products that belongs to an observation which observation\_id is '1342231052':

```
http://archives.esac.esa.int/hsa/aio/jsp/product.jsp?PROTOCOL=HTTP&OBSID=1342231052&PRODUCT_LEVEL=Auxiliary&PROTOCOL=HTTP
```

- Download all Auxiliary products that belongs to an observation which observation\_id is '1342231052' in a Tarball compressed:

```
http://archives.esac.esa.int/hsa/aio/jsp/product.jsp?PROTOCOL=HTTP&OBSID=1342231052&PRODUCT_LEVEL=Auxiliary&COMPRESSION=TARGZ&PROTOCOL=HTTP
```

- Download a XML ingestion request file name.

```
http://archives.esac.esa.int/hsa/aio/jsp/product.jsp?requestfile_xml=C-herschel.spire.ia.cal.SpireCal-23509.xml&PROTOCOL=HTTP
```

### 3.2 Product postcard requests

A Product postcard request will provide, as response, a JPG file. This file contains the product postcard associated to an observation and should be ingested into HSA. The file name shall follow some rules specified in the HSA-HCSS ICD:

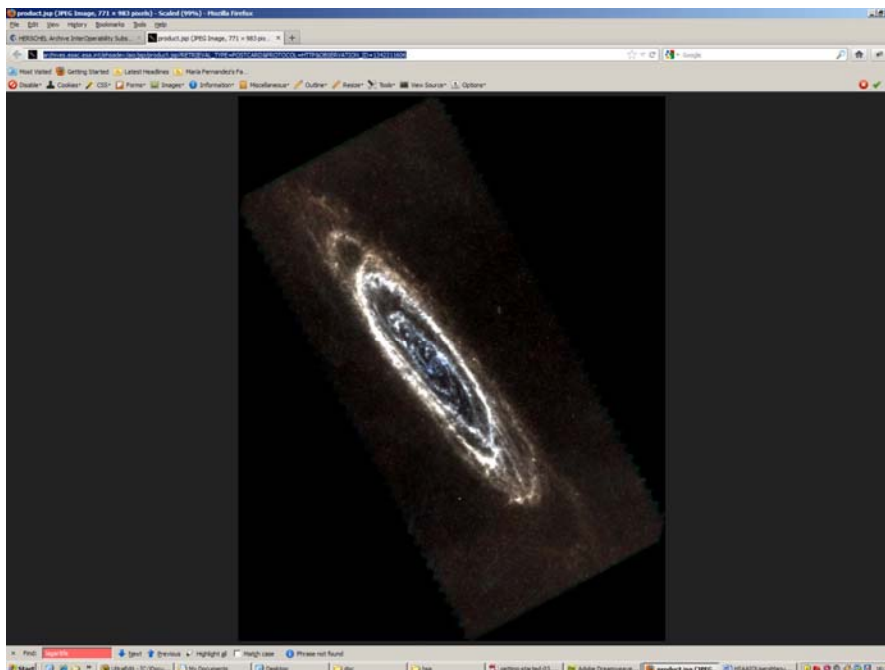
<i>Parameter Name(s)</i>	<i>Description</i>	<i>Mandatory?</i>	<i>Possible Values</i>
<b>OBSERVATION_ID</b>	Unique identifier of the observation we want to retrieve	YES	Any integer value
<b>RETRIEVAL_TYPE</b>	Type of retrieval. In this case 'POSTCARD'	YES	POSTCARD

*Table 4: Postcard request parameters*

Examples:

- Postcard of observation with identifier '1342211606':

```
http://archives.esac.esa.int/ehsadev/aio/jsp/product.jsp?RETRIEVAL_TYPE=POSTCARD&PROTOCOL=HTTP&OBSERVATION_ID=1342211606
```



*Figure 2: HTTP Product Postcard Request*



### 3.3 Login requests

A client can do login into the HAIO by invoking the url described in the example below.

The server will return a HTTP cookie that should be used in product retrieval in order to access proprietary data. In case of login problem, an HTTP error ( login error ) will be raised ( See HSA-HCSS ICD for further information ).

