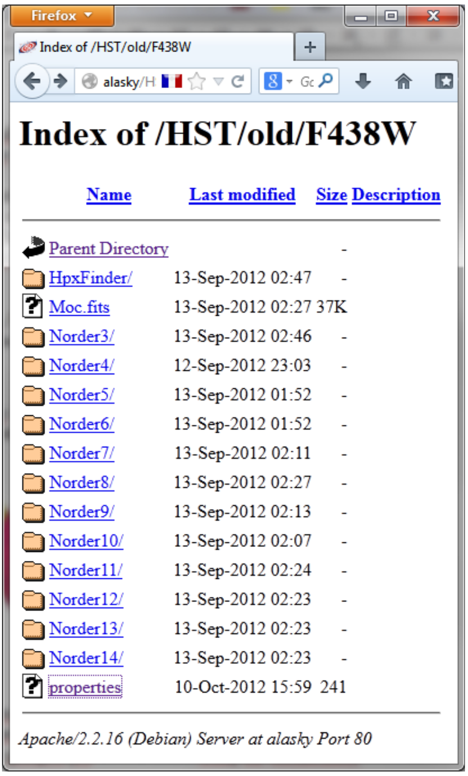
HiPS – Hierarchical Progressive Survey

This document describes the method for storing a HiPS (Hierarchical progressive Survey) as a collection of files. This method was developed in the frame of Aladin allsky study based on HEALPix sky tessellation. It is not an IVOA standard and it is still in progress, but it can already help the development of alternative HiPS generators/clients/browsers.

Author : Pierre Fernique

Date : 22th June 2015 – 18:35

HiPS version : 1.3



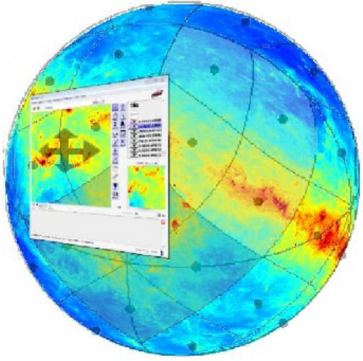
Index of /HST/old/F438W

Name	Last modified	Size	Description
Parent Directory		-	
HpxFinder/	13-Sep-2012 02:47	-	
Moc.fits	13-Sep-2012 02:27	37K	
Norder3/	13-Sep-2012 02:46	-	
Norder4/	12-Sep-2012 23:03	-	
Norder5/	13-Sep-2012 01:52	-	
Norder6/	13-Sep-2012 01:52	-	
Norder7/	13-Sep-2012 02:11	-	
Norder8/	13-Sep-2012 02:27	-	
Norder9/	13-Sep-2012 02:13	-	
Norder10/	13-Sep-2012 02:07	-	
Norder11/	13-Sep-2012 02:24	-	
Norder12/	13-Sep-2012 02:23	-	
Norder13/	13-Sep-2012 02:23	-	
Norder14/	13-Sep-2012 02:23	-	
properties	10-Oct-2012 15:59	241	

Apache/2.2.16 (Debian) Server at alasky Port 80

1 Description

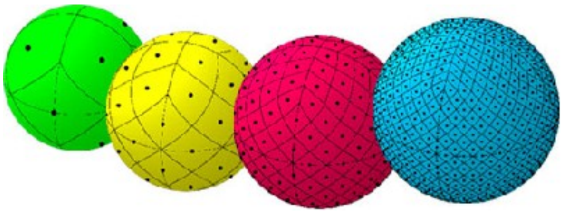
A “Hierarchical Progressive Survey”, also called HiPS, allows a dedicated client/browser tool to access and display a survey progressively, based on the principle that “*the more you zoom in on a particular area the more details show up”*.
This method is based on HEALPix sky tessellation. It consists to “map” a set of astronomical images at various resolutions into a collection of HEALPix “tiles” hierarchically organized as directories and files. A HiPS is generally published via a classical HTTP server without requiring any specific CGI. A HiPS browser, displaying a such survey, is able to access directly to the tiles corresponding to the current user view, without requiring any data base access for localizing the data, it just needs an HEALPix lib for computing the list of HEALPix tile index covering the current user view, and by this way, it can build the proper URLs and get the required tiles.



The HiPS has been created for image surveys with the idea to keep as much as possible the scientific information (astrometry and photometry). Various extensions has been done for other data types : astronomical source catalogs, collection of cubes, … Other extensions are still in progress.

2 HEALPix principle

HEALPix tessellation consists to divide the celestial sphere in 12 diamonds at the first level, and subdivide each of them in 4 sub-diamonds recursively. For instance, at the order 3 the sky is divided in 768 diamonds. HEALPix documentation and libraries are available on the net.



3 HiPS directory structure

A HiPS stores all its tiles as a collection of directories. The structure of these directories follows the simple hierarchy: order → tiles, by using respectively the prefix "Norder" for orders, and "Npix" for tiles. To avoid too large directories, the tiles are grouped by 10 000 items, using the subdirectory prefix "Dir".

A HiPS follows this directory structure:

```
…
Norder3
    Dir0
        Npix0
…
Norder6
    Dir0
        Npix0
        Npix1
        …
    Dir10000
        Npix10000
…
```

4 Tile format

A HiPS tile contains the data (pixels, catalog sources...) located on its sky area (one tile correspond to one HEALPix diamond, fully localized on the sky)
The tile format depends on the survey data type: FITS, JPEG, PNG for image or cube surveys, TSV, VOTable for catalogs, JSON for meta information...

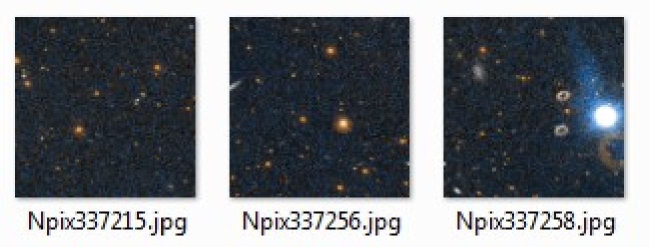
4.1 Image tile format:

An image tile is a bitmap, containing NxN pixels, usually N=512. These pixels represents the HEALPix sub-orders of the tile (9th sub-order for N=512, ie 2^9).

