
MountWizzard4
Version 4.0.0a19

Table of Contents

Welcome to MountWizzard4!

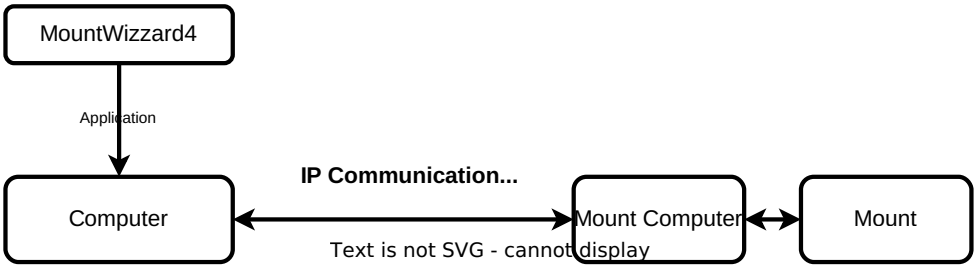
MountWizzard4 is a general utility for 10micron users for improving the workflow for astronomy work. It runs on Windows11, Windows10 (Win7 should be fine, but it will be not tested), Mac OSX (beginning from 10.12 to 14.x) including Mx variants if Rosetta is used and Linux (Ubuntu from 16.04 to 22.04). If you have some knowledge around Raspberry Pi's and other SOC, you might be able to install MountWizzard4 on a RPi3+, RPi4 or RPi5.

PDF Documentation:

https://mworion.github.io/MountWizzard4/mountwizzard4.pdf

Before starting

First let us have a look to the basic architecture: MountWizzard4 is an application installed on your external computer which is connected to the mount computer via an IP connection. The best choice is to use a wired connection. As the 10micron mounts also support a serial line, please be reminded MountWizzard4 does not! Many of the features are handled on the mount computer itself and MountWizzard4 does the GUI frontend for the user by using the command protocol provided by 10micron.



The basic idea is that MountWizzard4 will try to generate “digital twin” for the mount. All parameter changes for the mount will be sent to it and changes of it’s state are polled to make status visible in MountWizzard4. Therefore regular polling of data is needed.

Overview

Beside this documentation there is a youtube channel available with descriptions, previews, explanations:

https://www.youtube.com/channel/UCJD-5qdLEcBTCugltqw1hXA

For full operation MW4 supports several frameworks: INDI / INDIGO, ASCOM, Alpaca and in addition Sequence Generator Pro and N.I.N.A. as camera device.

Known limitations

MountWizzard4 does support python 3.10 - 3.12 right now. The reason for that is the lack of precompiled packages. Some features are limited to windows as they need the original 10micron updater program for execution.

On windows please check if you are working in a 32bit or 64bit environment. You need to choose the ASCOM setup (drivers etc.) and the python install accordingly.

If you are using the 10micron updater features on windows, MountWizzard4 remote controls the updater application. Please do not interrupt this automation.

Reporting issues

To have an eye on your setup here are some topics which you could check:

- Mount connection available and stable. Wifi might have performance problems. Look for right network settings in mount and local setup.
- Good counter check is review settings, status bars, message window if something is going wrong.

To improve quality and usability any feedback is highly welcome! To maintain a good transparency and professional work for my, please respect the following recommendations how to feed back.

Note

Please report issues / bugs here:

https://github.com/mworion/MountWizzard4/issues .

And if you have feature requests discussions or for all other topics of interest there is a good place to start here:

https://github.com/mworion/MountWizzard4/discussions

In case of a bug report please have a good description (maybe a screenshot if it's related to GUI) and add the log file(s). Normally you just could drop the log file (or PNG in case of a screen shot) directly to the webpage issues on GitHub. In some cases GitHub does not accept the file format (unfortunately for example FITs files). I this case, please zip them and drop the zipped file. This will work. If you have multiple files, please don't zip them to one file! I need them separated and zipped causes more work.

If changes are made due to a feedback, new releases will have a link to the closed issues on GitHub.

Table of contents

Feature Overview

For being fully operational, MountWizzard4 needs either:

- INDI server(s) (see: https://indilib.org) where your devices are connected to.
- INDIGO server(s) (see: http://www.indigo-astronomy.org) where your devices are connected to.
- ASCOM Alpaca remote server (see: https://ascom-standards.org/FAQs/Index.htm) abstracting your ASCOM devices or devices which speak native ASCOM Alpaca if you want to connect over IP with your environment.
- Running versions of Sequence Generator Pro or N.I.N.A. as frontend for camera device.
- For the core devices there is native ASCOM support (Windows platform only). Please be reminded, that ASCOM has 32bit and 64bit driver implementations and MountWizzard4 could also be installed in 32bit or 64 bit python environment. They could be not be mixed! 32bit python supports only 32bit drivers and vice versa. Normally this should not be an issue...
- In addition an internet connection is used for some services which might be very helpful.

It is recommended to use mount firmware 3.x or later as some of the functions don't work with older firmware versions. It should not be a problem using older versions. A HW pre2012 might also have some issues. MountWizzard4 supports also older firmwares from 2.x onwards, but with limited features and untested.

It is recommended to use mount firmware 2.16 or later as some of the functions don't work with older firmware versions.

Here is an overview of the functionality available in MountWizzard4:

- Many settings and features of the mount can be shown and changed.
- Control movement of the mount as well as tracking speeds.
- Coordinates in J2000 as well as in JNow.
- Virtual keypad

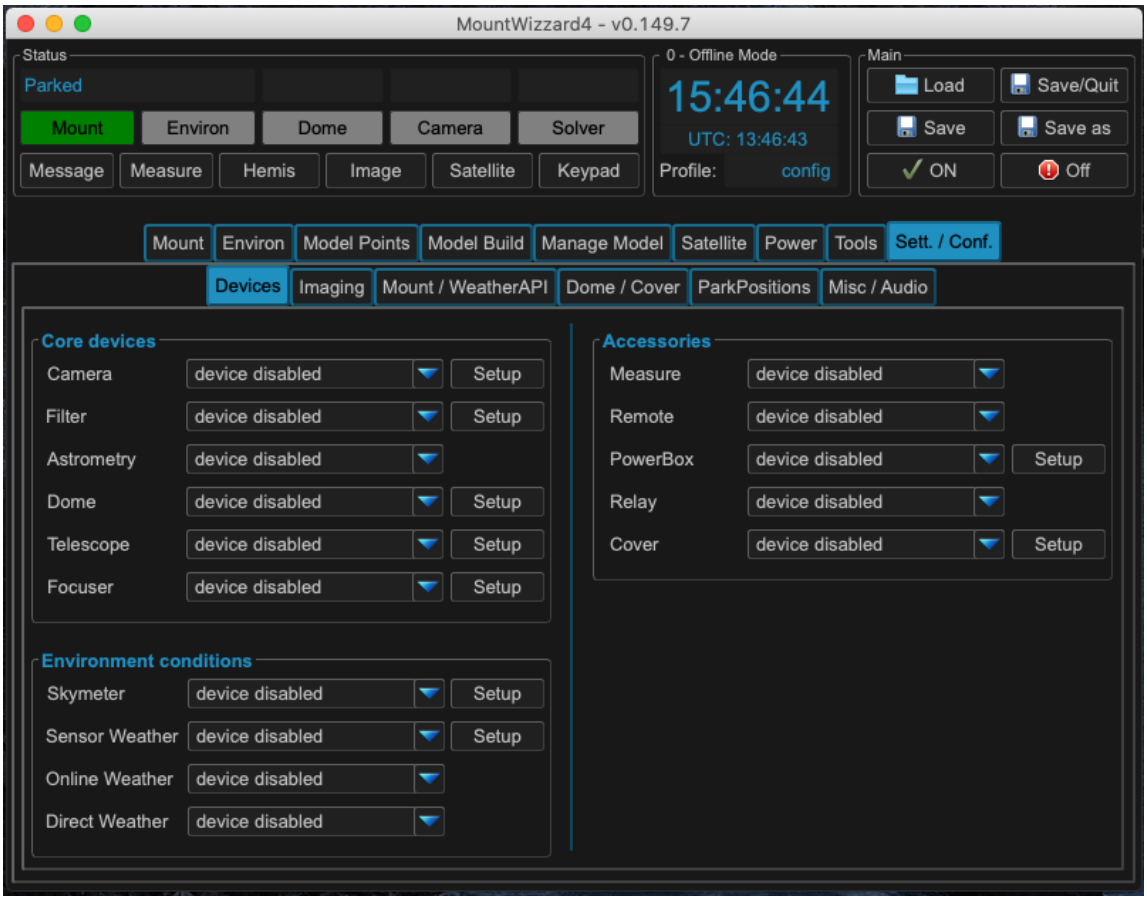
- Model building with different model setups and model generating capabilities. Sorting points for effective slew paths or dome situations.
- Model building is done in parallel threads (imaging, plate solving, slewing) to reduce time.
- Show the actual model and alignment error. Give hints on how to improve the raw polar alignment.
- Model optimisation: deleting points, automatic removing point for target RMS etc.
- Manage models stored in the mount (save, load, delete).
- Dome geometry integration (MountWizzard4 knows about target flip side and slews dome correctly as well as any geometrical constraints).
- Environment data: MountWizzard4 shows data from OpenWeatherMap, ClearOutside, External Sensors like MBox, Stickstation, UniHedronSQR as well as direct linked sensors like MGBox.
- Refraction handling external / internal from the above sources.
- Satellite: searching, displaying, programming, updating tracking.
- Tools: FITS Header renaming, Park positions, etc.
- Remote shutdown of MountWizzard4 and Mount via IP commands.
- Measurements and CSV saving for most environment and mount data
- Imaging: control of connected camera / cooler / filter.
- WOL (wake on LAN) boot for mount. MountWizzard4 catches MAC address automatically on first manual start.
- Audio signals for different events (end slew, finished modeling, alert, etc.)
- Updater for all MountWizzard4 functions.
- Generate / load / save as many profiles as you would like.
- Show alignment stars. Choose and automatically center for polar or orthogonal adjustments.
- Imaging: expose one or N images, auto solve or auto stack these images.
- Imaging: show distortion grid, astrometric calculations (flux, roundness, sharpness)

Configuration

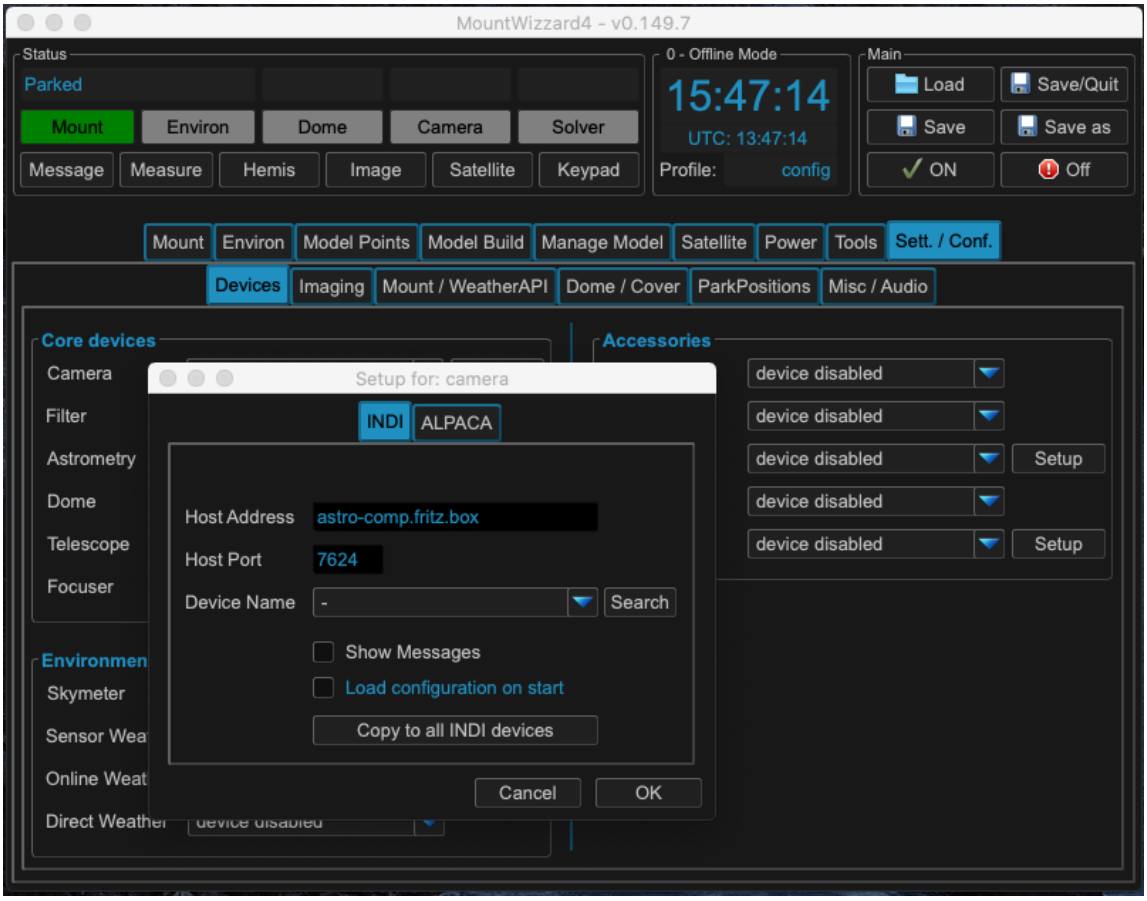
The configuration part of the documentation is divided into several sections, each of which describes a different aspect of the configuration of the observatory setup. There were several aspects of tuning MountWizzard4 to your needs, and this section will help you to understand how to do that. Still there is many more information in tooltips and help texts in the application itself.

Adding the camera

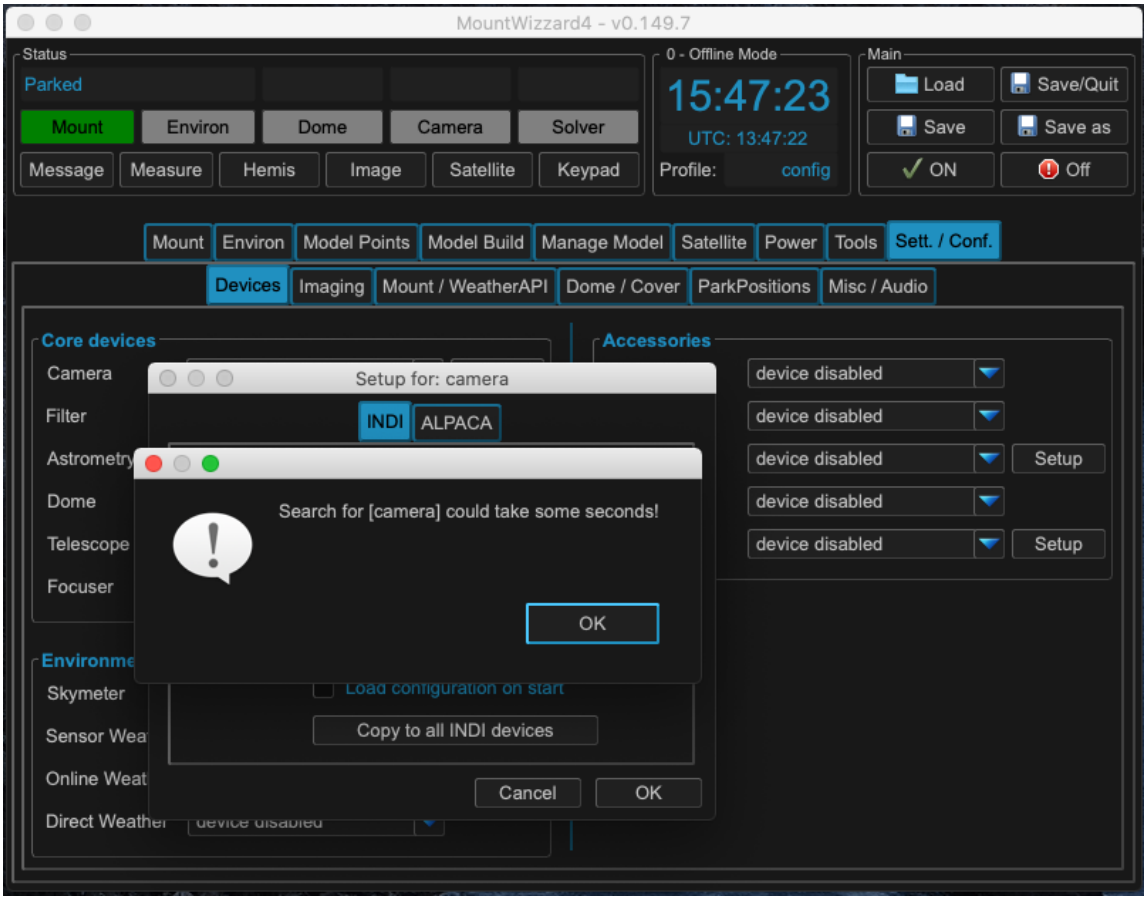
For adding devices like the camera select the Sett. / Conf. tab in the main menu and there select the devices tab. For the following explanation we would like to connect a camera, a filter and adding a link to the mount as well. The mount link is only used for reading the parameters of the mount driver of your setup (e.g. focal length, aperture).



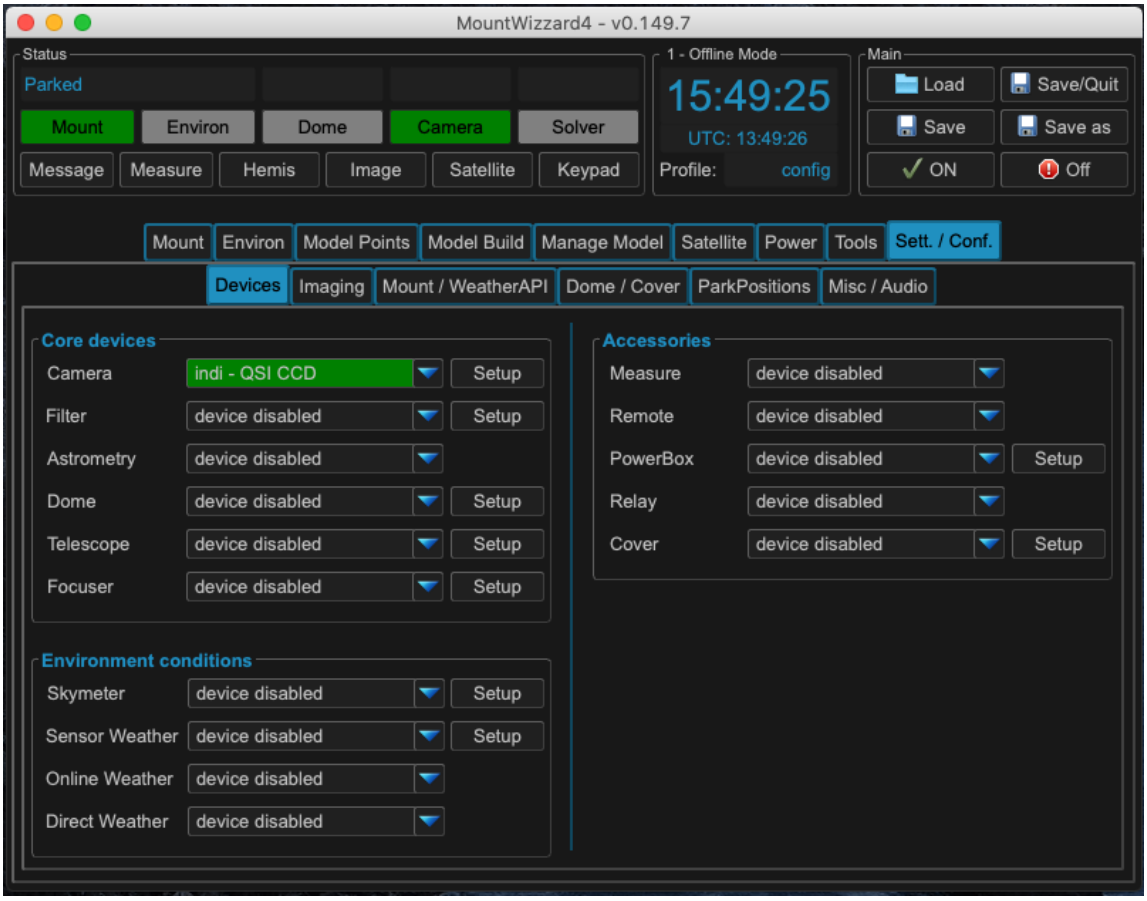
In core devices select Setup für Camera. A popup shows up. Please fill in the ip address of the INDI or ALPACA server, where your devices are connected to.



If you are using INDI, you search for INDI camera devices by clicking on the search button. Once you finished searching by pressing OK button, a list of available devices will be populated. From the list choose the device and finish the setup with OK button.



The selected camera will be highlighted green in the drop down menu and in the status as well. Once you configured the camera, the selection list will be stored for later use. If you want to disable the camera, please select device disabled in the menu.



MountWizzard4 will now try to connect to the device and show green light whenever a connection is established.

Do the steps 1 - 5 for all devices you need to configure. All configuration are save when leaving MountWizzard4 with Save/Quit button or just when saving the profile. You can add or change any config later on at any time.

Using SGPro or N.I.N.A. as camera

In addition to the standard frameworks to interface to devices, MountWizzard4 could use Sequence Generator Pro (SGPro) an Nighttime Imaging (N.I.N.A.) as a camera driver for devices attached to them. Unfortunately they support only a minimum set of devices through their API and only with a limited feature set. But the provided basic API is sufficient to do the modeling job.

Note

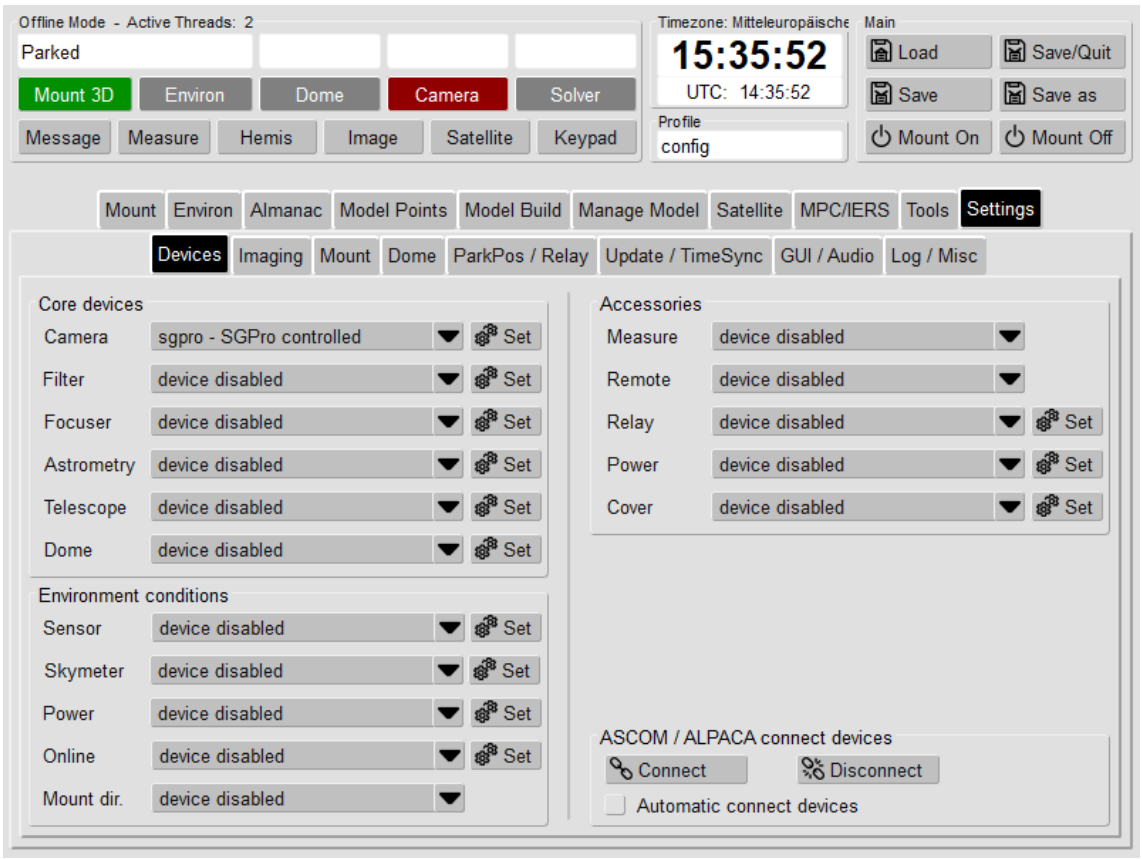
MountWizzard4 uses all necessary data from the FITS of the images taken by the external apps. Please make sure, that the FITS header contains this information, especially the focal length, the pixel size. Otherwise plate solving will fail. As both applications do not transfer their images to MountWizzard4, you have to ensure that the FITS files are stored on your local disk and MountWizzard4 has access.

Basically MountWizzard4 interface these apps and let them control the devices. Selection and connecting the devices have to be done manually in the regarding application. MountWizzard4 just recognizes of a device is connected or not and if connected uses it as is.

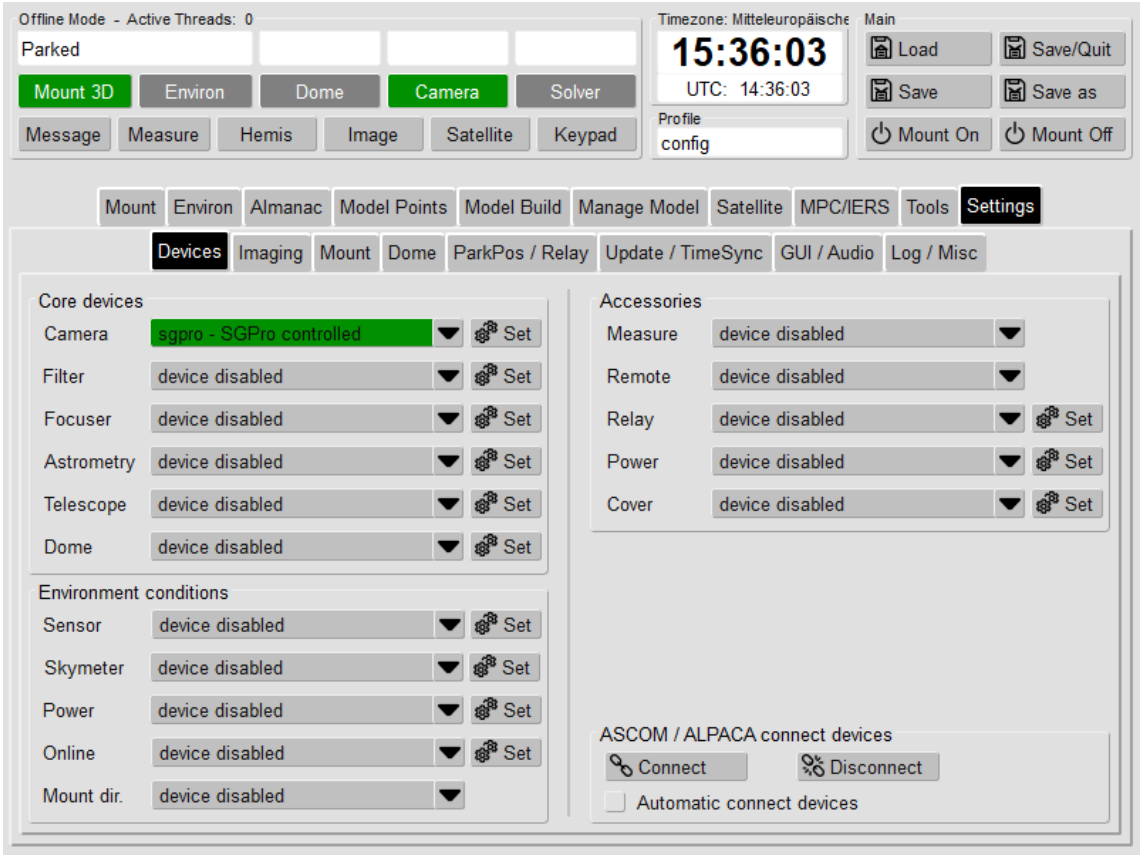
SGPro controlled mode

First in camera driver setup you choose to use the SGPro controlled mode:

You recognize the setting in device tab:



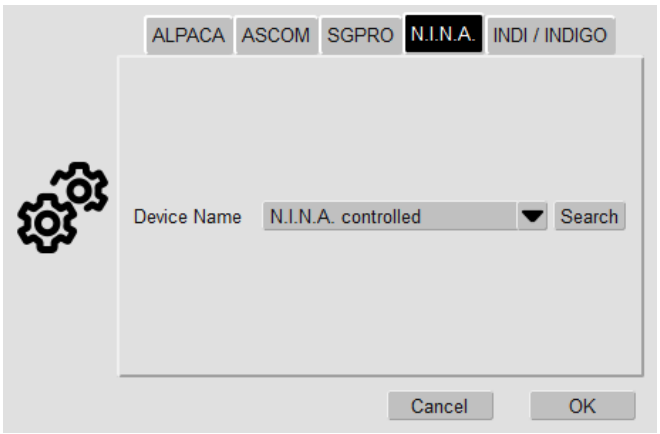
Once you connect a camera in SGPro



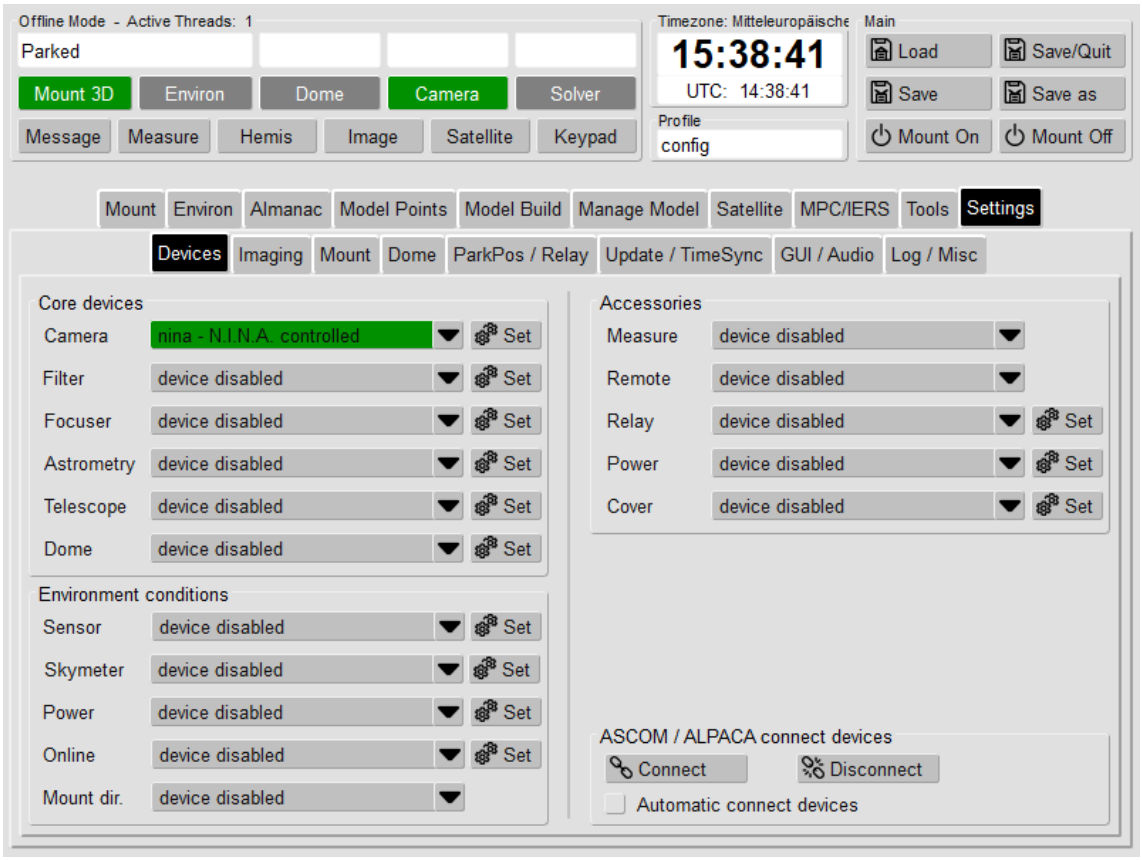
the status in MountWizzard4 will change to connected as well.