All the data requests have the following HTTP prefix:

`http://archives.esac.esa.int/hsa/aio/jsp/login.jsp?`

Please note that:

- Login Request syntax and parameters are separated by a question mark (?)
- Login Requests are build using parameter=value sentences separated by an ampersand (&)

<b>Parameter Name(s)</b>	<b>Description</b>	<b>Mandatory?</b>	<b>Possible Values</b>
<b>USERNAME</b>	User Name	YES	Any alphanumerical combination.
<b>PASSWORD</b>	User Password	YES	Any alphanumerical combination but not less than four characters.
<b>PROTOCOL</b>	If this parameter does not appear, an intermediate html page will be displayed in a human readable way.	NOT	HTTP

**Table 5: Latest available postcard parameters**



## Examples:

- Example of login request:

```
http://archives.esac.esa.int/hsa/aio/jsp/login.jsp?username=sathsa&password=[password]
```

- Example of login request with the parameter 'PROTOCOL' specified:

```
http://archives.esac.esa.int/hsa/aio/jsp/login.jsp?username=sathsa&password=[password]&protocol=HTTP
```





## 4 LIST OF HSA ENTITIES AND ATTRIBUTES

Below you can find the list of all the HSA entities and attributes. They can be used in the metadata request, related to certain parameters:

- Entity names can appear as **SELECTED\_FIELDS** (only one at a time)
- Entity names can appear as **RESOURCE\_CLASS** (only one at a time)
- Attribute names can appear as **SELECTED\_FIELDS** (one or a combination of them)
- Attribute names can appear as **ORDER\_BY** (one or a combination of them)
- Simple attribute restrictions can be added as "Attribute Name"="value"

<b>Entity Name</b>	<b>Attribute Name</b>	<b>Alias Name</b>	<b>Attribute Type</b>
<b>OBSERVATION</b>	OBSERVATION.OBSERVATION_ID	OBSERVATION_ID	INTEGER
	OBSERVATION.URN_VERSION	URN_VERSION	INTEGER
	OBSERVATION.SPG_ID	SPG_ID	STRING
	OBSERVATION.RA	RA	DOUBLE
	OBSERVATION.DEC	DEC	DOUBLE
	OBSERVATION.PA	PA	DOUBLE
	OBSERVATION.DURATION	DURATION	DOUBLE
	OBSERVATION.PROP_END	PROP_END	DATE
	OBSERVATION.PROP_END	EXPIRATION	DATE
	OBSERVATION.START_TIME	START_TIME	DATE
	OBSERVATION.END_TIME	END_TIME	DATE
	OBSERVATION.OBSERVER	OBSERVER	STRING
	OBSERVATION.OD_NUMBER	OD_NUMBER	INTEGER
	OBSERVATION.AOR	AOR	STRING
<b>PRODUCT</b>	PRODUCT.HCSS_URN	URN	STRING
	PRODUCT.HCSS_URN	HCSS_URN	STRING
	PRODUCT.HCSS_FILE_NAME	FILENAME	STRING
	PRODUCT.HCSS_FILE_NAME	HCSS_FILE_NAME	STRING
	PRODUCT.HCSS_TRACK_VERSION	VERSION	INTEGER
	PRODUCT.HCSS_TRACK_VERSION	HCSS_TRACK_VERSION	INTEGER
	PRODUCT.HCSS_URN_VERSION	HCSS_URN_VERSION	INTEGER
	PRODUCT.HCSS_TRACK_ID	HCSS_TRACK_ID	STRING
	PRODUCT.HCSS_SIZE	HCSS_SIZE	INTEGER
	PRODUCT.HCSS_TOTAL_SIZE	HCSS_TOTAL_SIZE	INTEGER
PRODUCT.HCSS_HASH	HCSS_HASH	STRING	
PRODUCT.HCSS_CLASS_PATH	HCSS_CLASS_PATH	STRING	
<b>OBSERVING_MODE</b>	OBSERVING_MODE.OBSERVING_MODE_NAME	OBSERVING_MODE_NAME	STRING
<b>INSTRUMENT</b>	INSTRUMENT.NAME	INSTRUMENT_NAME	STRING
<b>OBSERVATION_PRODUCT</b>	OBSERVATION_PRODUCT.DISTRIBUTION_PATH	DISTRIBUTION_PATH	STRING
<b>PRODUCT_METADATA</b>	PRODUCT_METADATA.ACMSMODE	acmsMode	STRING
	PRODUCT_METADATA.ACTIVESTRID	activeStrId	STRING
	PRODUCT_METADATA.AORLABEL	aorLabel	STRING
	PRODUCT_METADATA.AOT	aot	STRING
	PRODUCT_METADATA.APERTURE	aperture	STRING