Example. The pixel (0,0) of the tile Npix32 of the Order3 correspond to the original pixel located on the sky at the HEALPix position Npix(32*512*512) for the Order(3+9).

Note: The method for evaluating each pixel value depends on the HiPS creation algorithm. It can be the nearest pixel, a bilinear interpolation or any other algorithm that the HiPS generator tool implements. Also, in case of overlapping original images, an average method, weighted or not, can be applied.
=> Aladin/HipsGen uses by default a bilinear interpolation, and the value average in case of overlapping images weighted by the distance to the nearest original image border (fading effect)

Inside a tile, the pixels are arranged in the natural spacial ordering (do not follow the NESTED HEALPix scheme). By this way, an image tile can be displayed – in a first approximation - as a classical bitmap image.



An image tile can be stored as FITS keeping the original full survey dynamic, in JPEG for a good compression, on in PNG for managing transparency. The tile file extension corresponds to the data type (respectively .fits, .jpg, .png). The fits tiles can be compressed (gzip, zip... in live by the http server or precompressed on the server) but keep the .fits extension. The client is responsible to test the magic code corresponding to a possible compression.

A HiPS can be delivered simultaneously in various formats (ex: FITS and JPEG). In this case each tile type are provided, differing only by their file extension.

Image tile drawing method: to draw a region, a HiPS image browser should implement this algorithm:

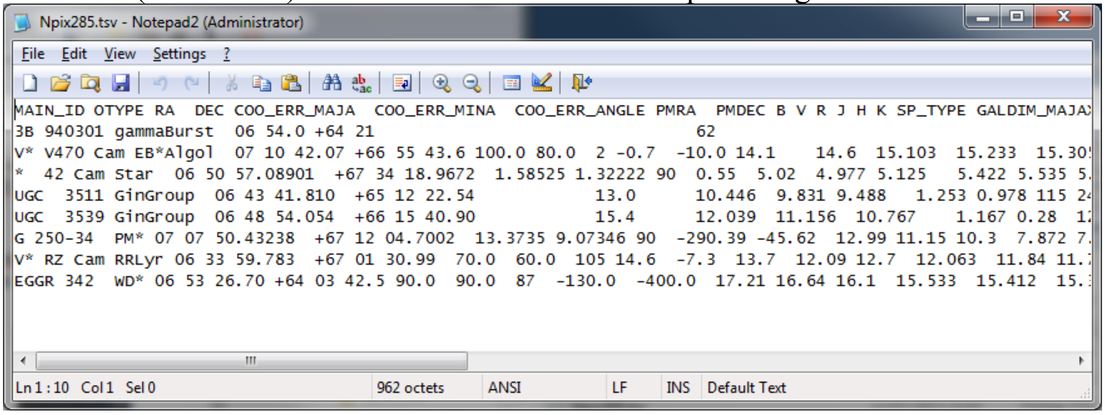
1. Compute the relevant HiPS order (one screen pixel should be cover by one tile pixel)
2. Compute the list of HEALPix tile index covering the region (HEALPix lib function)
3. Retrieve the corresponding tiles (locally or via the net)
4. Draw each tile on its sky projected location (based on the 4 corner sky localizations => affine transformation), or based on more control points by subdividing each tile in sub HEALPix tiles)

4.2 Catalog tile format:

A catalog tile is a list of sources (ie. a list of catalog rows). These sources are located on the sky in the corresponding HEALPix diamond.

If the number of sources in a tile overtakes a certain limit, the other sources are stored in the 4 sub-tiles from the next order, and thus, recursively. This limit is not necessary a constant depending of the algorithm for generating the HiPS cat (local density, level, …).

A catalog tile is stored in TSV (Tab Separated Value) with the extension “.tsv”. All tiles must have the same column (same order). The first line is a header line providing the column names.



Npix285.tsv - Notepad2 (Administrator)
File Edit View Settings ?

```
MAIN_ID OTYPE RA  DEC COO_ERR_MAJA  COO_ERR_MINA  COO_ERR_ANGLE PMRA  PMDEC B V R J H K SP_TYPE GALDIM_MAJA
3B 940301 gammaBurst  06 54.0 +64 21                                     62
V* V470 Cam EB*Algol  07 10 42.07 +66 55 43.6 100.0 80.0  2 -0.7  -10.0 14.1    14.6  15.103  15.233  15.30
*  42 Cam Star  06 50 57.08901  +67 34 18.9672  1.58525 1.32222 90  0.55  5.02  4.977 5.125   5.422 5.535 5.
UGC  3511 GinGroup  06 43 41.810  +65 12 22.54            13.0      10.446  9.831 9.488   1.253 0.978 115 2
UGC  3539 GinGroup  06 48 54.054  +66 15 40.90            15.4      12.039  11.156  10.767    1.167 0.28  1
G 250-34  PM* 07 07 50.43238  +67 12 04.7002  13.3735 9.07346 90  -290.39 -45.62  12.99 11.15 10.3  7.872 7.
V* RZ Cam RRLyr 06 33 59.783  +67 01 30.99  70.0  60.0  105 14.6  -7.3  13.7  12.09 12.7  12.063  11.84 11.
EGGR 342  WD* 06 53 26.70 +64 03 42.5 90.0  90.0  87  -130.0  -400.0  17.21 16.64 16.1  15.533  15.412  15.
```

Catalog tile drawing method: to display the sources in a region, a HiPS cat browser should implement this algorithm:

1. From the lower order:
    1. Compute the list of HEALPix tile index covering the region (HEALPix lib function)
    2. Retrieve the corresponding tiles (locally or via the net)
    3. Draw the sources inside the tiles (position on the sky and/or associated measurement information)
    4. Continue with the next order if required (depending of the algorithm: number of drawn sources, order number compared to the view size...)

4.3 Cube tile format:

For cubes, a cube tile is divided in frame tiles (separated files). The number of frame tiles corresponds to the depth of the original cubes (necessarily homogeneous for all the cube collection). Each of frame tile will have a suffix "_n" where "n" is the depth index. The tile name without suffix corresponds to the first frame (depth = 0). By this way HiPS client not HiPS cube compatible will be able to display at least the first frame of the HiPS cube like a simple HiPS image. As HiPS image, the cube frame tiles can be in FITS, in JPEG and/or PNG.