N.I.N.A. controlled mode

First in camera driver setup you choose to use the N.I.N.A. controlled mode, you recognize the setting in device tab:



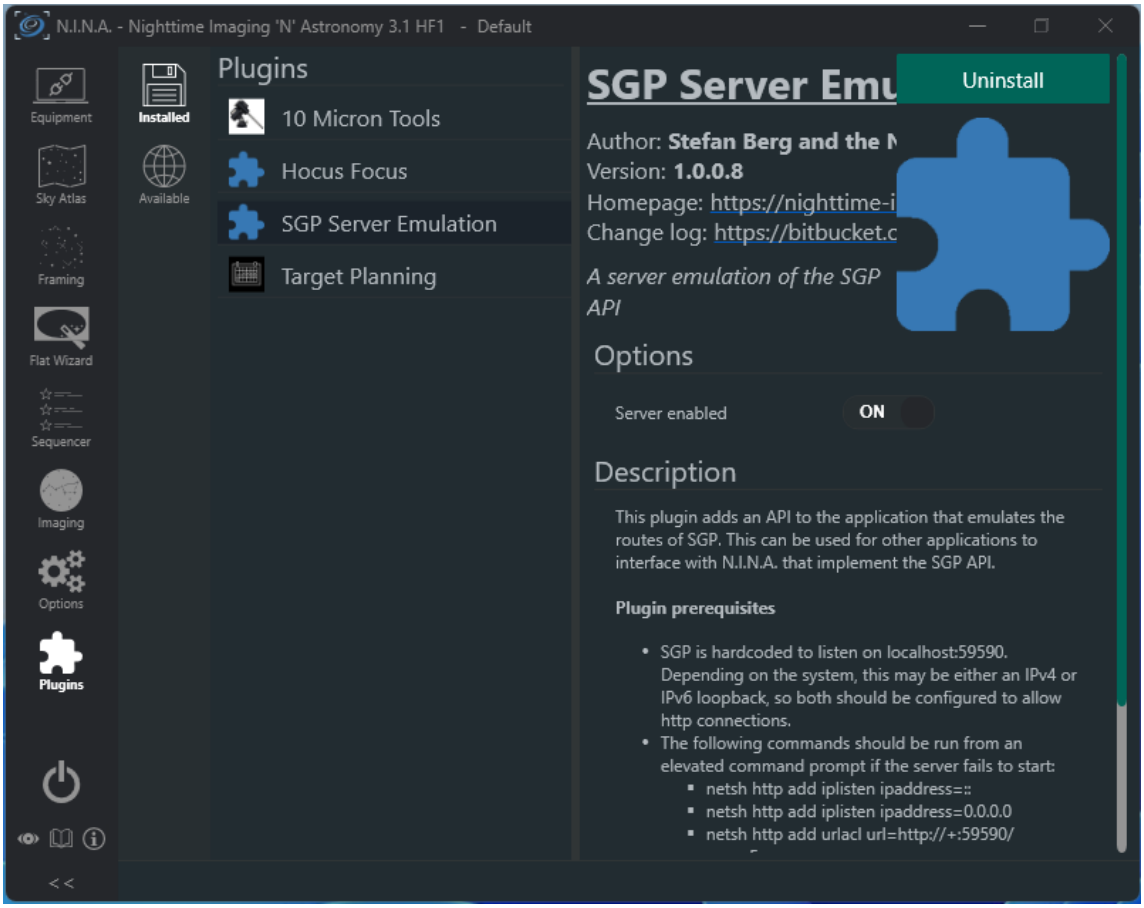
Once you connect a camera in N.I.N.A.



the status in MountWizzard4 will change to connected as well.

Preparation for using NINA 3.x as remote camera

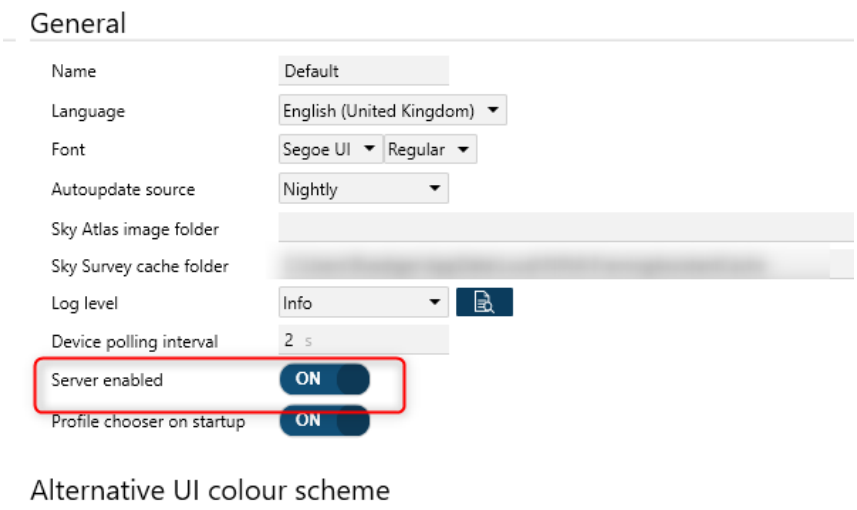
N.I.N.A. 3.x realizes this feature in a separate plugin. The plugin is called SGPro server emulator:



Pleas install this plugin first and enable server mode.

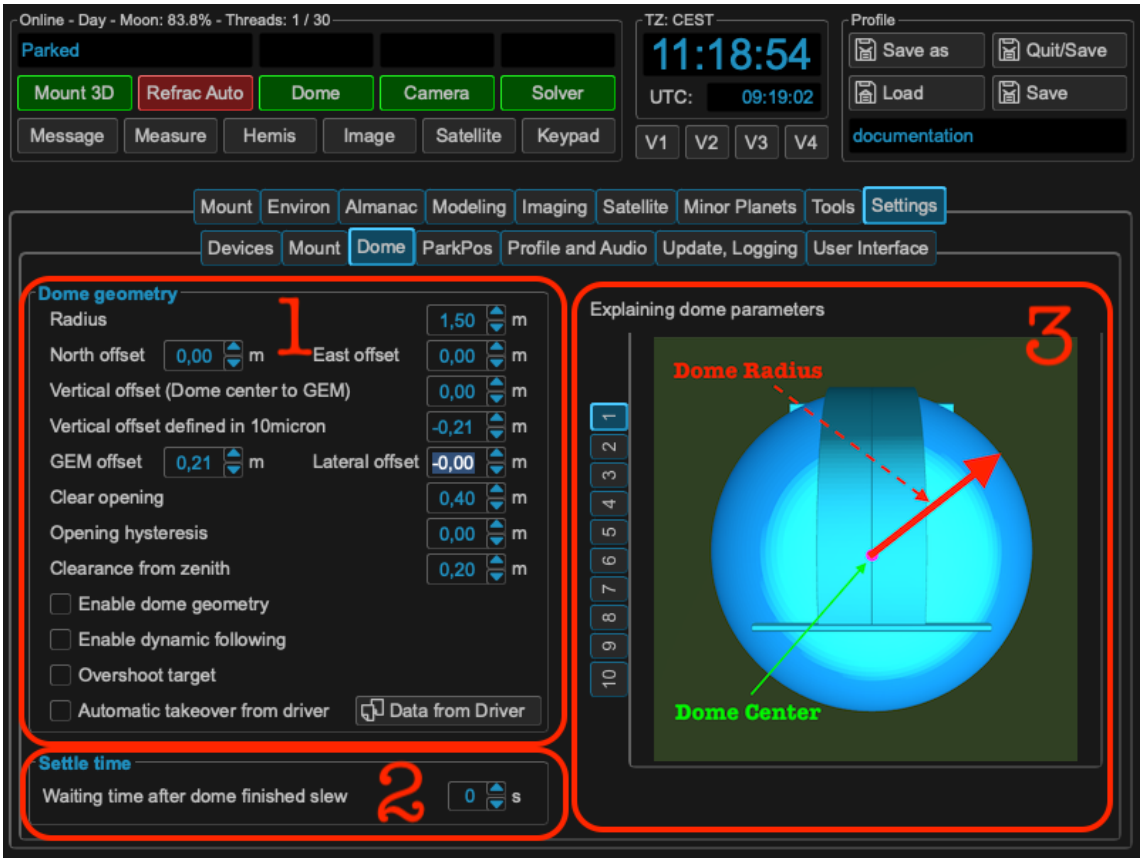
Preparation for using NINA 2.x as remote camera

In N.I.N.A. 2.x you have to enable server mode directly in main program:



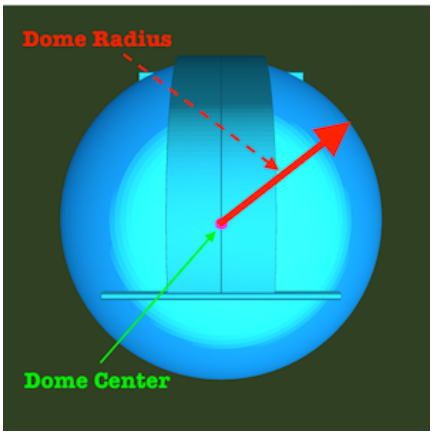
Dome geometry

If you using dome, you should set up the parameters to get a good performance of the slew calculations. To use the calculations, please check the “Enable dome geometry offset calculations”. If checked, MountWizzard4 will calculate the correct dome azimuth in relation to the desired target of the mount. It take the pierside already into account, so you could slew mount and dome at the same time.

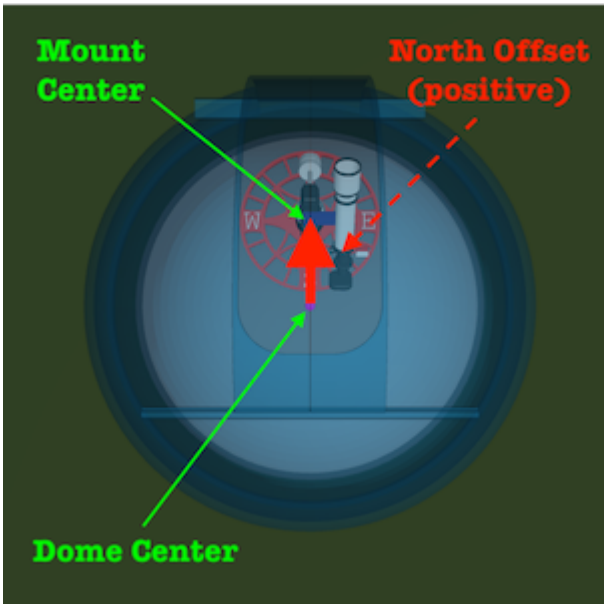


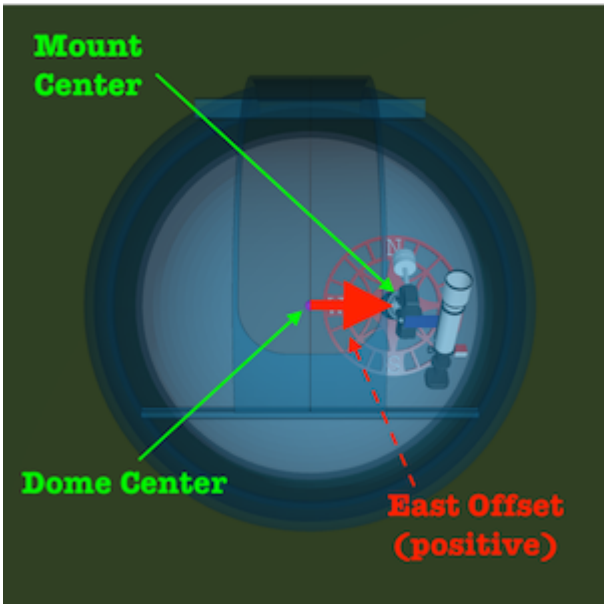
Area 1: dome geometry parameters

I this section the geometry parameters of the dome and mount in relation to the dome are set. The dome center - the center point of the dome hemisphere - is always the reference point from where the measurements are taken. All measure are in meters. It starts with the radius

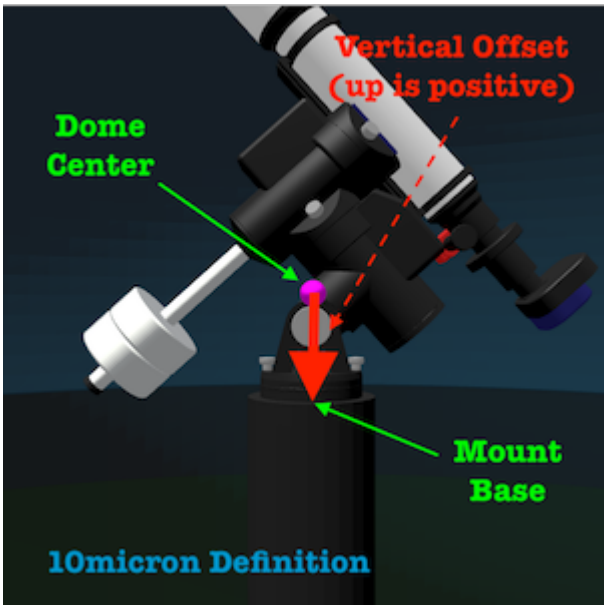


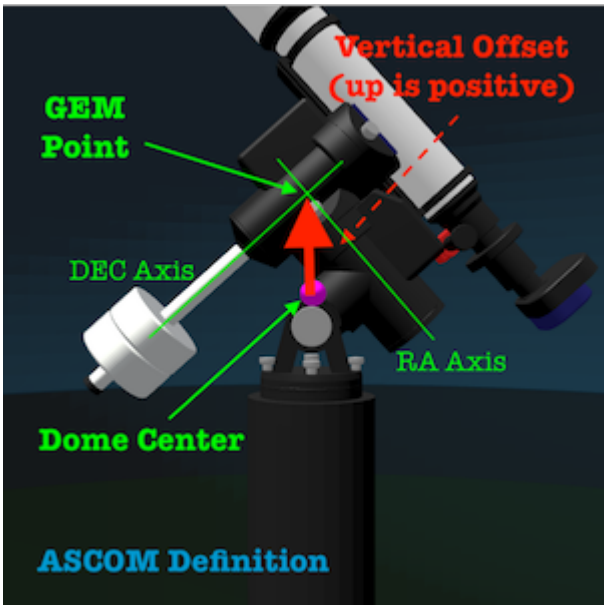
followed by the displacement of the mount in north and east direction. A displacement to the north is treated positive as well as a displacement to the east.



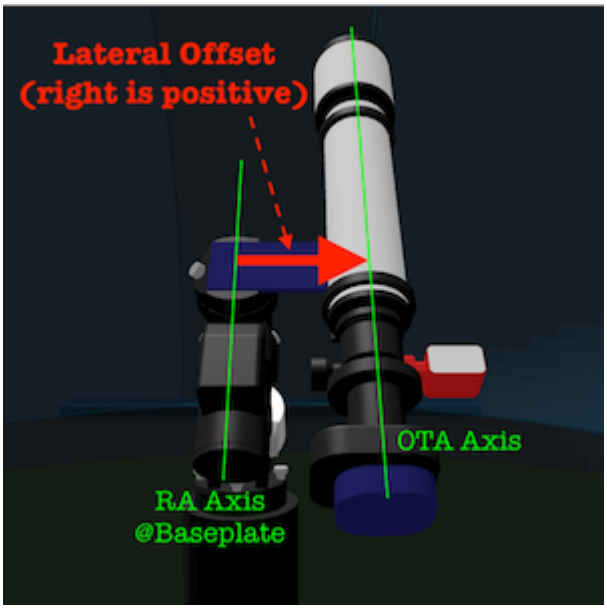


For the vertical displacement you have to take car about different definitions of how to measure this value. A common way many ASCOM drivers do is measuring the distance between the dome center and the GEM point of the mount. The GEM point of the mount is the intersection of the RA and DEC axis. Ideally you will have that point fitting to the dome center. In the 10micron handbook, the measurement is done between the dome center and the base of the mount. This value is mostly negative, because the mount base level is on lower height than the dome center.





If you have mor than one telescope mounted, you will experience a lateral displacement of the OTA you would like to use for reference. The measurement is taken from base plate of the mount to the OTA axis. Displacement to the right (east) is treated positive.



Area 2: dome settle time

With the dome settle time you define the waiting time from reaching the target azimuth until the dome emits slew finished. This could be used for avoiding mechanical influence on the mount. This settle time in handled in MountWizzard4 only. If you could enter a settle time in your dome driver etc. the times will add up.

Area 3: dome geometry help

As geometry parameters are difficult to remember, MountWizzard4 will show you a short explanation picture of the parameter you are currently changing.

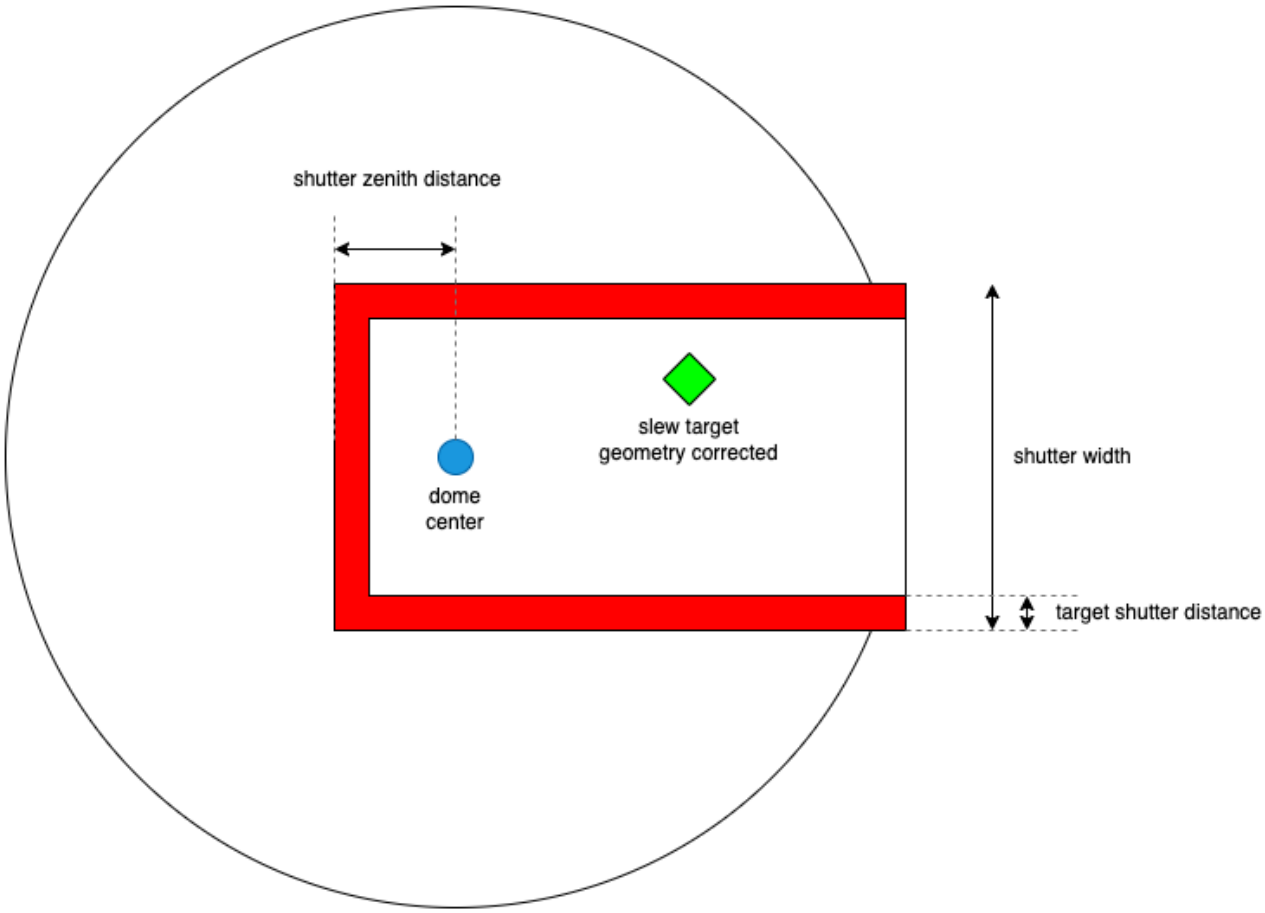
Dome dynamic follow

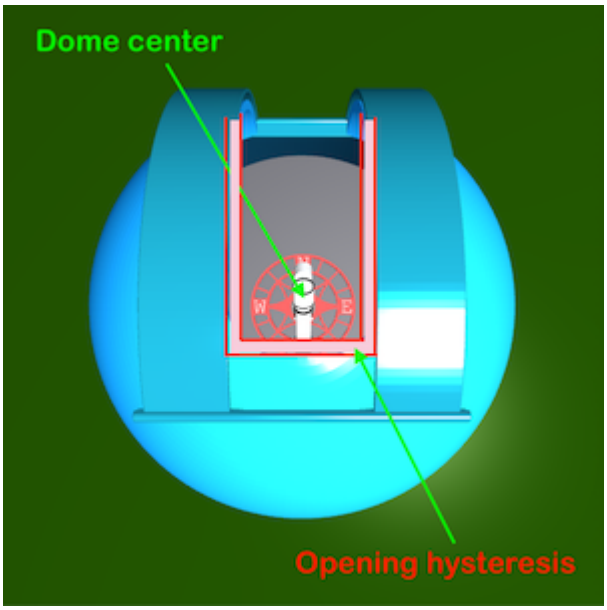
For satellite tracking moving the dome adequately is very important. If you have following on your dome controller, you might have set a hysteresis to avoid continuous dome movements. These values normally are between 1 and 3 degrees. MountWizzard4 could handle this different. If you set your dome geometry correctly, you could add parameters for target shutter distance and shutter zenith distance (better terminus ???). With these parameters MountWizzard4 will calculate if the viewing spot of you telescope moves with the next slew command in the “red zone”. If so, the dome azimuth will be corrected and the dome will slew to it’s new centered position. If there is still a reasonable distance, your telescope still could view your target and no dome movement is necessary.

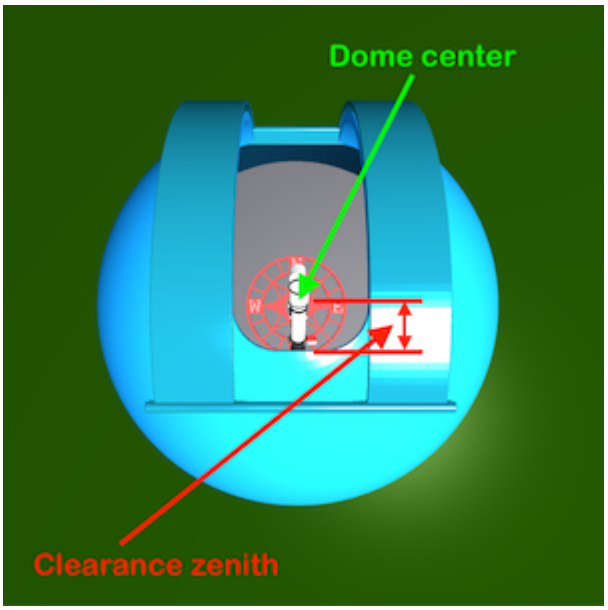
Hint

See also on youtube: https://youtu.be/ZmDz-rtvFzc

Top view of your dome setup:

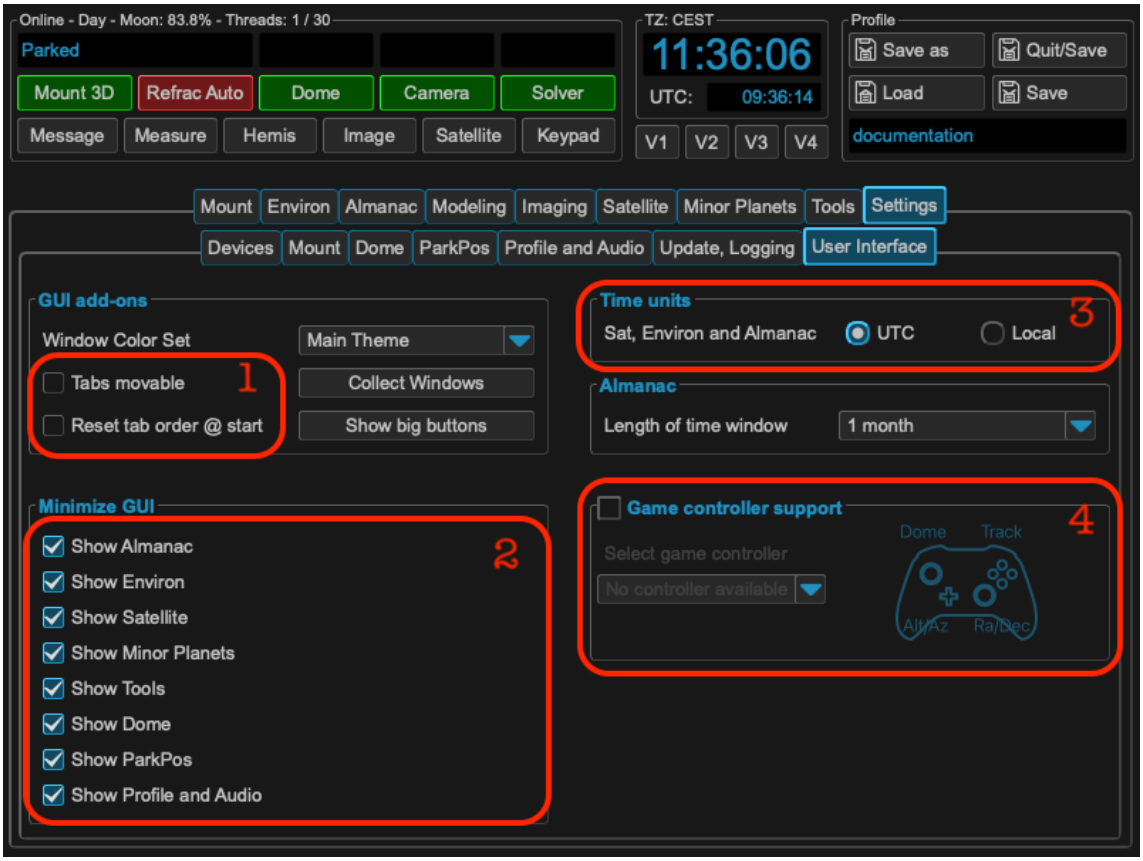






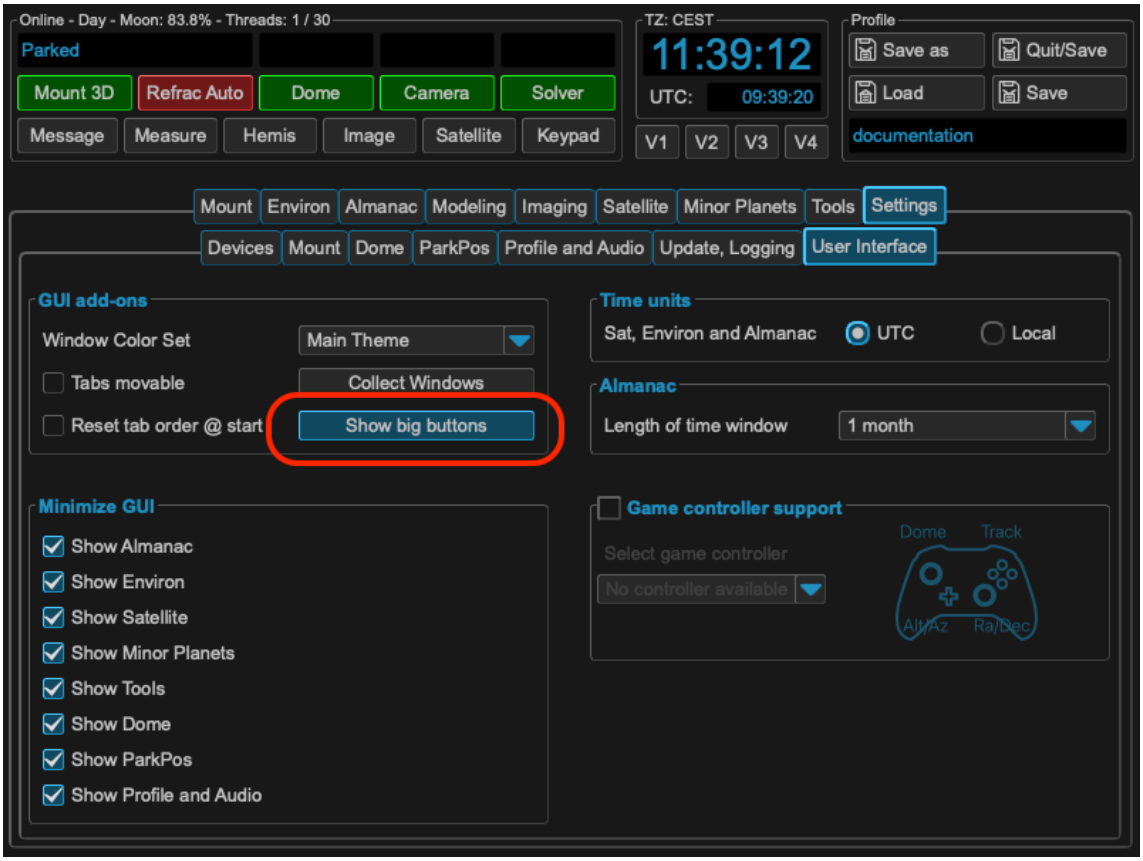
GUI adaptations

The GUI of MountWizzard4 could be configured to your needs in many ways. You could reorder your tabs (1) or disable not needed menu parts to get a much simpler interface (2)

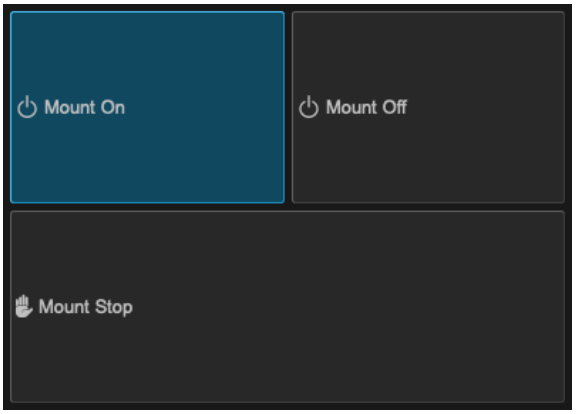


Depending on your usage, all times could be display in local time or in UTC (3). In addition for some functions you could add a game controller to move your mount (4).

