
InstallerMW4
Version 4.0.1

Table of Contents

Welcome to InstallerMW4!

InstallerMW4 is a general utility for installing and starting MountWizzard4 on your computer and it comes also with an PDF documentation:

https://mworion.github.io/InstallerMW4/InstallerMW4.pdf

Before starting

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.

Hint

Please report issues / bugs here:

https://github.com/mworion/InstallerMW4/issues .

Hint

Feature requests and discussions or for all other topics of interest there is a good place to start here:

https://github.com/mworion/InstallerMW4/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.

If you on the way of installing MountWizzard4 to your windows system, please be aware of the 32bit / 64bit limitations of ASCOM / drivers and python. If you are using 64bit drivers (most likely with the new large scale CMOS cameras), you need to install 64bit python as well as windows does not mix both variants flawless.

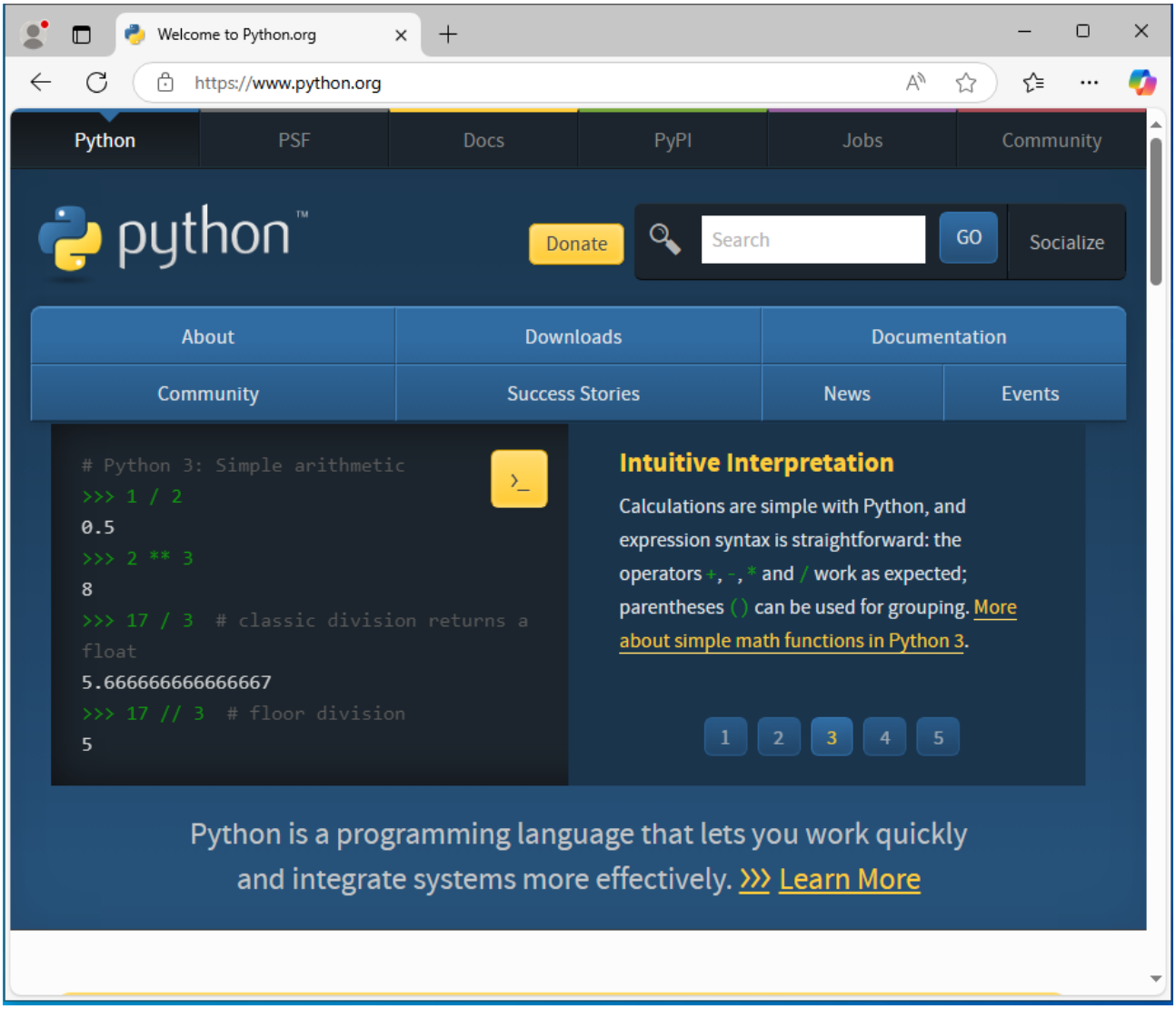
Warning

I strongly recommend not using whitespace in filenames or directory paths. Especially in windows handling them is not straight forward and I hardly could do all the tests needed to ensure it's functionality.

Install Python3

MountWizzard4 is a python3 application based on some python libraries and uses Qt5/6 as framework for GUI. Different to past versions of MountWizzard4 there will be no one box solution (MAC bundle, EXE File, etc.) available. As MountWizzard4 is python3 and comes with internal update functionality, it uses a standard python3 environment. Ideally it is recommended in a virtualenv.

The first step is to install the appropriate python3 package if not already installed. For all platforms installer are available. Please look here:



And the link to the python website: https://www.python.org .

Please follow the descriptions that comes with the installers. To give a short overview here are some quick installation hints for all platforms.

Warning

Please do not use a newer version of python than recommended for the MountWizzard4 version you would like to use. Some libraries bring precompiled binaries with them and they might not be available for a newer python version.

If there is already an appropriate python3 installed, you can skip this section and go directly to the MountWizzard4 installation process. If you have to install python3 this has to be done only once for as many MountWizzard4 installations you might want.

There are videos for multiple use cases on the youtube channel: https://www.youtube.com/channel/UCJD-5qdLEcBTCugltqw1hXA

Note

On windows there are some new features which supports comet, earth rotation and asteroids update for the mount. These functions are available from python 3.8.2 on. Earlier python versions have issues. If you would like to upgrade an older python installation, please see the comments below for windows. On other OS there is no need for doing that.

The actual recommended python version for MountWizzard4 is 3.10.