5 Allsky packaging

Two tricks has to be taken into account at low resolutions (order0 to order3) :

- the distortion with a basic 4 corners bilinear drawing method is high
- generally, all tiles are required for drawing the user view (whole sky at this level)

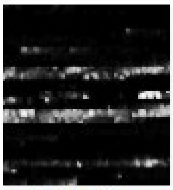
For facilitate the browsing algorithm, these improvements **MAY** be implemented for the first 4 lower orders:

Omission: the low orders: Order0 (12 tiles), Order1 (48 tiles), Order2 (192 tiles) **MAY** be simply omitted, meaning that the survey is not provided at these resolutions.

Allsky shortcuts: at low orders (0 to 3), the contain of tiles **MAY** be "concatenated" in a unique file called "Allsky" located in the NorderX corresponding directory. Notice that the associated regular tiles do not be removed (for instance for basic clients).

HiPS image allsky:

- The "Allsky" file is build as an array of tiles, stored side by side in the natural reading order. The width of this array is the square root of the number of the tiles of the order. For instance, the width of this array at order 3 is 27 ((int)sqrt(768)). To avoid too large Allsky file, the resolution of each tile can be reduced (typically 64x64 pixels rather than 512x512).



Allsky.jpg

Client display algorithm adaptation: the client/browser drawing a HiPS at very low resolution (whole sky area) should look first if a Allsky.xxx file is existing in the NorderX directory, and if yes, it should load it, split it as a very low resolution tile collection, and draw them. A client can decide to also use the regular individual NorderX tiles as an intermediate zoom level before drawing the Norder(X+1) tiles.

HiPS catalog allsky:

- All sources at the order may be stored in a unique file called "Allsky".

  Client display algorithm adaptation: the client/browser who must display a HiPS cat at very low resolution (whole sky area) should look first if a Allsky.xxx is existing in the NorderX, and if yes, it should load it and display the corresponding sources. In this case, the regular NorderX tiles must be ignored for avoiding doubloons.

HiPS cube allsky :

- *The "Allsky" file adapted to HiPS cubes follows the same rule that regular cube frame tiles: it is divided in "Allsky_nnn" files for providing each frame separately. Without suffix, the*

"Allsky" file corresponds to the frame 0.

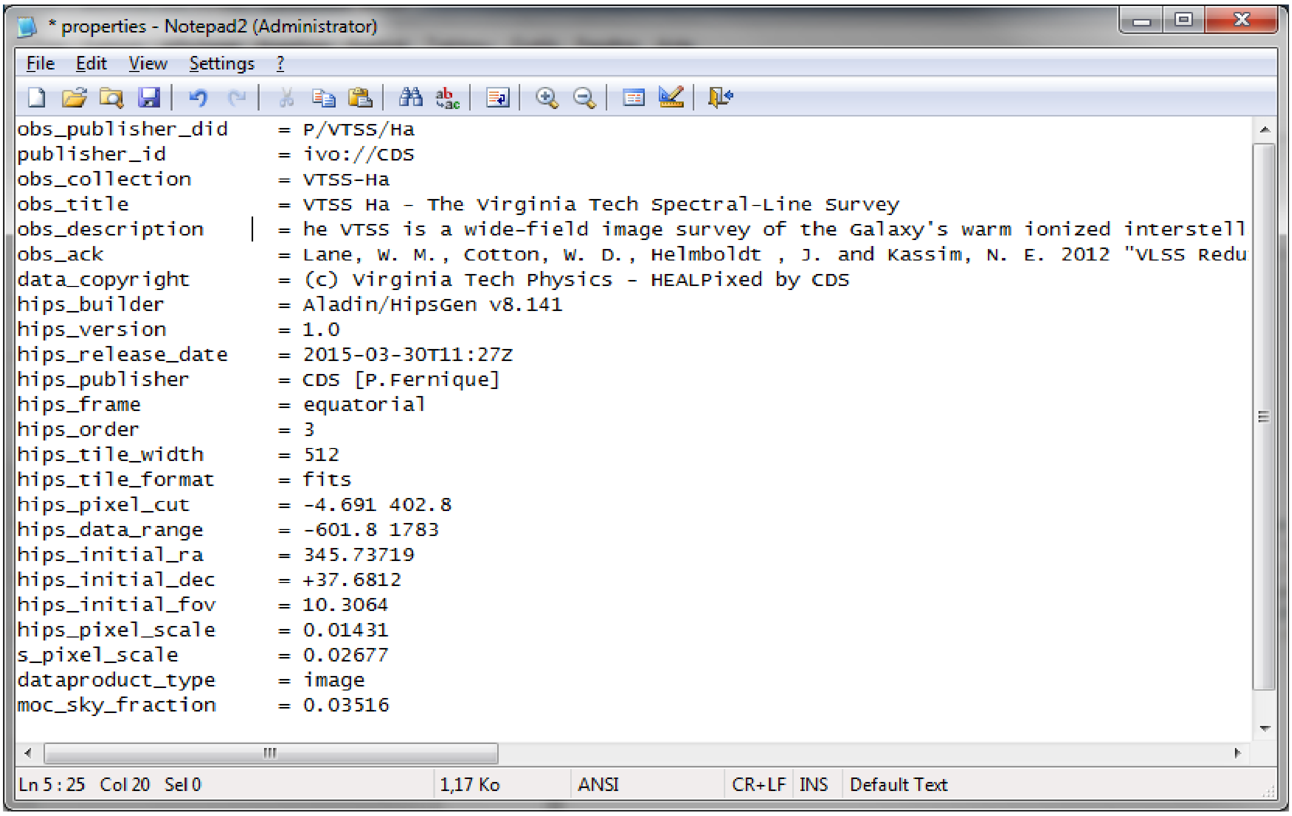
6 Meta data

Four complementary methods **SHOULD** or **MAY** be used for specifying survey meta data: 1- in the "properties" file, 2 - in the "metadata" file, 3 - in the MOC file, 4 – in the "index.html" file.