If you ar using a touch based system, the main GUI elements could be shown in a separate window.



You open this window by clicking on the “Show big buttons” button.

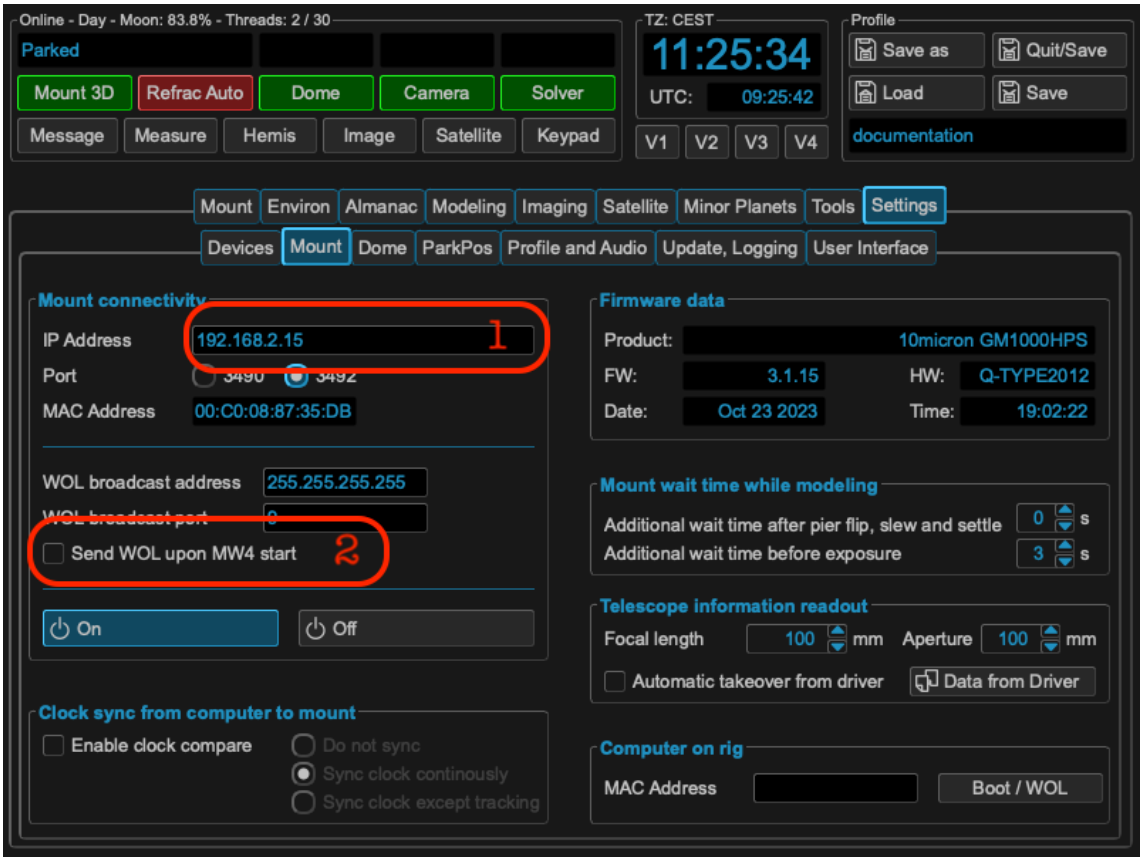


Imaging setup

Mount setup

Basic connectivity

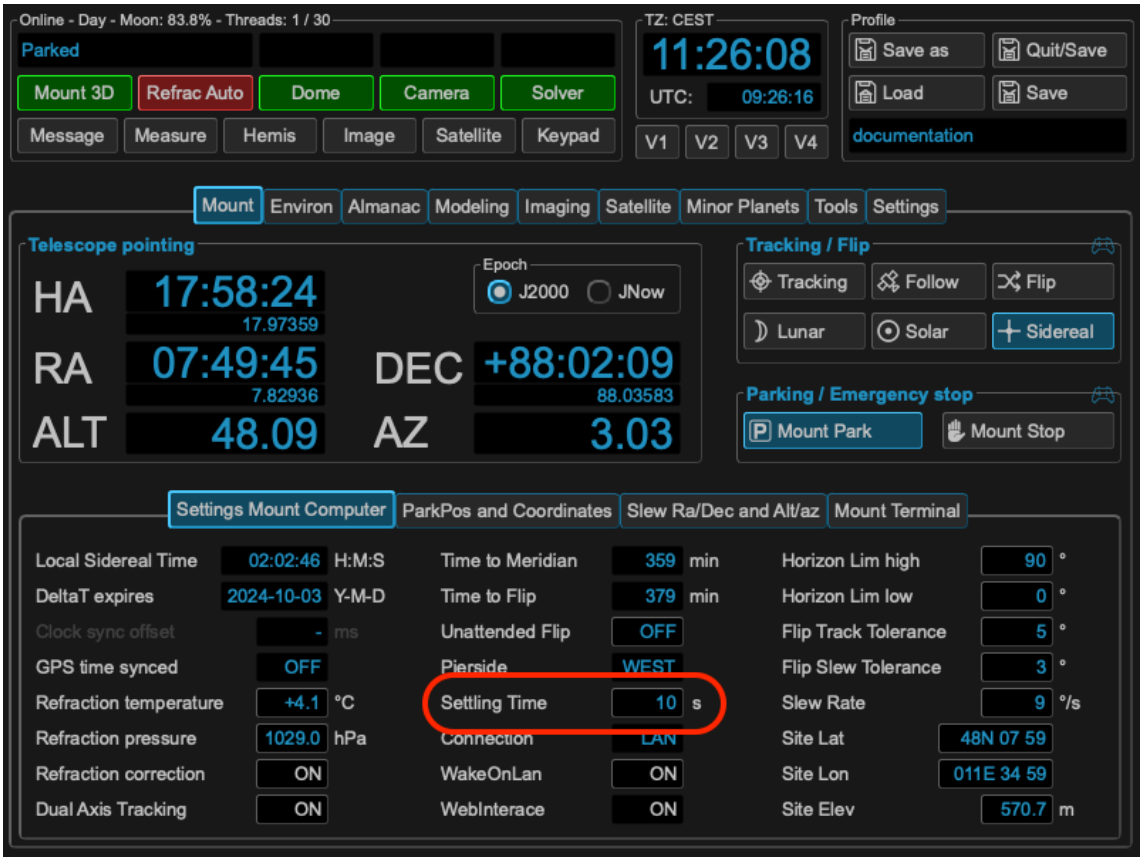
With the first start you have a clean installation. First step should be to enable the mount connectivity. In the settings tab goto Mount and enter under Mount connectivity the IP address of you mount (1). Boot the mount manually and wait until the mount computer is ready.



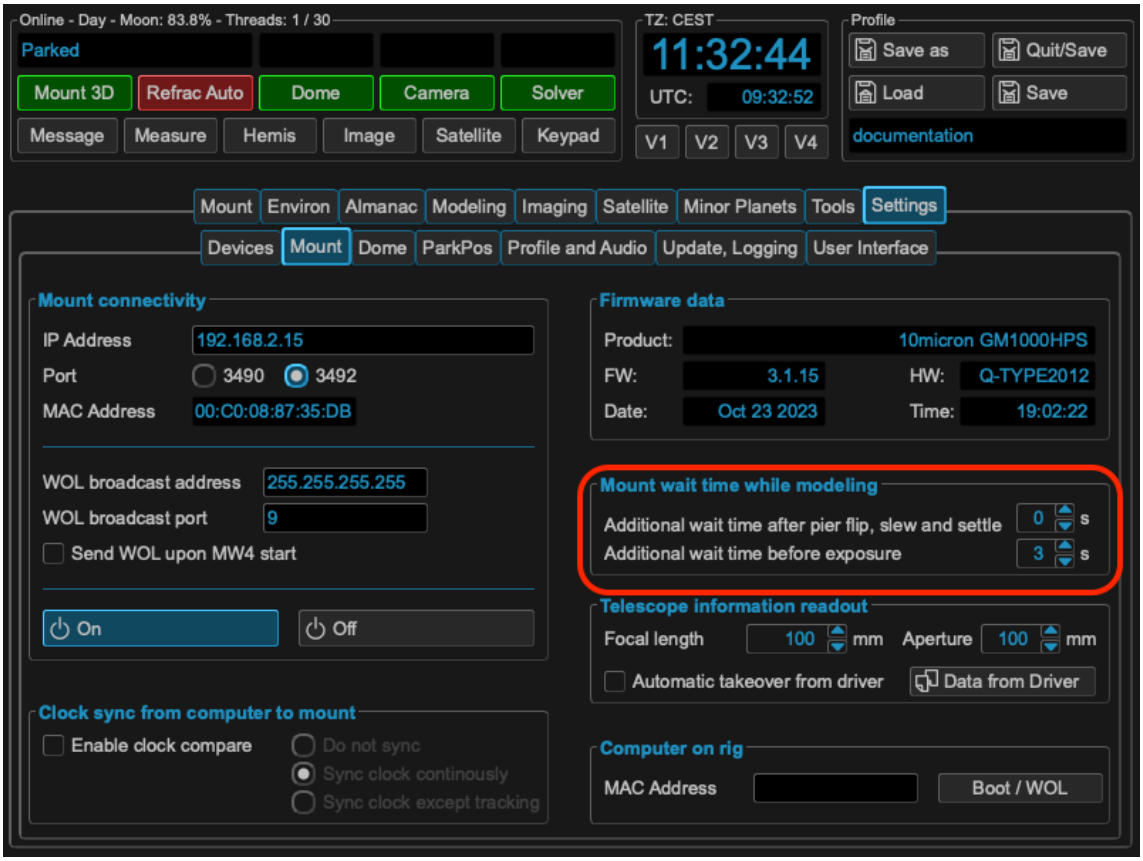
MountWizzard4 will show for Mount connection a green light and enters the MAC Address for remote boot via wake-on-lan (WOL). You should select WakeOnLan on startup in the menu (2). All configuration are save when leaving MountWizzard4 with Save/Quit button or just when saving the profile. You can add or change any config later on at any time.

Settling Time / Waiting Time

To accommodate several different use cases MountWizzard4 implements additional waiting times to the core settling time, which is implemented and user directly from the mount computer. The following image shows the setting of this parameter, which could be also set and altered through 10micron tools.



This settling time is valid for all slews and movements of your mount once set. Please have a look to the 10micron spec where this behaviour has to be taken into account. Nevertheless for the modeling part MountWizzard4 add two more parameters as the modeling process need heavy movement of the mount. Therefore MountWizzard4 call these parameters not settling time but waiting time. These parameters could be set under the mount parameters:



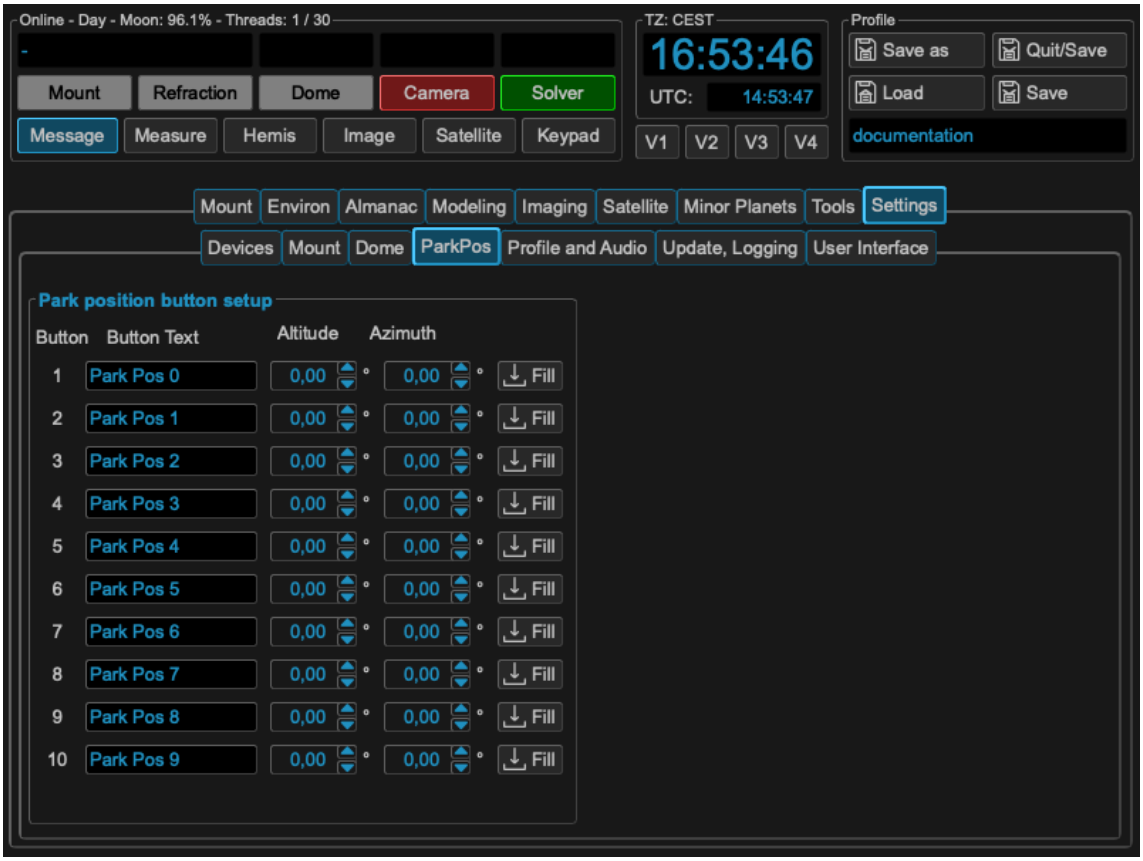
The working principle is as follows: MountWizzard4 initiates a slew. This command is run by the mount computer and takes the internal settling time into account. This means after the mount came to physical stop, the mount computer will send the signal slew finished after this time period (upper image). This is the case in all used cases and will applied also during modeling process.

For the modeling process MountWizzard4 **adds** a waiting time before moving on after slew, which means waiting the addition set time before starting a next exposure (you know that MountWizzard4 runs asynchron for slew, expose and plate solve to improve speed). The wait is only applied during the modeling process.

Furthermore MountWizzard4 will differentiate if the mount starts and stops on the same pierside or if the was a meridian flip of the mount. For both cases you could set the waiting time.

Park Positions

MountWizzard4 supports setup and definition of several park positions of the mount. This is useful for mounts which are used in different locations and where the user wants to store the park position for each location.



Terrain image & horizon mask

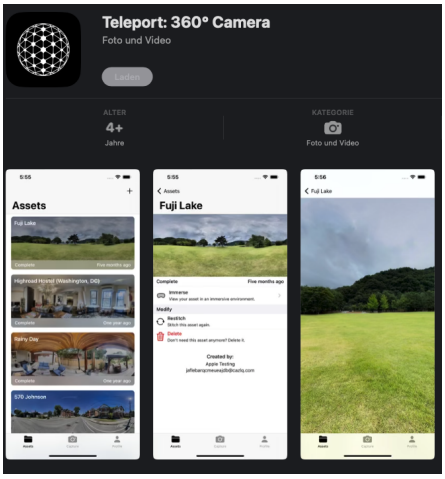
For an optimal use of model points, slew path etc. it is necessary to define a horizon mask for the use in MountWizzard4. There are several way to define and use a mask based on your actual environment

Hint

See also on youtube: https://youtu.be/EqDNNMNMS0w

Using terrain image from Teleport: 360 app

Please use Teleport: 360 app to generate a 360 degrees image around your mount rig. You can save as a .jpeg then rename it to terrain.jpg. I then had to adjust the azimuth paying attention to high points using compass readings.



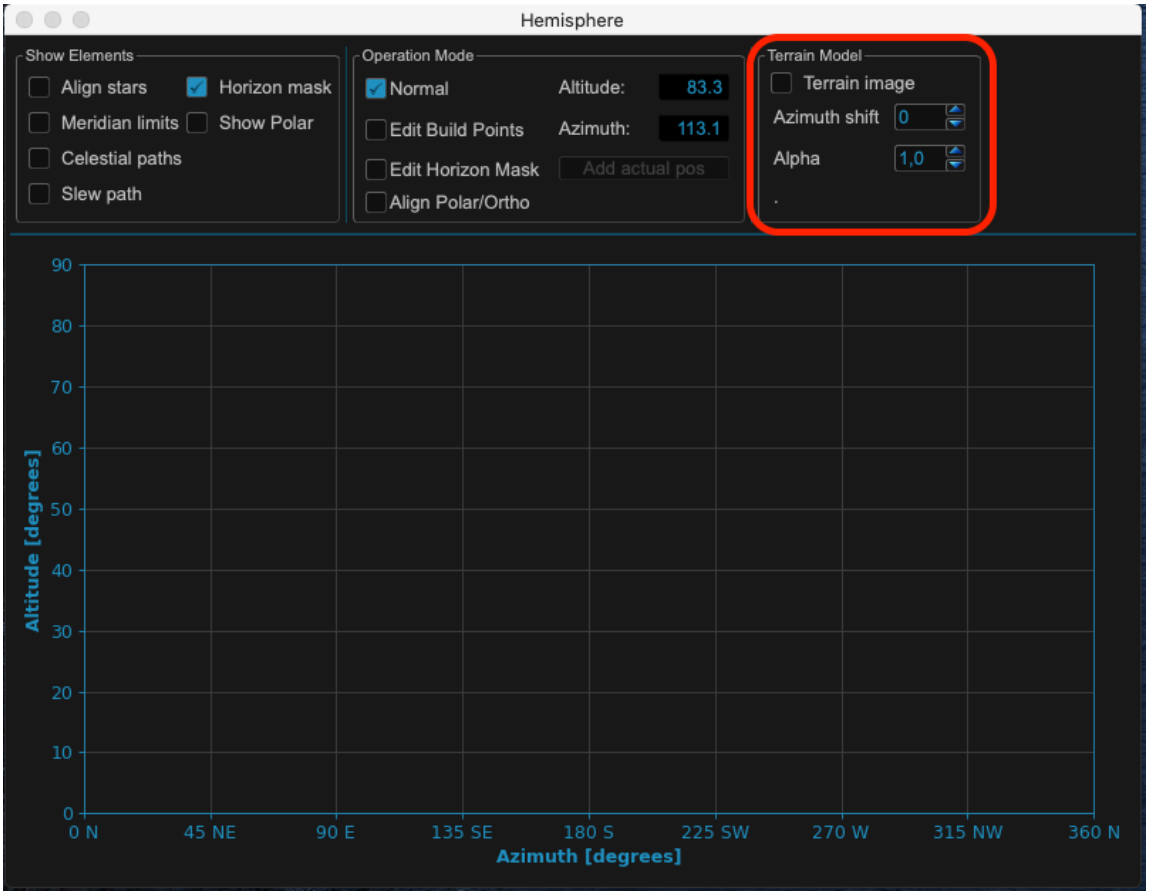
If you finished the image and transferred it to you computer it might look like:

Please put this image to you config dir in your MountWizzard4 working folder. The file format output from streetview is 4096 X 2048 pixel color as JPG. The horizon for altitude 0 degree cuts the image in an upper and lower half. MountWizzard4 uses only the upper half of the image as it expects the image to be taken at the height of the horizon line.

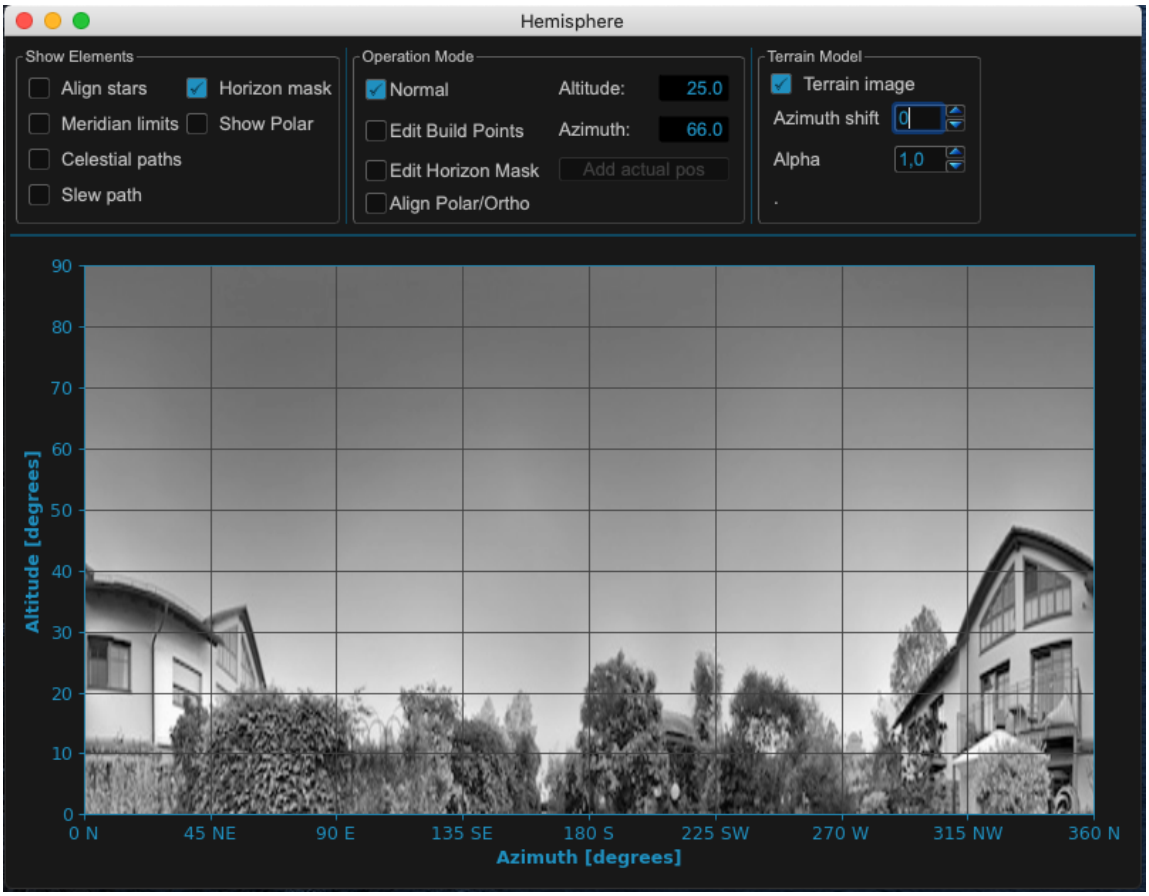
Note

The image file has to be named to: **terrain.jpg** and should be in JPG format.

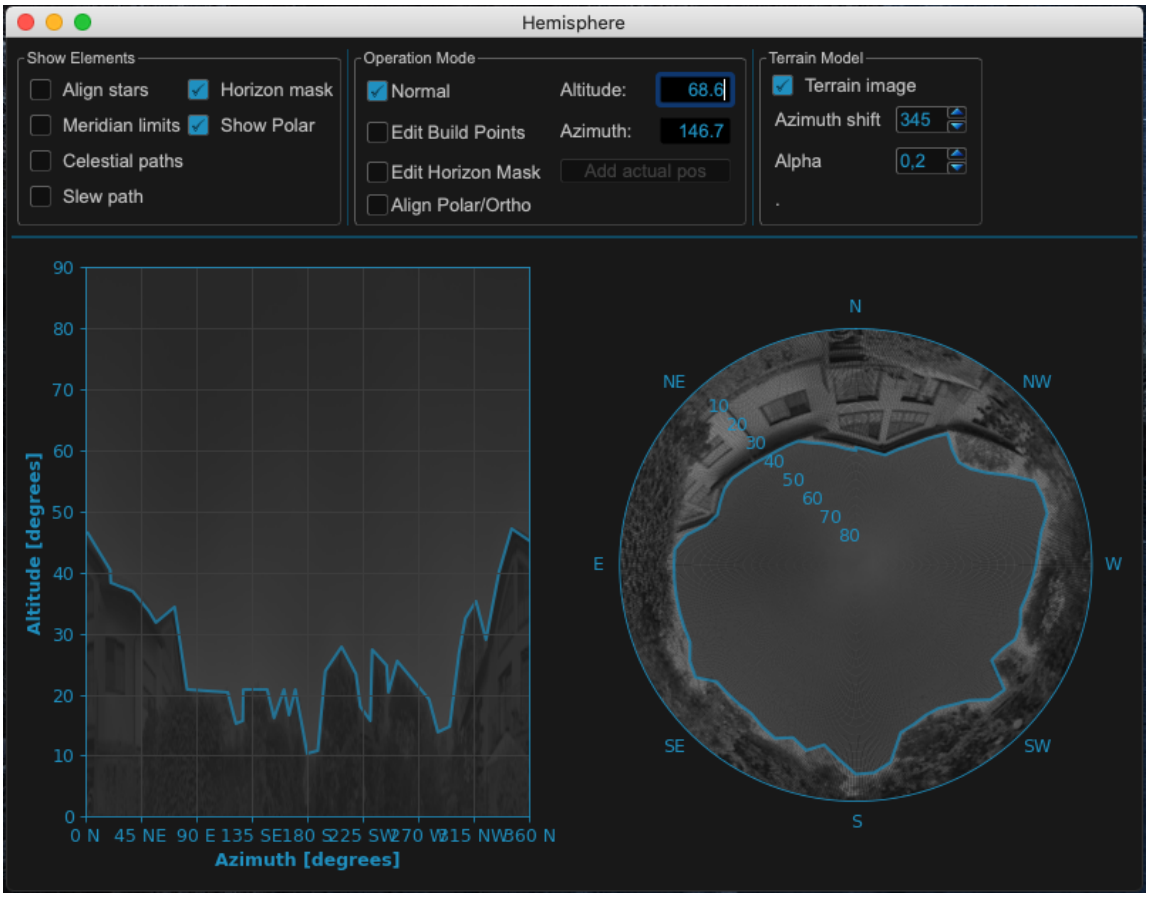
MountWizzard4 will use the image as greyscale image. You could play around how you prepare the image before you copy it to the config directory. Good ideas of improving the image are: make the sky transparent, equalize gamma / lightness settings to avoid highlights in the image, etc. Once you open the hemisphere window you see the setups for the terrain background.



After **use terrain** is checked and a terrain.jpg image is available in config directory, the image will be shown as background of hemisphere.