Windows Platform

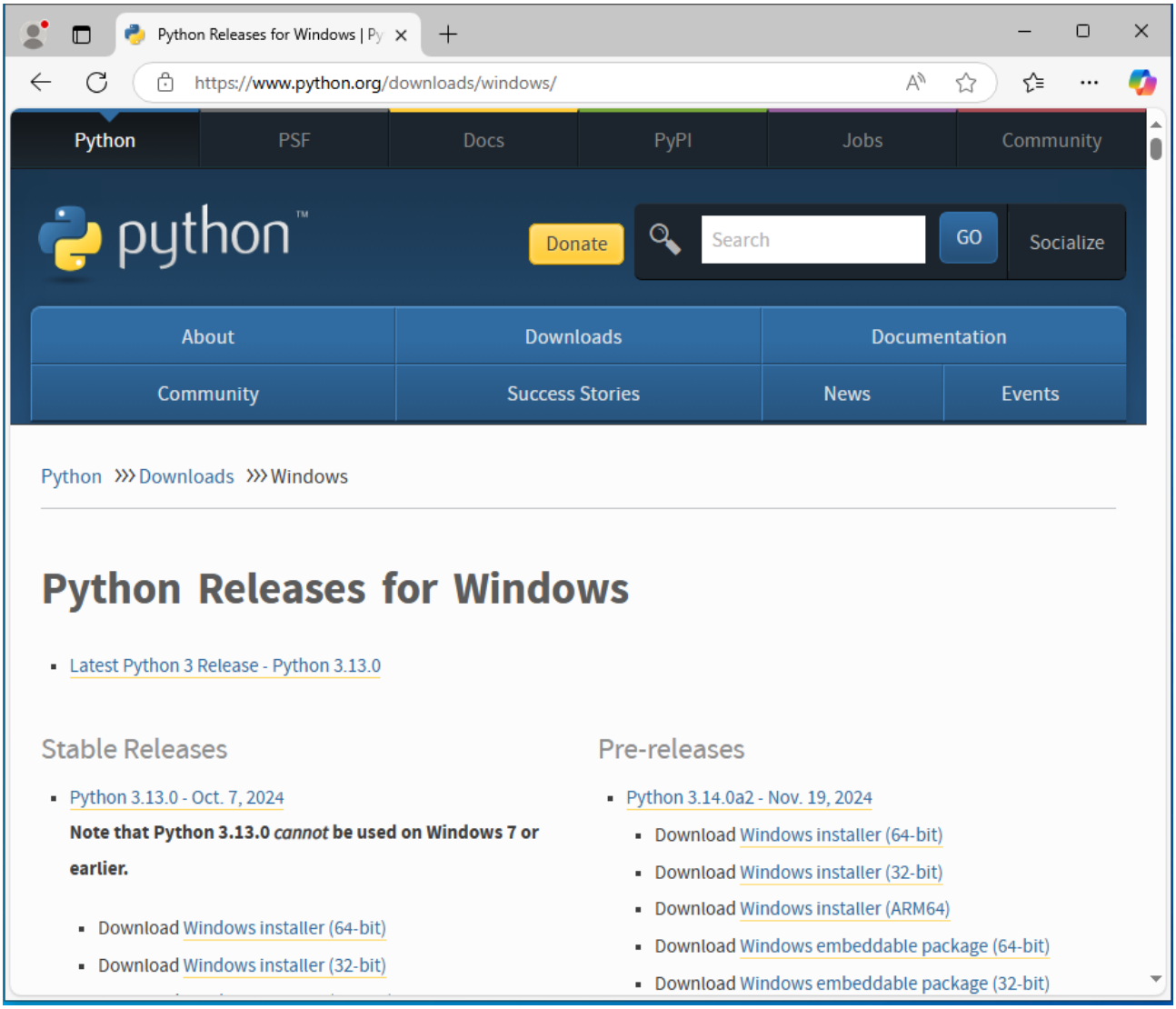
Note

Windows makes a hard split between 32bit and 64bit versions. If your drivers and setup uses 64bit solutions, please install 64bit python!

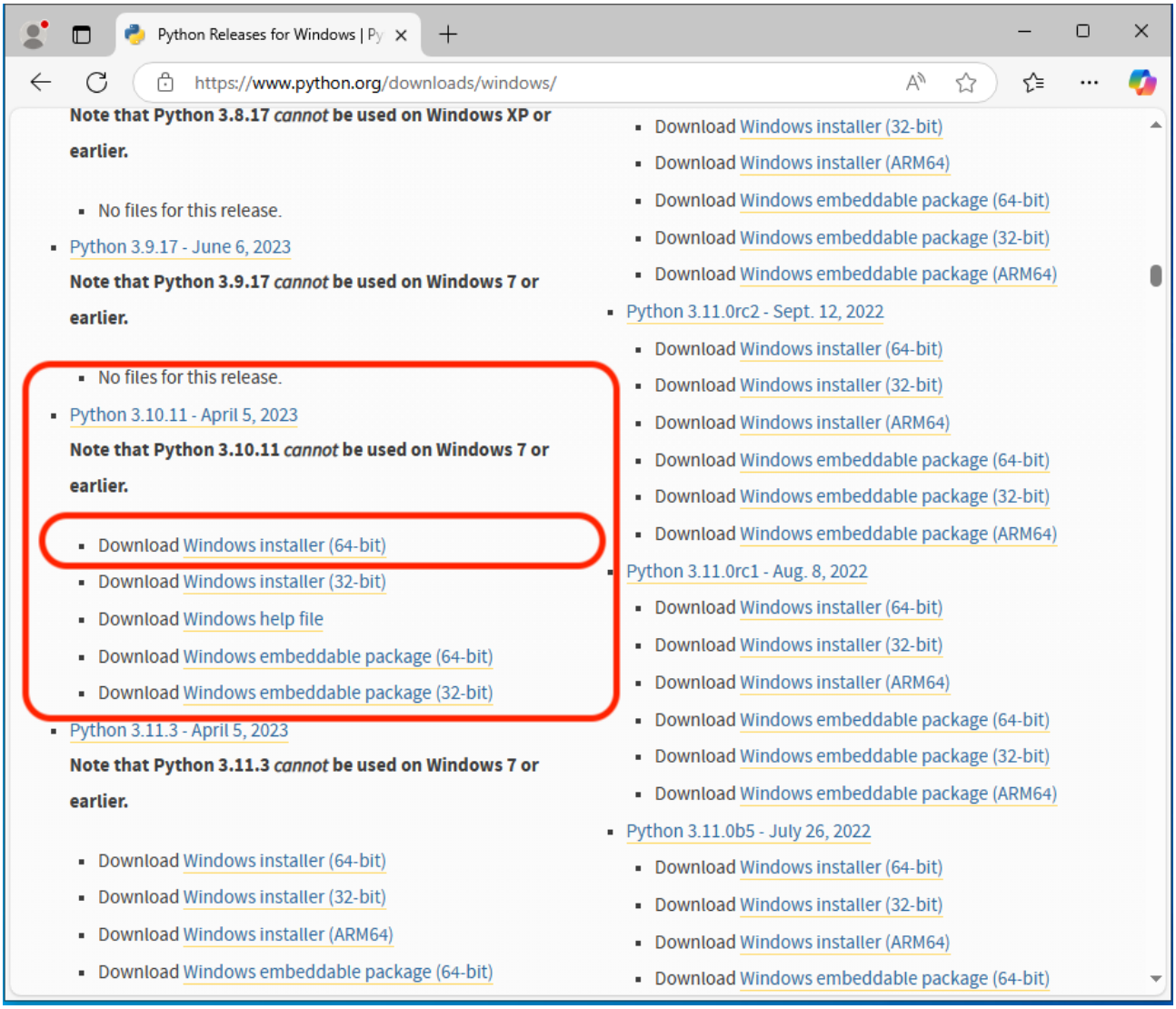
Depending on your Windows version please download the installer for python3 from:

https://www.python.org/downloads/windows/

Switch to the windows part



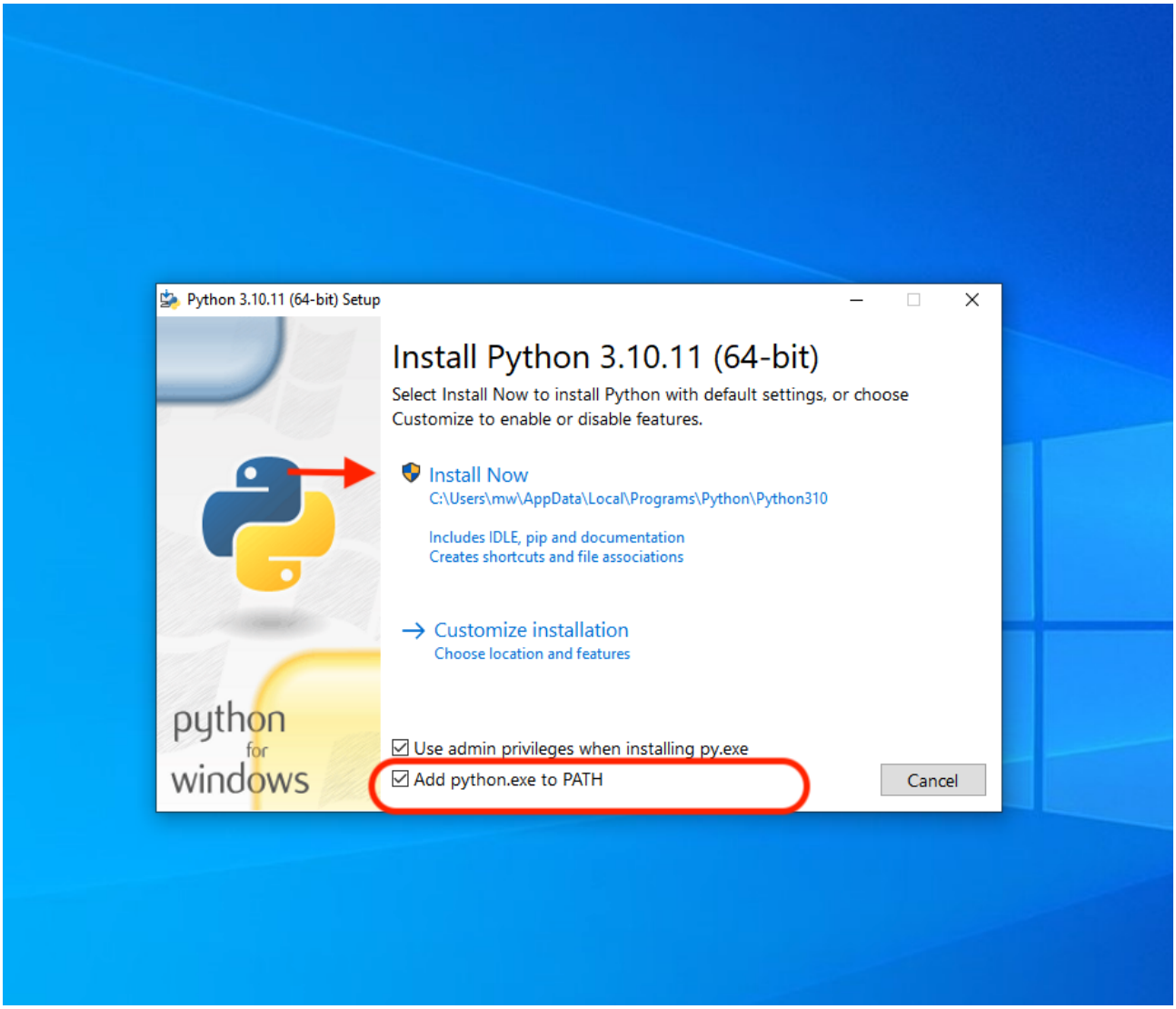
choose the appropriate installer,



and follow the installation procedure.

Hint

Please take care that during the installation the checkbox “Add Python.exe to Path” is selected and to install for a single user if you want to use the scripts.



MountWizzard4 does not need admin rights to run!

Mac OSx platform

Depending on your OSx version please download the installer for python3 from:

https://www.python.org/downloads/mac-osx/

and follow the installation procedure. MountWizzard4 does not need admin rights to run!

Warning

Using a Mac with Apple silicon need special treatment. There is rather any experience with these setups. Actually MountWizzard4 only support Intel architecture so you need to use the Rosetta emulator.

Ubuntu platform

Referring to Ubuntu 20.04 LTS as it comes with python3.8. This should work, but you could upgrade to a newer python3 version. This could be done by adding an appropriate repo, which enables this version.

Hint

If you update to a higher python version, please update to the latest supported python3 version (3.10 for MountWizzard4 3.x and 3.12 for MountWizzard4 4.x). There are many descriptions out, so please search for it in case you don't know exactly.

An example is from: https://linuxize.com/post/how-to-install-python-3-7-on-ubuntu-18-04/

```
sudo add-apt-repository ppa:deadsnakes/ppa
sudo apt-get update
sudo apt-get upgrade
sudo apt-get install python3.10
```

Please check the right version and the availability of virtualenv in your setup. If virtualenv is not present in your setup, please install it prior to run the install scripts with:

```
sudo apt-get install python3-virtualenv
```

Install Python3 expert add-ons

This section is for users who are familiar with python and have some experience.

Updating python in your existing environment

This is a step which should be done if you are familiar with some pc experience. Hence the steps are not complicated, the setups of you environment might be somehow special and need a adjusted treatment. The following steps explain a standard procedure.

Update python version on your windows computer

Please go to the python website an download the appropriate python version. On windows please check the selection of the 32bit or 64bit correctly. It should be the version you have already chosen.

Start the python installer. If everything went right, it will show an update offer. If so, please chose that and you get the upgrade. If you would like to switch from 32bit to 64bit or vice versa, the updater only shows a new install. In this case please deinstall the old version manually. Than it's like a new python3 installation, please see above.

Having your python version updated on you computer, you have to update the new version to you work environment(s), too. There are two ways to do that. First you could use the install script provided and install MW in a new work dir. You could copy all you settings (except the 'venv' folder) to the new workdir. Another way is to open a command window, change to your work directory and run the command:

```
python -m venv --upgrade venv
```

This will upgrade your work environment to the python version of your computer (so the updated one)

Note

Before doing any changes or updates, please do a backup of your environment to be safe in case of errors in the update process. This could simply be done by making a copy of your work folder.

Install MountWizzard4

When starting with the installation of MountWizzard4, python 3.8-3.10 for versions 3.x and python 3.10-3.13 for versions 4.x should be successful installed. To check, open a terminal (available on all platforms) and run the command

```
python --version
virtualenv --version
```

On windows you can't call python3, but you have to run the command

```
python --version
```

In one of the choices you should see the version number of the installed and available packages. For python it should say 3.8.x ... 3.12.x (depending on the version of MountWizzard4). Starting with InstallerMW4 version 4.x, the script does not support installing versions of MountWizzard4 < 3.0.0 anymore.

Hint

MountWizzard4 does not need admin rights to install or run. To avoid problems with accessing directories or file please ensure, that you run install and MountWizzard4 itself as normal user!

To install MountWizzard4 on your computer, there are some support available for Windows, MacOSx and Ubuntu to make it a little bit easier to install and run MountWizzard4. The scripts are online, and available from Github.

Installing with installer version 4.x:

The install procedure also got improved: You will have only a single compressed python script (startup.pyz) which is valid for all platforms and does all things the different existing scripts stand for. Additionally some support files are present to make for Linux a good starter Please download the package and unzip it to get the content. You will find four files:

- startup.pyz -> the script for doing all the work
- mountwizzard4.desktop -> support for ubuntu / linux running the script
- mw4.png -> icon for mountwizzard4.desktop
- mw4.ico -> icon to customize the link in windows for running the script