6.1 Properties

A file called "properties", stored in the HiPS root **SHOULD** be provided for generic meta information such as copyright, creation date, pixel range... The syntax is ASCII, one line per property, following the syntax KEY = value.

Note : Contrary to java property file, the values need **not** backslash mechanism for ':', '/' characters.



```
obs_publisher_did     = P/VTSS/Ha
publisher_id          = ivo://CDS
obs_collection        = VTSS-Ha
obs_title             = VTSS Ha - The Virginia Tech Spectral-Line Survey
obs_description      | = he VTSS is a wide-field image survey of the Galaxy's warm ionized interstell
obs_ack               = Lane, W. M., Cotton, W. D., Helmboldt , J. and Kassim, N. E. 2012 "VLSS Redu
data_copyright        = (c) Virginia Tech Physics - HEALPixed by CDS
hips_builder          = Aladin/HipsGen v8.141
hips_version          = 1.0
hips_release_date     = 2015-03-30T11:27Z
hips_publisher        = CDS [P.Fernique]
hips_frame            = equatorial
hips_order            = 3
hips_tile_width       = 512
hips_tile_format      = fits
hips_pixel_cut        = -4.691 402.8
hips_data_range       = -601.8 1783
hips_initial_ra       = 345.73719
hips_initial_dec      = +37.6812
hips_initial_fov      = 10.3064
hips_pixel_scale      = 0.01431
s_pixel_scale         = 0.02677
dataproduct_type      = image
moc_sky_fraction      = 0.03516
```

The table below describes the list of available keyword. The HiPS version 1.3 uses the **ObsCore IVOA** vocabulary and syntax.

This list is not limited and new keywords **MAY** be added by HiPS generator tool if required. Some keywords **MAY** be repeated for specifying several values (cf. *)

publisher_did	Unique ID of the data set - Format : IVORN - Ex : ivo://CDS/P/2MASS/J
publisher_id	Unique ID of the HiPS publisher – Format : IVORN - Ex : ivo://CDS
obs_collection	Short name of original data set – Format : one word – Ex : 2MASS
obs_title	Data set title – Format : free text, one line – Ex : HST F110W observations
obs_description	Data set description – Format : free text, one paragraph
obs_ack	Acknowledgment mention.
prov_progenitor (*)	Provenance of the original data – Format : free text
bib_reference (*)	Bibliographic reference

bib_reference_url (*)	URL to bibliographic reference
obs_copyright	Copyright mention – Format : free text
obs_copyright_url	URL to a copyright mention
obs_regime (*)	General wavelength – Format: radio, IR, optical, UV, EUV, X-ray, Gamma-ray
data_ucd (*)	UCD describing data contents
hips_version	Number of HiPS version – Format : 1.3
hips_builder	Name and version of the tool used for building the HiPS – Format : free text
hips_publisher	Institute or person who built the HiPS – Format : free text – Ex : CDS (T.Boch)
hips_creation_date	HiPS first creation date - Format : ISO 8601 => YYYY-mm-ddTHH:MMZ
hips_release_date	Last HiPS update date - Format : ISO 8601 => YYYY-mm-ddTHH:MMZ
hips_status	Mirror HiPS status – Format : master, mirror, partial, private – Default : master
hips_estsize	HiPS size estimation – Format : integer – Unit : KB
hips_frame	Coordinate frame reference – Format : equatorial, ecliptic, galactic – No default => must be provided
hips_order	Highest HiPS order – Format : integer
hips_order_min	Lowest HiPS order – Format : integer
hips_tile_width	Tiles width in pixels – Format : integer – Default : 512
hips_tile_format	List of available tile formats, blank separated. The first one is the default suggested to the client – Format : jpeg, png, fits
hips_pixel_cut	Default pixel cut range (physical values) – Format : min max – Ex : 10 300
hips_data_range	Default data range (physical values) – Format : min max – Ex : -18.5 510.5
hips_sampling	Sampling applied for the HiPS generation – Format : none, nearest, bilinear
hips_overlay	Pixel composition method applied on the image overlay region during HiPS generation – Format : add, mean, first, border_fading, custom
hips_skyval	Sky background subtraction method applied during HiPS generation – Format : none, hips_estimation, fits_keyword
hips_pixel_bitpix	Fits tile BITPIX code – Format : -64, -32, 8, 16, 32, 64 (FITS convention)
data_pixel_bitpix	Original data BITPIX code - Format : -64, -32, 8, 16, 32, 64 (FITS convention)
dataproduct_type	Type of data – Format : image, cube, catalog, meta
dataproduct_subtype	Subtype of data – Format : color, live
hips_progenitor_url	URL to an associated meta progenitor HiPS
hips_cat_nrows	Number of rows of the HiPS catalog – Format: integer
hips_cube_depth	Depth of the HiPS cube – Format : integer
hips_cube_firstframe	Initial first index frame to display for a HiPS cube – Format : integer – Default : 0
data_cube_crpix3	Coef for computing physical canal value (see FITS doc) – Format : real
data_cube_crval3	Coef for computing physical canal value (see FITS doc) – Format : real
data_cube_cdelt3	Coef for computing physical canal value (see FITS doc) – Format : real
data_cube_bunit3	Third axis unit (see FITS doc) – Format : string
hips_rgb_red	Original HiPS red component parameters of a HiPS color composition – Format: HipsID [cutLow cutMiddle cutHight TransferFunction]
hips_rgb_green	Original HiPS green component parameters of a HiPS color composition – Format: HipsID [cutLow cutMiddle cutHight TransferFunction]
hips_reg_blue	Original HiPS blue component parameters of a HiPS color composition – Format: HipsID [cutLow cutMiddle cutHight TransferFunction]
hips_initial_ra	Default RA display position – Format : real (ICRS frame) – Unit : degrees

hips_initial_dec	Default DEC display position – Format : real (ICRS frame) – Unit : degrees
hips_initial_fov	Default display size – Format : real – Unit : degrees
hips_pixel_scale	HiPS pixel angular resolution at the highest order – Format : real – Unit : degrees
s_pixel_scale	Best pixel angular resolution of the original images – Format : real – Unit : degrees
t_min	Start time of the observations – Format : real – Unit: MJD (*)
t_max	Stop time of the observations – Format : real – Unit: MJD (*)
em_min	Start in spectral coordinates – Format: real – Unit: meters
em_max	Stop in spectral coordinates – Format: real – Unit: meters
client_category	'/' separated keywords suggesting a display hierarchy to the client – Ex : Image/InfraRed
client_sort_key	Sort key suggesting a display order to the client inside a « client_category » – Format : free text – Sort : alphanumeric
moc_sky_fraction	Fraction of the sky covers by the MOC associated to the HiPS – Format : real between 0 and 1

