

# 1- Various use cases on planetary surfaces with Aladin & TOPCAT

Follow-up of previous ObsParis / CDS meeting

Initially using Aladin prototype v9.6 (27/6/2017) and TOPCAT v4.4 on MacOS

- See also page: [Aladin & planetary surfaces](#) for other use cases / examples.

Updated 7/2019 using Aladin prototype v10.1 / based on v10.128 (12/5/2019) and TOPCAT v4.6-3 on MacOS

## Set up for planetary mapping

Assuming longitude is E-handed (EPN-TAP convention, and IAU convention in planetocentric frames):

### • Standard setup for planetary maps in TOPCAT SkyPlot:

(in Axes / Projection)

Projection = sin (i.e. sphere, for orthographic projection), but car (plate carrée / cylindrical) and Aitoff are also possible (will not rotate)

Uncheck Reflect longitude axis, (keep it checked only if coordinates are provided with W longitudes)

View Sky System: Equatorial,

(in Axis / Grid)

Uncheck Sexagesimal

Increase Grid Crowding cursor value to 30 or 60° tick

For cylindrical maps:

(in Axes / Projection)

Projection = car (for 0° at center, range = 0-360°)

Projection = car0 (for 0° on left border)

### • Standard setup for planetary maps in TOPCAT PlanePlot (cylindrical):

(in Axes / Range)

Set Max X value to 360

Central meridian will be on left border

- To put the central meridian at the center (range = -180 to 180):

(in Axes / Range)

Min / Max X = -180 / 180

(in Mark / Position)

X: LON > 180 ? Lon - 360 : Lon

(in Axes / Labels)

X Label = Longitude

- From v4.6-3, the SkyPlot window is more versatile for maps (see above)

### • Standard setup for planetary maps and HiPS in Aladin:

In the fields on top of the window, set Frame to "Planet" or "Planet deg" - this will use a grid in degrees (counted E and W from 0 - not the IAU convention, but OK)

Set Projection to Sphere (actually: orthographic) or Cartesian (actually: cylindrical) for most applications. Aitoff is ~ similar to sinusoidal, with all the surface visible at once; keep it centered on (0°,0°) !

In properties, .longitude should be set to "ascending" for correct orientation.

## Healpix generation from a table in TOPCAT

Healpix is a tessellation system on the sphere, which depends on a resolution/scale parameter. It provides regular sampling in surface, with no singularity (poles).

When using this system, TOPCAT will integrate elements included in the corresponding cells. The benefit is to easily get a combined map from sparse data; the result is also scalable in resolution: at least degradation is performed correctly by combining healpix cells.

Cell size (in FoV size, ie in longitude at Equator): scale 6 ~ 1°, 12 ~ 1', 18 ~ 1"

From the SkyPlot window, several map projections are available with this mode (in Axes panel): "sin" (actually: orthographic, 3D), Aitoff (all surface visible, minimize deformations near the poles), and "car" (cylindrical).

In TOPCAT, load a table containing a list of observations located in lon/lat (the working example is a large table of VIRTIS/Rosetta observations of 67P)

A detailed step by step tutorial is now available: [Use case: mapping sparse spatial data with TOPCAT](#)

## 1) Standard way (density map from a lon/lat table):

- Open the SkyPlot panel, select table in Position panel, as well as Lon and Lat parameters (C1min and C2min for an EPN-TAP table)
- Go to Form panel, click on the big Forms button (green cross) and "Add SkyDensity" from the local menu. Uncheck the previous Mark Form
- In the Form panel, select parameter to plot as Weight (will be weighted by density), select combine mode, and adjust scale. Healpix generation is performed on the fly, so all scales can be selected.
- Possibly adapt color scale in the Aux Axis tab (left menu)
- In the SkyDensity / Form panel, play with HEALPix level to set a convenient integration scale.