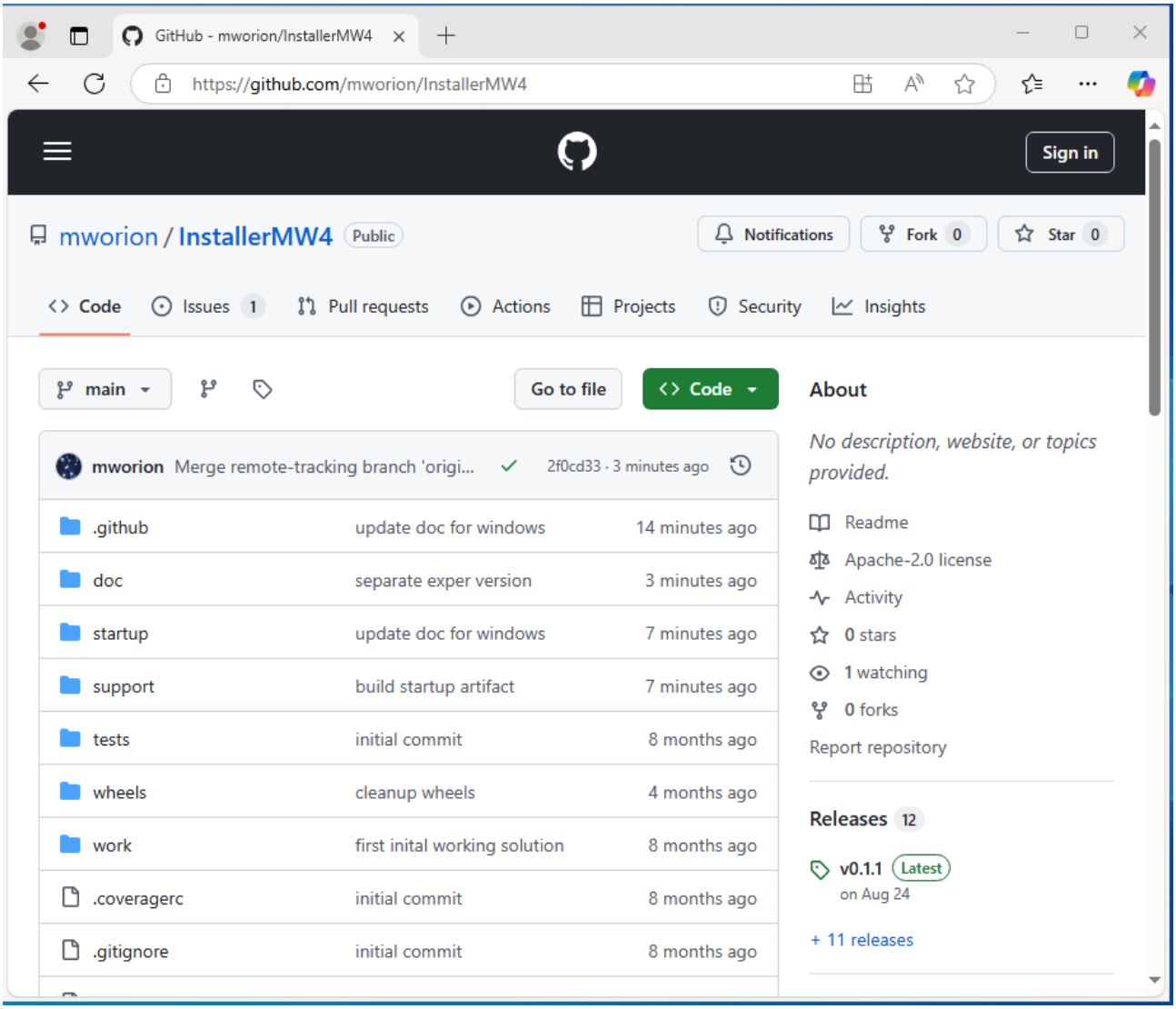
Process for installation

Step 1

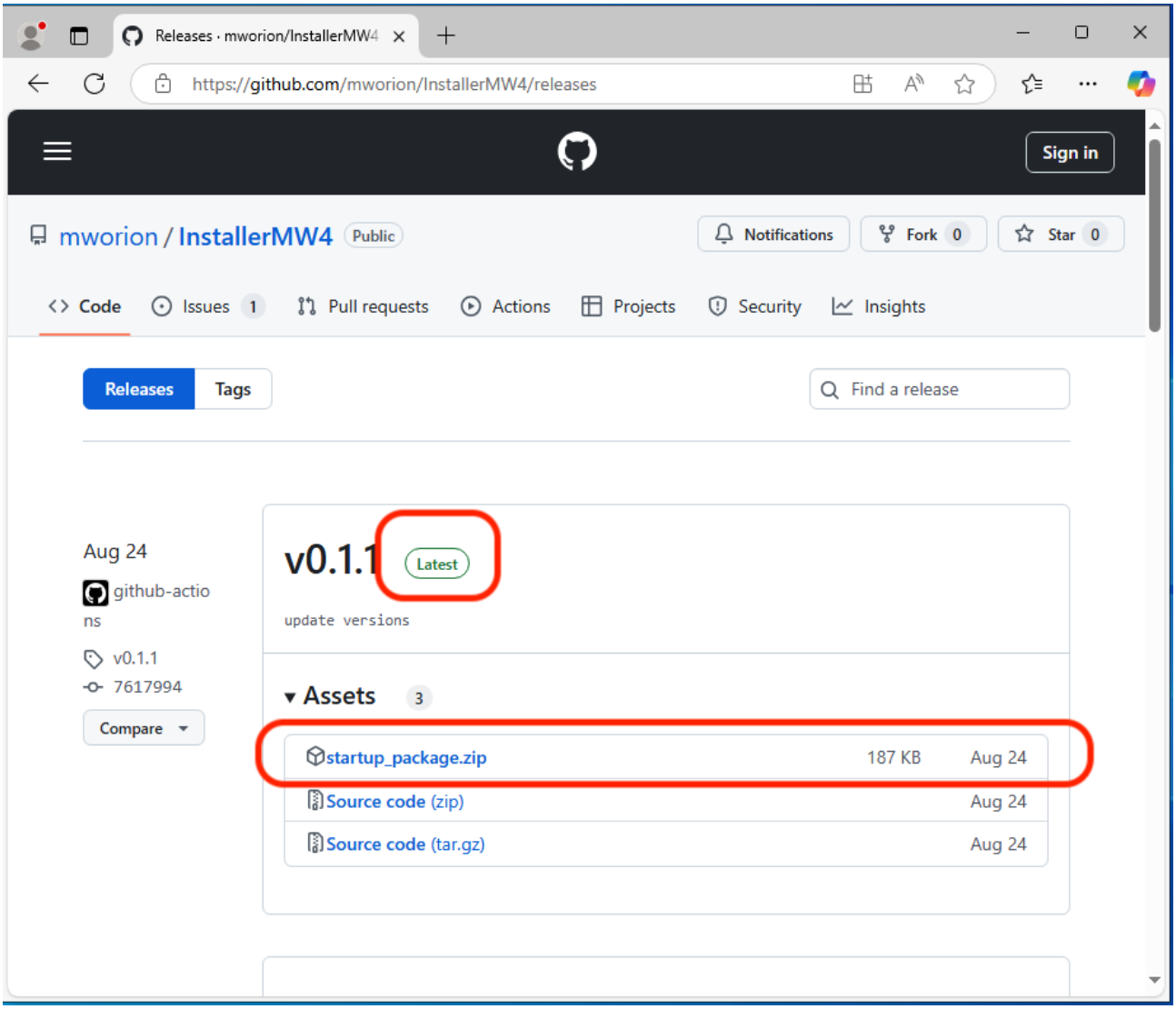
Please create a working directory of your choice and location. For MacOSx I would recommend not using a location on the desktop as it might cause troubles with execution right in newer OSx installations. The directory can be renamed later on, it also can also be moved to any other location. Copy the scripts for your platform into this directory.

Step 2

Download the **lastest** release of the installer script and unzip it's content to your work dir. You will find it here:

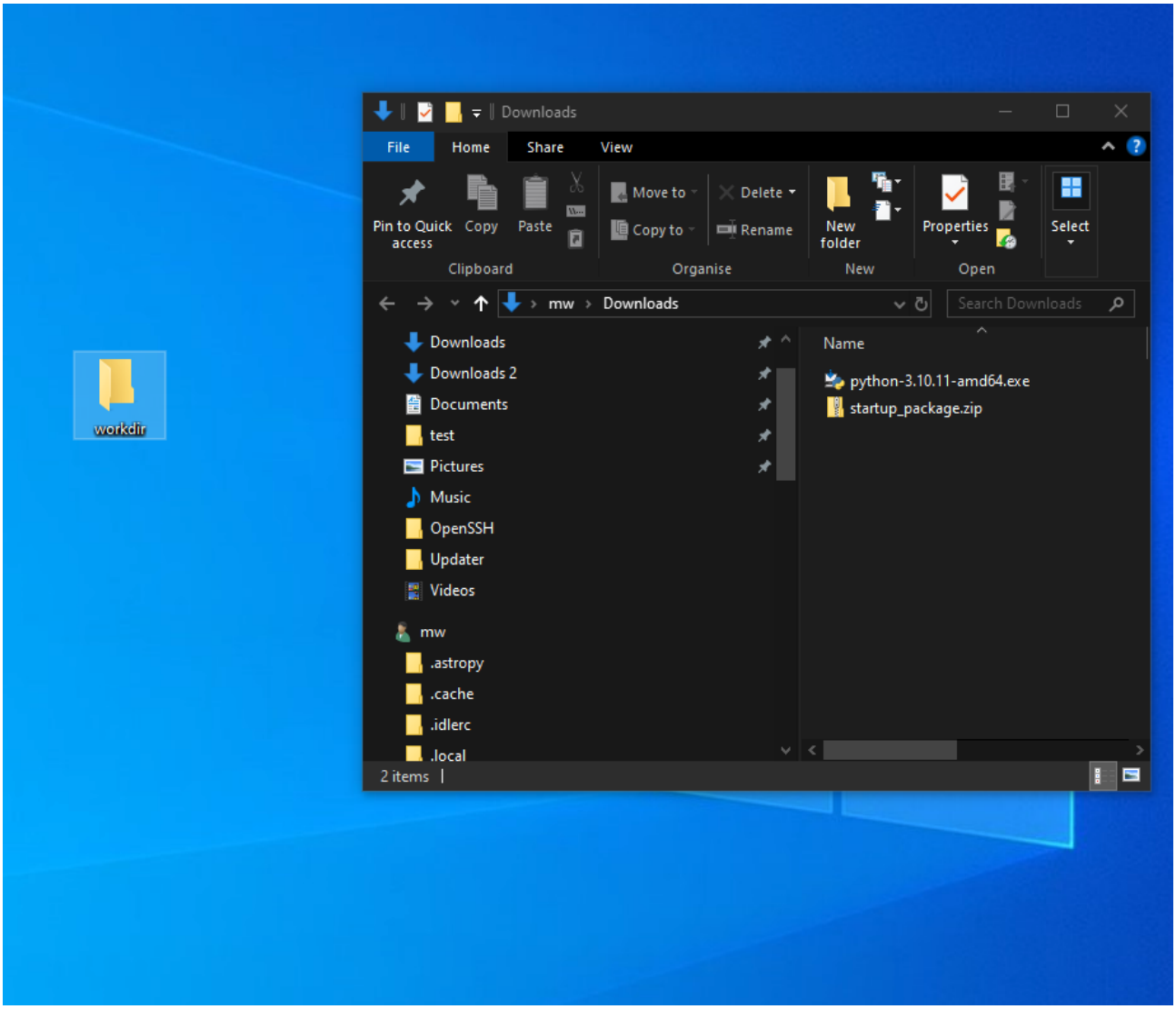


and there select the latest release:

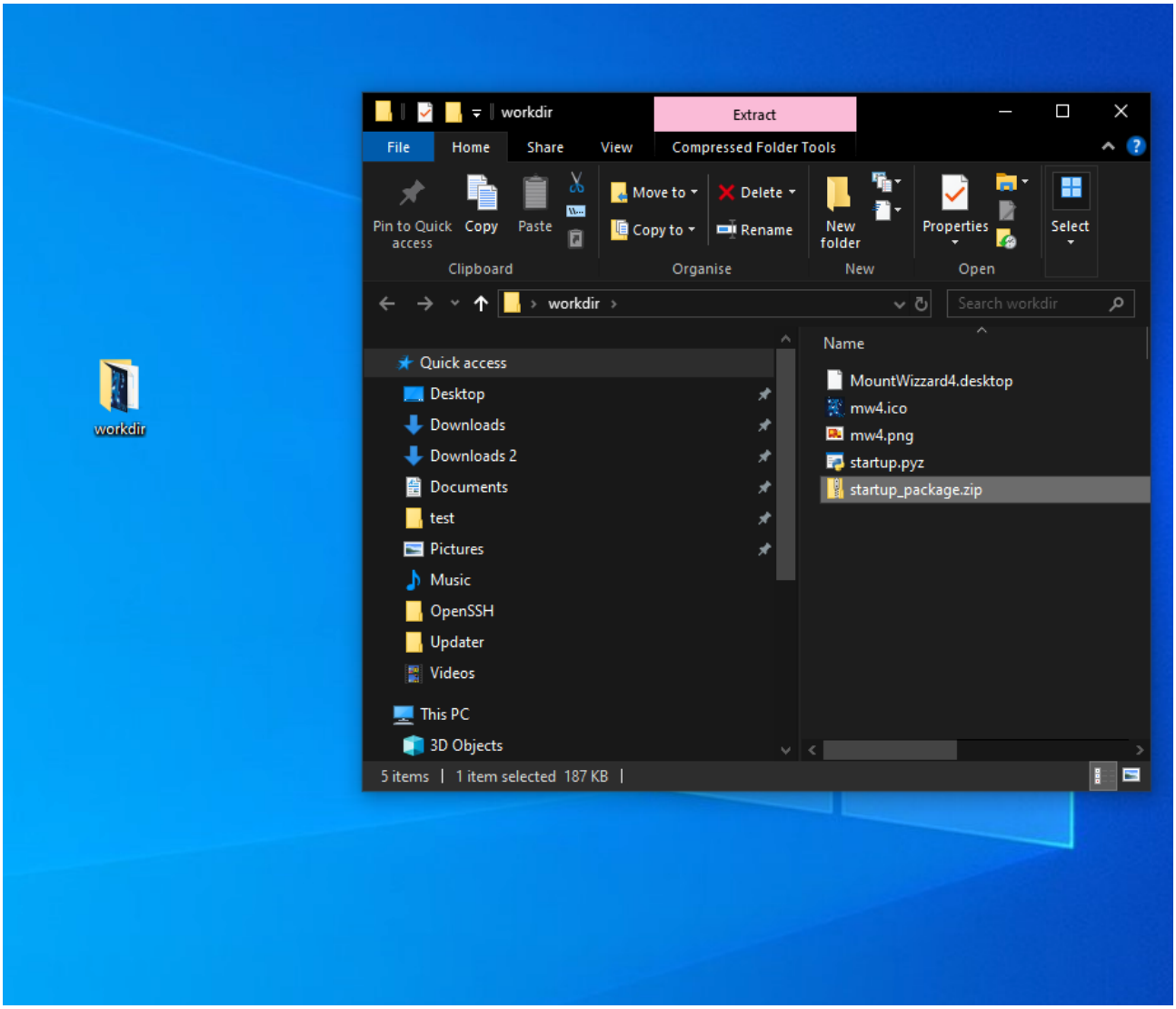


Here is the direct link to the latest release: https://github.com/mworion/InstallerMW4/releases

Please move the downloaded file to your work dir

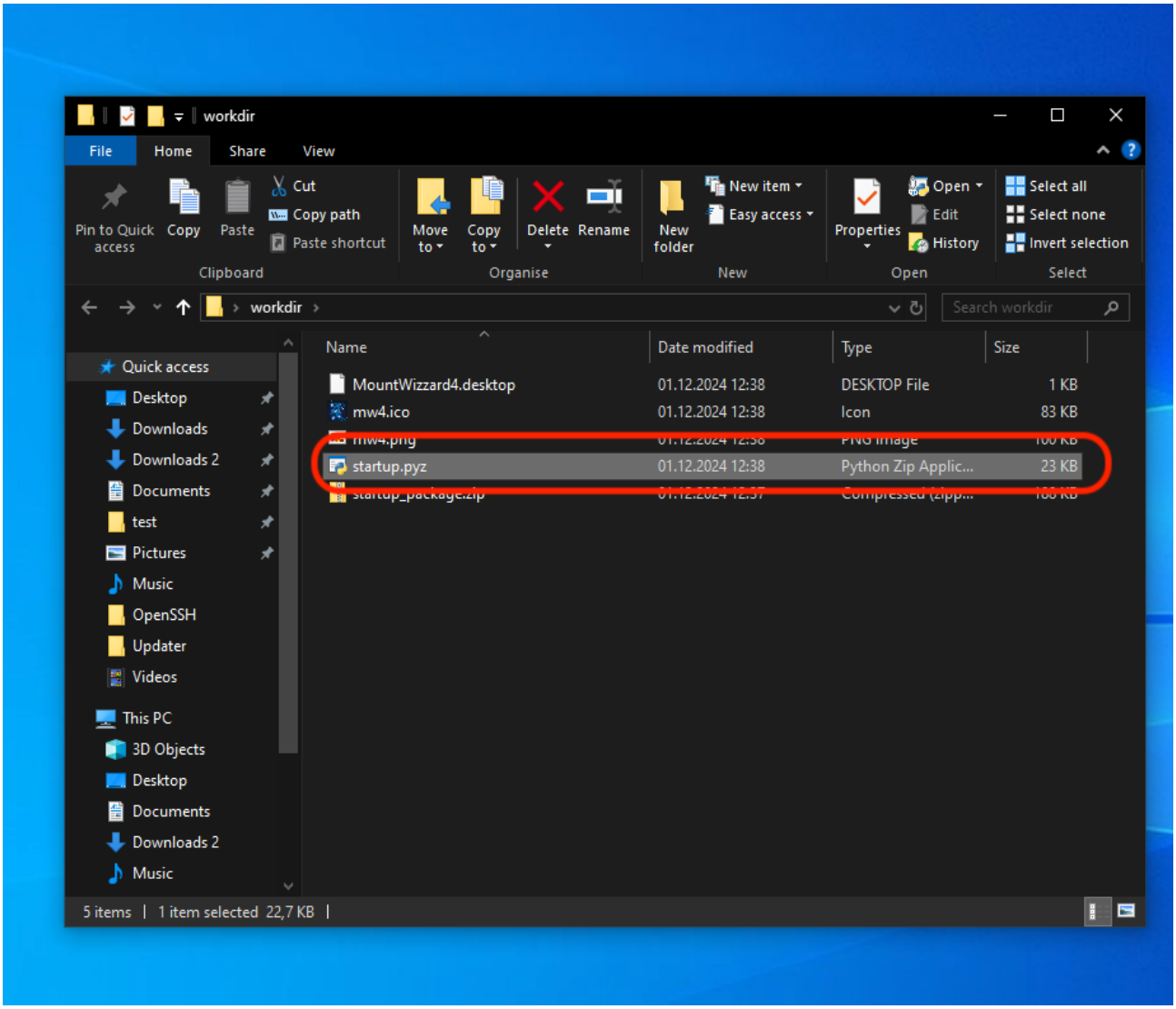


and unzip it there



Step 3

Run the installer script, on windows you should be able to start the script just be double click on it (you see the filetype is already linked to a python interpreter):



in all other platforms you start it with:

```
python startup.pyz  # Windows
python3 startup.pyz # Ubuntu, ARM64 and OSx
```

This script will prepare an virtual environment, install all necessary libraries - if applicable for arm64 platforms (RPi's) also precompiled wheels and MountWizzard4 itself. After a successful installation, the script will start MountWizzard4 the first time. During this first run MountWizzard4 will create some subdirectories in your working folder. When starting, a splash screen shows the progress of it's initialization. After first start the directory should look like (example Windows)

For newer MountWizzard4 versions, there is no need for precompiled wheels anymore as they are provided directly from PyPi.