(*) MJD can be easily computed from Unix time: MJD = (Unixtime / 86400) +40587

Previous HiPS versions used another vocabulary. Its usage is deprecated. The table below provided the correspondences with the new version :

id	**publisher_did**
copyrigth	**obs_copyright**
copyrightUrl	**obs_copyright_url**
HiPSBuilder	**hips_builder**
publisher	**hips_publisher**
firstProcessingDate	**hips_creation_date**
processingDate	**hips_release_date**
coordsys	**hips_frame**
maxOrder	**hips_order**
minOrder	**hips_order_min**
nside	**hips_tile_width**
format	**hips_tile_format**
pixelCut	**hips_pixel_cut**
pixelRange	**hips_data_range**
cubeDepth	**hips_cube_depth**
firstFrame	**hips_cube_firstframe**
target	See **hips_initial_dec, hips_initial_ra**
targetRadius	**hips_initial_fov**
category	**client_category**
isColored	See **dataproduct_subtype**
isCatalog	See **dataproduct_type**
isCube	See **dataproduct_type**
liveUpdate	See **dataproduct_subtype**
red	See **hips_rgb_red**

green	See **hips_rgb_green**
blue	See **hips_rgb_blue**

6.2 Metadata

A file called «metadata.xxx», stored in the HiPS root **MAY** be provided for specific meta information such as column description, FITS keywords... They depends on the survey data types:

HiPS image: the meta data are stored as a FITS header (FITS convention) in "metadata.fits" file for providing generic FITS header keywords. This file can be reduced to a FITS header or even a regular FITS file from the original survey containing the FITS header used for global metadata information – in the second case the image array is simply ignored by the client.

HiPS catalog: the meta data are stored in "metadata.xml" as a fully defined VOTable (column descriptions, units, ucds,...). By this way all column information associated to the source tiles can be provided via the "metadata" file. In this case the internal header in each individual tile is ignored by the client (VOTable metadata richer than the default simple header line)

HiPS cube: same as Hips image.

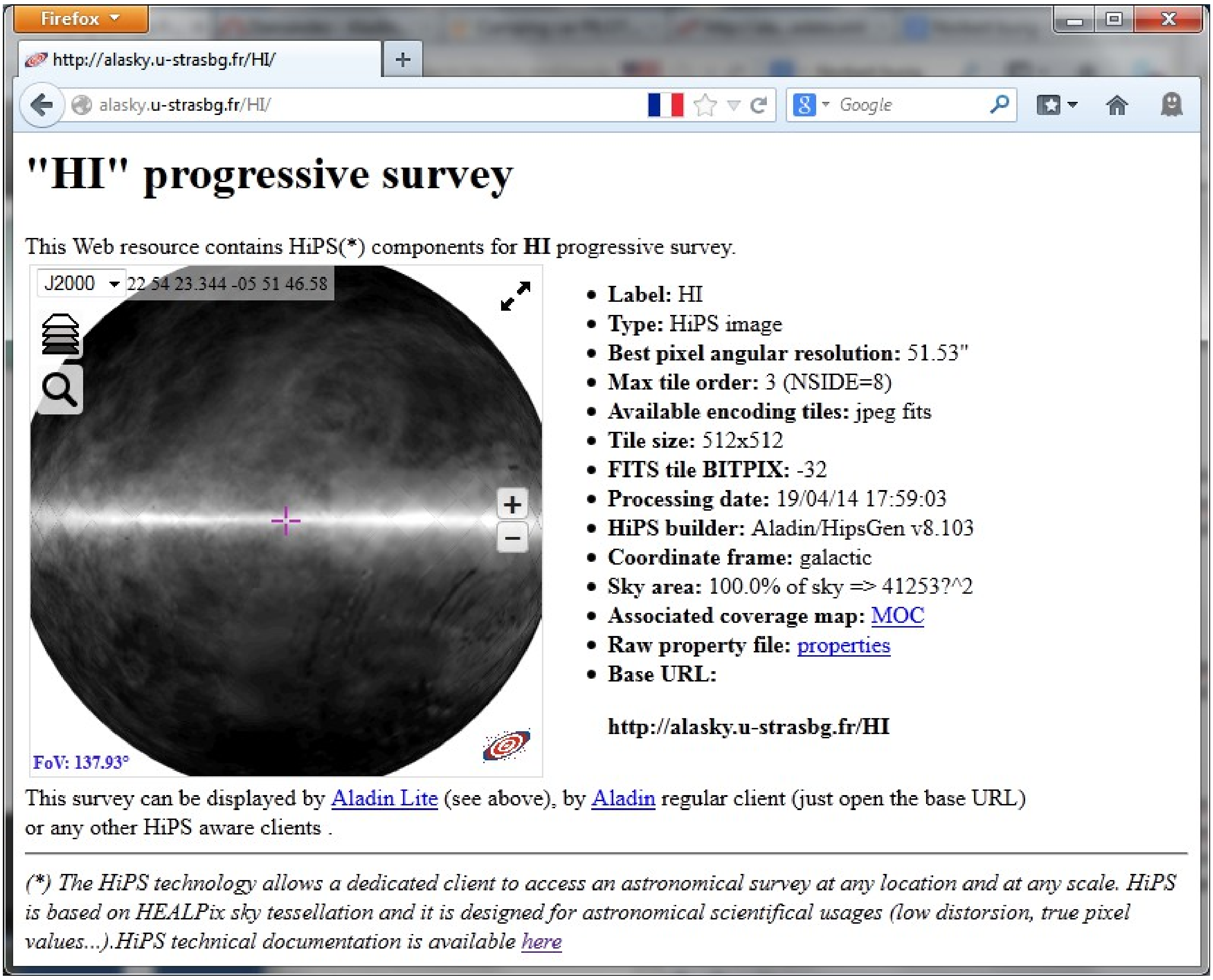
6.3 MOC

A file called "Moc.fits", stored in the HiPS root, **MAY** be provided for delivering the coverage map of the survey. This file follows the IVOA MOC standard. The client can be used it for avoiding to try to load the tiles outside the HiPS. The MOC coordinate reference (equatorial, ecliptic, galactic) **MUST** be the same that the HiPS.