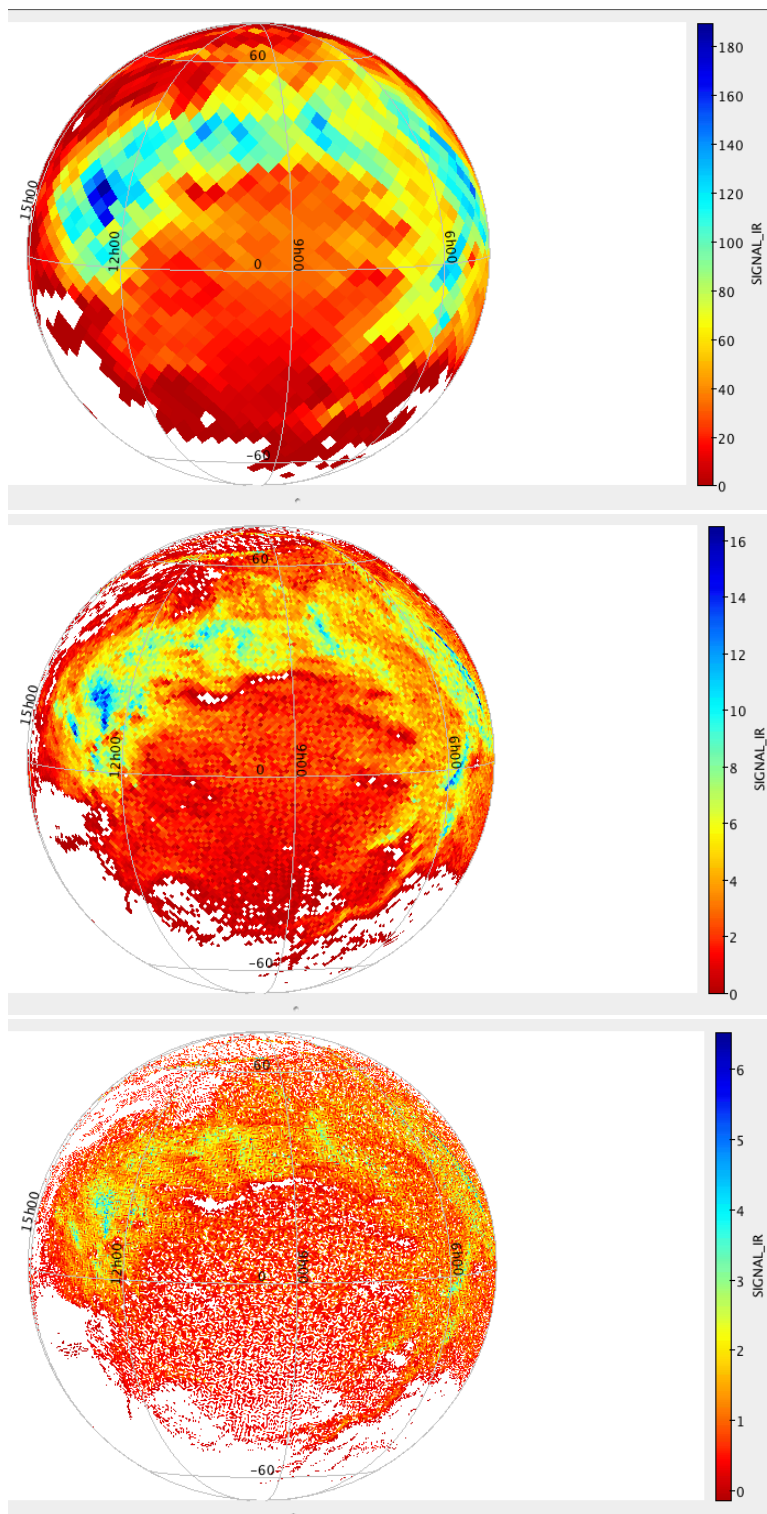
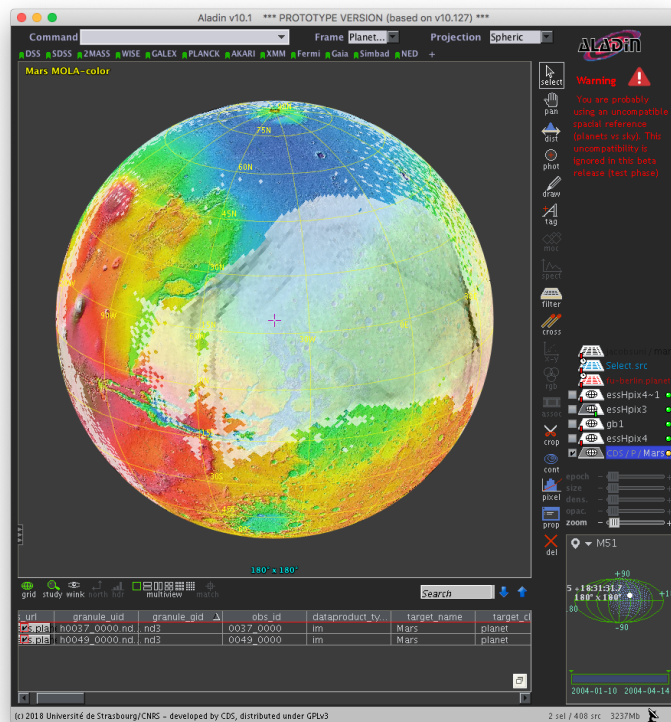