Remarks for installation

MountWizzard4 is installed inside the virtual environment venv in your work dir. Once installed, the startup.pyz script is also used for starting MountWizzard4 at any time.

Hint

Please check if an online connection is available on your computer during installation as the libraries and MountWizzard4 is installed from online sources.

Over time, there might be some improvements also made for these scripts. So if you had installed MountWizzard4 some time ago and will install new setups, it might be helpful to check if some new scripts are available for better handling. When running the script, it will check for updates and gives you some hints.

The new script 4.x supports multiple platforms (Windows, MacOSx, ARM64 and x86 Linux distributions! Still for some you need to do some preparations.

Short videos for installation

For a better impression of how MountWizzard4 could be installed, there are some special videos showing a installation on different platforms.

Windows10: https://youtu.be/Tzob8ZSnMH0

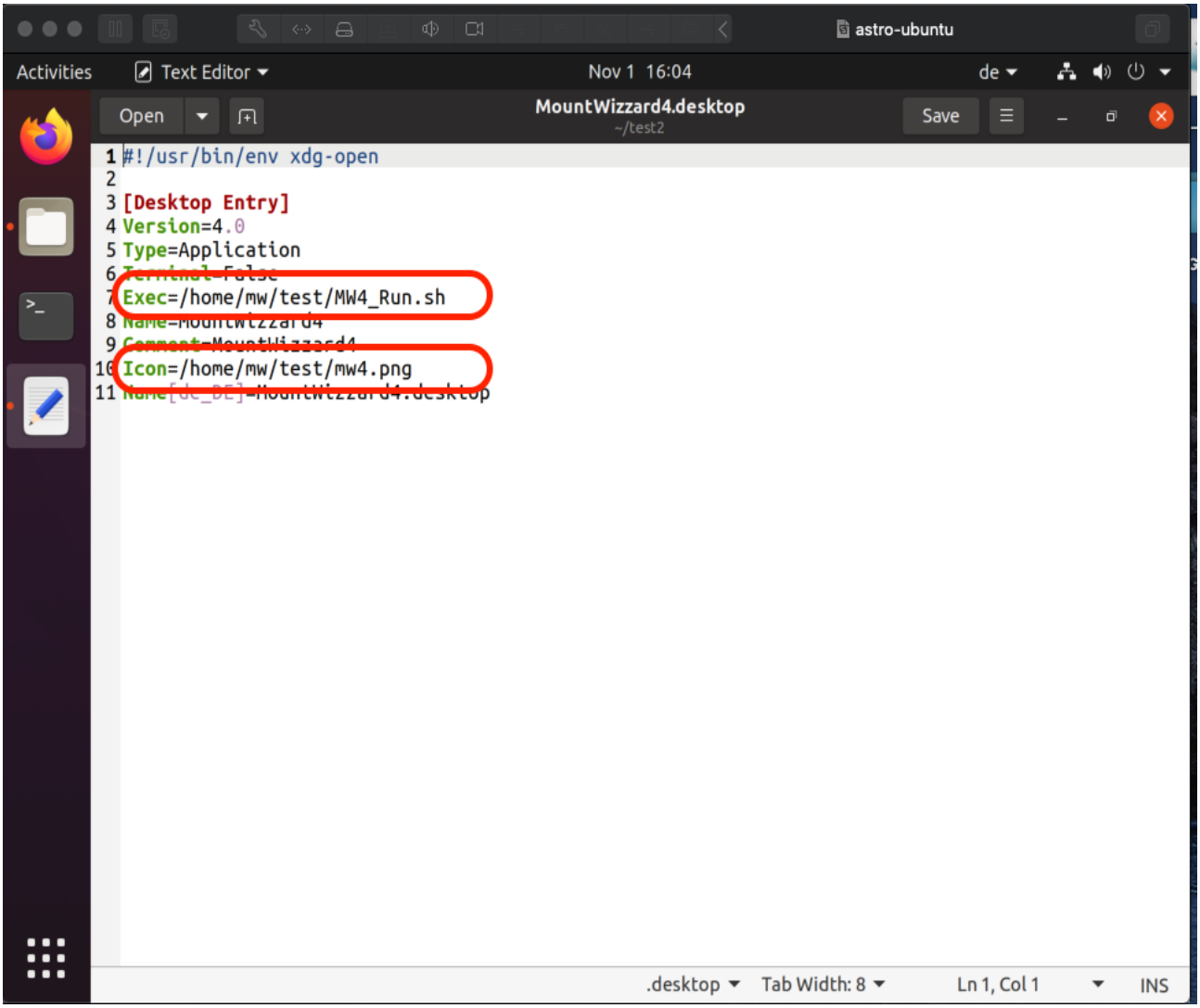
Mac OS Catalina: https://youtu.be/bbZ9_yLm1TU

Ubuntu 18.04: https://youtu.be/kNfLrtJtkq8

If you see the upper window, you succeed and from now on you are able to customize your setup of MountWizzard4 and it's features. Please refer to the MountWizzard4 documentation for further information.

Setting up Ubuntu

For Ubuntu the scripts also include an icon file (mw4.png) as well as a desktop description file (MountWizzard4.desktop). In order to use this add-on, please adjust the directories used in this file:



Unfortunately this is broken un Ubuntu 20.04LTS, see (including the workaround):

https://askubuntu.com/questions/1231413/basic-desktop-actions-are-not-available-on-ubuntu-20-04

If you install nemo (hint as workaround) as file manager, the desktop icons will work.

Install Plate Solvers

Supported platesolvers are astrometry.net, ASTAP and Watney. All solvers should be installed locally. There is no support for astrometry.net online. If you install a plate solver, please be reminded that you have to install their index files as well. Unfortunately all are using different index files and methods, also depending on your optical setup. MW4 helps you in finding the necessary index files for your setup:

install/image/information.png

System Message: WARNING/2 (/home/runner/work/InstallerMW4/InstallerMW4/doc/install/platesolvers.rst , line 10)

Cannot scale image! Could not get size from "install/image/information.png": [Errno 2] No such file or directory: 'install/image/information.png'

Based on optical and Sensor data. MW4 will make a prognosis, which index selection might be good. You also have direct internet links to the sources.

Astrometry.net

Astrometry.net is useful on Linux and Mac installations. On Windows there is no good setup possible. There are many solutions available (e.g. ANSRV), but these could not be used through MW4. You will find astrometry.net here:

http://astrometry.net

ASTAP

ASTAP is an application available for all platforms from Han. Great software! It is available as application with GUI and as pure command line interface solution (CLI). If you don't use ASTAP for other things as well, I would recommend installing the CLI version locally in a folder of your choice. You find all information on:

https://www.hnsky.org/astap.htm

Watney

Watney is a new solver from Jusas, based in the algorithm Han published for ASTAP. It is available as API and command line interface (CLI) version for all platforms. For the use with MW4, please install the CLI version in a folder of your choice. You will find all information on:

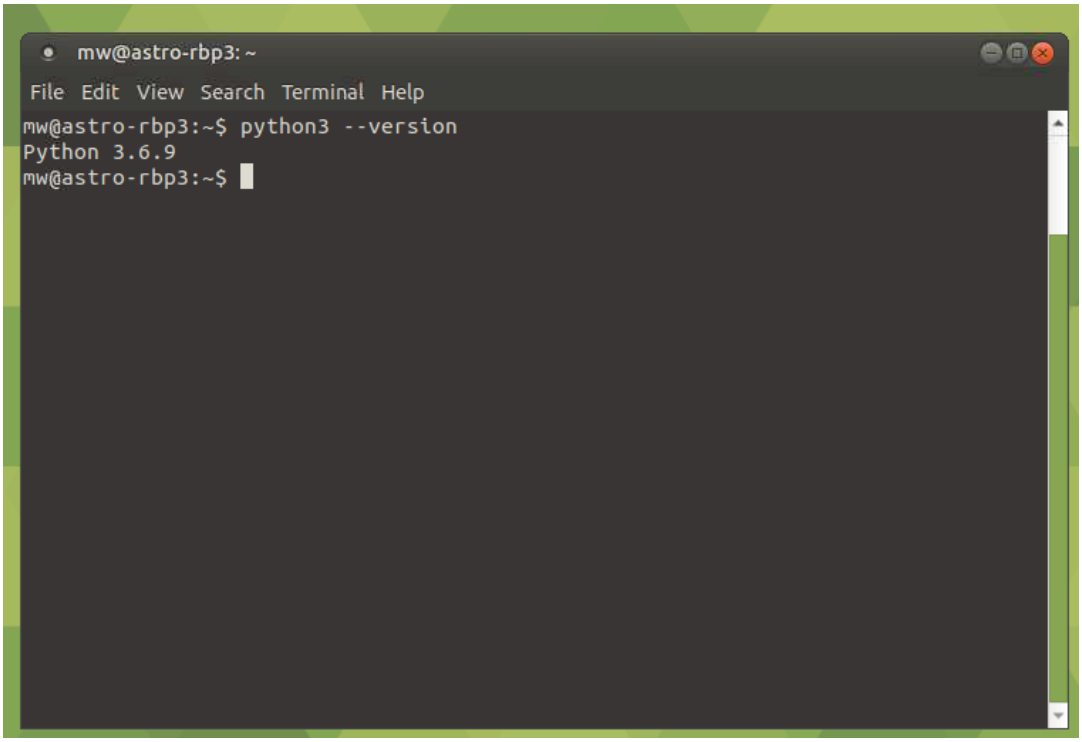
https://github.com/Jusas/WatneyAstrometry

and

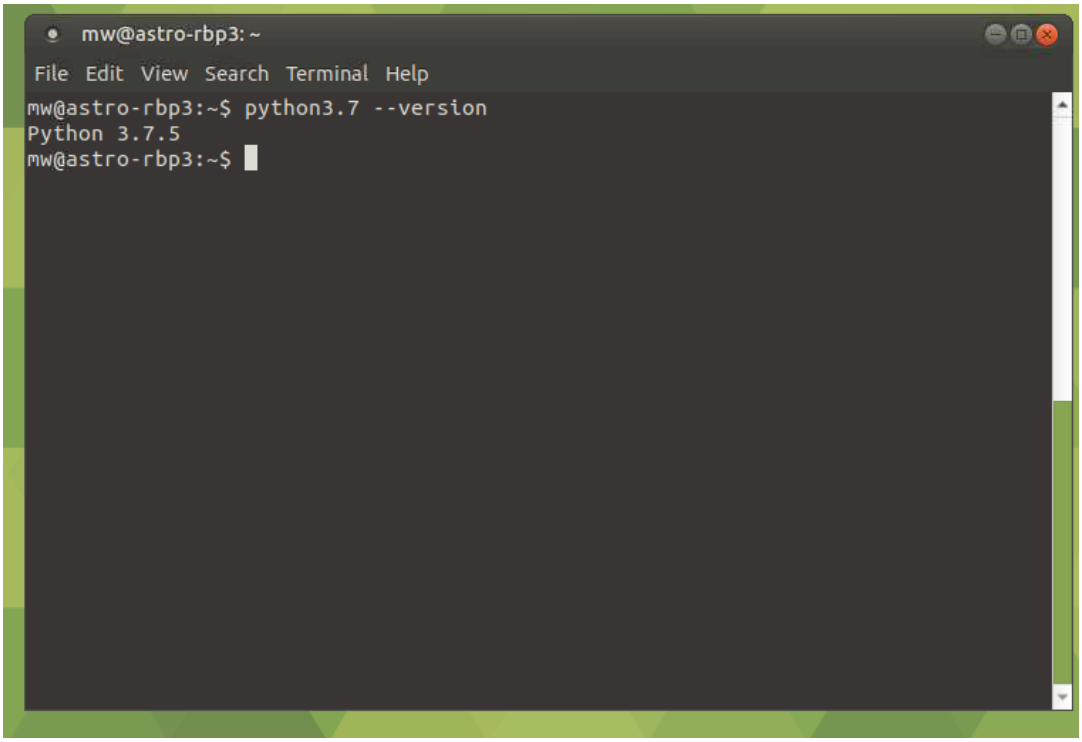
https://watney-astrometry.net

Install on RaspberryPi 3

To get MW4 installed on RPi3 you will follow the instructions of Robert Lancaste (many thanks to him fore this work!) on https://github.com/rlancaste/AstroPi3 with installing AstroPi3 scripts. The installation procedure I describe is based on Raspbian Buster with desktop. should give you the following result:



In addition you have to take care, that python3.8 is installed. The actual Ubuntu mate 18.04.2 distribution comes with python 3.6, so we need to update this. Please follow the description: Ubuntu . After that you should get an python3.8 or newer available on your system:



If everything went fine, we can proceed to the next step.

Installing PyQt5 on RPi3

As on arm the installation of PyQt5 could not be done through pip, the actual tested path is to install Qt directly via apt-get on your RBP3. As result, you cannot install MW4 easily in a virtual environment as apt-get will install all libraries in a system path.

As there were no compiled binaries for actual Qt version available, you have to compile it yourself.

```
sudo apt-get update
sudo apt-get install python3.8-dev
sudo apt-get install qt5-default
sudo apt-get install sip-dev

cd /usr/src
sudo wget https://www.riverbankcomputing.com/static/Downloads/sip/
4.19.19/sip-4.19.19.tar.gz
sudo wget https://www.riverbankcomputing.com/static/Downloads/PyQt5/
5.13.2/PyQt5-5.13.2.tar.gz

sudo tar xzf sip-4.19.19.tar.gz
sudo tar xzf PyQt5-5.13.2.tar.gz

cd sip-4.19.19
sudo python3.7 configure.py --sip-module PyQt5.sip
sudo make -j4
sudo make install
```

```
cd PyQt5_gpl-5.13.2
sudo python3.7 configure.py
sudo make -j4
sudo make install
```

There are in different packages to be downloaded and installed. They build on each other, so keep the order of compiling and install. This procedure take about 2 hours or more, depending on the system.

⚠ Warning

So far PyQtWebEngine does not build on RPi3! So I removed for the build from 0 .138 on the capabilities, who need the PyQtWebEngine package. This is basically the Keypad. So you will have limited features!

So before you could actually run MW4 you need to install some mor libraries:

```
sudo apt-get install libgfortran5
sudo apt-get install libjpeg-dev zlib1g-dev
python3.7 -m pip install -U Pillow
```

Once you are set, make a work directory, cd to this directory and install MW4 by

```
python3.8 -m pip install mountwizzard4
```

and run MW4 with the command

```
python3.8 ~/.local/lib/python3.8/site-packages/mw4/loader.py
```

If everything went fine, you should see MW4 on RPi3:

Install on RaspberryPi 4/5

We are installing MW4 on an ubuntu 20.04.1LTS 64Bit system. In relation to the RPi3 it seems to be much simpler to do. Nevertheless some of the big packages will be compiled on your system during installation, which means this will take some time (hours). There is the opportunity to use precompiled packages out of the install scripts provided.

Another big step forward is that you could use now a virtual environment for installing MW4.

To get MW4 installed on RPi4 you will follow the instructions of Dustin Casto:

https://homenetworkguy.com/how-to/install-ubuntu-mate-20-04-lts-on-raspberry-pi-4/

to get Ubuntu Mate 20.04.1 LTS on your RPi4.

Hint

Some users experience problems with KStars/EKOS on original ubuntu-mate desktop. So the recommendation is to use a KDE bases desktop like kubuntu. The easiest way to install a desktop on top of the server installation is using: https://github.com/wimpysworld/desktopify

After you have finished the setup and got the desktop up and running, the command

```
python3 --version
```

should give you the following result 3.8.5: Please take care, that a python version 3.8.5 or later is installed.