The image will also show up in polar diagram!



Two adjustments could be changed to make the image fit for use: Most important the azimuth adjustment.

- You shift the image by a number of degrees (0 - 359) to get your image fit to the cardinal points of the hemisphere.
- You could change the alpha channel of the image to get a nice view on your screen.

Then you could add the horizon mask quite easily:

Note

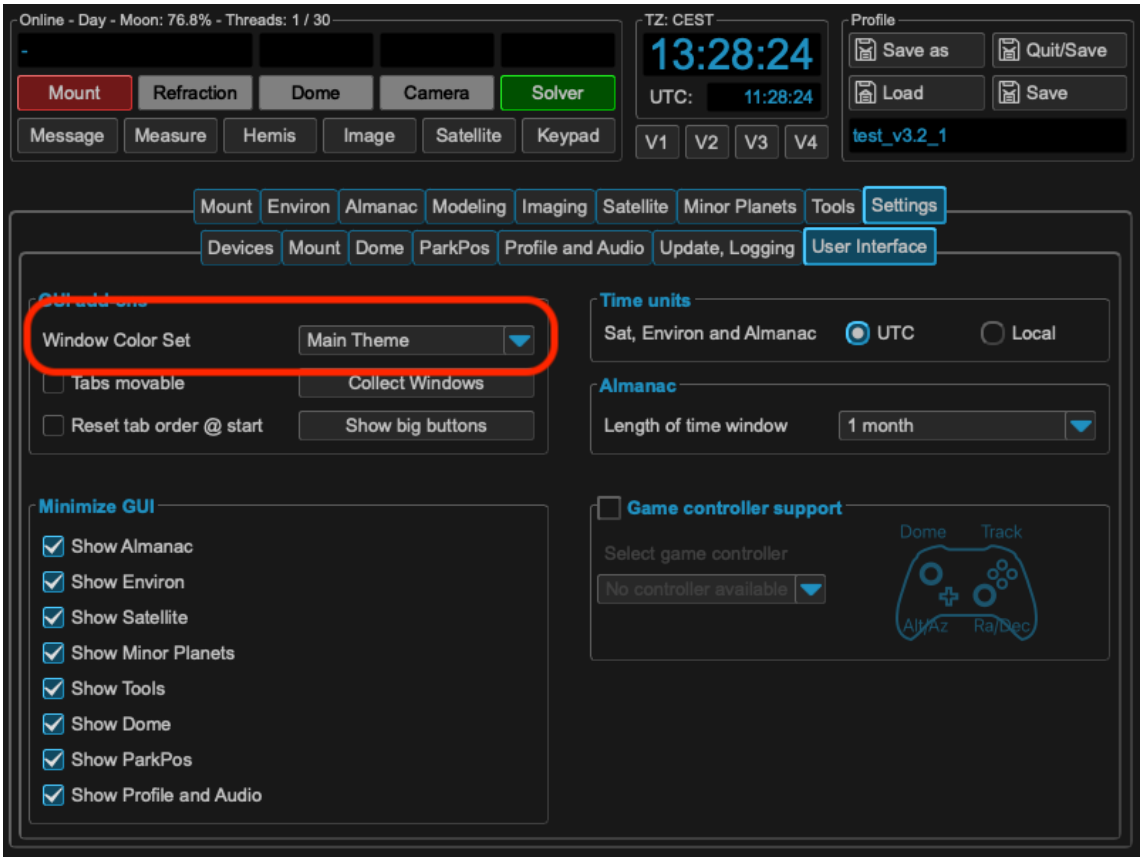
The horizon line (altitude = 0) is set within streetview. Please try to shoot the image at the height of your mount to make it fit. Still there might be some deviations from “real life”.

Warning

All calculations which respect a horizon mask reference to a given mask not the image itself. So even if you have an image set, you need to define horizon mask points!

Themes selection

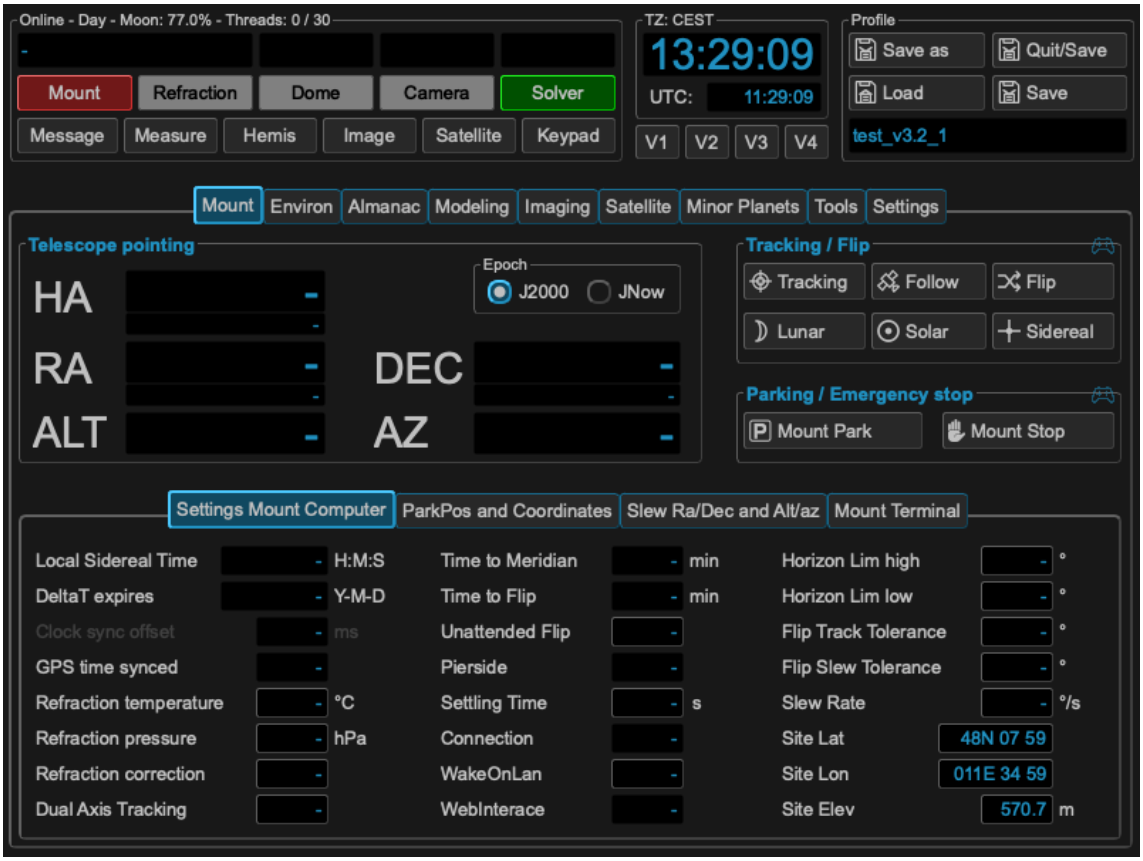
MountWizzard4 supports now 3 different color sets. These could be selected in settings:



and will be persisted after closing the application. They could be changed during runtime.

Main theme

It the theme which was present from the beginning on.

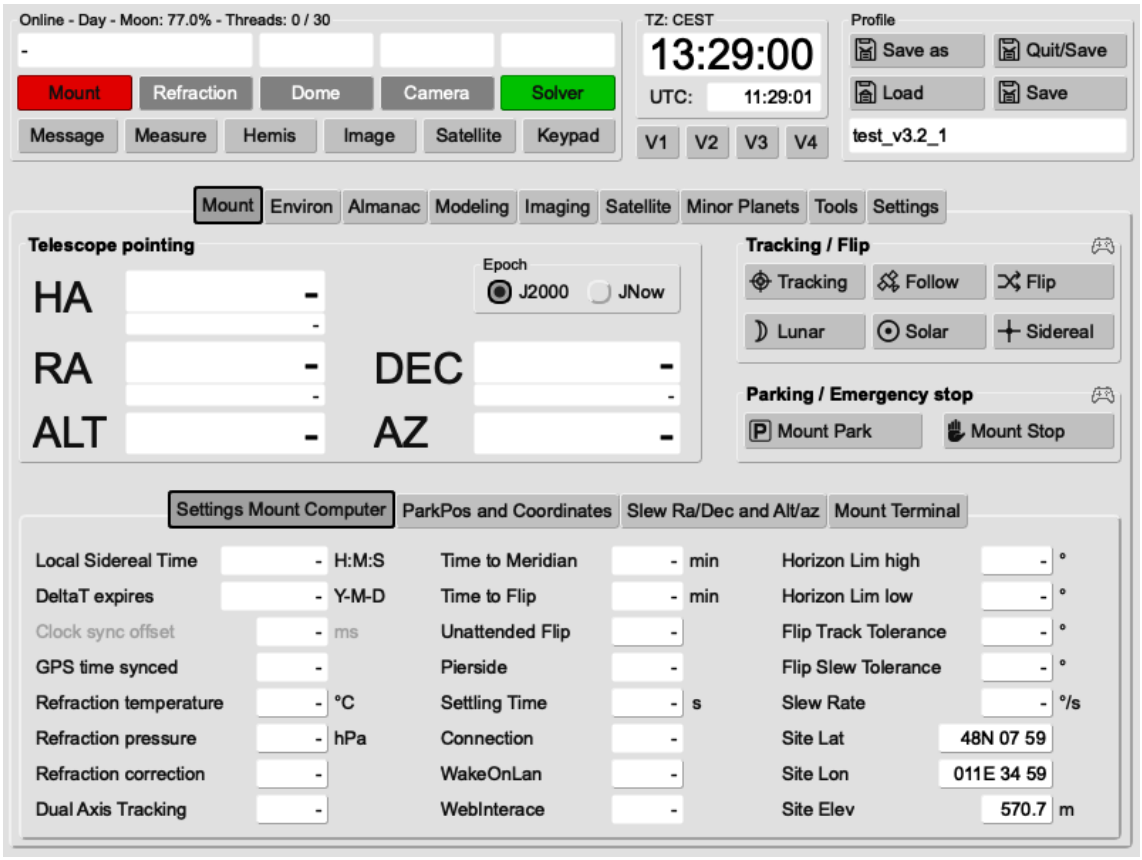


Red night theme

Getting a nor dim color set for those, who use the application in the field.

Light day theme

For those ones who sit more inside and like lighter themes as well, but need not to take care about brighter coloring.

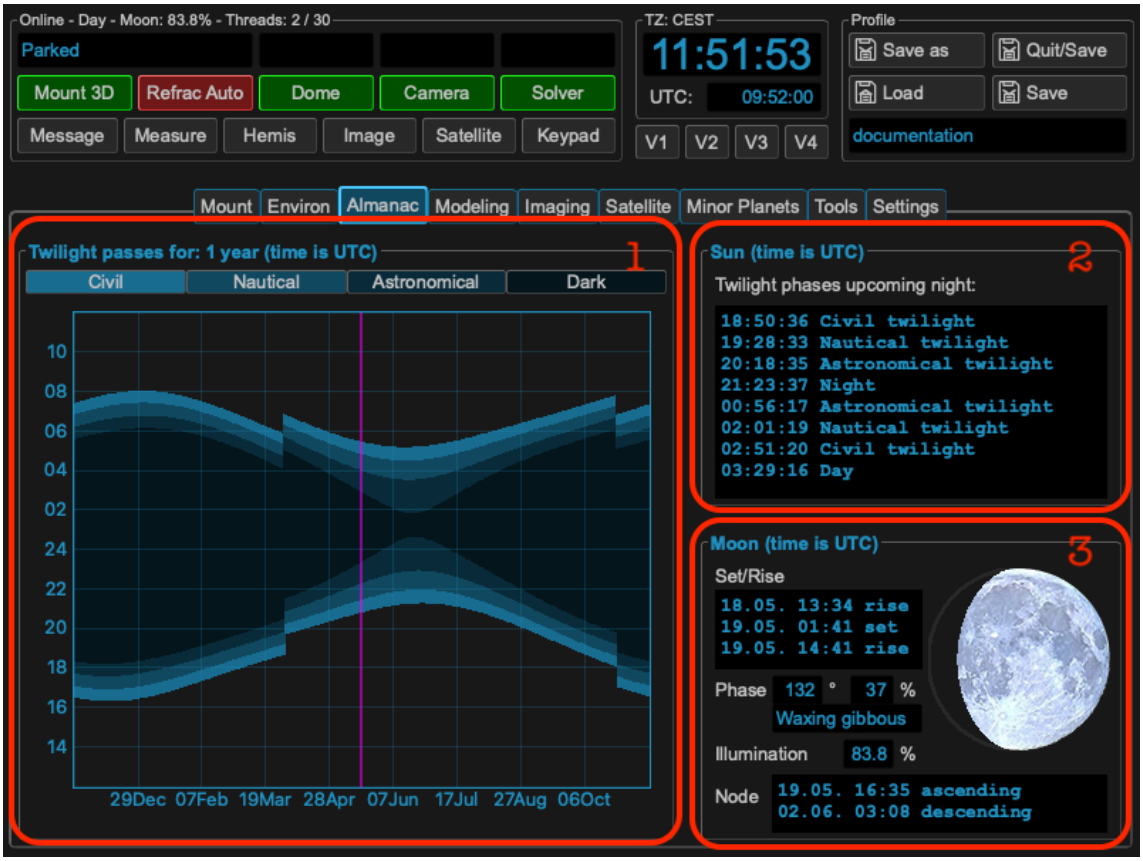


Workflows and functions

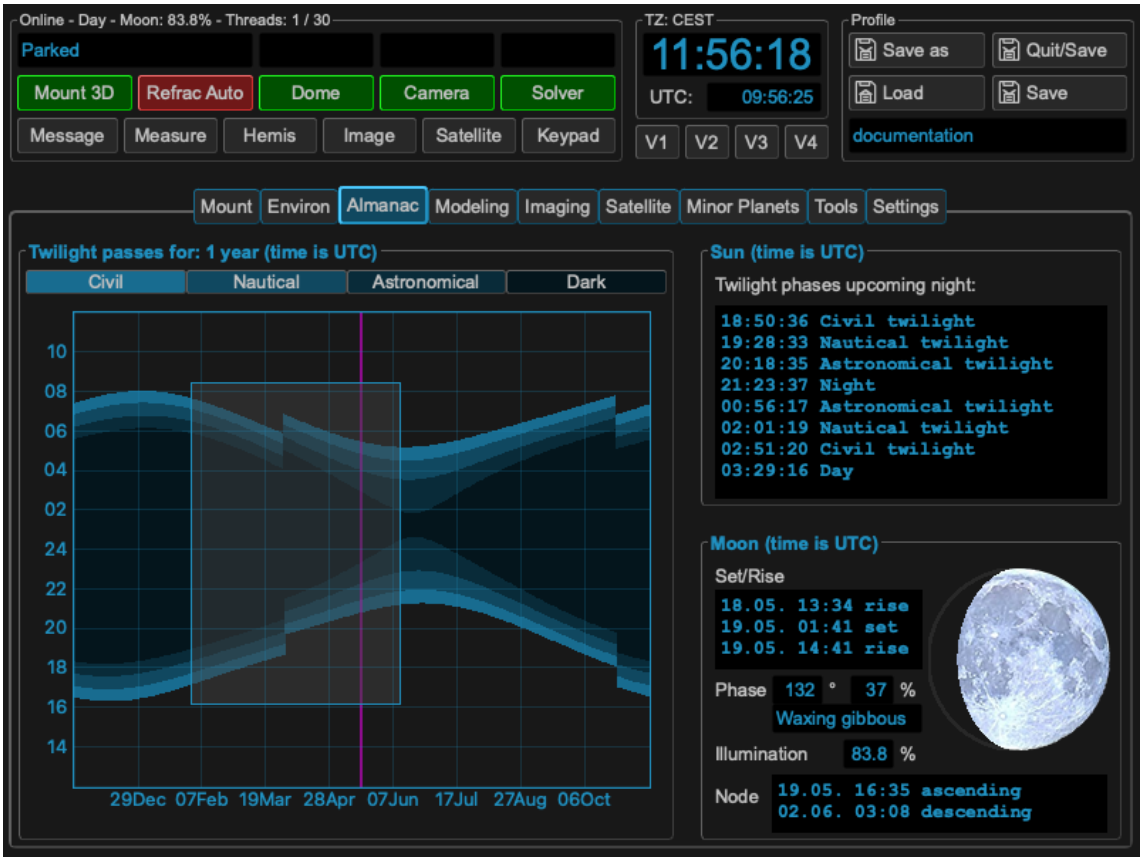
The following part should describe and explain the workflows and functions of the different parts of the software. Often some hints and explanations are given to understand and they are based of the experience of the author or other users.

Almanac

MountWizzard4 provides some basic information about twilight (1), (2) and moon (3) phases.



You could always select an area for zoom in or just use the mouse wheel to zoom.



Dome

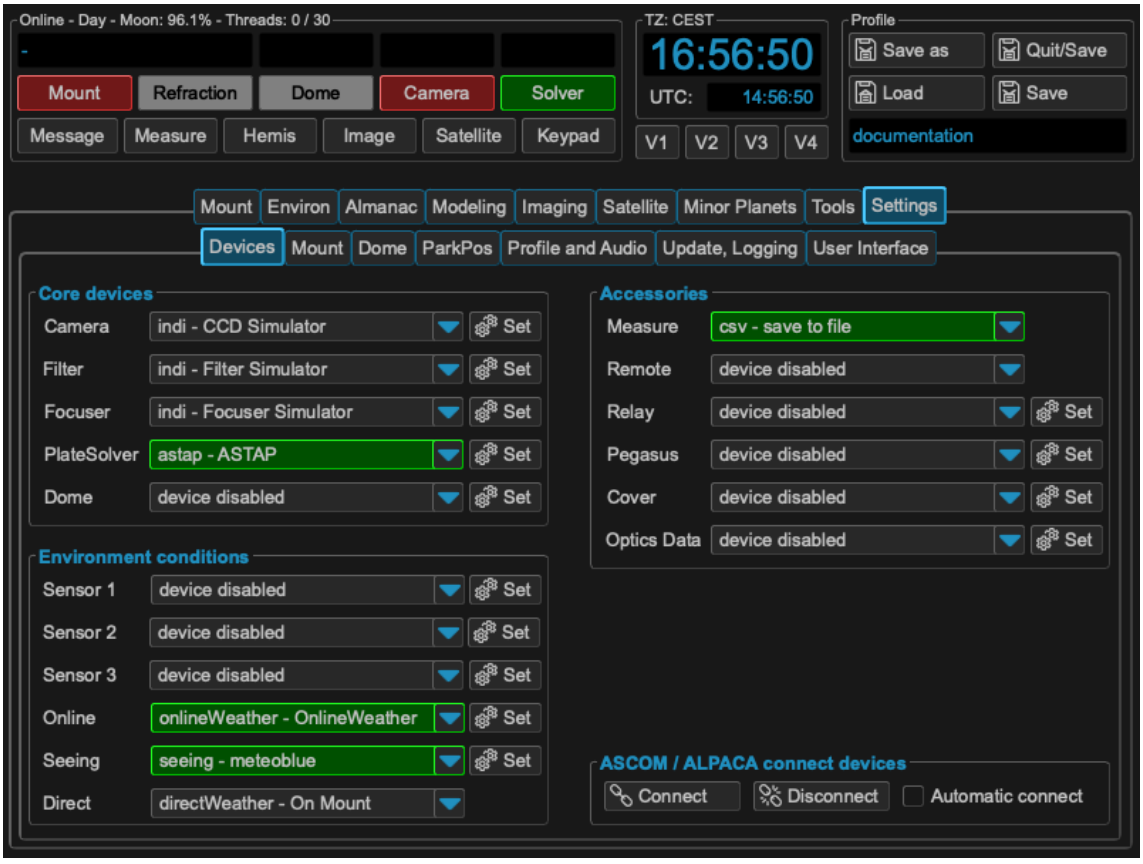
MountWizzard4 starts slewing the dome in parallel to the mount as it knows where the mount will land. Normally it should be able to detect how long the dome slews. The waiting time for the dome should be only relevant if dome movements influence mount by vibrations. It could be the case that the slewing signal has some specialities.

Sorting there are some optimizations to be set in MountWizzard4. Default ist without dome and it sorts for minimum mount slews distance.

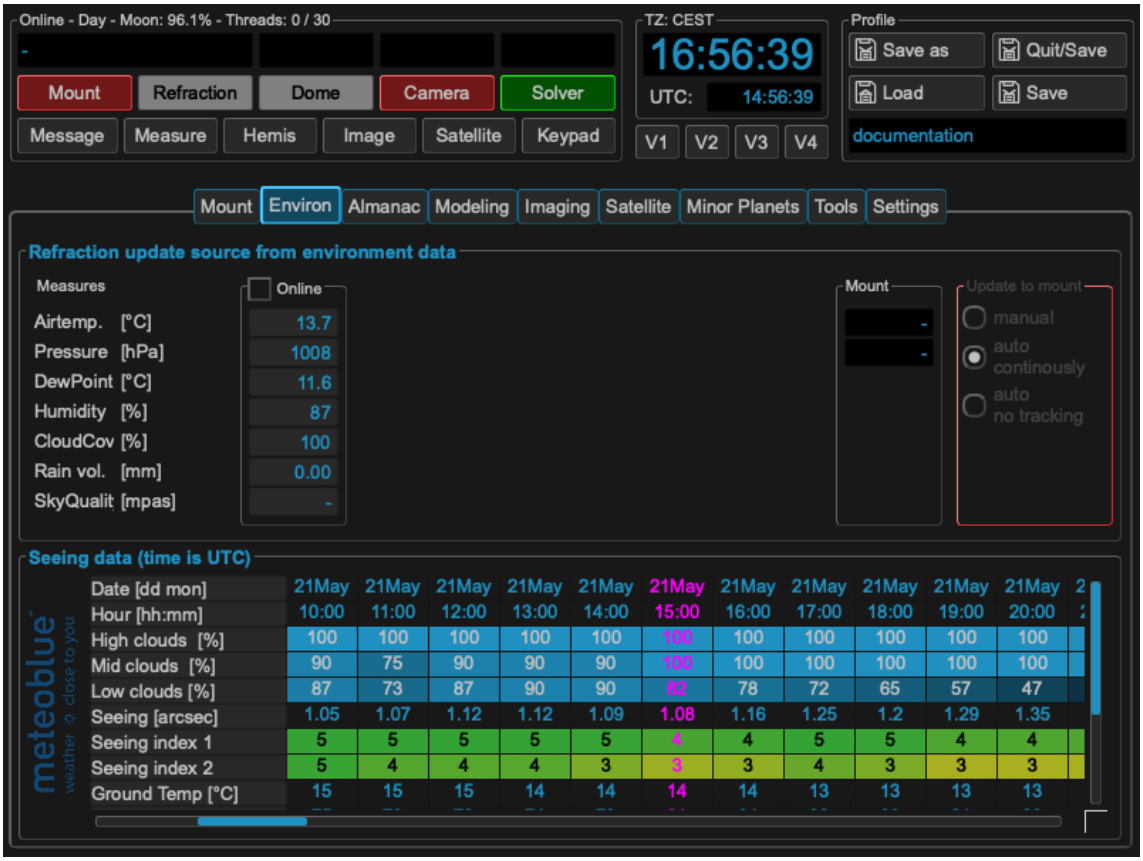
workflows/dome/image/dome1.png

Environment

Setup the environment:



Showing:



Imaging

The imaging window shows FITS files loaded from disk or images exposed manually or during model build. It is split in different areas to work with.

Area 1: Image exposing and solving

MountWizzard4 supports single (expose 1) and multiple (expose N) exposures. Continuous imaging could be stopped with abort. You also could explicitly load a fits file (extension .fit or .fits). If you have a plate solver (e.g. ASTAP) installed, you could solve the actual displayed image. The solved results are shown in message window. If you would like to add the results to the image, please check “embed data”. This will make MountWizzard4 to write the plate solving results in the fits header of the file.

Warning

Please be aware that MountWizzard4 will write the data to be embedded directly in the FITS file without making a copy of the file!

When “auto solve” is checked, MountWizzard4 will automatically plate solve every new picture and show or embed the results in the message window or fits header.

A simple stacking method is available when mount is in tracking and keeps point accurate. When “stacking” is checked, MountWizzard4 will add all exposed images (expose N running) and calculate the mean of the image.

Area 2: FITS Header entries

Some of the FITS header entry of the actual image are shown.

Area 3: Image attributes

MountWizzard4 calculates and extracts some image attributes. For example if a WCS header information is available, the distortion parameters are present or the actual image is flipped with regard to real position in sky.

Area 4: Image display

Show the image and it's different view selected in area 5. For standard view the scale is pixel with 0/0 to be the center of the image. There will be a colorbar in each view with the values of the image.

Area 5: View options

For the image you have different options to alter the main view of the image:

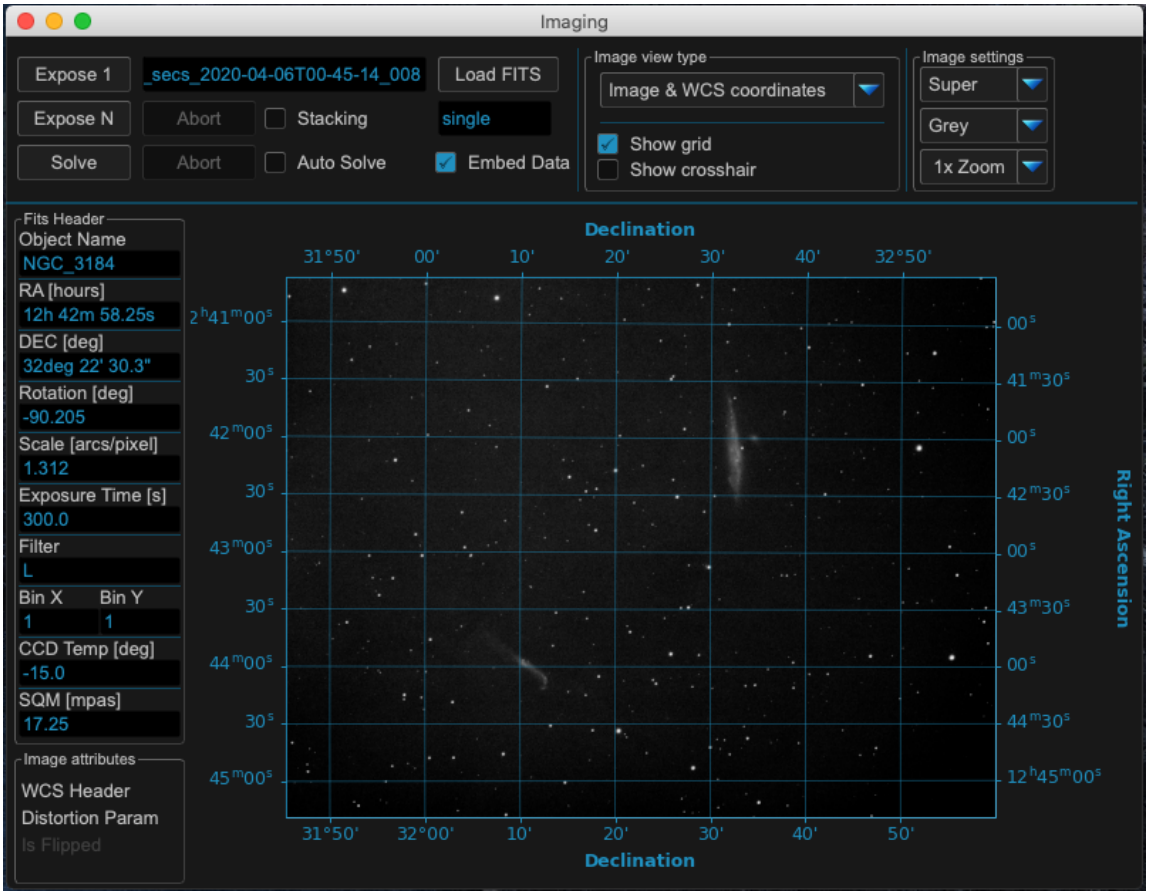
Image views

Drop down entry	Explanation
Image Raw	Standard visualization of the image in greyscale. MountWizzard4 does not support colors
Image with Sources	An overlay of the image with the extracted sources (stars) as circles
Photometry: HFD Value	SEP: 50 top Sources with HFD values
Photometry: Background level	SEP: Image of the background level
Photometry: Background noise	SEP: Image of the background oise level
Photometry: Flux	SEP: Value for flux of the detected sources

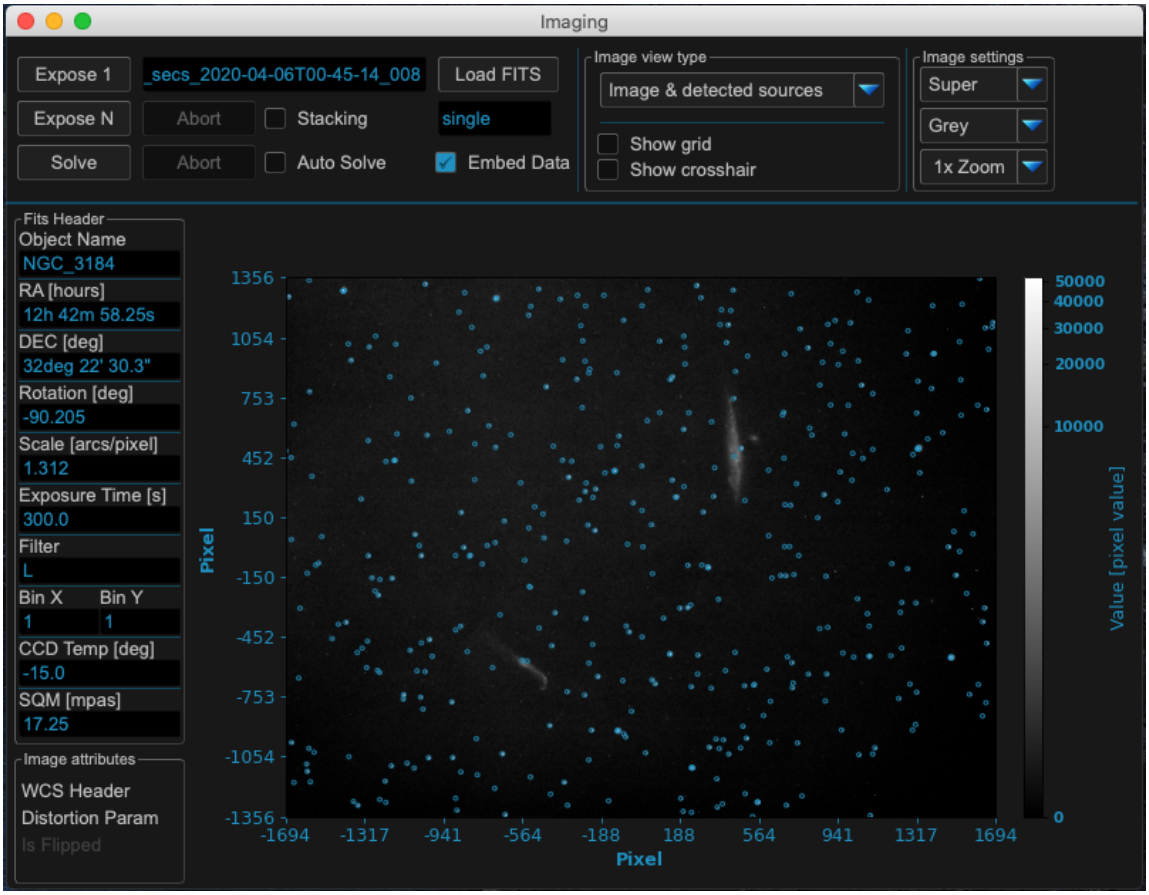
If distortion parameters are included, you could check “UseWCS” and MountWizzard4 will show the RA / DEC coordinates for the first threed image view options.