Figure: VIRTIS/Rosetta measurements on 67P, located in lon/lat

- In the Form panel, you can either "Import" to create the integrated HEALPix table in TOPCAT at the current scale (2 columns: cell #, plotting parameter; only filled cells are present), or "Save" in a file in HEALPix-fits format (FITS formatting is more efficient than VOTable, as it is compressed).

- HEALPix-fits maps from TOPCAT (>v4.6-3) open in Aladin and can be, e.g., superposed on a planetary HiPS. Aladin prototype v10.1 (based on v10.127) is required to plot the HEALPix map with correct orientation (set longitude = ascending in Properties panel of the HiPS).



## 2) Alternatively (from a binned table):

This other method is adapted when you have only 1 or 0 data / healpix cell, ie if the grouping has already been done in a previous step (e. g., with an ADQL query to group data by HEALPix cells, or when opening a previous HEALPix-fits map). It is not intended to group or average table rows in the cells interactively.

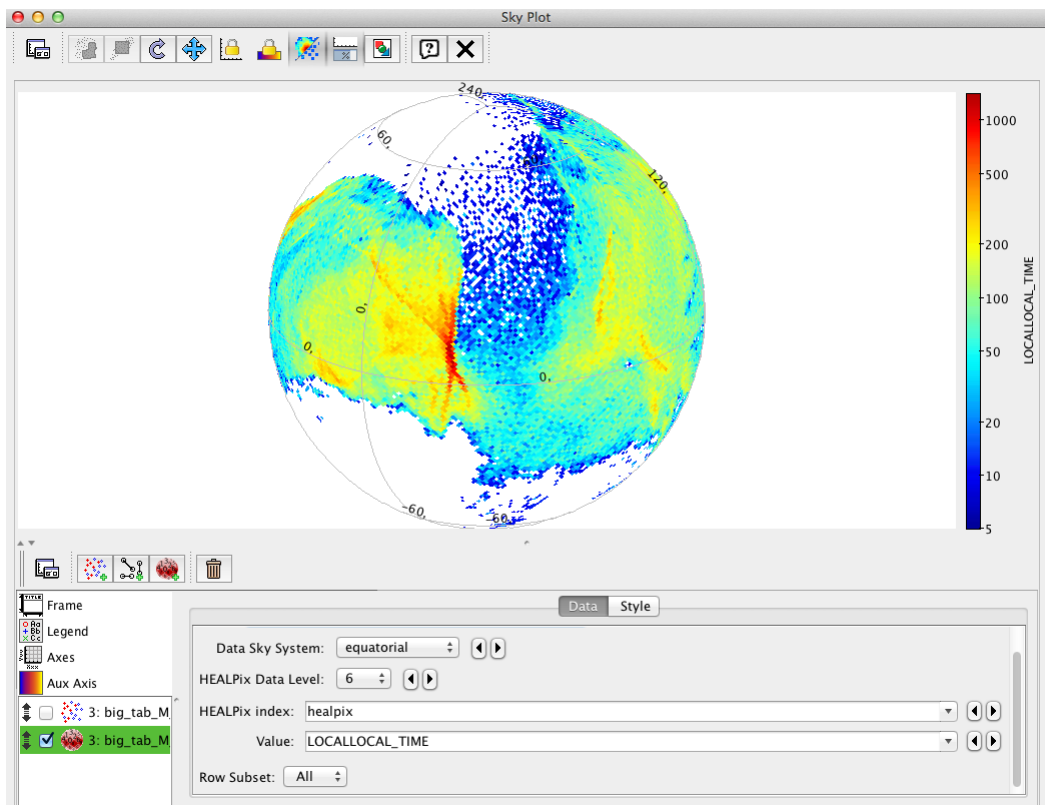
- If starting from a lon/lat table, first add a new synthetic column providing healpix cell number (right-click on table column header and enter new column description):

```
healpixNestIndex( 6, longitude, latitude )
```