6.4 index.html

A file called "index.html", stored in the HiPS root, **MAY** be provided for offering a basic HTML presentation of the survey. By this way a simple Web browser, loading the root HiPS directory will be able to display information on the HiPS itself.

For instance, Aladin/HipsGen tool generates a default "index.html" file implementing "Aladin Lite" panel + survey information.



"HI" progressive survey

This Web resource contains HiPS(*) components for **HI** progressive survey.

- **Label:** HI
- **Type:** HiPS image
- **Best pixel angular resolution:** 51.53"
- **Max tile order:** 3 (NSIDE=8)
- **Available encoding tiles:** jpeg fits
- **Tile size:** 512x512
- **FITS tile BITPIX:** -32
- **Processing date:** 19/04/14 17:59:03
- **HiPS builder:** Aladin/HipsGen v8.103
- **Coordinate frame:** galactic
- **Sky area:** 100.0% of sky => 41253?^2
- **Associated coverage map:** MOC
- **Raw property file:** properties
- **Base URL:**

  http://alasky.u-strasbg.fr/HI

This survey can be displayed by Aladin Lite (see above), by Aladin regular client (just open the base URL) or any other HiPS aware clients .

(*) The HiPS technology allows a dedicated client to access an astronomical survey at any location and at any scale. HiPS is based on HEALPix sky tessellation and it is designed for astronomical scientifical usages (low distorsion, true pixel values...).HiPS technical documentation is available here

7 Progenitors & HpxFinder

In case of HiPS images, an additional directory called “HpxFinde” **MAY** be also provided in the HiPS root. This directory offers to advanced HiPS browsers an efficient way to access to the “original image” and other related information.

Note : This method is still in development. It is based on the Aladin/HipsGen HiPS generator which creates a HEALPix spatial index to know the list of original images covering such or such HEALPix tile. This spatial index is built as a set of HEALPix tiles, in JSON format, stored in the HpxFinder directory. Each tile contains the names and the paths of the original images covering, even partially, the tile.

The idea consist to also publish this list of tiles generated during the HiPS creation phase for retrieving the list of the original image ID covering the current user view.
As the HpxFinder directory structure is the same that the HiPS itself, it is straightforward for a client to retrieve the associated HpxFinder tiles (the same tile index), extract the data set information, removes the possible doubloons to get the list of the original images, also called « progenitors » - their IDs, their footprints on the sky...

7.1 HpxFinder tile format

An HpxFinder tile is a list of ASCII lines, **one line per image**, following a JSON record syntax { "key": "value", …}.

The default keywords are :

- ra: ICRS RA center of the image
- dec: ICRS DEC center of the image
- name: the ID/name of the image
- path: the path of the image during the HiPS creation process
- stc: the footprint of the image in STC IVOA standard.

Depending of the HiPS generator capacity, additional keywords can be also present for storing meta data information associated to each original image such as exposure time, wavelength,... For instance, Aladin/HiPSgen tool allows to specified a list of FITS keywords for which the values will be extract from original images and stored in the corresponding HpxFinder tiles.

Example :

```
{ "ra": "221.040", "dec": "42.60632", "name": "hst_12018_18_wfc3_uvis_f438w_
drz", "path": "/staging/sciproc2/1/durand/HSTSKY_FINAL/extraction/F438W/hst_
12018_18_wfc3_uvis_f438w_drz.fits[2048,4096-327x44]", "stc": "POLYGON J2000
221.07105 42.60602 221.03554 42.60602 221.03553 42.65159 221.07107 42.65159" }
```

7.2 VOTable progenitor template

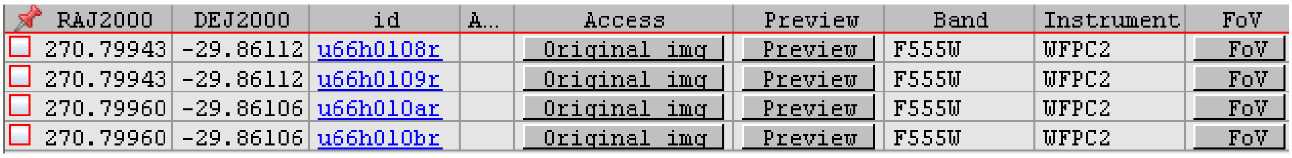
For formatting and publishing the information stored in the HpxFinder JSON tile, the progenitor mechanism extends the same idea developed for HiPS catalog via the “metadata.xml” file (see above). A VOTable template is provided in the root HpxFinder directory for providing to the clients progenitor metadata information as a classical VO table with its own metadata (units, field descriptions, links,...), plus some rules to extract data from the JSON tile. Concretely the VOTable header follows the classical VOTable standard (like HiPS catalog “metadata.xml” file), plus a unique generic DATA record providing a rule per field for extracting data. This rule is a string template with $[xxx] variables referring JSON fields, possibly followed by a regular expression for extract subfields or the JSON corresponding value (ex : *prefix $[key1] suffix $[key2:.*/(\w+)]...*)