Some examples for the windows

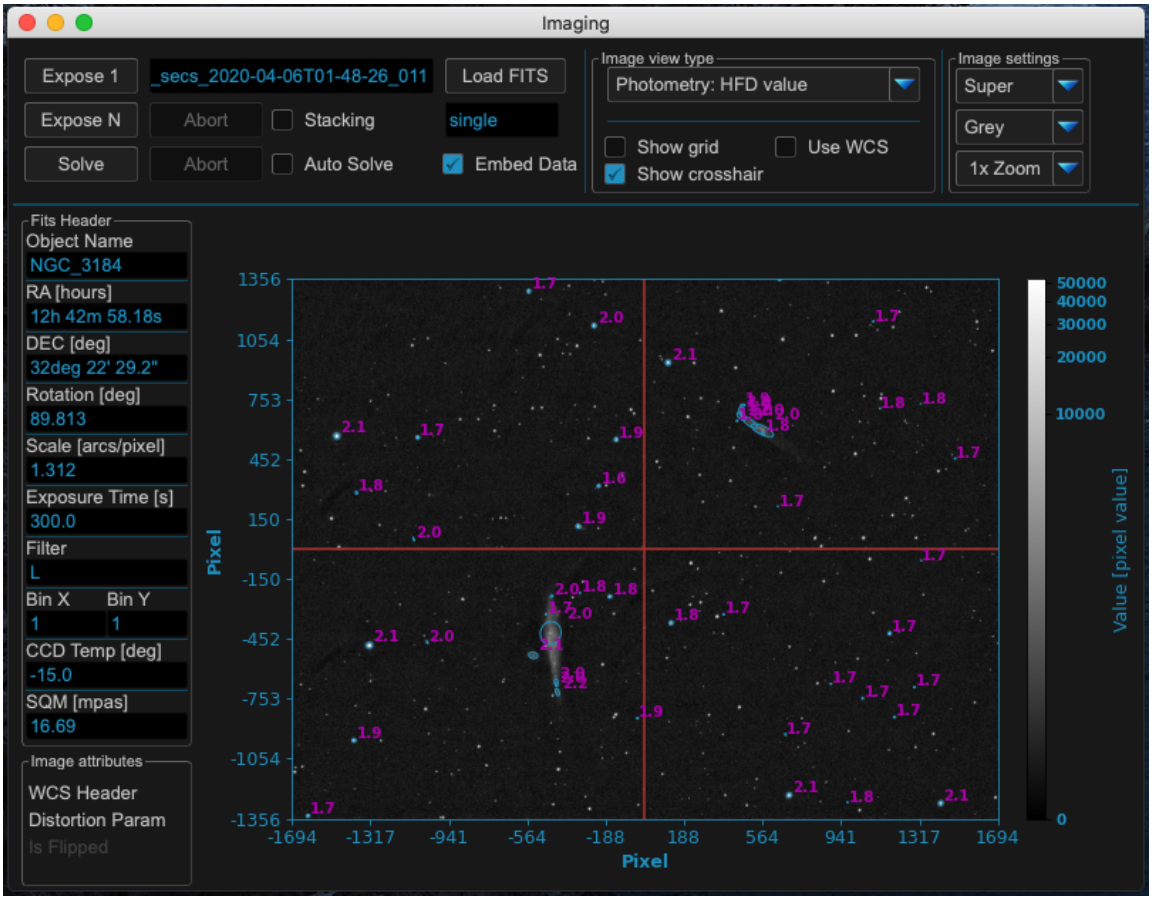
View Image with WCS distortion:



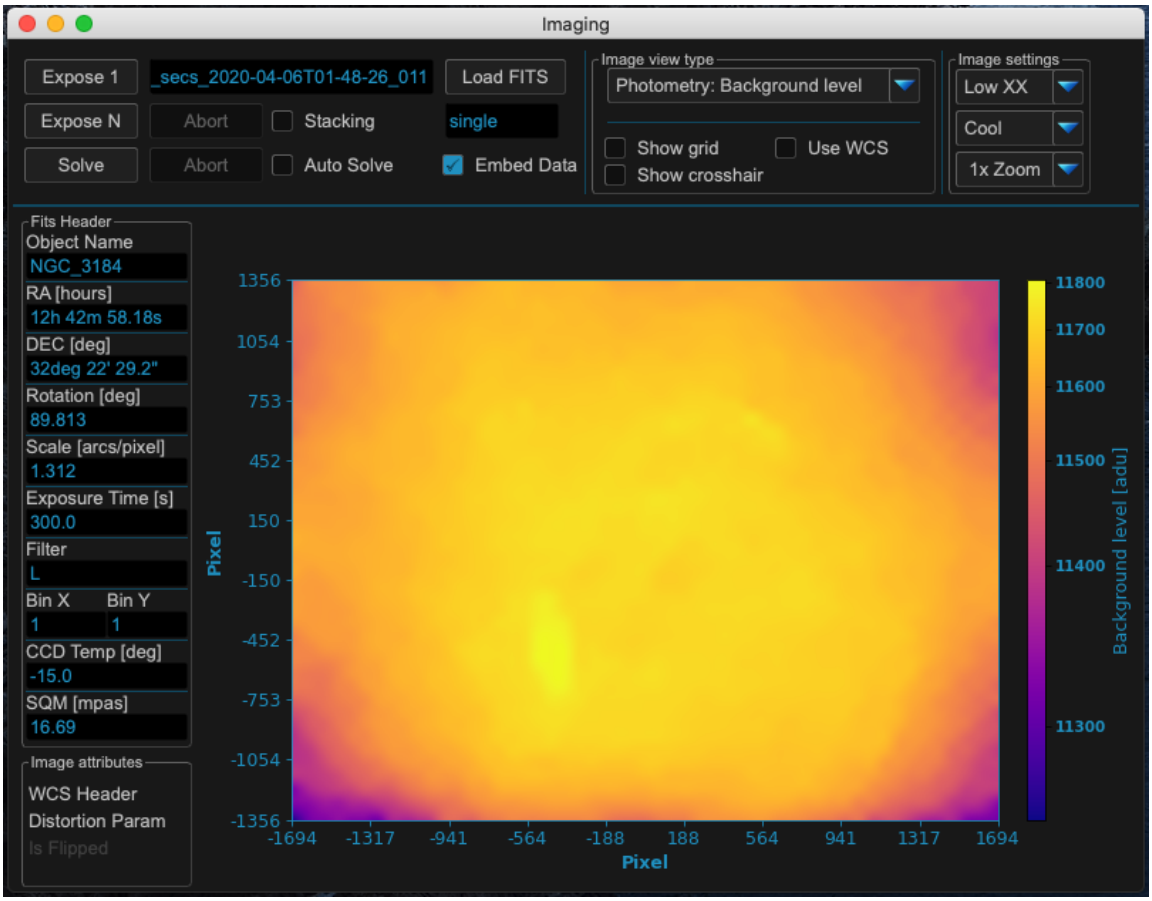
View Image with sources:



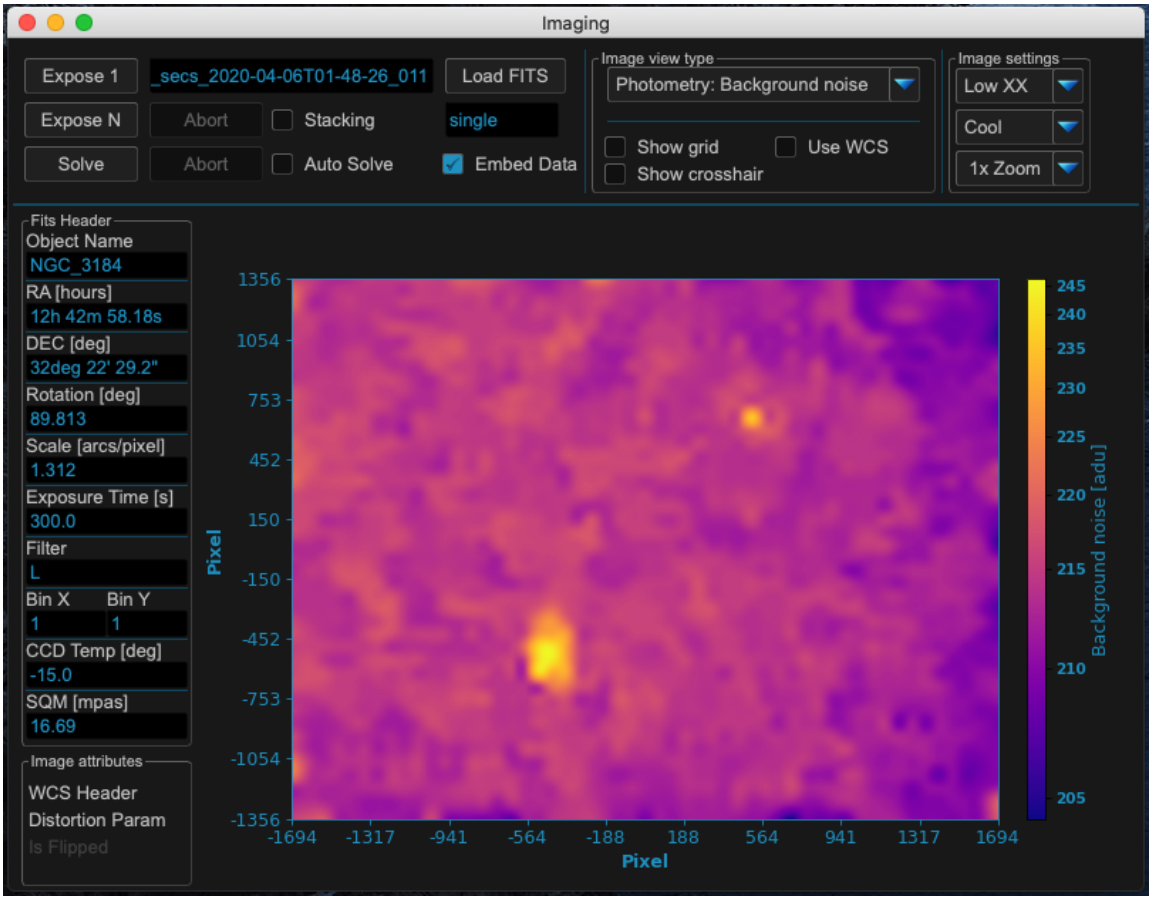
View image with HFD values



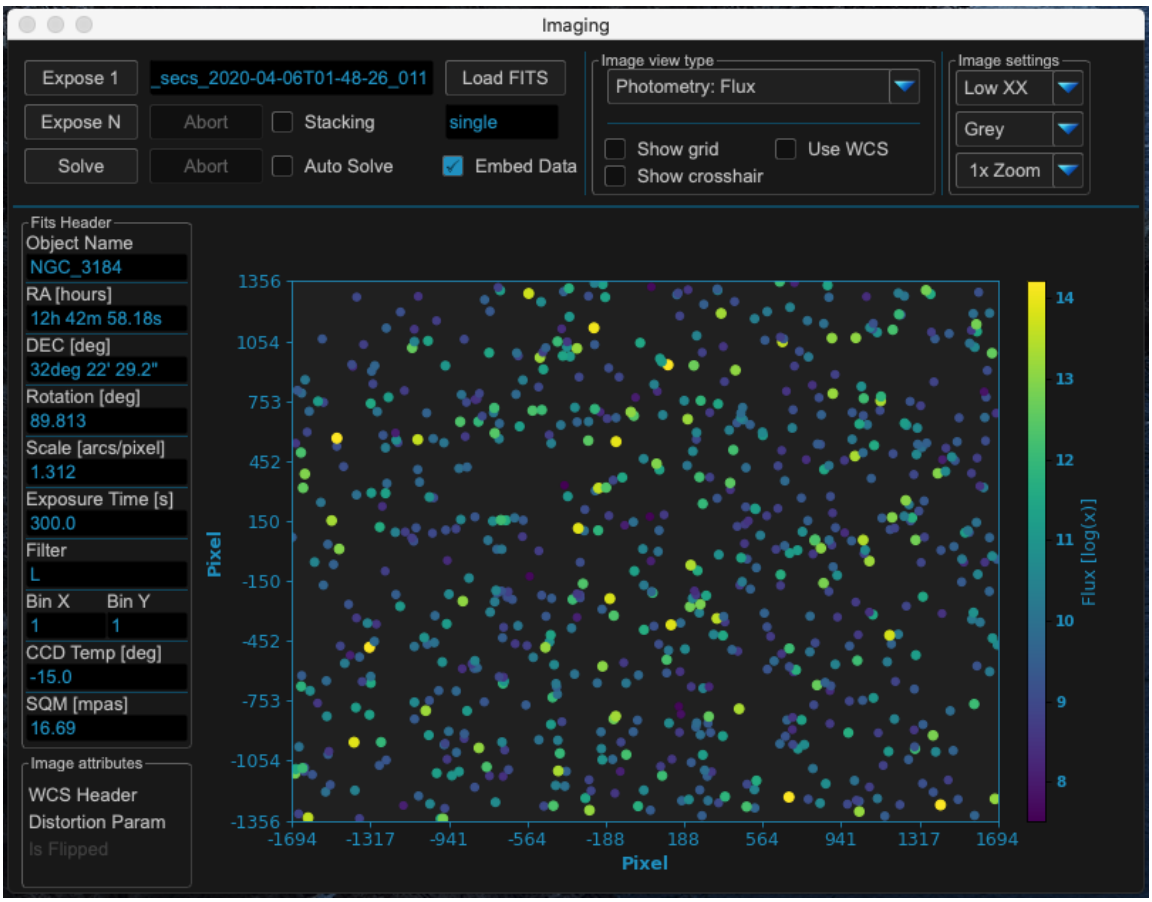
View image with background level



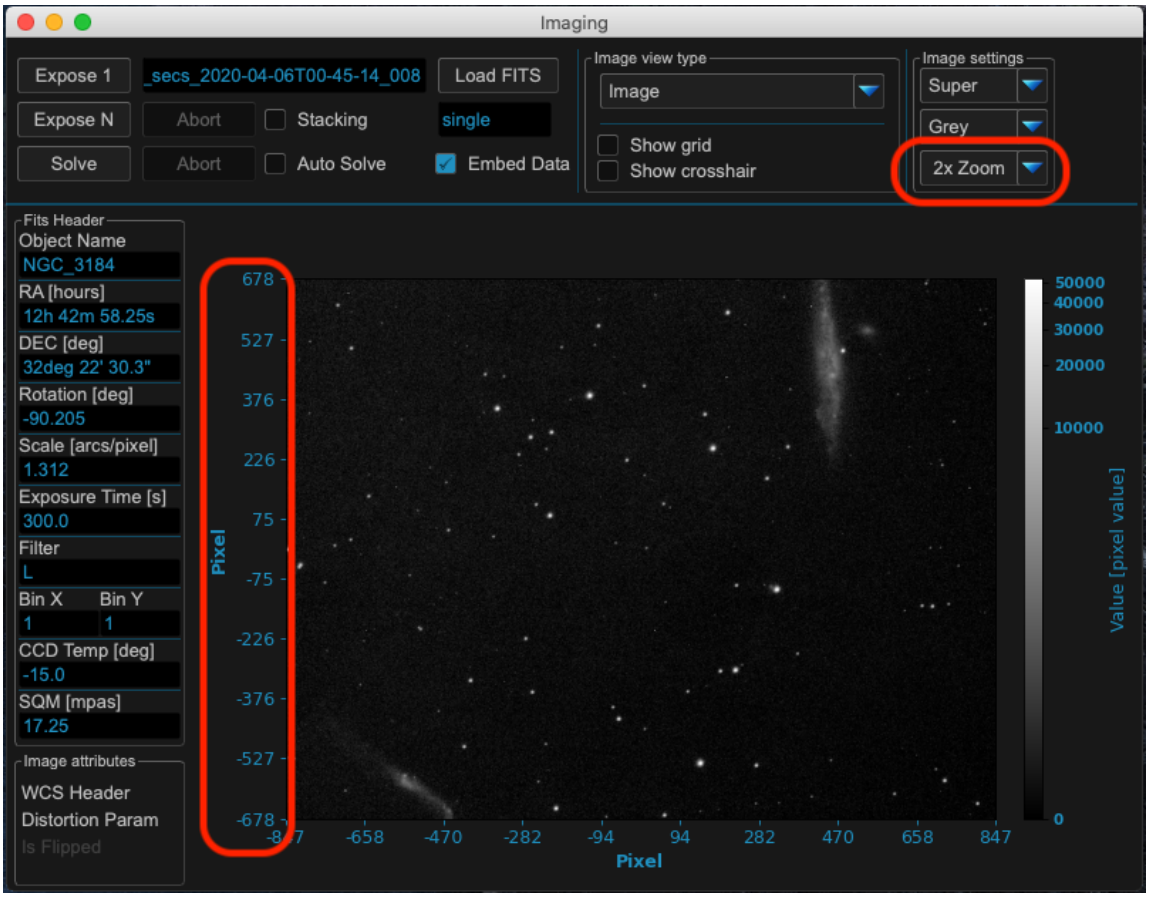
View image with background noise



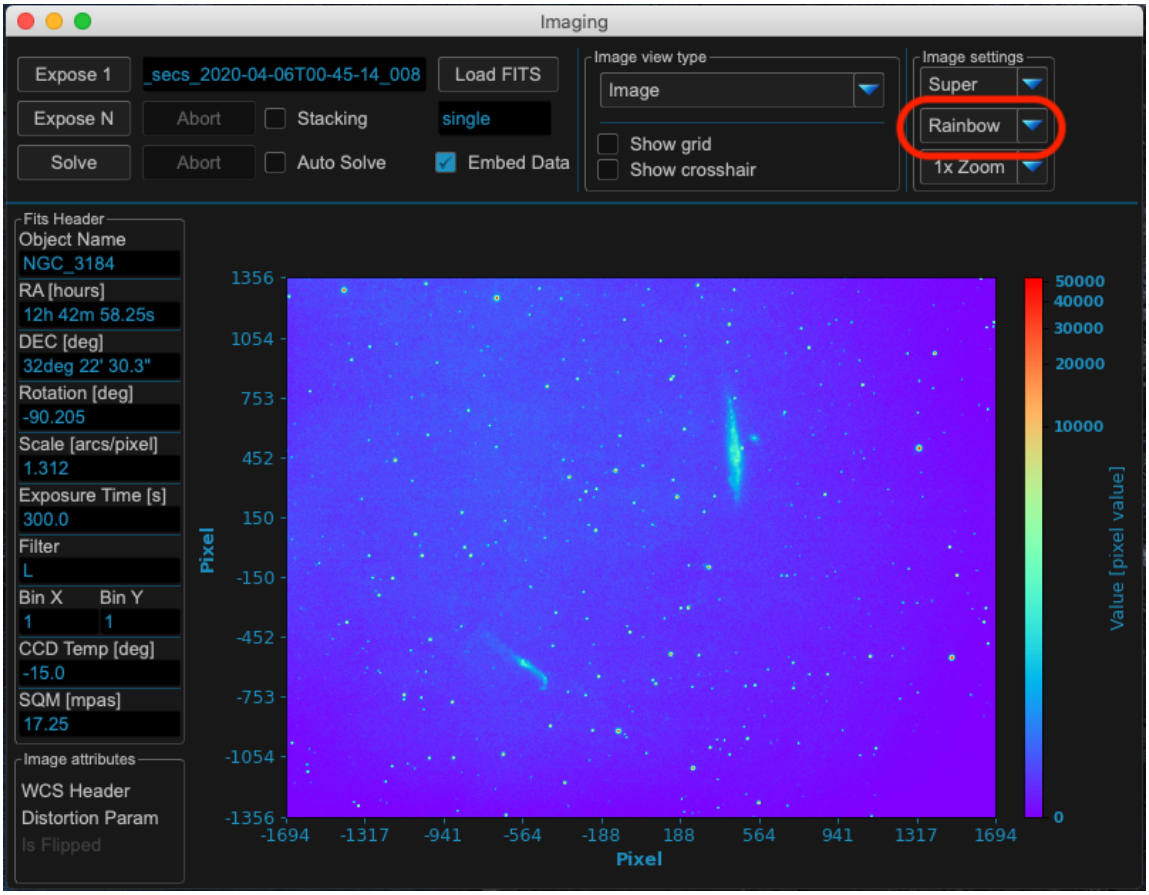
View image with photometry flux



View image with different zoom

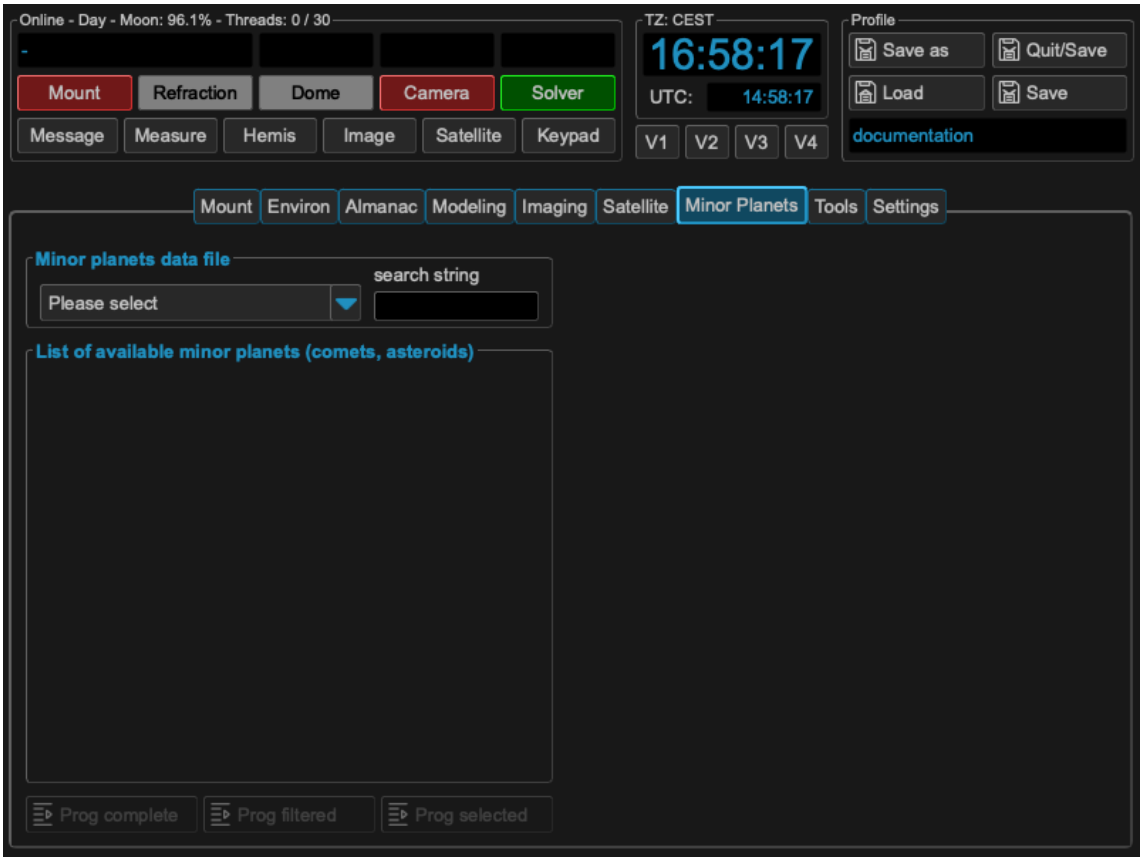


View image with different color scheme

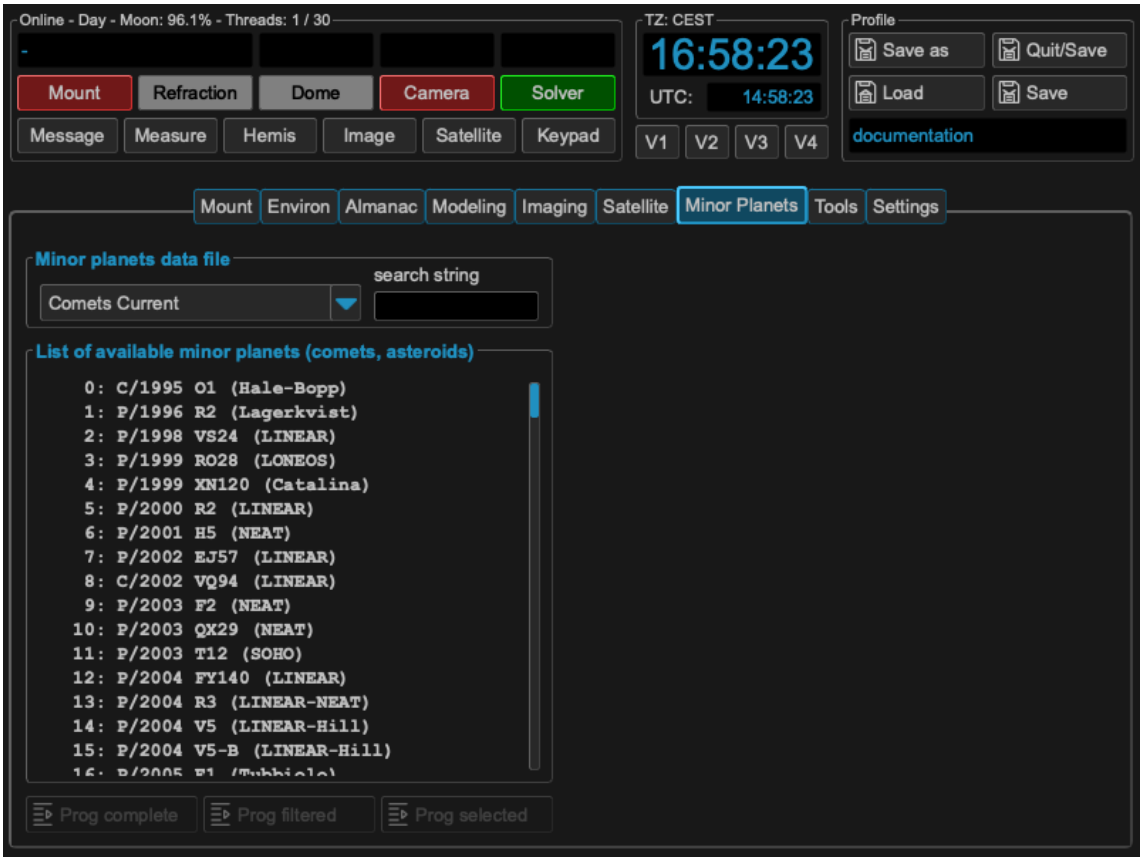


Minor Planets

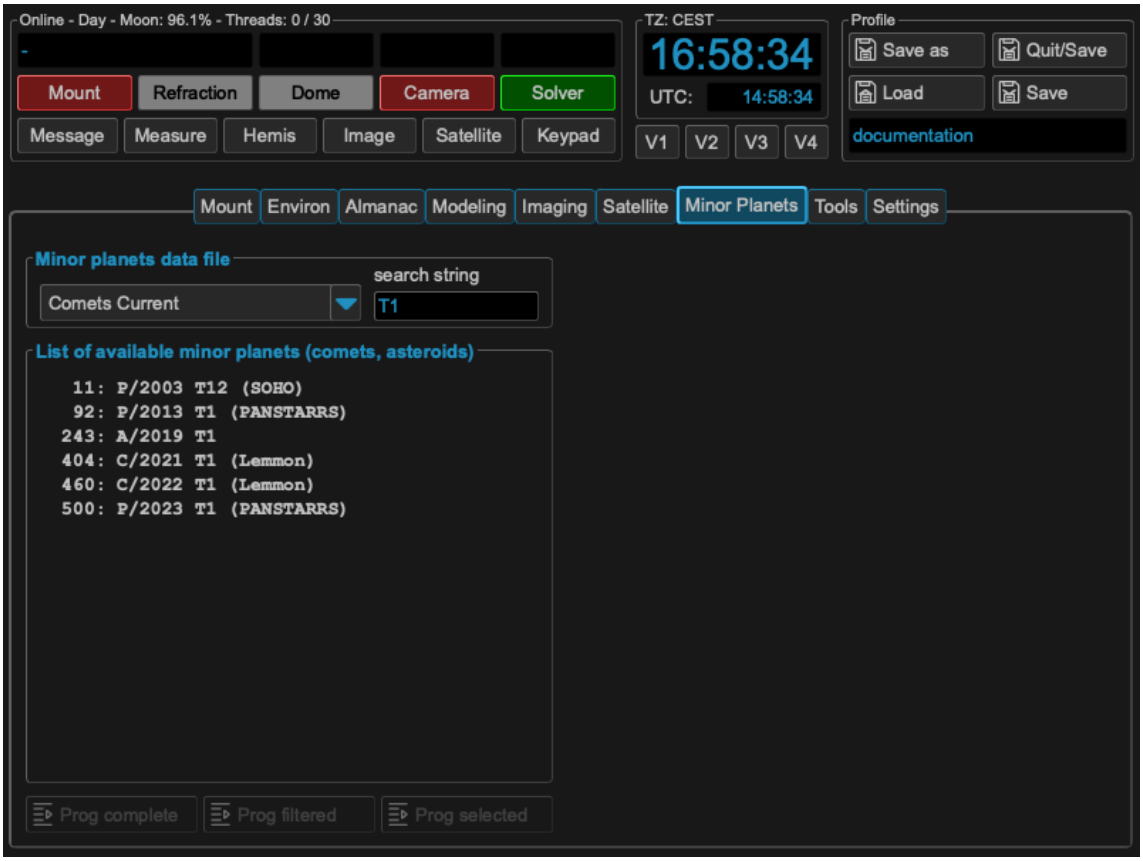
Menu



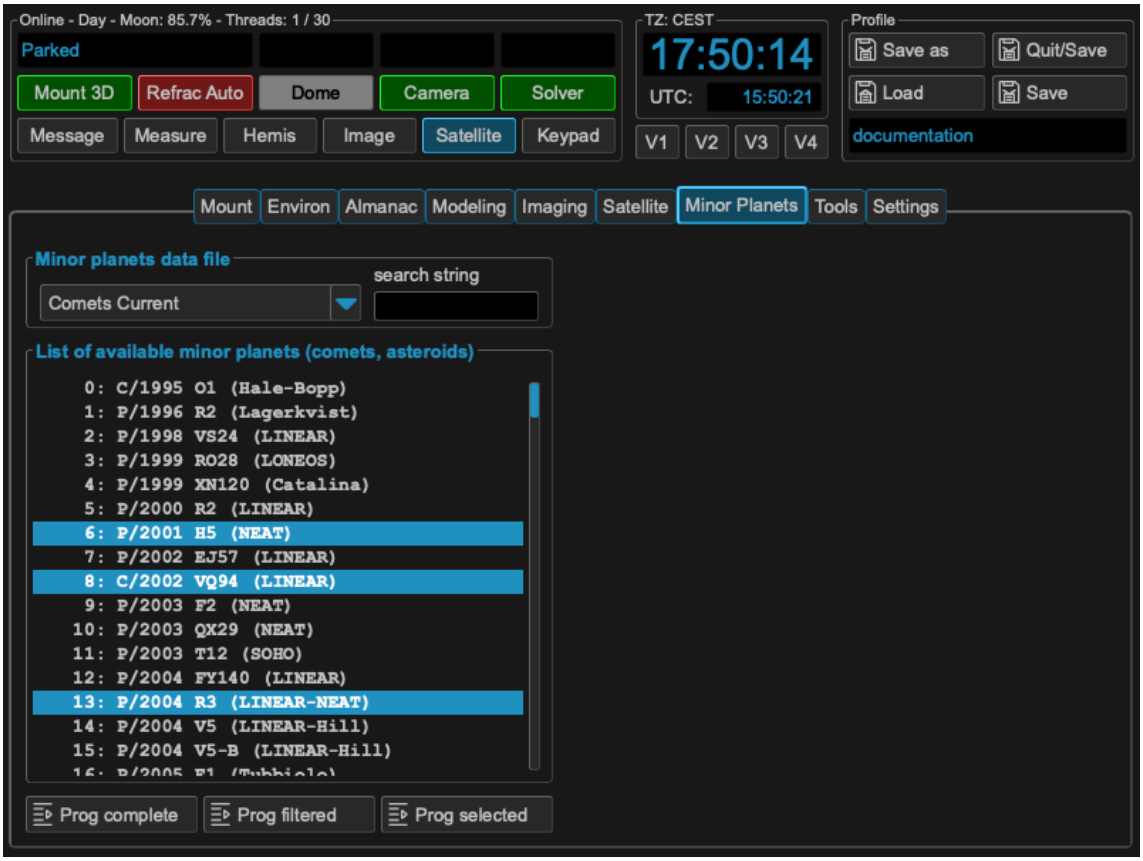
Downloaded Minor Planets database (example comets)



Selecting a minor planet by text search



Selecting a minor planet by clicking in the list



Modeling

Overview

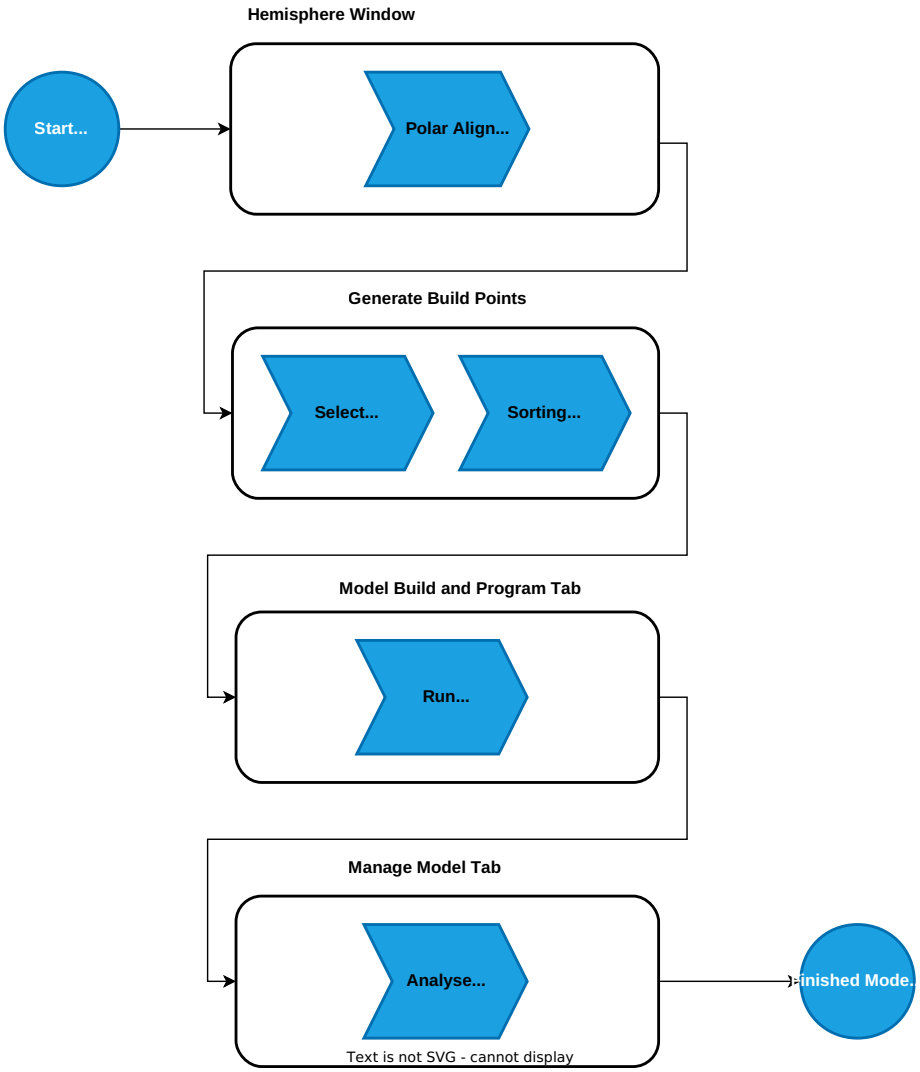
I only focus on model building with tool support, so no words about manual model building. I don't know the internal algorithms of the 10micron mount how the calculate their corrections. So many of the hints derive from pure logical or mathematical approaches and even there I personally might have some misconceptions or make some errors.

So my goal in model building is quite simple: I'm lazy in doing setups, so I want a solutions which gives me a correction model most accurate in minimum of time automatically. I rely heavily on the corrections capability on the 10micron mounts, so I use them always with dual tracking on. For doing a setup there are many things to think of beside the model (leveling, rigidity etc.). Keep them perfect, but I don't talk about them. So this results in two tasks I have to do to get a model to do unguided images: Polar alignment and the model for correction itself. I refer to the Blog Filippo Riccio from 10micron:

https://www.10micron.eu/forum/viewtopic.php?f=16&t=846

All the hints you get from the mount (how to turn knobs, alignment star) improve the alignment. As the model is only an approximation for the error correction, it will be not an one step approach. If you aim for the best result, please think of 2-3 iterations of the whole procedure. In my setup I normally need 2 iterations for doing an alignment which is good for 20-30 min exposures and have round stars.

As Overview: MountWizzard4 has a straight forward approach for building models. The following chart shows each basic step.



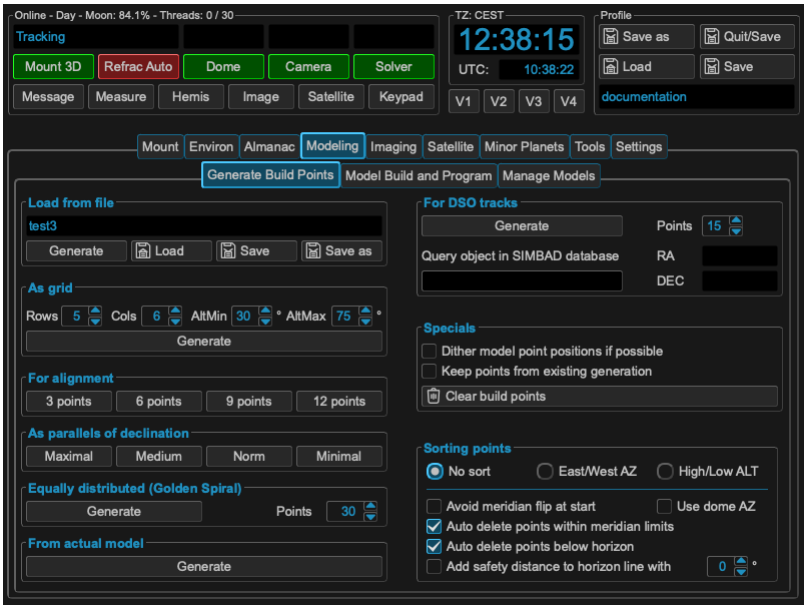
The following sections describe each step in detail:

Step 1: Polar alignment

Detailed description: Polar align your mount

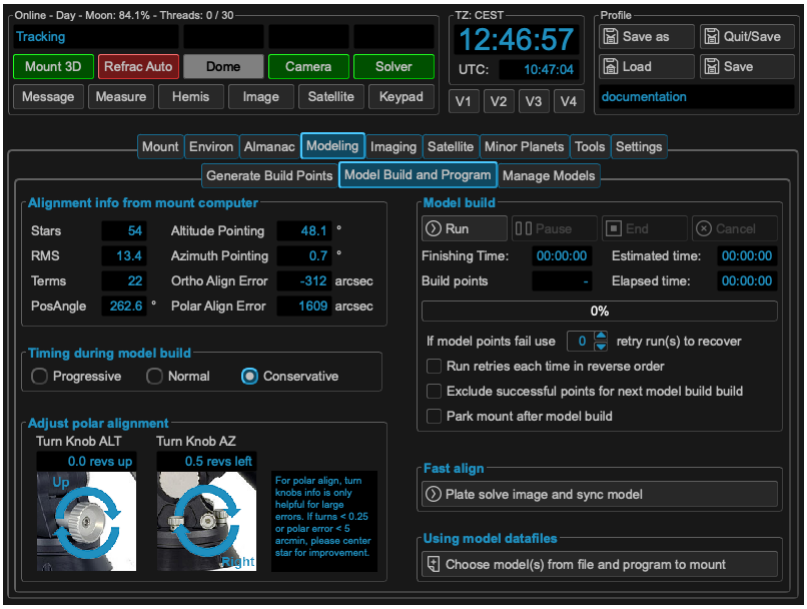
Step 2: Selecting build points

Detailed description: Selecting build points



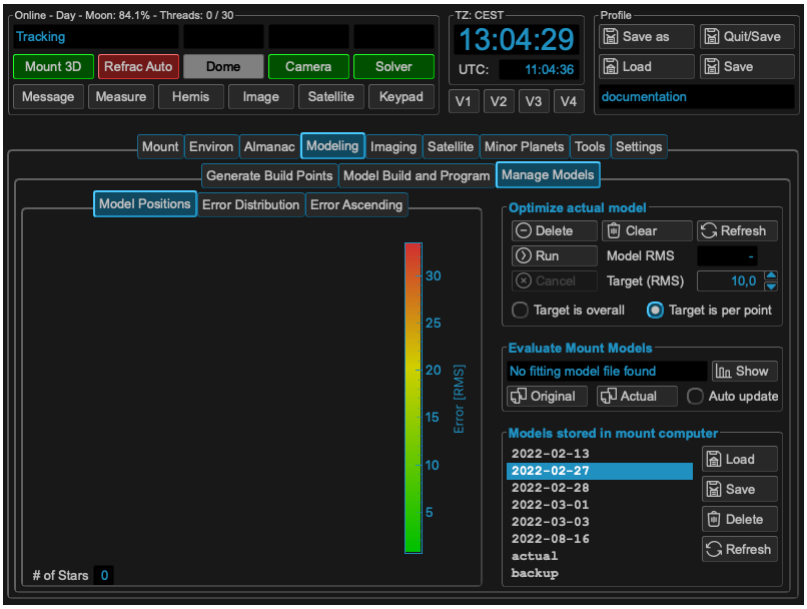
Step 3: Running the model build

Detailed description: Build a model



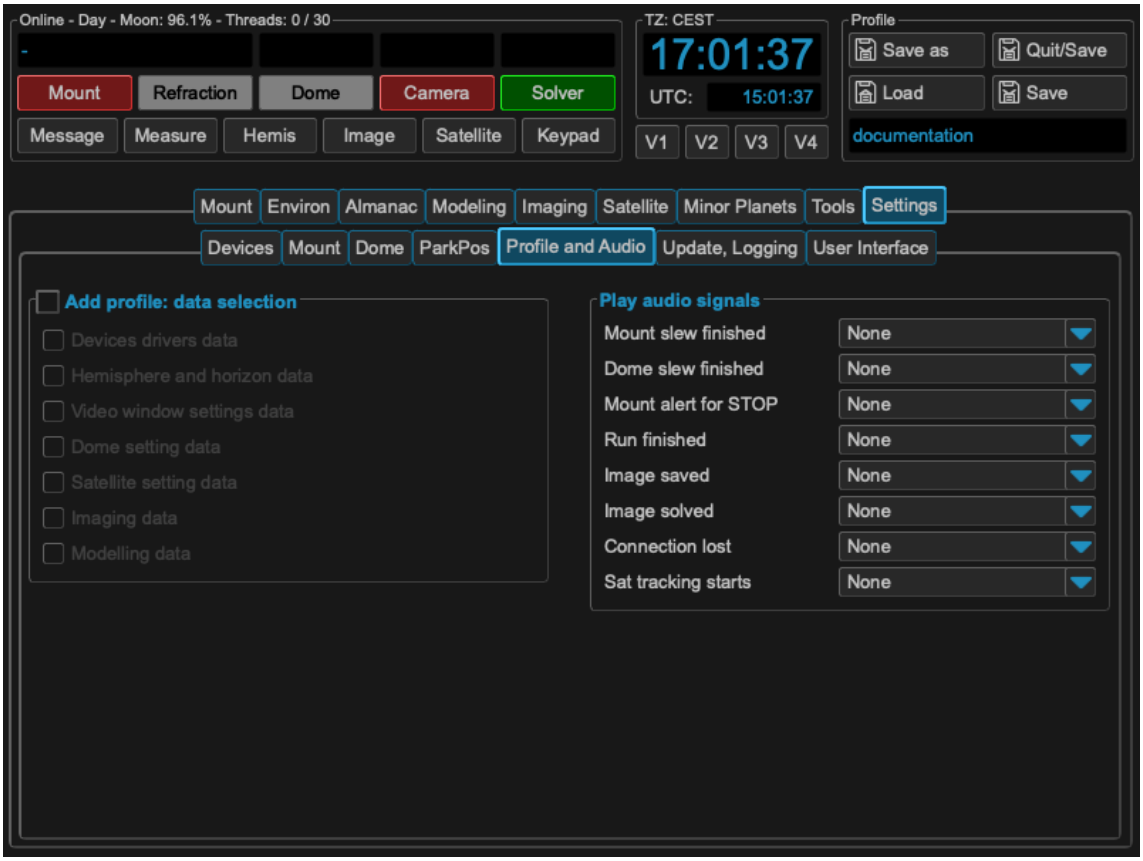
Step 4: Managing mount models

Detailed description: Managing mount models

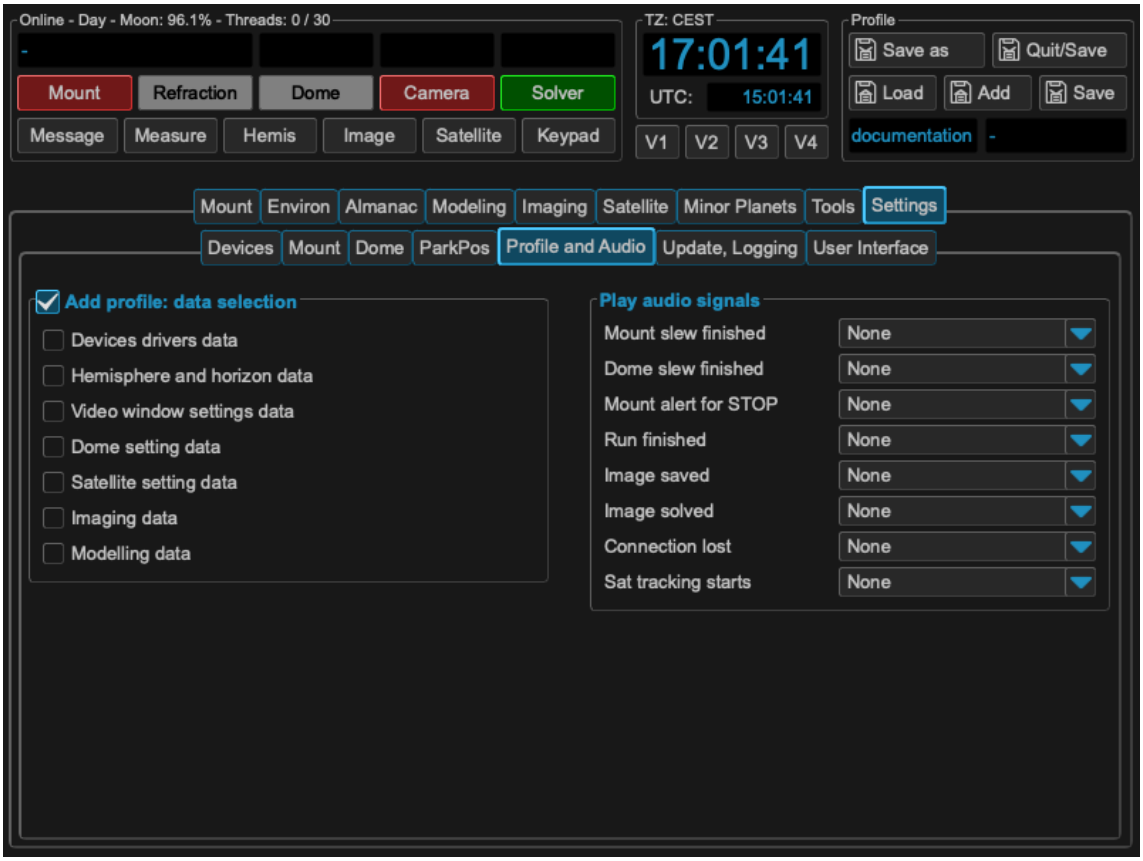


Profile handling

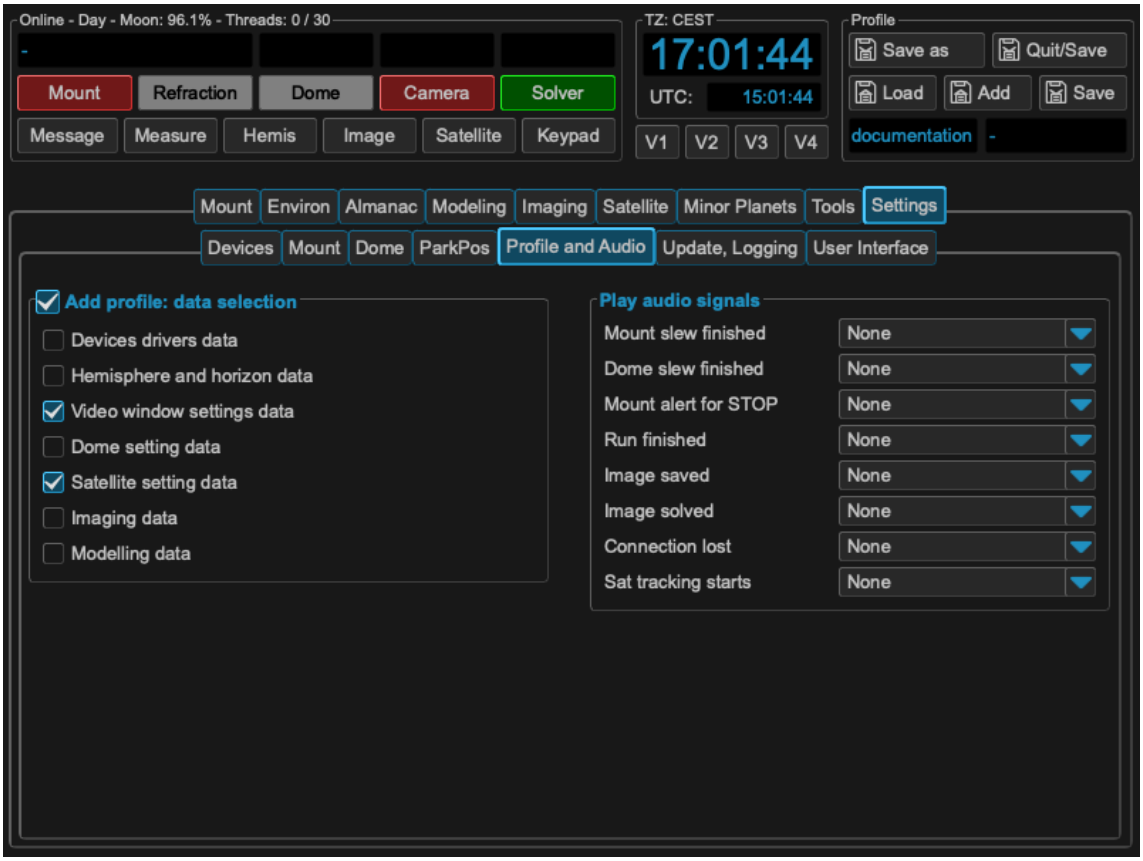
Menu



Adding subprofiles



Selecting the clusters for subprofiles



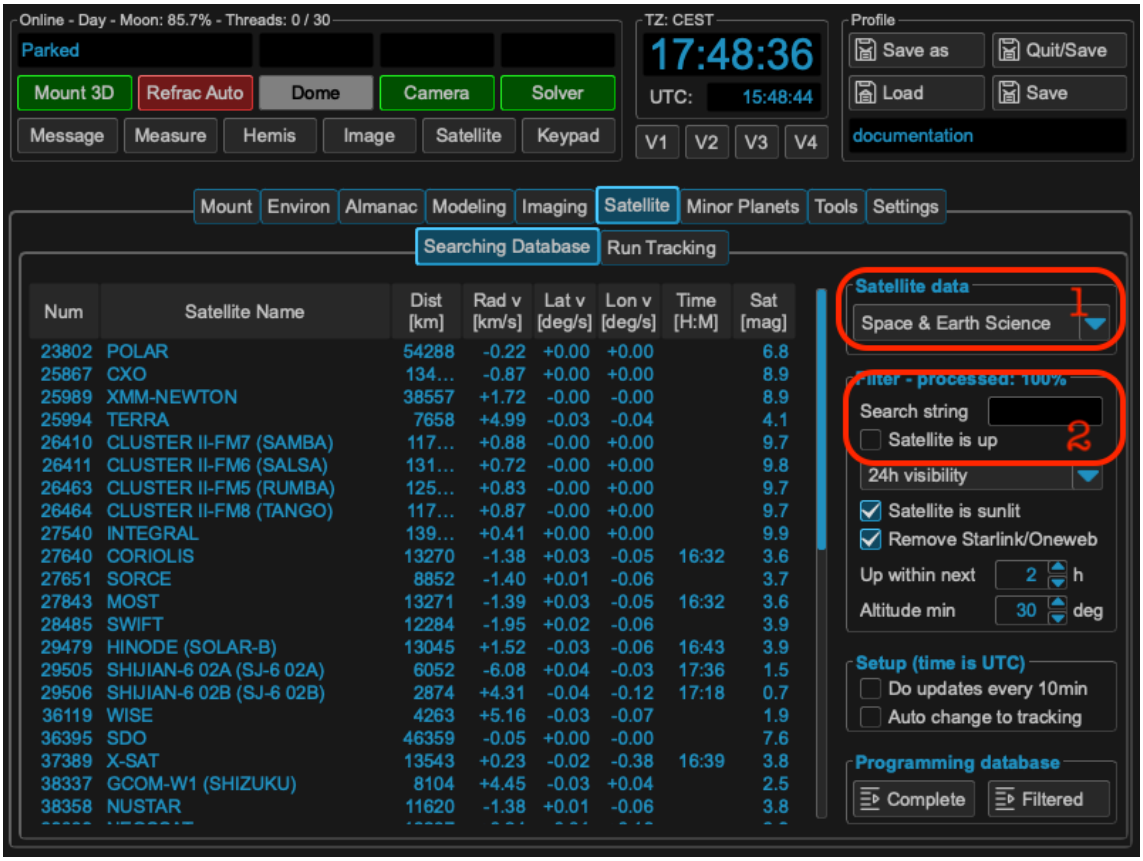
Satellite tracking

MountWizzard4 supports the mount capability of tracking satellites. The planning is based on Two Line Elements (TLE), which are provided by organisations (e.g. http://www.celestrak.com) and deliver a set of orbital parameters for calculating the satellite track with regard to your location in Alt / Az coordinates. Since the introduction, 10micron does the calculation internally. Therefore it is needed to upload the satellite TLE data to the mount and let it calculate. Before FW 3.x this is the only possibility (using the internal satellite database works similar) to do tracking.

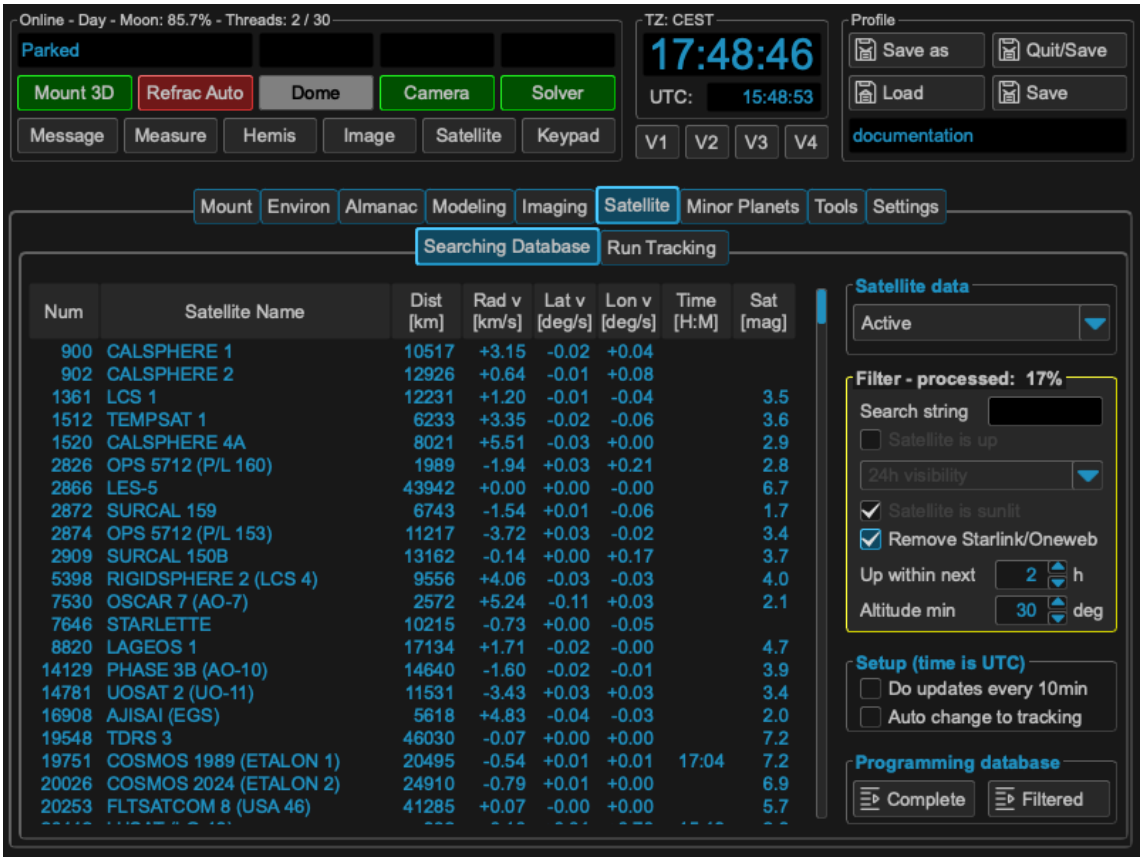
Note

As the orbital elements of a satellite change over time, it is necessary to get most actual data for get a good tracking. Data which is older than 10 days is definitely outdated. MountWizzard4 marks them red. Older than 3 days may work, MountWizzard4 marks them yellow. Ideal is to get them right in time.