(adapt scale to data in use, 6 provides a resolution of  $\sim 1^\circ$  at the Equator)

- (once you have a table with healpix cell # included)  
Open Skyplot panel, add New healpix layer, provide inputs  
Correct Healpix Data Level in particular (TOPCAT will try to guess, but this is uncertain)

- In Skyplot / Style panel, use Degrade and Combine mode to reduce resolution on the fly (super-resolution can no longer be achieved)



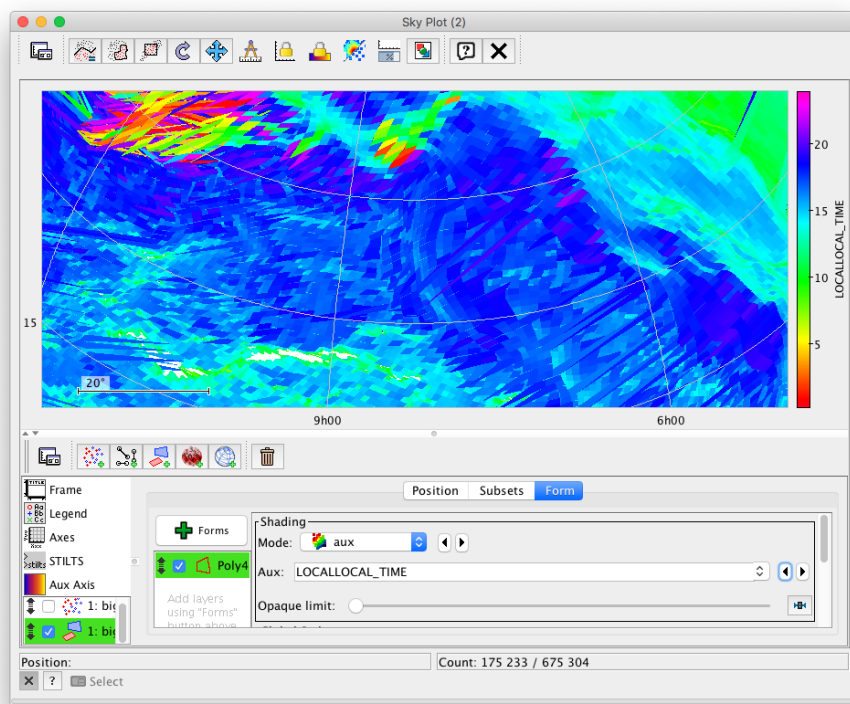
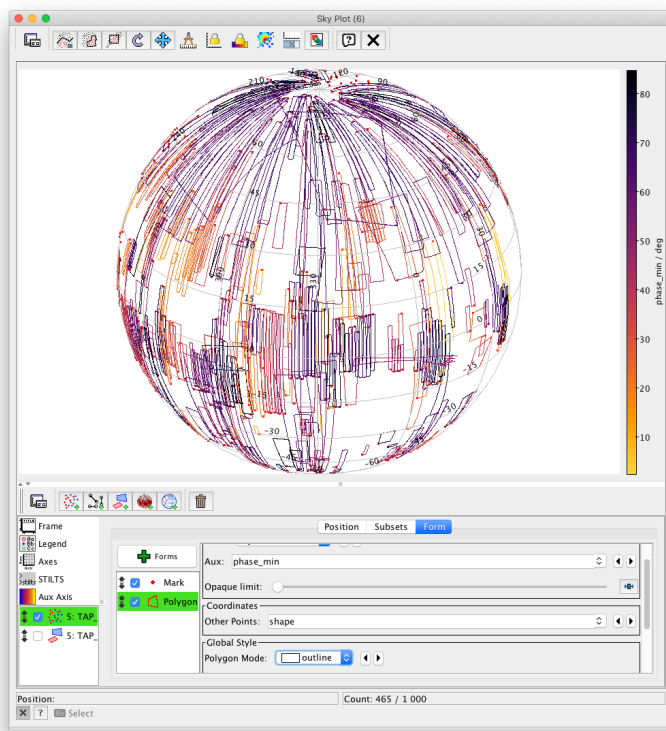
## Complex footprints in TOPCAT

This mode allows display of either file footprints (series of images or spectral cubes) or extended pixels footprints (individual spectra from a cube).