	PRODUCT_METADATA.APID	apid	STRING
	PRODUCT_METADATA.ARRAYNAME	arrayName	STRING
	PRODUCT_METADATA.AUTHOR	author	STRING
	PRODUCT_METADATA.AVERAGING	averaging	STRING
	PRODUCT_METADATA.BAND	band	STRING
	PRODUCT_METADATA.BASELINEMODEL	baselineModel	STRING
	PRODUCT_METADATA.BASELINEPARAMS	baselineParams	STRING
	PRODUCT_METADATA.BBCOUNT	bbCount	INTEGER
	PRODUCT_METADATA.BBID	bbid	INTEGER
	PRODUCT_METADATA.BBTYPENAME	bbType	INTEGER
	PRODUCT_METADATA.BBTYPENAME	bbTypeName	STRING
	PRODUCT_METADATA.BIASMODE	biasMode	STRING
	PRODUCT_METADATA.BIASFREQ	biasFreq	DOUBLE
	PRODUCT_METADATA.BIASVOLTAGE	biasVoltage	DOUBLE
	PRODUCT_METADATA.BITPOS	bitPos	INTEGER
	PRODUCT_METADATA.CALFILEID	calFileId	STRING
	PRODUCT_METADATA.CALFILEVERSION	calFileVersion	INTEGER
	PRODUCT_METADATA.CALTHRESHOLD	calThreshold	DOUBLE
	PRODUCT_METADATA.CAMERA	camera	STRING
	PRODUCT_METADATA.CAMERAMODEL	cameraModel	STRING
	PRODUCT_METADATA.CD1_1	cd1_1	DOUBLE
	PRODUCT_METADATA.CD1_2	cd1_2	DOUBLE
	PRODUCT_METADATA.CD1_3	cd1_3	DOUBLE
	PRODUCT_METADATA.CD2_1	cd2_1	DOUBLE
	PRODUCT_METADATA.CD2_2	cd2_2	DOUBLE
	PRODUCT_METADATA.CD2_3	cd2_3	DOUBLE
	PRODUCT_METADATA.CD3_1	cd3_1	DOUBLE
	PRODUCT_METADATA.CD3_2	cd3_2	DOUBLE
	PRODUCT_METADATA.CD3_3	cd3_3	DOUBLE
	PRODUCT_METADATA.PC1_1	pc1_1	DOUBLE
	PRODUCT_METADATA.PC1_2	pc1_2	DOUBLE
	PRODUCT_METADATA.PC1_3	pc1_3	DOUBLE
	PRODUCT_METADATA.PC2_1	pc2_1	DOUBLE
	PRODUCT_METADATA.PC2_2	pc2_2	DOUBLE
	PRODUCT_METADATA.PC2_3	pc2_3	DOUBLE
	PRODUCT_METADATA.PC3_1	pc3_1	DOUBLE
	PRODUCT_METADATA.PC3_2	pc3_2	DOUBLE
	PRODUCT_METADATA.PC3_3	pc3_3	DOUBLE
	PRODUCT_METADATA.CDELT1	cddelt1	DOUBLE
	PRODUCT_METADATA.CDELT2	cddelt2	DOUBLE
	PRODUCT_METADATA.CHANGELOG	changelog	STRING
	PRODUCT_METADATA.CHOPPERPLATEAU	chopperPlateau	INTEGER
	PRODUCT_METADATA.CONSTVELFLAG	constVelFlag	BOOLEAN
	PRODUCT_METADATA.CONVERSIONFACTOR	conversionFactor	DOUBLE
	PRODUCT_METADATA.CREATIONDATE	creationDate	DATE
	PRODUCT_METADATA.CREATOR	creator	STRING
	PRODUCT_METADATA.CROTA2	crota2	DOUBLE
	PRODUCT_METADATA.CRPIX1	crpix1	DOUBLE
	PRODUCT_METADATA.CRPIX2	crpix2	DOUBLE