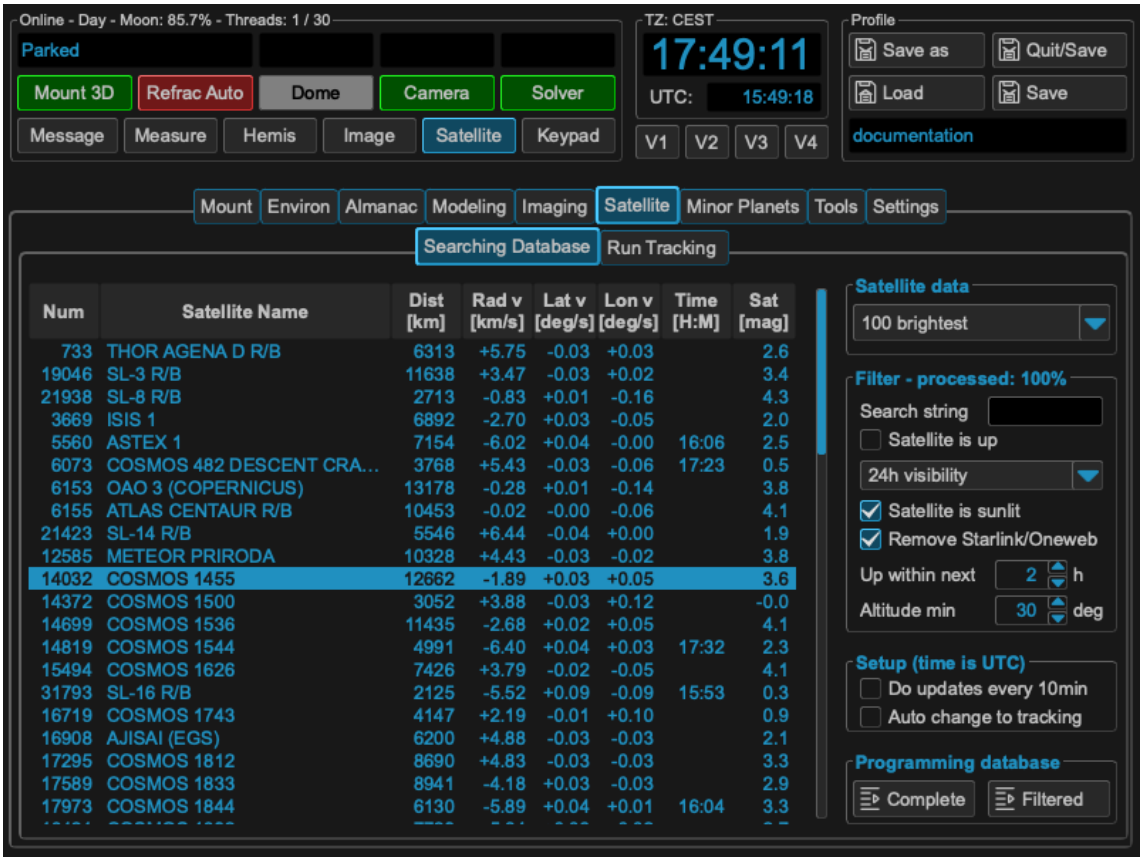
You could select different databases on the right upper part of “Searching Database” tab {1}. Once selected and online enabled, MountWizzard4 will download the newest data and offers the list of included satellites. For finding the satellite of your choice you could use the search field (2) to reduce the list. The string is *not* case sensitive and will be stored persistent.



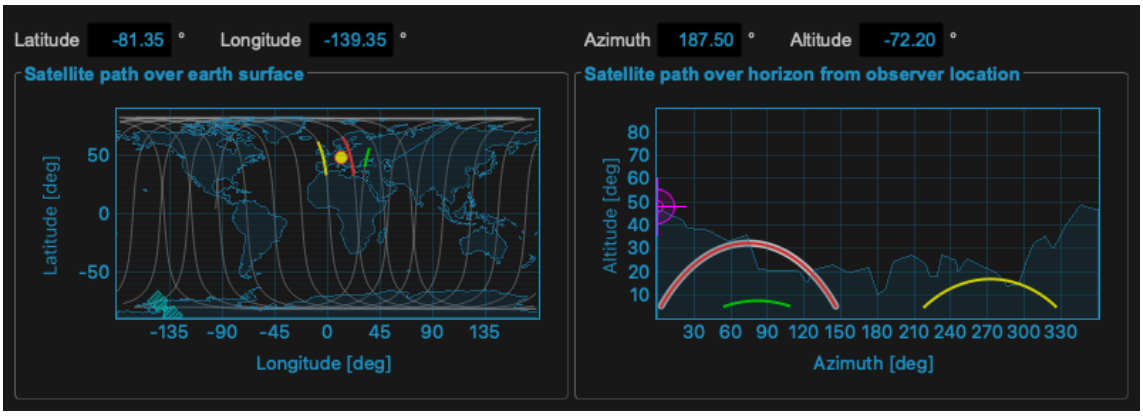
There are many more selection criteria available to filter the list of satellites with real-time calculations of some properties like velocity, distance and a glimpse of the apparent magnitude. This calculation takes some time and you will see a yellow frame and progress counter in the filter area.



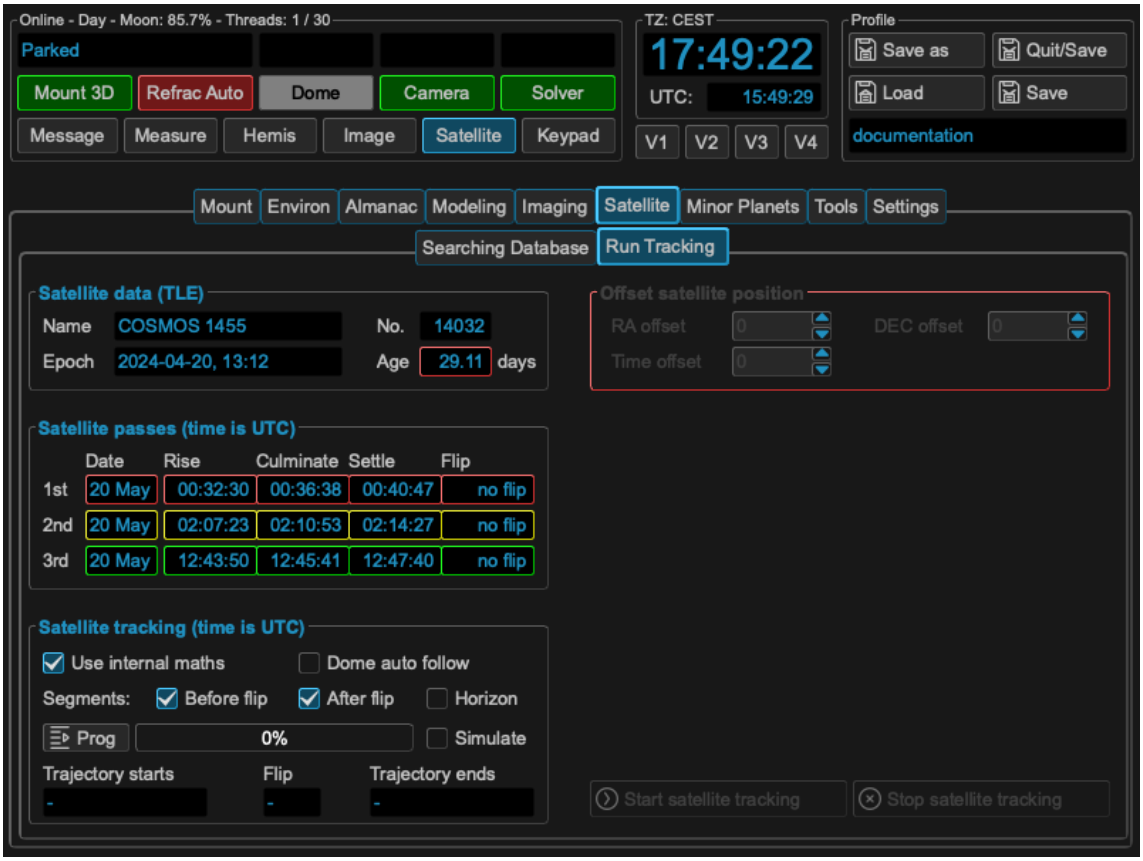
Once you choose a satellite with double click, data is programmed to mount controller, parameters are displayed, MountWizzard4 calculated the next 3 orbits of the satellite with rise / culminate / settle and if it occurs the flip time when crossing the meridian.



The selected satellite will also be shown in the **Satellite** window:



After the mount has done it's calculations as well, the result will be shown in the **Trajectory starts** and **Trajectory ends** fields and a possible flip will be announced on the "Run Tracking" tab, MountWizzard4 offers now (need Firmware 3.x) some adjustments for centering the satellites in you imaging FOV.



As soon as a valid tracking path is present in the mount, the **Start satellite tracking** and **Stop satellite tracking** button are enabled. Once started, the mount will slew to the begin of the tracking path and wait for the satellite to rise. Selecting partial tracks and respecting constraints is not possible.

Since FW 3.x the command protocol offers the programming of a custom satellite track. This offers the capability of takings care of avoiding flips, respect horizon and other constraints. The operation is different to the classic approach: Instead of programming TLE data to the mount,

MountWizzard4 programs Alt / Az coordinates in a one second interval to the mount (max. 900s) which the mount after start tracking will follow. You could enable this feature with checking **Use internal maths** if the firmware is 3.x or higher. After enabling, additional elements will be enabled.

workflows/satellite/image/sat_new.png
As the calculation now happens outside the mount, we could take a look to the difference between tracks calculated by the 10micron mount and MountWizzard4 based on the same satellite TLE data! You will find some comparison under the architecture / math section: precision of internal calculations .

If you are using the internal math as well as classic mode, you could open the satellite window. There you could see the next three orbits and for internal math the resulting satellite track with an white underlay. If you change any setup, MountWizzard4 will recalculate all data and updates the plots accordingly.

Note

The solid line shows the track before meridian transit, the dotted line the track after meridian transit. The resulting white background shows the resulting track, which takes flip track tolerance into account. If you change settings in satellite (or even limits), MountWizzard4 will update the tracks path in plot accordingly and reset the prog state.

Select only a segment before a meridian transit (and therefore avoiding a flip during tracking):

workflows/satellite/image/sat_af.png workflows/satellite/image/sat_af_track.png
Select only a segment after a meridian transit (and therefore avoiding a flip during tracking):

workflows/satellite/image/sat_be.png workflows/satellite/image/sat_be_track.png
Select both segments of the meridian (this might cause a flip during tracking):

workflows/satellite/image/sat_be_af.png workflows/satellite/image/sat_be_af_track.png
Select respecting the horizon line. This filters out additional all segments, which are below the given horizon mask.

workflows/satellite/image/sat_hor.png workflows/satellite/image/sat_hor_track.png
MountWizzard4 will take into account the meridian track limits of your mount. Here set to 1 degree (which is close to meridian)

workflows/satellite/image/sat_lim_1.png workflows/satellite/image/sat_lim_1_track.png
MountWizzard4 will take into account the meridian track limits of your mount. Here set to 15 degrees (which could avoid a meridian flip or at least extend the tracking time)

workflows/satellite/image/sat_lim_15.png workflows/satellite/image/sat_lim_15_track.png

Warning

The meridian track limits have to be chosen carefully as the mount might hit your setup !

The biggest change in using satellite tracking with the new implementation is how the data is handled to the mount: whereas in classic mode only the TLE data has to be uploaded (which is quick) now the whole track coordinates have to be programmed . As this takes up to 10 seconds, MountWizzard4 does not automatically start the transfer. Once your setup (choice of segment, horizon etc.) is made, you have to start the programming by pushing the **Prog** button.

workflows/satellite/image/sat_prog.png
After a successful upload, the trajectory data is populated and the Start / Stop tracking buttons are enable like in classic mode.

workflows/satellite/image/sat_result.png

Tools for the Job

Updating the app

Troubleshooting Q&A and hints

Based on many feedbacks and solved issues, please check first if your question already has some answers or at least some hints how to improve the situation.

Installation

Despite MW4 will run on many platforms, your setup might have some special constrains which need to taken into account. In the following I try to refer to the important ones.

Python

MW4 runs on python 3.7 - 3.9. On other versions scripts will will fail and MW4 will not run.

Windows version needs at least 3.8.2 to allow automation.

Windows 7 might be using, but is not tested. Other windows versions are not supported.

On windows, you need to select if you are using 32bit or 64bit python depending on drivers you are using for your devices. 32bit and 64bit could not be mixed !

Normally you will use a preinstalled python version (if that fits) or use a python version from python.org. Please do not use other sources.

As MW4 lives in a virtual environment, updating python does not automatically update the virtual environment. If you need to update python for any reason there are two possibilities: New install of MW4 in a new work dir or deleting the venv folder in your actual work dir and running the install script again.

MW4 App

There is no need for running MW4 with admin rights. If so, something is wrong.

MW4 will run in a virtual environment. Please do not try to install MW4 as a system application as this might interfere with other installations.

MW4 does not behave as expected: please post a log and describe the procedure in steps. If you could add screen shots this helps a lot.

MW4 Updating does fail

Since v2 MW4 should be able to handle all updates / downgrades with the internal updater. From v1 to v2 windows needs the MW4_Update.bat script as the internal updater can't free used windows libraries. MacOSx and Linux should be fine.

Scripts

Installation does not need admin rights. If so, please check the folder locations. In windows10 desktop and some other folder are not writable for applications.

The scripts do nothing special, you could use for many topics manual commands as well. Unfortunately the scripts could not manage all special setups, but feel free to change them accordingly.

On RaspberryPi4 (arm64), the scripts try to use precompiled wheels from github. This increases speed. But in some circumstances, compile on your system might be necessary. If so, you need to have a compiler and environments installed as well.

On RaspberryPi3 you need to compile the environment partly yourself. An installation only with scripts will not work.

Mount connectivity

MW4 only supports IP links. As data latency is a critical topic, please use a wired connection to the mount. Wireless connection might have some drops in connections (you will see this with mount button switching red / green multiple times)

Please check your IP settings, gateways if first connections fail.

If your WOL does not work, please check MAC address, WOL being enabled. If you switched your mount manual off and cut the power supply, sometimes WOL does not work the first time. You need a redundant path if you are in a remote site!

Basically multiple instances of MW4 could be up and running, but MW4 take up to 6 parallel connections to the mount. The documentation allows in total 10 connection each of the two ports (3490, 3492). This might overload the 10 micron system.

Device connectivity

ASCOM Device does not work / connect

ASCOM uses different types of drivers for the devices. Some of them need an environment in 32bit or 64bit like you application. So if you are using an 64bit application for imaging your drivers will be 64bit compatible. In this case the python installation also needs to be 64bit. Otherwise the connection will fail.

Modern CMOS cameras with large sensors normally work on 64bit mode.

Many device driver only support one connection at the same time. So if you imaging application already *took* a device, MW4 might be not able to connect anymore.

Please test your setup running with ASCOM suite (included in ASCOM platform installation) or any other programs you good know to test device functions outside MW4.

Model building

Updating IERS/SAT/MPC data

Data could not be uploaded

All data uploads within MW4 use the 10micron updater. The updater is only available on windows and has to be installed.

MW4 does a windows automation. So it steers the original application and automates all user interactions. This might take time. Please wait until MW4 has finished it's job. Please do not interact on your PC during this time with mouse or keyboard as they disturb the automation process.

In any case MW4 has downloaded or prepared the data for upload. This is also valid for non windows platforms.

Data could not be fetched from internet

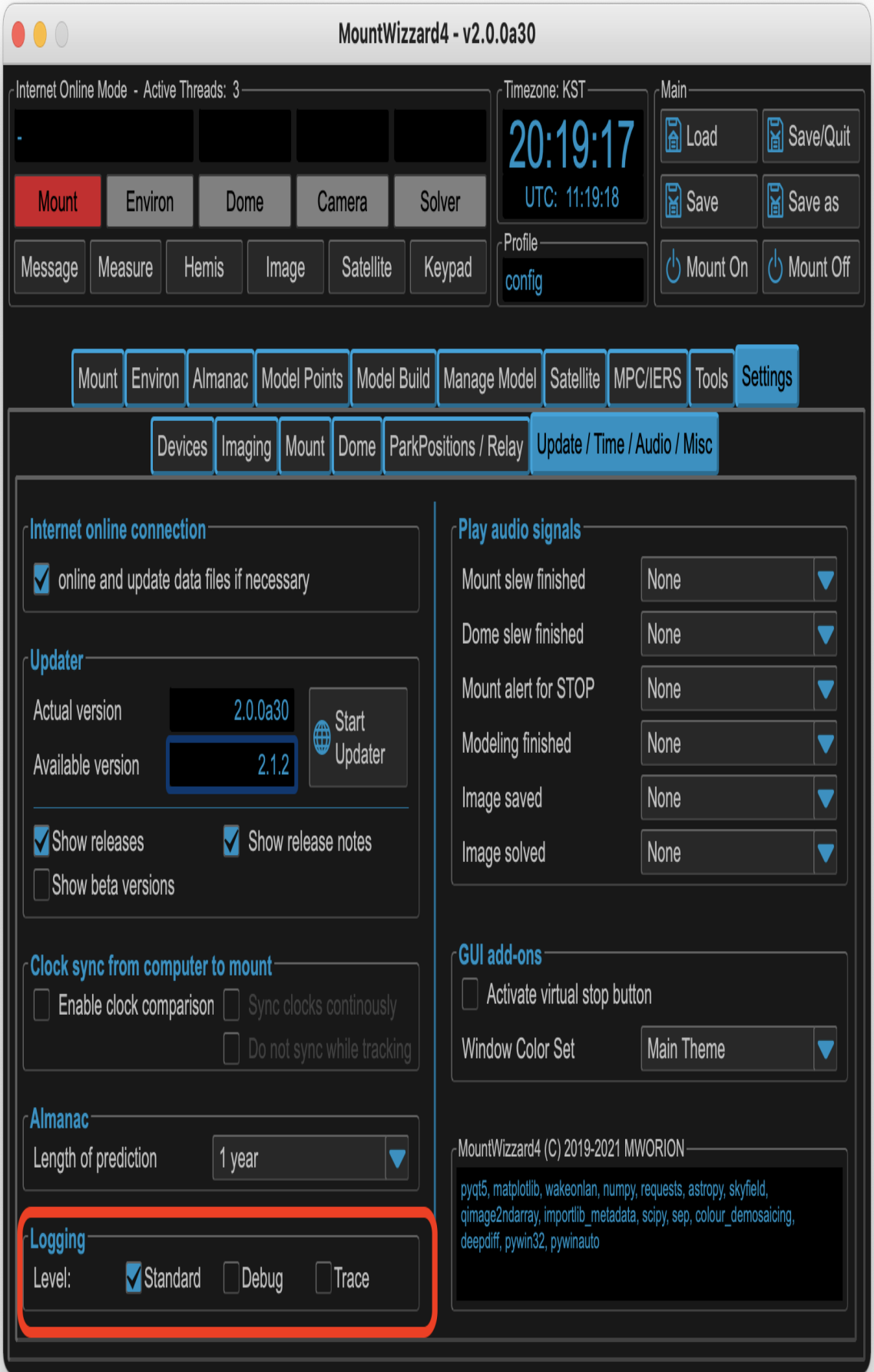
You need an internet connection and set MW4 in online mode to download new data for upload.

Tracking satellites

Logfiles and reports

Where could I change the log level ?

The log level could be changed under settings misc. The default setting is warning. If you need analyzes, please go to debug. If a driver or mount connectivity is related as well, please go to trace. Please be aware that log file especially in trace mode could become big.

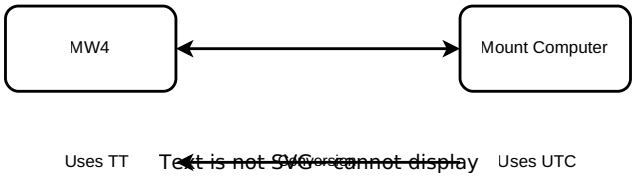

MountWizzard4 - v2.0.0a30
Internet Online Mode - Active Threads: 3
Mount
Environ
Dome
Camera
Solver
Message
Measure
Hemis
Image
Satellite
Keypad
Timezone: KST
20:19:17
UTC: 11:19:18
Profile
config
Main
Load
Save/Quit
Save
Save as
Mount On
Mount Off
Mount
Environ
Almanac
Model Points
Model Build
Manage Model
Satellite
MPC/IERS
Tools
Settings
Devices
Imaging
Mount
Dome
ParkPositions / Relay
Update / Time / Audio / Misc
Internet online connection
online and update data files if necessary
Updater
Actual version
2.0.0a30
Available version
2.1.2
Start Updater
Show releases
Show release notes
Show beta versions
Clock sync from computer to mount
Enable clock comparison
Sync clocks continously
Do not sync while tracking
Almanac
Length of prediction
1 year
Logging
Level:
Standard
Debug
Trace
Play audio signals
Mount slew finished
None
Dome slew finished
None
Mount alert for STOP
None
Modeling finished
None
Image saved
None
Image solved
None
GUI add-ons
Activate virtual stop button
Window Color Set
Main Theme
MountWizzard4 (C) 2019-2021 MWORION
pyqt5, matplotlib, wakeonlan, numpy, requests, astropy, skyfield, qimage2ndarray, importlib_metadata, scipy, sep, colour_demosaicing, deepdiff, pywin32, pywinauto

Architectural topics and math

Within these pages I would explain how and why I made the architecture decisions for linking it with the 10micron mount computer. This might help for setting up or just explain the behavior you experience when using MountWizzard4. I do this also as my development documentation. There might be some faults and error in it. If you find one, please let me know. I would like to get MountWizzard4 from it's technical base as clean as possible.

Handling time

One basic definition is that MountWizzard4 will use at any time the clock of the mount computer. Therefore MountWizzard4 polls julian date, difference utc - ut1, time sidereal. This allows full sync for any calculation to be made. No time from computer to mount is necessary, but could be done at any time (except during model build run). The mount mostly use the julian date representation except for model build where a local sidereal time (LST) is used. In this case MountWizzard4 just stores the value and feed it back when the model is programmed. That's the reason why you should not change time during model run.



One important difference between MountWizzard4 and Mount exists. As I use skyfield as on of the frameworks with it's units for Angle, Coords, Time etc. I have to take the time definition of skyfield into account. Skyfield chooses TT (Terrestrial Time) as it's basic concept, whereas the mount uses UTC (Coordinated Universal Time) as reference. TT is a modern astronomical time standard defined by the International Astronomical Union. TT is distinct from the time scale often used as a basis for civil purposes, UTC. TT is indirectly the basis of UTC, via International Atomic Time (TAI).

Precision of internal calculations

MountWizzard4 is using for all calculations the skyfield (https://rhodesmill.org/skyfield/) from Brandon Rhodes. As for the new command set offered with 10microns FW3.x it needs to calculate the alt/az coordinates for a satellite track each second for the entire track. As you would like to follow the as precise as possible I made some comparisons between the internal calculations done in 10micron mount and the results provided by skyfield.

In skyfield there is a chapter about satellite calculations and precision: https://rhodesmill.org/skyfield/earth-satellites.html#avoid-calling-the-observe-method Despite the fact that the observe method is expensive the difference in calculation time for a 900 step track is on my computer 120ms (using more precise observe method) to 7ms (using the less precise difference).

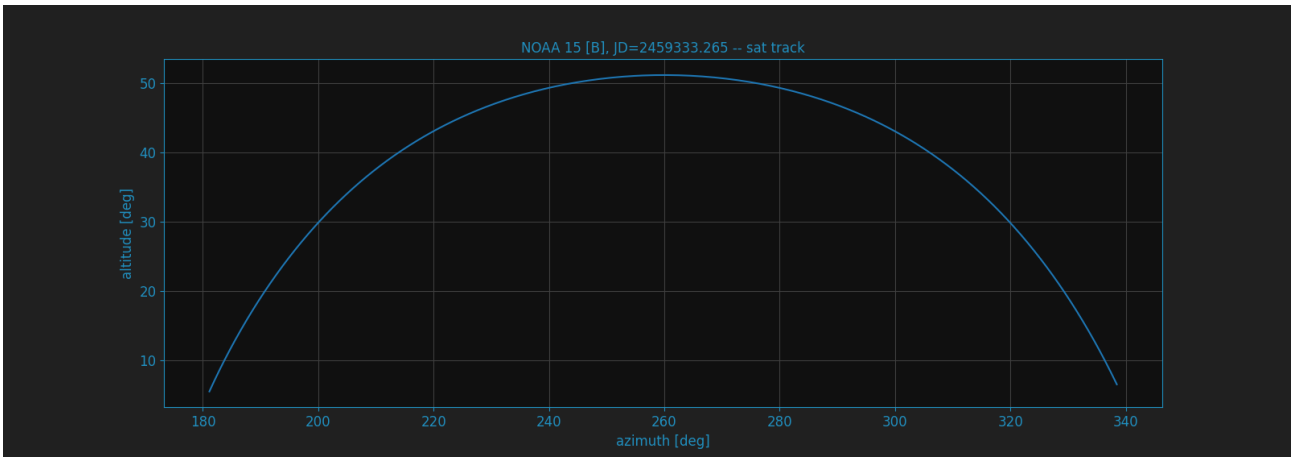
Brandon writes about it:

> While satellite positions are only accurate to about a kilometer anyway, accounting for light travel time only affected the position in this case by less than an additional tenth of a kilometer. This difference is not meaningful when compared to the uncertainty that is inherent in satellite positions to begin with, so you should neglect it and simply subtract GCRS-centered vectors instead as detailed above.

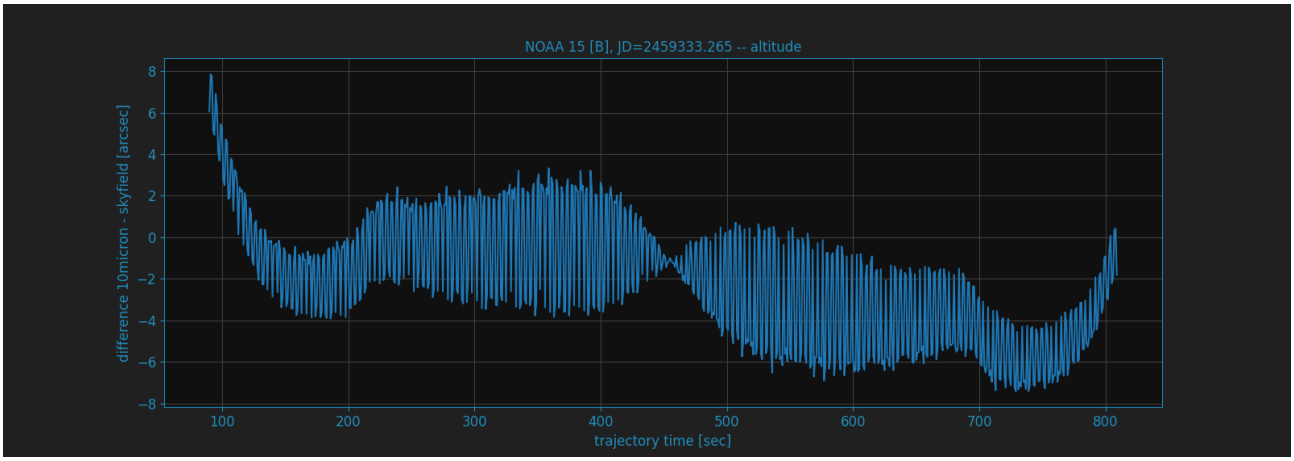
Here the charts for NOAA 15 [B] at julian date JD=2459333.26498 for the transit happening. The used TLE data was:

```
NOAA 15 [B]
1 25338U 98030A   21104.44658620  .00000027  00000-0  29723-4 0  9990
2 25338  98.6888 133.5239 0011555 106.3612 253.8839 14.26021970192127
```

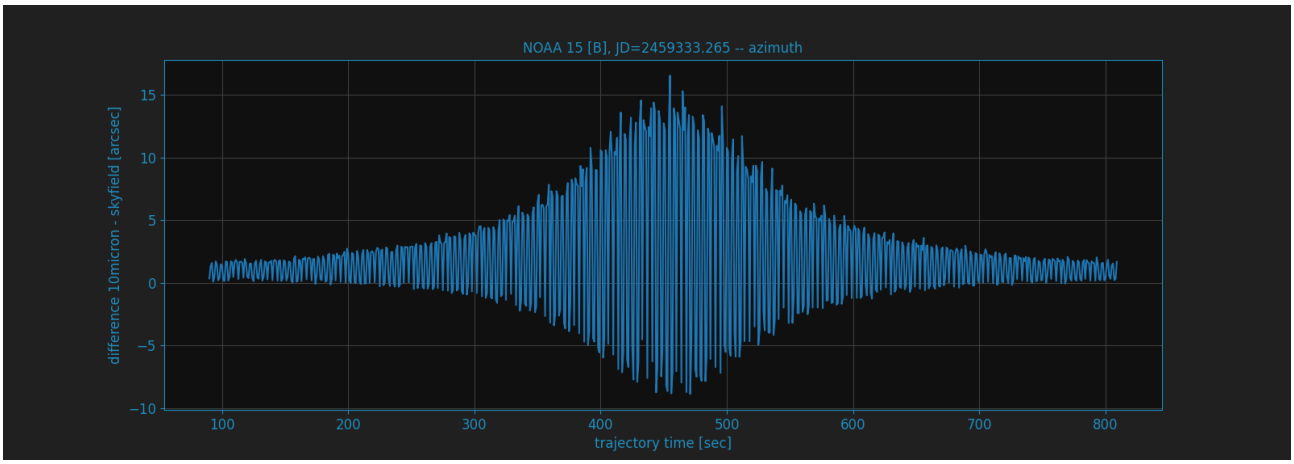
You could see the alt/az of the sat track.