Set up is easier if table columns are associated with relevant UCDs (typically pos.bodyrc.lon/lat).

- When FoV have 4 corners (or only 3), use Add New Quadrilateral Plot in a SkyPlot or PlanePlot window
- In the Position tab, set the 4 Lon/Lat pairs to corners values
- In the Form tab, set Mode = Aux and select the adequate parameter as usual
- Set the other graphical parameters as usual

If contours are more complex, use the Polygon Form.





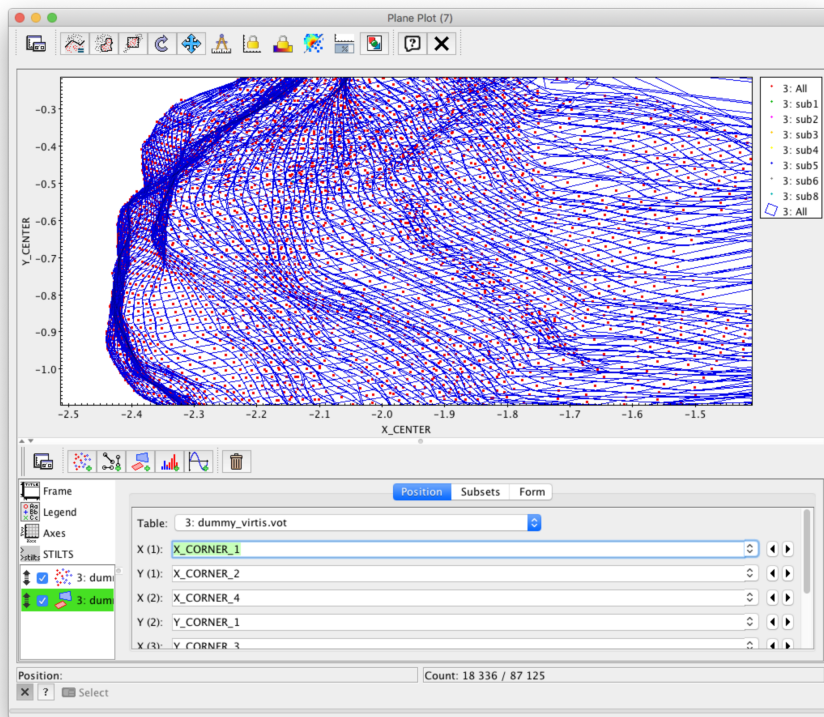


Figure XX (a) HSRC/MEx image footprints. (b) individual spectra (iFoV) from VIRTIS/Rosetta cubes, in SkyPlot with lon/lat. (c) same in PlanePlot with Cartesian coord.

- If the table contains footprints provided through an STC-S *Polygon* string (in particular s\_region in EPN-TAP tables):
  - define a new column (e.g. "shape") with expression "parseDoubles(s\_region)"
  - use Lon=shape[0], Lat=shape[1] as positions in Skyplot
  - add a new Form of type Polygon, and fill in Other Points field as "shape"

This applies either to image contours, to individual FoV for spectra, and to other situations:

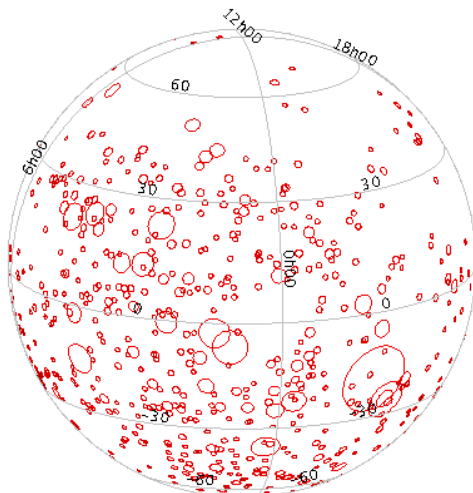


Figure: Mars craters contours.