	PRODUCT_METADATA.CRVAL1	crval1	DOUBLE
	PRODUCT_METADATA.CRVAL2	crval2	DOUBLE
	PRODUCT_METADATA.CTYPE1	ctype1	STRING
	PRODUCT_METADATA.CTYPE2	ctype2	STRING
	PRODUCT_METADATA.CUSMODE	cusMode	STRING
	PRODUCT_METADATA.DATAANALYST	dataAnalyst	STRING
	PRODUCT_METADATA.DEC	dec	DOUBLE
	PRODUCT_METADATA.DECNOMINAL	decNominal	DOUBLE
	PRODUCT_METADATA.DECOBJECT	decObject	DOUBLE
	PRODUCT_METADATA.DELTAPIX	deltaPix	DOUBLE
	PRODUCT_METADATA.DESCRPTION	description	STRING
	PRODUCT_METADATA.ENDDATE	endDate	DATE
	PRODUCT_METADATA.ENDWAVELENGTH	endWavelength	DOUBLE
	PRODUCT_METADATA.EPOCH	epoch	DOUBLE
	PRODUCT_METADATA.EQUINOX	equinox	DOUBLE
	PRODUCT_METADATA.ERROR	error	DOUBLE
	PRODUCT_METADATA.FILTER	filter	STRING
	PRODUCT_METADATA.FINETIME	fineTime	LONG
	PRODUCT_METADATA.FILENAME	fileName	STRING
	PRODUCT_METADATA.FORMATVERSION	formatVersion	STRING
	PRODUCT_METADATA.GYROPROPQUALIDX	gyroPropQualIdx	DOUBLE
	PRODUCT_METADATA.INSTRUMENT	instrument	STRING
	PRODUCT_METADATA.INSTMODE	instMode	STRING
	PRODUCT_METADATA.INTERPMETHOD	interpMethod	STRING
	PRODUCT_METADATA.JIGGLEID	jiggleId	LONG
	PRODUCT_METADATA.KEYWAVELENGTH	keyWavelength	DOUBLE
	PRODUCT_METADATA.LEVEL	level	STRING
	PRODUCT_METADATA.MAXWAVELENGTH	maxWavelength	DOUBLE
	PRODUCT_METADATA.MINWAVELENGTH	minWavelength	DOUBLE
	PRODUCT_METADATA.MISSIONCONFIG	missionConfig	STRING
	PRODUCT_METADATA.MODELNAME	modelName	STRING
	PRODUCT_METADATA.NAIFID	naifId	STRING
	PRODUCT_METADATA.NODCYCLENUM	nodCycleNum	LONG
	PRODUCT_METADATA.NUMCHOPCYC	numChopCyc	LONG
	PRODUCT_METADATA.NUMHIFISAA	numHifiSaa	LONG
	PRODUCT_METADATA.NUMJIGGLEPOS	numJigglePos	LONG
	PRODUCT_METADATA.NUMNODCYC	numNodCyc	LONG
	PRODUCT_METADATA.NUMPACSSAA	numPacsSaa	LONG
	PRODUCT_METADATA.NUMRASTERCOL	numRasterCol	LONG
	PRODUCT_METADATA.NUMRASTERLINES	numRasterLines	LONG
	PRODUCT_METADATA.NUMSCANLINES	numScanLines	LONG
	PRODUCT_METADATA.NUMSPECTRA	numSpectra	LONG
	PRODUCT_METADATA.NUMSPIRESAA	numSpireSaa	LONG
	PRODUCT_METADATA.OBSID	obsId	INTEGER
	PRODUCT_METADATA.OBJECT	object	STRING
	PRODUCT_METADATA.OBJECTTYPE	objectType	STRING
	PRODUCT_METADATA.OBSERVER	observer	STRING
	PRODUCT_METADATA.OBSMODE	obsMode	STRING
	PRODUCT_METADATA.ODNUMBER	odNumber	LONG