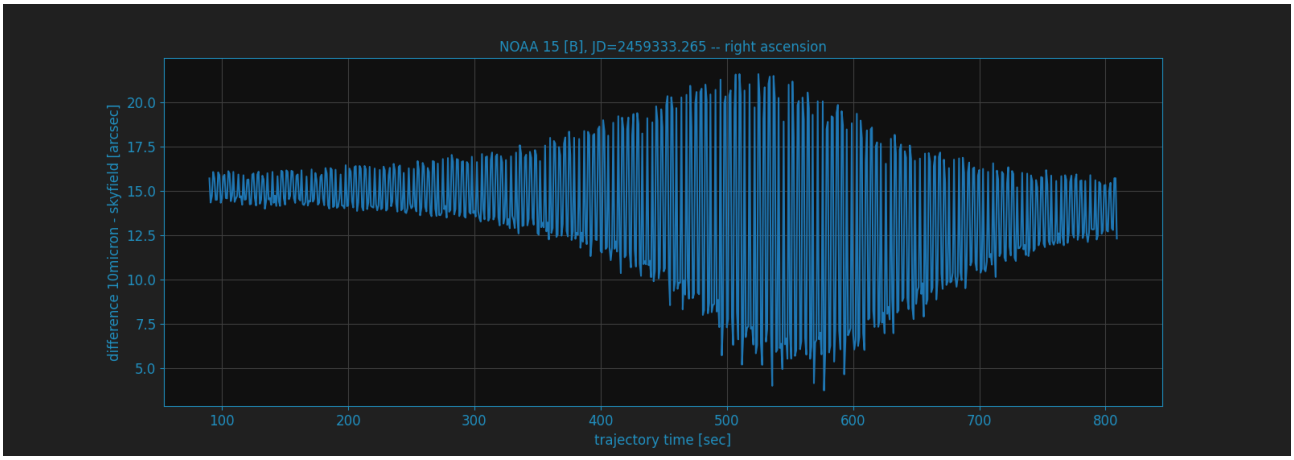
the difference for altitude between 10micron and skyfield



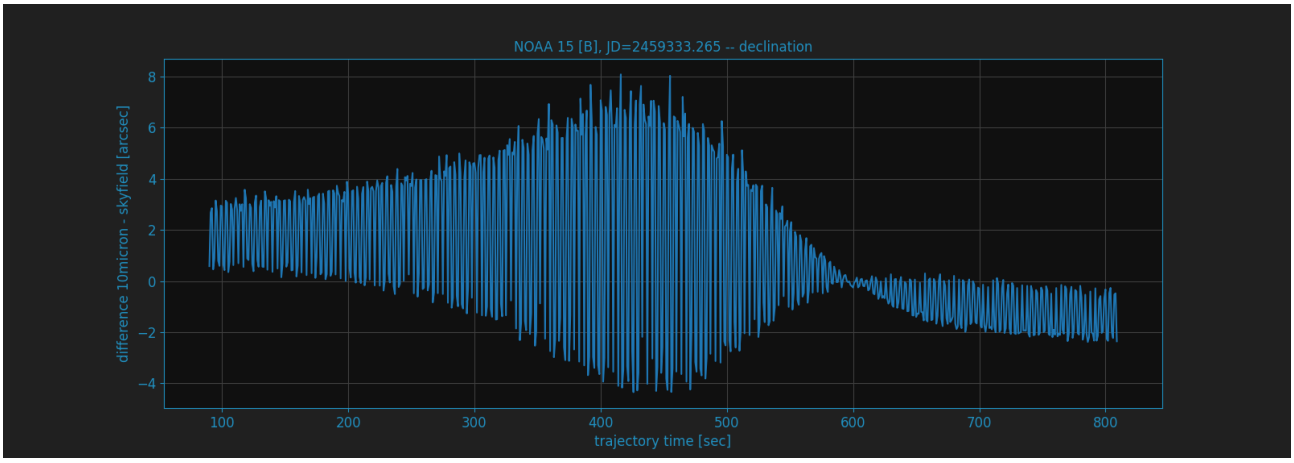
the difference for azimuth between 10micron and skyfield



the difference for right ascension between 10micron and skyfield



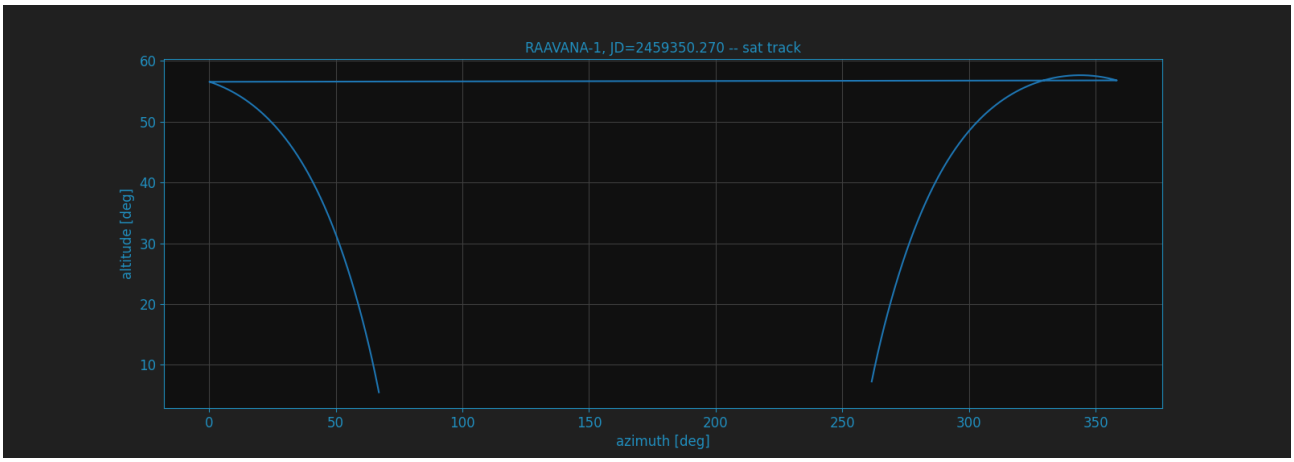
the difference for declination between 10micron and skyfield



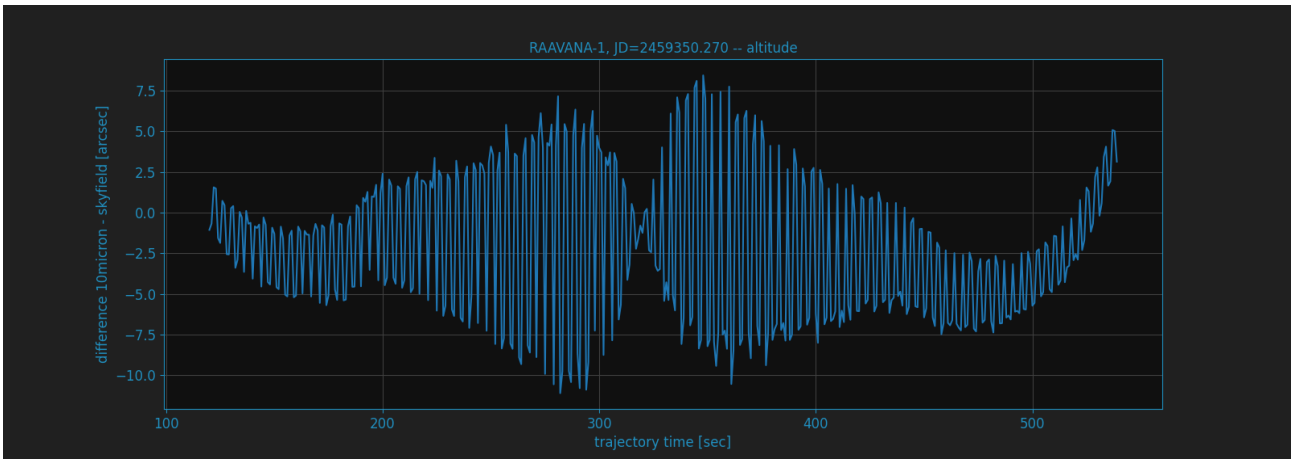
There is a set of plots for another satellite, which shows the same behavior. The used TLE data was:

```
RAAVANA-1
1 44329U 98067QE  21134.29933328  .00044698  00000-0  30736-3 0  9995
2 44329  51.6342 100.9674 0004554 122.3279 237.8162 15.74179130108776
```

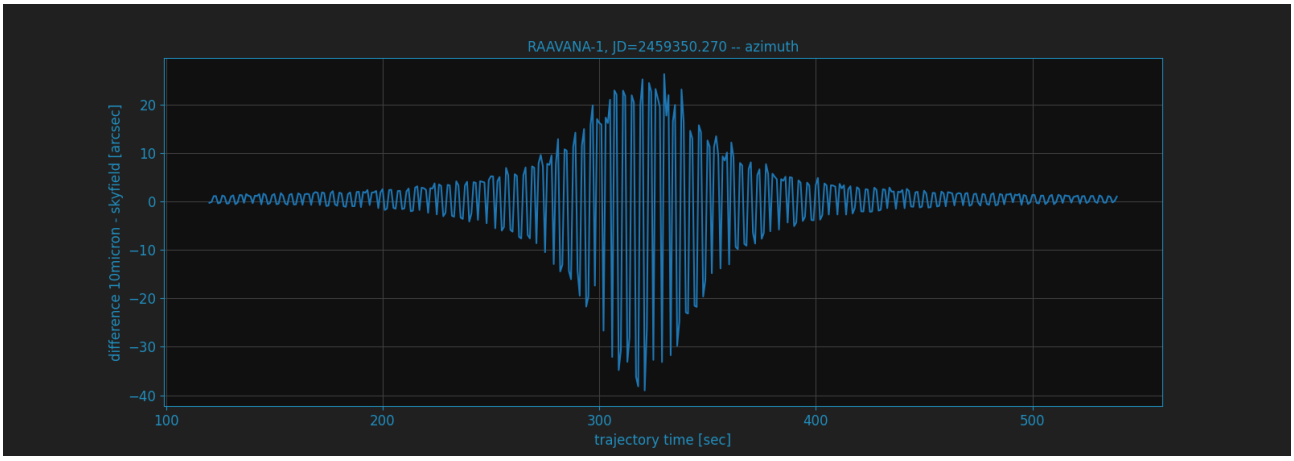
You could see the alt/az of the sat track.



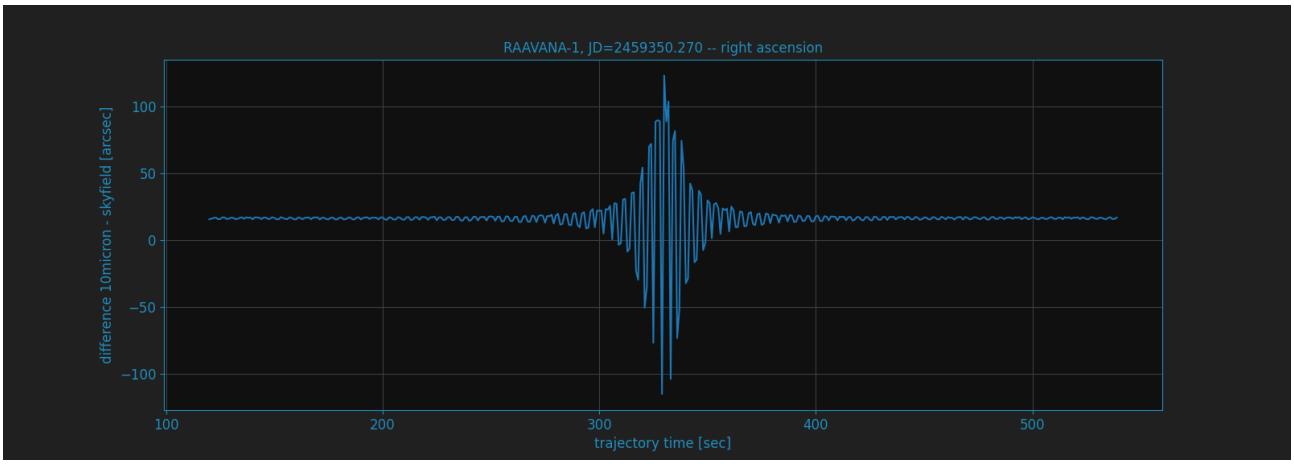
the difference for altitude between 10micron and skyfield



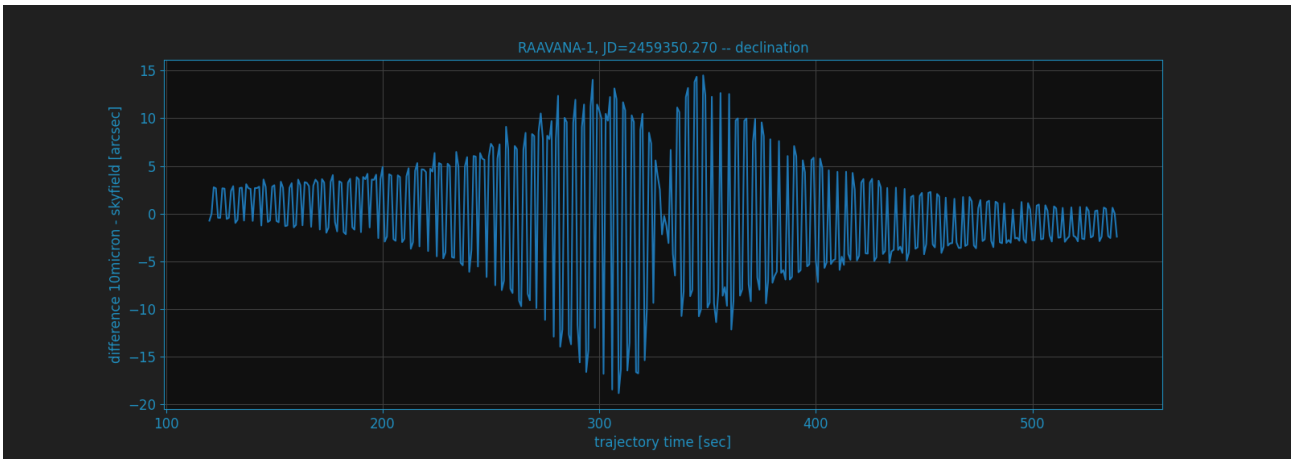
the difference for azimuth between 10micron and skyfield



the difference for right ascension between 10micron and skyfield



the difference for declination between 10micron and skyfield



For all calculations is valid:

- they are using refraction correction with the same values.
- the coordinates from 10micron are gathered with :TLEGEQJD#, :TLEGAZJD# commands
- julian date is in UTC time system
- 10micron firmware 3.0.4

- skyfield version 1.39

Changelog

Beta versions

Version 4.x

4.0.0b0

- this is a major release ! please try out in a separate work folder for test!
- add: support for python 3.11 and 3.12, remove 3.7, 3.8 and 3.9
- add: support automation with 10micron webservices
- add: support uploading mechanism for databases for macos and linux
- change: faster and more reliable uploading mechanism for databases
- change: celestrak interface url's and retrieval strategy
- change: moving PyQt5 to major version PySide6
- change: moving libraries to latest versions
- change: remove old windows automation
- change: remove embedded documentation and replace with online link
- refactoring: environment devices (now 3 generic ones)
- refactoring: remove installer to separate repo (InstallerMW4)
- improve: reduced size of app
- improved: don't delete message list when color change
- remove: automation of 10micron installer
- fix: typos

Released versions

Version 3.x

3.2.6

- add: support for INDI Pegasus Uranus Meteo sensor
- add: wait time after slew finished before exposing
- change: writing pointing coordinates to fits header from MW4 now
- change: celestrak interface url's and retrieval strategy
- improve: add waiting time for image file save for NINA and SGPro
- improve: logging for NINA / SGPro controlled cameras
- improve: gain handling when missing values in camera settings
- improve: lower the dome radius to 0.8m
- fix: typos and some minor bugs

3.2.5

- improve: add more information to the log file seeing
- improve: openweathermap data handling (API)
- improve: add support for pegasus uranus meteo sensor

3.2.4

- add: support for astap D80 database
- improve: more robust implementation against touptek drivers
- improve: add more information to the log file seeing
- improve: openweathermap data handling (API)

3.2.3

- fix: correct editing points, when slew path is not selected
- improve: sort horizon points when loading a file

3.2.2

- change: switch from forecast to weather api on openweathermap

3.2.1

- fix: change humidity and dewpoint value in driver as there were mixed up

3.2.0

- add: editable mount settling time for 10micron box (UI change!)
- add: waiting time used w/o meridian flip
- add: bring “keep scale” when doing exposeN
- improve: some refactoring for speed
- improve: watney checking allows for multiple sets in one directory

3.1.0

Version 3.1 brings aarch64 support for arm back if using the new installer 3.1

- add: support for aarch64 on raspi for python 3.8 - 3.10 (needs installer 3.1)
- add: support for ASTAP new databases D50, D20, D05
- improve: speedup launch if INDI server not ready
- improve: support for catalina
- improve: ParkPos with 2 digits precision
- fix: download sources IERS
- fix: switching UTC / local times
- fix: seeing entries visibility upon startup

3.0.1

- fix: ASCOM cover: brightness status.
- fix: ASCOM cover: setting / reading brightness / max brightness
- fix: almanac: text for “rise” and “set” were mixed
- fix: DNS resolving
- improve: add a hint for optimal binning to keep reasonable image sizes

- improve meteoblue behavior: correct text and undisplayed if disabled
- improve minor planets selection: adding multiple selection by mouse
- improve refraction: when selecting internal sensor, go to automatic

3.0.0

Version 3.0 is a major release! Please update with care! No ARM7 support / ARM64 only Python 3.8 - 3.9

- add: GUI: all charts could be zoomed and panned
- add: GUI: all tab menu entries could be customized in order and stored /reset
- add: GUI: all open windows could be collected to visual area
- add: GUI: separate window with big buttons are available
- add: GUI: reduced GUI configurable for a simpler user interface
- add: video: support for up to 4 external RTSP streams or local cameras
- add: video: adding authentication to video streams
- add: video: adding support for HTTP and HTTPS streams
- add: almanac: now supports UTC / local time
- add: almanac: support set/rise times moon
- add: environment: integrate meteoblue.com seeing conditions
- add: analyse: charts could show horizon and values for each point
- add: analyse: alt / az charts with iso 2d contour error curves
- add: audio: sound for connection lost and sat start tracking
- add: model points: multiple variants for edit and move points
- add: model points: set dither on celestial paths
- add: model points: generate from actual used mount model
- add: model points: existing model files could be loaded
- add: model points: golden spiral with exact number of points
- add: polar align: adding hint how to use the knobs measures right

- add: plate solve: new watney astrometry solver for all platforms
- add: hemisphere: selection of terrain file
- add: hemisphere: show actual model error in background
- add: hemisphere: edit horizon model much more efficient
- add: hemisphere: show 2d contour error curve from actual model
- add: hemisphere: move point with mouse around
- add: dome: control azimuth move CW / CCW for INDI
- add: satellites: all time values could be UTC or local time now
- add: MPC / IERS: adding alternative server for download
- add: measure: window has max 5 charts now (from 3)
- add: measure: more values (time delta, focus, cooler power, etc.)
- add: image: photometry functions (aberration, roundness, etc.)
- add: image: tilt estimation like ASTAP does as rectangle and triangle
- add: image: add flip H and flip V
- add: image: show RA/DEC coordinates in image if image was solved
- add: image: center mount pointing g to any point in image by mouse double click
- add: image: center mount pointing to image center
- add: image: support for reading XISF files (simple versions)
- add: imaging: separate page for imaging stats now
- add: imaging: stats: calcs for plate solvers (index files etc.)
- add: imaging: stats: calcs for critical focus zones
- add: drivers: polling timing for drivers could be set
- add: drivers: game controller interface for mount and dome
- add: system: support for python 3.10
- add: help: local install of documentation in PDF format
- add: profiles: automatic translation from v2.2.x to 3.x

- improve: GUI: layout for main window optimized and consistent and wording updates
- improve: GUI: complete rework of charting: performance and functions
- improve: GUI: clean up and optimize IERS download messages
- improve: GUI: get more interaction bullet prove for invalid cross use cases
- improve: GUI: moved on / off mount to their settings: avoid undesired shutoff
- improve: GUI: show twilight and moon illumination in main window
- improve: INDI: correcting setting parameters on startup
- improve: model points: optimized DSO path generation (always fit, less params)
- improve: model run: refactoring
- improve: model run: better information about status and result
- improve: hemisphere: improve solved point presentation (white, red)
- improve: plate solve: compatibility checks
- improve: system: all log files will be stored in a separate folder /log
- improve: system: enable usage of python 3.10
- improve: system: use latest PyQt5 version
- improve: system: adjust window sizes to be able to make mosaic layout on desktop
- improve: system: moved to actual jpl kernel de440.bsp for ephemeris calcs
- remove: system: matplotlib package and replace with more performant pyqtgraph
- remove: system: PIL package and replace with more powerful cv2
- remove: system: move from deprecated distutils to packaging
- remove: system: support for python 3.7 as some libraries stopped support
- remove: imageW: stacking in imageW as it was never used
- remove: testing support for OSx Mojave and OSx Catalina (still should work)
- fix: drivers: device selection tab was not properly positioned in device popup

Version 2.x

2.2.9

- fix: internal updater shows only alpha versions instead of betas

2.2.8

- fix: updates for supporting newer ASTAP versions
- fix: model run will cancel if solving fails
- fix: workaround ASTAP FITS outputs which are not readable via astropy
- update ephemeris file

2.2.7

- fix: text labels
- fix: getting min / max values from indi devices
- fix: updates for supporting newer ASTAP versions
- fix: model run will cancel if solving fails

2.2.6

- fix: reduce load in debug trace mode
- fix: model process stalls in some cases in normal mode
- fix: text labels
- fix: getting min / max values from indi devices

2.2.5

- fix: reduce load in debug trace mode
- fix: model process stalls in some cases in normal mode

2.2.4

- fix: remove race condition for large image file causing solve error in ASTAP
- fix: reduce load in debug trace mode

2.2.3

- fix: mount orientation in southern hemisphere

2.2.2

- fix: almanac moon phase drawing error

2.2.1

- update: builtin data for finals200.all
- fix: download iers data: fix file not found feedback

2.2.0

- add: support SGPro camera as device
- add: support N.I.N.A. camera as device
- add: two modes for SGPro and N.I.N.A.: App or MW4 controlled
- add: debayer (4 modes) all platforms (armv7, StellarMate, Astroberry)
- add: filter satellites for twilight visibility settings
- add: setting performance for windows automation (slow / normal / fast)
- add: auto abort imaging when camera device is disconnected
- add: missing cursor in virtual keypad window
- add: support for keyboard usage in virtual keypad window
- add: screenshot as PNG save for actual window with key F5
- add: screenshots as PNG save for all open windows with key F6
- add: query DSO objects for DSO path setting in build model
- improved: flexible satellite handling when mount not connected
- improved: show selected satellite name in satellite windows title
- improved: 3D simulator drawing
- improved: updater now avoids installation into system package
- improved: GUI for imaging tab - disable all invalid interfaces
- improved: redesign analyse window to get more space for further charts
- improved: Tools: move mount: better UI, tooltips, multi steps in alt/az
- improved: gui in image window when displaying different types

- improved: reduced memory consumption if display raw images
- improved: defining park positions with digit and improve gui for buttons
- improved: when pushbutton shows running, invert icons as well
- improve: moon phases in different color schemes
- upgrade: pywin32 library to version 303 (windows)
- upgrade: skyfield library to 1.41
- upgrade: numpy library to 1.21.4
- upgrade: matplotlib to 3.5.1
- upgrade: scipy library to 1.7.3
- upgrade requests library to 2.27.2
- upgrade importlib_metadata library to 4.10.0
- upgrade deepdiff library to 5.7.0
- upgrade wakeonlan library to 2.1.0
- upgrade pybase64 library to 1.2.1
- upgrade websocket-client library to 1.2.3
- fix: simulator in southern hemisphere

2.1.7

- add: 12 build point option for model generation
- add: grouping updater windows upper left corner
- add: support for languages other than english in automation
- add: minimize cmd window once MW4 is started
- fix: KMTronic Relay messages

2.1.6 - add: explicit logging of automation windows strings for debug - add: showing now detected updater path and app - revert: fixes for german as they do not work

2.1.5

- fix: checking windows python version for automation

2.1.4

- add: enabled internal updater for astroberry and stellarmate
- add: temperature measurement for camera
- improved: logging for ASCOM threading
- improved: image handling
- fix: DSLR camera devices

2.1.3

- add: config adjustments for astroberry and stellarmate devices (no debayer)
- improved: logging for UI events

2.1.2

- fix: non connected mount influences camera on ASCOM / ALPACA
- fix: logging string formatting

2.1.1

- fix: for arm64 only: corrected import for virtual keypad
- fix: arrow keys on keypad did accept long mouse press

2.1.0

- add: hemisphere window: help for choosing the right star for polar alignment
- add: hemisphere terrain adjust for altitude of image beside azimuth
- add: angular error ra / dec axis in measurement
- add: device connection similar for ASCOM and ALPACA devices
- add: extended satellite search and filter capabilities (spreadsheet style)
- add: estimation of satellite apparent magnitude
- add: extended satellite tracking and tuning capabilities
- add: enabling loading a custom satellite TLE data file
- add: command window for manual mount commands
- add: sorting for minimal dome slew in build point selection

- add: setting prediction time of almanac (shorter reduces cpu load)
- add: providing 3 different color schemes
- add: virtual keypad available for RPi 3/4 users now
- improve: check if satellite data is valid (avoid error messages)
- improve: better hints when using 10micron updater
- improve: simplified signals generation
- improve: analyse window plots
- improve: rewrite alpaca / ascom interface
- improve: gui for running functions
- improve: test coverage
- remove: push time from mount to computer: in reliable and unstable
- fix: segfault in qt5lib on ubuntu

2.0.6

- fixes

2.0.5

- fix: bug when running “stop exposure” in ASCOM

2.0.4

- improvement: GUI for earth rotation data update, now downloads
- improvement: performance for threads.
- improvement: added FITS header entries for ALPACA and ASCOM
- fix: removed stopping DAT when starting model

2.0.3

- improvement: GUI for earth rotation data update, now downloads
- improvement: performance for threads.

2.0.2

- fix: robustness against errors in ALPACA server due to memory faults #174

- fix: robustness against filter names / numbers from ALPACA server #174
- fix: cleanup import for pywinauto timings import #175
- improvement: avoid meridian flip #177
- improvement: retry numbers as int #178

2.0.1

- fix: MW4 not shutting down when dome configured, but not connected
- fix mirrored display of points in polar hemisphere view

2.0.0

- add new updater concept
- add mount clock sync feature
- add simulator feature
- add terrain image feature
- add dome following when mount is in satellite tracking mode
- add dome dynamic following feature: reduction of slews for dome
- add setting label support for UPB dew entries
- add auto dew control support for Pegasus UPB
- add switch support for ASCOM/ALPACA Pegasus UPB
- add observation condition support for ASCOM/ALPACA Pegasus UPB
- add feature for RA/DEC FITS writing for INDI server without snooping
- add completely revised satellite tracking menu gui
- add partially satellite tracking before / after possible flip
- add satellite track respect horizon line and meridian limits
- add tracking simulator feature to test without waiting for satellite
- add alt/az pointer to satellite view
- add reverse order for failed build point retry
- add automatic enable webinterface for keypad use

- add broadcast address and port for WOL
- add new IERS and lead second download
- add more functions are available without mount connected
- add change mouse pointer in hemisphere
- add offset and gain setting to imaging
- add disable model point edit during model build run
- update debug standard moved from WARN to INFO
- update underlying libraries
- update GUI improvements
- fix for INDI cameras sending two times busy and exposure^0
- fix slewing message dome when disconnected
- fix retry mechanism for failed build points
- fix using builtins for skyfield and rotation update
- fix plate solve sync function

Version 1.x

1.1.1

- adding fix for INDI cameras sending two times BUSY, EXP^0

1.1.0

- adding release notes showing new capabilities in message window
- adding cover light on / off
- adding cover light intensity settings
- reversing E/W for polar diagram in hemisphere window
- adding push mount time to computer manual / hourly
- adding contour HFD plot to image windows
- adding virtual emergency stop key on time group
- update build-in files if newer ones are shipped

- auto restart MW4 after update
- adding OBJCTRA / OBJCTDEC keywords when reading FITs
- upgrade various libraries