## Plotting on shape models in TOPCAT

If the table contains the Cartesian coordinates of individual measurements, TOPCAT can easily display this in 3D: (the shape model is then actually included in the data - but only where measurements exist)

- Open a 3D Cartesian plot window. X/Y/Z coordinates are preselected if the names are explicit enough. This also works in 2D plots, with all Z values projected in the X/Y plane.
- The usual plot options are available to superpose an Aux parameter.

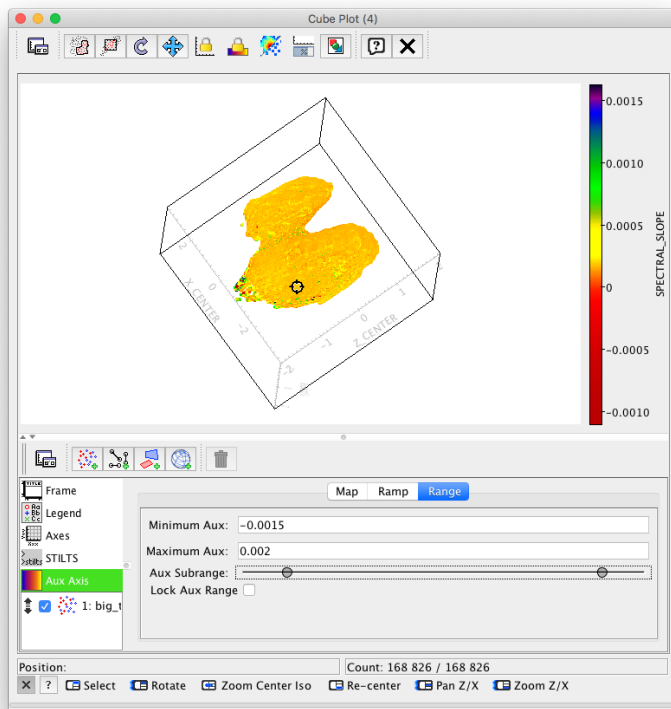


Figure: VIRTIS/Rosetta measurements on 67P, in Cartesian coordinates

## Looking for spatial overlaps between datasets

This uses TAP functions on the s\_region parameter. Such requests can be issued from TOPCAT using a subset table select in the VESPA portal.

A tutorial is available here:

[https://github.com/epr-vespa/tutorials/blob/master/surfaces/HRSC\\_vs\\_OMEGA/HRSC\\_vs\\_OMEGA-tutorial.md](https://github.com/epr-vespa/tutorials/blob/master/surfaces/HRSC_vs_OMEGA/HRSC_vs_OMEGA-tutorial.md)

This includes mapping examples in TOPCAT and Aladin.

## Planetary HiPS generation in Aladin

HiPS (Hierarchical progressive survey) are associations of healpix maps at different scales. They provide efficient multiresolution scaling on the fly in Aladin.

### 1) From a single file (updated May 2019, v10.128)

The use is mostly to get a 3D spherical model from a map, where contours and objects can be overplotted. See Io use case for more details: [Aladin & planetary surfaces](#)

In Aladin, load file Mars\_MGS\_colorhillshade\_mola\_1024.jpg (MOLA integrated relief map)