“metadata.xml” HpxFinder example :

```
<RESOURCE>
  <TABLE name="HST">
    <DESCRIPTION>HST progenitors</DESCRIPTION>
    <FIELD name="RAJ2000" ucd="pos.eq.ra" datatype="double" width="9" precision="5" unit="deg">
      <DESCRIPTION>Right ascension</DESCRIPTION>
    </FIELD>
    <FIELD name="DEJ2000" ucd="pos.eq.dec" datatype="double" width="9" precision="5" unit="deg">
      <DESCRIPTION>Declination</DESCRIPTION>
    </FIELD>
    <FIELD name="id" ucd="meta.id;meta.dataset" datatype="char" arraysize="10*">
      <DESCRIPTION>Dataset name, uniquely identifies the data for a given exposure.</DESCRIPTION>
       <LINK gref="HST.obs ${id}"/>
     </FIELD>
    <FIELD name="Access" datatype="char" arraysize="9*">
      <DESCRIPTION>Load original this original image</DESCRIPTION>
       <LINK content-type="image/fits"
         href="http://www.cadc.hia.nrc.gc.ca/data/pub/HSTCA/${id}_drz.fits" title="Original img"/>
     </FIELD>
    <FIELD name="Preview" datatype="char" arraysize="9*">
      <DESCRIPTION>Load preview  image</DESCRIPTION>
       <LINK content-type="image/fits"
         href="http://www.cadc.hia.nrc.gc.ca/data/pub/HSTCA/${id}_prev.jpg" title="Preview"/>
     </FIELD>
    <FIELD name="Band" datatype="string" arraysize="10*">
      <DESCRIPTION>Wavelengh band</DESCRIPTION>
     </FIELD>
    <FIELD name="Instrument" datatype="string" arraysize="10*">
      <DESCRIPTION>Instrument</DESCRIPTION>
     </FIELD>
    <FIELD name="FoV" datatype="char"
        utype="stc:ObservationLocation.AstroCoordArea.Region" arraysize="7*">
      <DESCRIPTION>Field of View (STC description)</DESCRIPTION>
    </FIELD>
```

```
<DATA><TABLEDATA>
   <TR>
   <TD>$[ra]</TD>
   <TD>$[dec]</TD>
   <TD>$[name:(.*)_drz]</TD>
   <TD>$[name:(.*)_drz]</TD>
   <TD>$[name:(.*)_drz]</TD>
   <TD>$[path:.*extraction/(\w+)/\w+.*]</TD>
   <TD>$[path:.*extraction/\w+/(\w+).*]</TD>
   <TD>$[stc]</TD>
   </TR>
</TABLEDATA></DATA>
</TABLE>
</RESOURCE>
```

Example of corresponding client result:



	RAJ2000	DEJ2000	id	A...	Access	Preview	Band	Instrument	FoV
□	270.79943	-29.86112	u66h0108r		Original img	Preview	F555W	WFPC2	FoV
□	270.79943	-29.86112	u66h0109r		Original img	Preview	F555W	WFPC2	FoV
□	270.79960	-29.86106	u66h010ar		Original img	Preview	F555W	WFPC2	FoV
□	270.79960	-29.86106	u66h010br		Original img	Preview	F555W	WFPC2	FoV

*Note***:** the links to the original images, previews, additional data, VO datalink, etc are realized thanks to the LINK VOTable mechanism (see example below). Notice that the LINK format used is own url template (href=...) with their own variable mechanism ${xxx} referring the <FIELD id attribute. This mechanism is independent to the JSON extraction rules.

Note: the usage of the STC describing the field of the original image depends of the client. Aladin recognizes the dedicated utype attribute "stc:ObservationLocation.AstroCoordArea.Region" and is able to display it over a background image.

8 Miscellaneous

8.1 Fits tile compression

For reducing the size of Fits tile, and also the Allsky.fits file, a ZIP or GZIP compression algorithm may be applied on the tiles, or a part of tiles. This task can be very long, especially for large surveys, longer than the HiPS generation itself. A good compromise consists to compress only the low orders (order 0,1,2,3,4,5) and the corresponding Allsky.fits file. The time required to compress deeper orders grows a lot, and the benefit is decreasing (at low orders, notably for partial sky survey, the tiles are generally partial implying a good compression factor).

Note: Even if the fits tiles and the Allsky.fits file is gzipped, these files keep the same name extension (.fits). The browser must detect if the tile is compressed or not (for instance based on GZIP magic code).

8.2 Client cache

It is strongly recommended to implement a cache mechanism on the client side. Due to the mirror HiPS potential copies, it is preferable to take into account the date inside the “properties” file rather that the date of files.

If the HiPS is updated regularly as Simbad or NED, the “live” attribut on “dataproduct_subtype” property file must be set, and the client may apply the HTTP cache protocol for each individual tiles in order to download only new tiles.

8.3 HiPS examples

All CDS HiPS can be displayed thanks to a basic Web navigator via this table: http://aladin.unistra.fr/hips/list.

HiPS site & HiPS mirroring

As described below, the distribution of HiPS are realized via a classical HTTP server. Several astronomical institutes are deploying a such server distributing their own HiPS, or mirrored HiPS from other sites. In this context, HiPS version 1.3 describes a minimal basic protocol for managing HiPS site and HiPS mirror mechanism.

1 HiPS node, HiPS list

A "*HiPS node*" is a basic HTTP site which provides HiPS surveys.

A HiPS node **MUST** implement one dedicated URL (simple ASCII file, CGI, …) for publishing the list of HiPS surveys that he is delivering. This list is called the "*HiPS lis*t" of the node. This list is a collection of ASCII records (key1 = property1\nkey1 = property2...) blank line separated (same syntax and vocabulary that HiPS "properties" file. The lines beginning with # is considered as a comment. Each record **MUST** provided at least these properties: **publisher_did, hips_release_date, hips_service_url, hips_status**. Other properties **MAY** be provided: **hips_order, hips_tile_format, hips_estsize, moc_sky_fraction**,...

See previous table in the meta data section for property description. The table below describes two additional properties dedicated to HiPS node description:

hips_service_url	Base URL of the HiPS
hips_status	Status of the HiPS node – Format: list of keywords: **public|private :** allowed for all clients (public), or only for dedicated authorized clients (private) **master|mirror|partial :** original HiPS (master), full copy of HiPS (mirror), partial copy of HiPS (partial) **clonable|unclonable** : HiPS mirroring allowed (clonable) or not (unclonable) Default: public master clonable

Example:

```
# Hipslist of http://alasky.unistra.fr HiPS node
# Date: 2015-04-16T13:42Z
# Do not copy any HIPS with unclonable status !

publisher_did        = ivo://CADC/P/HST/F850LP/r3
hips_release_date    = 2014-10-14T12:00Z
hips_service_url     = http://alasky.u-strasbg.fr/HST/F850LP
hips_status          = public master clonable

publisher_did        = ivo://CDSP/2MASS/H
hips_release_date    = 2014-11-03T12:00Z
hips_service_url     = http://alasky.u-strasbg.fr/2MASS/H
hips_status          = public mirror unclonable
hips_estsize         = 1610612736
hips_order           = 9
hips_tile_format     = fits jpeg
dataproduct_type     = image
moc_sky_fraction     = 1

…
```

Warning: if the properties provided by the hipslist differs from individual properties file (in HiPS root directory), the hipslist properties takes precedence.