PRODUCT_METADATA.OFFPOSFLAG	offPosFlag	BOOLEAN
PRODUCT_METADATA.ONTARGETFLAG	onTargetFlag	BOOLEAN
PRODUCT_METADATA.ORIGIN	origin	STRING
PRODUCT_METADATA.OUTOFFIELDFLAG	outOfFieldFlag	BOOLEAN
PRODUCT_METADATA.PIXELROW	pixelRow	INTEGER
PRODUCT_METADATA.POINTINGMODE	pointingMode	STRING
PRODUCT_METADATA.POSANGLE	posAngle	DOUBLE
PRODUCT_METADATA.PROPOSAL	proposal	STRING
PRODUCT_METADATA.QUALITYFLAG	qualityFlag	INTEGER
PRODUCT_METADATA.RA	ra	DOUBLE
PRODUCT_METADATA.RADESYS	raDeSys	INTEGER
PRODUCT_METADATA.RAERR	raErr	DOUBLE
PRODUCT_METADATA.RANOMINAL	raNominal	DOUBLE
PRODUCT_METADATA.RAOBJECT	raObject	DOUBLE
PRODUCT_METADATA.RASTERCOLUMNNUM	rasterColumnNum	SHORT
PRODUCT_METADATA.RASTERLINENUM	rasterLineNum	SHORT
PRODUCT_METADATA.READOUTS	readouts	STRING
PRODUCT_METADATA.REFERENCES	references	STRING
PRODUCT_METADATA.REFPIXEL	refPixel	LONG
PRODUCT_METADATA.ROLL	roll	DOUBLE
PRODUCT_METADATA.SAA	saa	DOUBLE
PRODUCT_METADATA.SATURATION	saturacion	DOUBLE
PRODUCT_METADATA.SATVALUESSIGNED	satValuesSigned	INTEGER
PRODUCT_METADATA.SATVALUESUNSIGNED	satValuesUnsigned	INTEGER
PRODUCT_METADATA.SCANLINENUM	scanLineNum	SHORT
PRODUCT_METADATA.SCOPE	scope	STRING
PRODUCT_METADATA.SEDVERSION	sedVersion	STRING
PRODUCT_METADATA.SERENDIPIYFLAG	serendipityFlag	BOOLEAN
PRODUCT_METADATA.SIAMID	siamId	STRING
PRODUCT_METADATA.SKYRESOLUTION	skyResolution	DOUBLE
PRODUCT_METADATA.SLEWFLAG	slewFlag	STRING
PRODUCT_METADATA.SOURCE	source	STRING
PRODUCT_METADATA.SOURCEDETECTOR	sourceDetector	STRING
PRODUCT_METADATA.SOURCESMEC	sourceSmec	STRING
PRODUCT_METADATA.SPECNUM	specNum	LONG
PRODUCT_METADATA.SPECTRALRESOLUTION	spectralResolution	DOUBLE
PRODUCT_METADATA.STARTDATE	startDate	DATE
PRODUCT_METADATA.STARTWAVELENGTH	startWavelength	DOUBLE
PRODUCT_METADATA.STATUS	status	STRING
PRODUCT_METADATA.STRINTERLACINGSTATUS	strInterlacingStatus	BOOLEAN
PRODUCT_METADATA.STRQUALIDX	strQualIdx	DOUBLE
PRODUCT_METADATA.SUBINSTRUMENTID	subinstrumentId	STRING
PRODUCT_METADATA.SUBSYSTEM	subsystem	STRING
PRODUCT_METADATA.TELESCOPE	telescope	STRING
PRODUCT_METADATA.TEMPERATURE	temperature	DOUBLE
PRODUCT_METADATA.TYPE	type	STRING
PRODUCT_METADATA.VARIABILITY	variability	STRING
PRODUCT_METADATA.VERSION	version	STRING
PRODUCT_METADATA.WAVELENGTHID	wavelengthId	INTEGER



	PRODUCT_METADATA.WCSREFERENCE	wcsReference	STRING
	PRODUCT_METADATA.WCSTYPE	wcsType	STRING
	PRODUCT_METADATA.WHEELPOS	wheelPos	INTEGER
	PRODUCT_METADATA.ZEROPOINTOFFSET	zeroPointOffset	DOUBLE
	PRODUCT_METADATA.PLWBIA SAMPL	plwBiasAmpl	DOUBLE
	PRODUCT_METADATA.PSWBIA SAMPL	pswBiasAmpl	DOUBLE
	PRODUCT_METADATA.PMWBIA SAMPL	pmwBiasAmpl	DOUBLE
	PRODUCT_METADATA.PTCBIA SAMPL	ptcBiasAmpl	DOUBLE
	PRODUCT_METADATA.SSWBIA SAMPL	sswBiasAmpl	STRING
	PRODUCT_METADATA.SLWBIA SAMPL	slwBiasAmpl	STRING