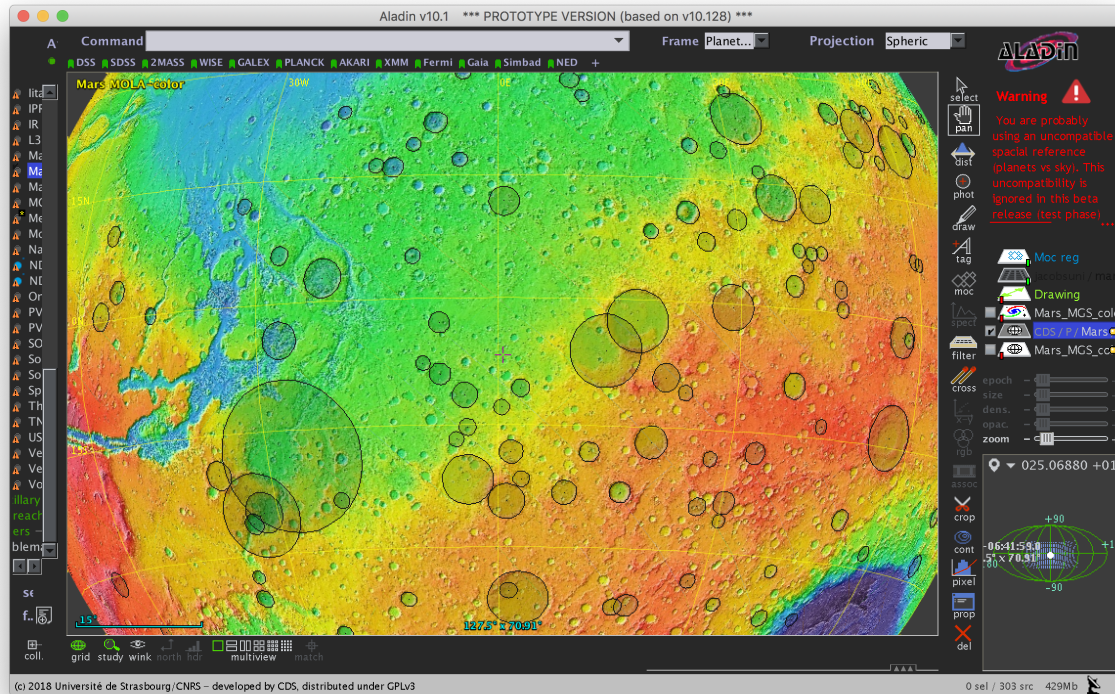
- Go to Image > astrometric calibration:  
Coord = 0.,0.  
pos x/y = 512, 256 (lon = 0°, lat = 0° at image center)  
size = 21.1' (= 360. / 1024)  
Equatorial, Cartesian  
RA symmetry = True => this is mandatory to get the image in the correct orientation.
- Convert to HiPS :  
Tool > Generate HiPS based on... > current image
- The hpx image is saved (somewhere) on disk automatically (you may have to search for it)



- Select the HiPS and set Properties => .longitudes = ascending ; will plot correctly with E longitudes eastward (the button exists but this option is *not* working in v10.127)

If color scale goes crazy, fix it by clicking the Pixel icon, then Reverse button.

- s\_region footprints sent from VESPA (with E longitudes) will plot at the right place.
- s\_region footprints retrieved directly by Aladin from an EPN-TAP service (with E longitudes) will also plot at the right place.



## 2) From a series of local files in Aladin

This method will provide adaptive resolution when using files of various scales

TBC, did not work on my machine early 2017.

## 3) Generate your HiPS from a planetary cartesian map thanks to Hipsgen CDS tool

(from P. Fernique, 9/7/2019, tutorial at 2nd Europlanet Planetary Mapping Workshop)

A tutorial concerning the generation of planetary HiPS from global CAR maps in JPEG, PNG or TIFF. This is the same method used by the CDS for 95% of the planetary HiPS available today. Notice that if you have them in FITS (via Chiara Marmo PDS2FITS conversion for instance) you can directly apply the regular Hipsgen method described [here](#).

(Warning: this tutorial uses undocumented Hipsgen parameter)  
P.Fernique - 2nd of July 2019 -

- 1) You need a cartesian map in JPEG, PNG format covering the whole body (or sky)  
Rq: if you have it in TIFF see 4
- 2) You need to generate a World Coordinate System (WCS) solution for you map.  
It is a small ASCII file with the same name and the .hhh extension containing the WCS solution of your image:

ex: Cartesian hhh file content for a 2880x1440 map

```
NAXIS1 = 2880      // ImageWidth
NAXIS2 = 1440      // ImageHeight
CRPIX1 = 1440      // ImageWidth/2    => central pixel x coord
CRPIX2 = 720       // ImageHeight/2   => central pixel y coord
CRVAL1 = -0.0625   // -(360/imageWidth)/2 => longitude of the central pixel
CRVAL2 = -0.0625   // -(180/imageHeight)/2 => latitude of the central pixel
CTYPE1 = RA---CAR
CTYPE2 = DEC--CAR
CD1_1 = 0.125      // 360/imageWidth   => pixel resolution
CD1_2 = 0
CD2_1 = 0
CD2_2 = 0.125      // 180/ImageHeight  => pixel resolution
```