Note: There is no constraints for HiPS list URL, but a default choice **MAY** be: **http://xxxx/hipslist** where xxxx is the HTTP server hostname (ex: http://alasky.unistra.fr/hipslist)

Help: This Perl script code: http://aladin.unistra.fr/hips/perl_hipslist can be adapted in order to generate on the fly the hipslist from a list of HiPS directories.

2 HiPS node registry

The HiPS registry is a list of available HiPS nodes. It is available via this URL: http://aladin.unistra.fr/hips/registry. This list is a collection of ASCII records (key1 = property1\nkey1 = property2...) blank line separated. The lines beginning with # is considered as a comment. Each record MUST provided at least these properties: **hips_node_id, hips_node_url**.

Example :

```
hips_node_id          = ivo://CDS/hipsnode/one
hips_node_url         = http://alasky.u-strasbg.fr/hips

hips_node_id          = ivo://ESAC/hipsnode
hips_node_url         = http://skies.esac.esa.int/list.txt
…
```

The **hips_node_id** provides a unique HiPS node identifier (IVORN syntax). The **hips_node_url** provides the HiPS list of the node (see above)

3 HiPS node mirroring policy

FIRST OF ALL: BEFORE DOING ANY KIND OF COPY OF HIPS SURVEYS, THE INSTITUTE HOSTING YOUR HIPS NODE MUST HAVE THE AGREEMENT OF THE ORIGINAL DATA PROVIDER FOR COPYING THE DATA PACKAGED AS HIPS AND/OR TO REDISTRIBUTE THE RESULTING MIRRORED HIPS. THAT'S WHY LOOK CAREFULLY THE COPYRIGHT MENTION OF EACH HIPS THAT YOU WOULD LIKE TO MIRROR. THE FACT THAT A COPY IS TECHNICALLY POSSIBLE DOES NOT MEAN THAT YOUR INSTITUTE OR YOURSELF ARE AUTHORIZED TO DO THE COPY.

After this legal prerequise, any HiPS node can mirror an HiPS survey from other HiPS node on condition that the HiPS survey is described by the HiPS node list, and without the property **"hips_status = … unclonable … ".**

There is no dedicated protocol for the mirroring process. It can be done via wget, hipssniffer, rsync or any other method used for synchronized HTTP web sites.

A HiPS node which provides a copy of HiPS survey **MUST** specify "**hips_status = … mirror ...**" for this HiPS.

A HiPS node which provides a copy of HiPS survey **MUST NOT** modify the **hips_release_date** of the master site in order to detect out of date copies.

If the mirroring is partial (lower hips_order, or only a subset of tile formats), the node who has done the mirroring must specify in its HiPS list "**hips_status = partial**", and adjust the relevant properties (**hips_order, hips_tile_format**), in the HiPS list and in the "properties" file of the concerned HiPS.

4 HiPS security

HiPS distribution is based on classical HTTP server It **MAY** be protected by HTTP authentication (BASIC or more advanced method) and/or encoded via HTTPS protocol.

5 HiPS statistics

An HiPS node **SHOULD** implements an URL providing cumulative statistics. These statistics provides the number of asked tiles per HiPS survey and per period (typically 1 month). They are presented as a ASCII records (key1 = property1\nkey1 = property2...) blank line separated (same syntax and vocabulary that HiPS "properties" file). The lines beginning with # is considered as a comment.

The method used for generating and updating these statistics is free. It could be a basic "grep" in the HTTP log file or any other more sophisticated method, repeated regularly via crontab or other equivalent method.

The record properties dedicated to statistics are:

stat_hips_tile_hits	Number of asked tiles (HTTP code 20X) – Format : integer
stat_t_min	Date of the beginning (include) of the measurement period – Format: real – Unit: MJD
stat_t_max	Date of the end (include) of the measurement period – Format: real – Unit: MJD

Example:

```
publisher_did         = ivo://CADC/P/HST/F850LP/r3
stat_hips_tiles_hits = 234569
stat_t_min            = 56444
stat_t_max            = 56475

publisher_did         = ivo://CADC/P/HST/F110W/r3
stat_hips_tiles_hits = 10543
stat_t_min            = 56444
stat_t_max            = 56475
...
```

Note: Several records can be associated to the same publisher_did for describing several periods.

Note: There is no constraints for HiPS stat URL, but a default choice **MAY** be: **http://xxxx/hipsstat** where xxxx is the HTTP server hostname (ex: http://alasky.unistra.fr/hipsstat)

6 Global HIPS list

The "*global HiPS list*" is an URL providing the full list of available HiPS over all HiPS node, with all associated properties. This list is a collection of ASCII records (key1 = property1\nkey1 = property2...) blank line separated (same syntax and vocabulary that HiPS "properties" file). The lines beginning with # is considered as a comment.

This "global HiPS list" is dedicated notably to the HiPS clients (Aladin desktop, Aladin Lite and derived tools [MMI, JUDO2, …], MIZAR,...).

In case of mirror sites, the property **hips_service_url** will be repeated with the suffixes "**_1**", "**_2**", etc. In case of partial mirror sites, some specific properties such as **hips_tile_format** and **hips_order** will also be derived with suffix.

The "Global HiPS list" is available at this URL: http://aladin.unistra.fr/hips/globalhipslist

An human readable alternative of this globalhipslist is provided at this URL : http://aladin.unistra.fr/hips/list

Note: According to the evolution of IVOA registry, it is possible that the role of this "global HiPS" will be partially assumed in the future by the VO registry.