- a) you can do that manually via any ASCII editor
- b) Or programmatically thanks to Hipsgen.jar itself. In this last case, just launch this kind of command:

```
java -jar Hipsgen.jar -hhh="path/FileName widthxheight"
ex: java -jar Hipsgen.jar -hhh="path/Titan.jpg 2880x1440"
```

Note: if your longitude origin is not localized at the central pixel line of your map, you can add its x coordinate as a third parameter (ex: java -jar Hipsgen.jar -hhh "path/Titan.jpg 2880x1440 0")

- 3) After that you can generate your HiPS by this kind of command

```
java -jar Hipsgen.jar in=path out=MyHips id="YourAUTH/P/Titan" color=jpg
```

Rq: you can also do steps 2 and 3 in one unique Hipsgen command

```
java -jar Hipsgen.jar -hhh="path/Titan.jpg 2880x1440" in=path out=MyHips id="YourAUTH/P/Titan" color=jpg
```

- 4) If your original image is really big, and/or in TIFF format, it is preferable to convert and to split it in a collection of small JPEG subimages before launching Hipsgen process. ImageMagick toolkit (free on Linux) does that magically thanks to this command:

```
convert srcImage.tif -crop widthCellxheightCell +repage TargetDir/GenericName.jpg
```

It is better to have subimages with the same size (one divisor factor of the source image)

ex :

```
convert Lunar_LRO_LROC-WAC_Mosaic_global_100m_June2013.tif -crop 10112x10112 +repage SplitMg/Moon.jpg
```

And thus, launch Hipsgen.jar with -hhh parameter with one additional parameter as size of subimages

ex:

```
java -jar Hipsgen.jar -hhh="SplitMg/Moon.jpg 92160x40448 10112x10112"
```

or in one Hipsgen command like this:

```
java -jar Hipsgen.jar hhh="SplitMg/Moon.jpg 92160x40448 10112x10112" in="SplitMg" creator_did="XXX/P/Moon/LROC-WA-GLD100-118m" -f color=jpg
```

- 5) in order to get the longitude direction right in Aladin, you may have to edit the generated index.html and change the following line :

```
aladin.setImageSurvey(aladin.createImageSurvey('2019-0712-1456', '2019-0712-1456', hipsDir, 'equatorial', 3, {imgFormat: 'png'}));
```

to:

```
aladin.setImageSurvey(aladin.createImageSurvey('2019-0712-1456', '2019-0712-1456', hipsDir, 'equatorial', 3, {imgFormat: 'png' ,
longitudeReversed: true}));
```

You can alternatively set hips\_frame to planet name (will change coord.sys). In Aladin Desktop you can also set properties to .longitude = ascending

## MOC use case in Aladin

(updated May 2019)

MOC (Multi-Order Coverages) are footprints defined in terms of a healpix cells list mixing various resolutions. They provide immediate location information e.g. about intersections, and can accommodate footprints of arbitrary complexity (hollowed, non-connex, etc...). There is no explicit contour associated, so the usage is different from the s\_region parameter.

- ☐ Should also retain info about original file coverages, for callback purpose - TBC

- Load previous Mars MOLA HiPS in Aladin

- Select two large craters in Mars Craters service from VESPA client
- send as s\_region from VESPA client (to Aladin) — (s\_region was then provided in this service with W longitudes, so this projected correctly)
- select them in Aladin with pointer
- Generate common MOC for these 2 craters (Coverage > Generate a MOC based on... > selected drawing objects)
- You can save it from File > Export Planes (save preferably as fits for reuse)

This is straightforward and very quick.

This is not a direct alternative to s\_region however: the result is a list of healpix cells which is much larger than a contour, and can't be included in a table. But we can either have precomputed MOC stored somewhere (for complete datasets? Especially with time direction added), or compute them on the fly when required.

- Next step: test intersections with something else - see example in the next pages