The actual Ubuntu mate 20.04.1LTS distribution comes with python 3.8.5, so everything should be OK. Next we have to do is to install a virtual environment capability, the packet manager pip and the development headers for python to be able to compile necessary packages:

```
sudo apt-get install python3.8-venv
sudo apt-get install python3-pip
sudo apt-get install qt5-default
```

Note

You need to have both packages installed as otherwise the install script or later does MW4 not run.

After that, please proceed with the installer script

Display configuration

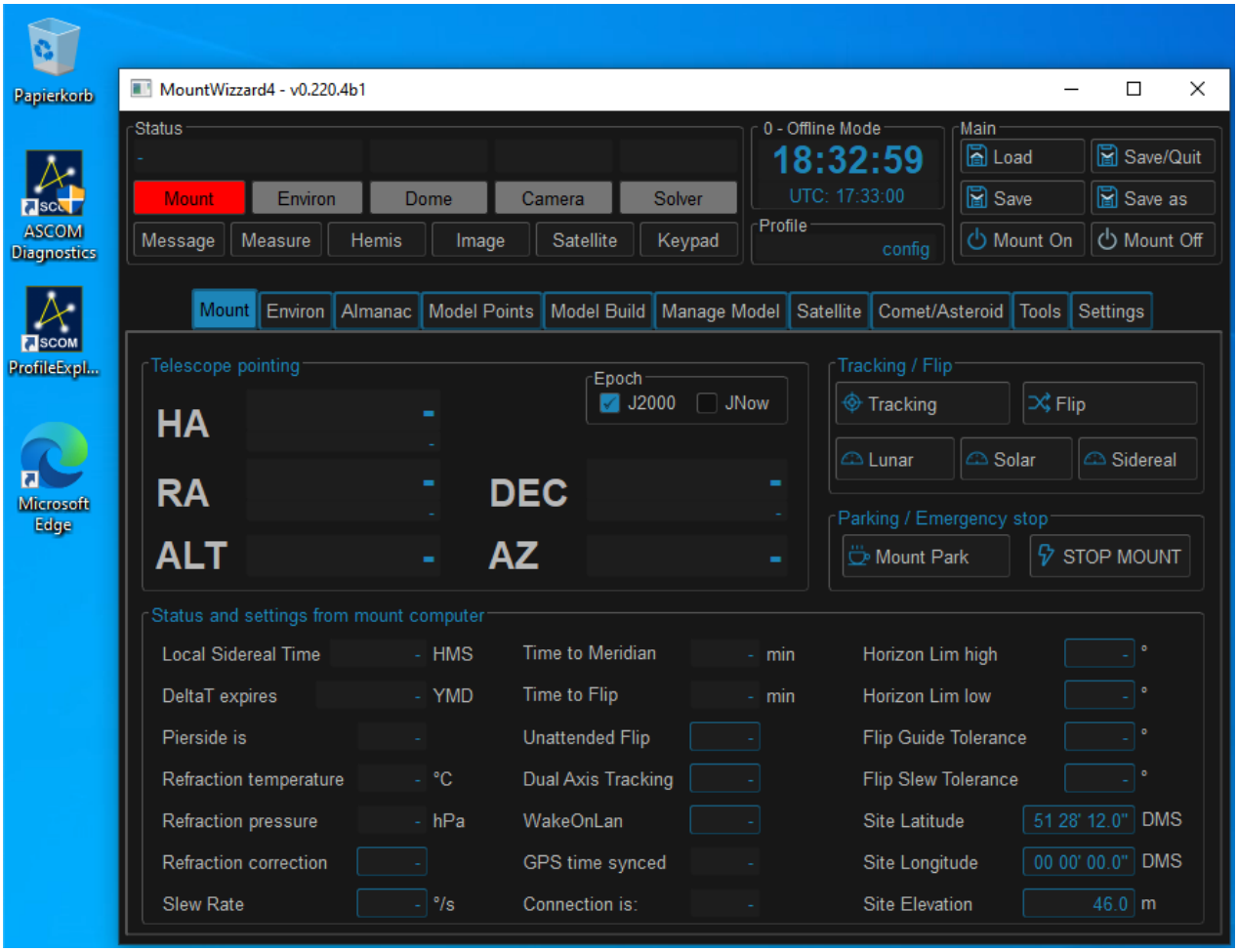
DPI scaling on Windows

If you are running a windows machine with setting the zoom factor for you display settings different to 100%, you might notice inadequate font sizes etc. Unfortunately this could not be

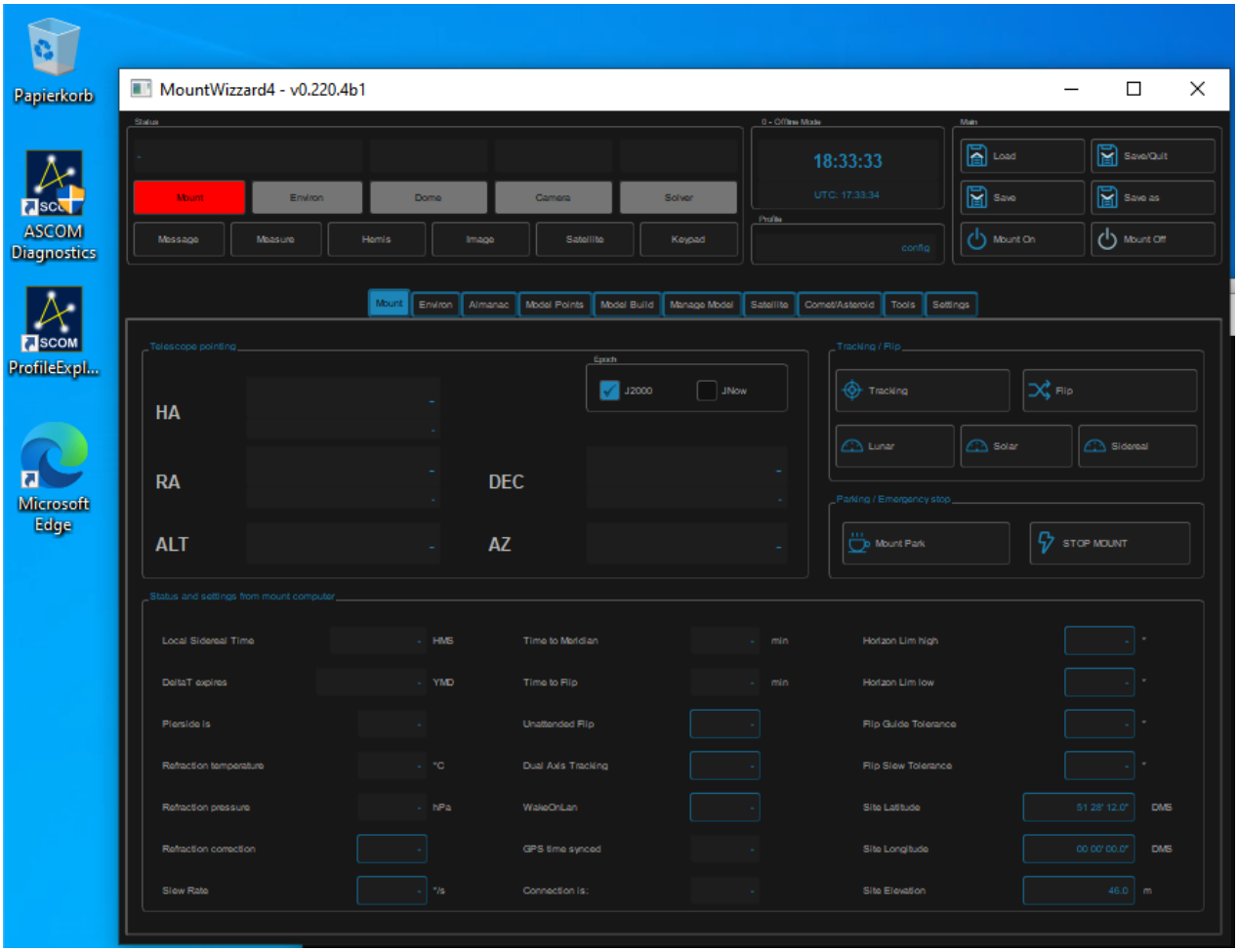
worked around within MountWizzard4 itself, but ou could change some environment variables to omit this problem. The actual script already contain some setting to keep the resolution to 100% even if you choose to increase this value for other applications. You want to play with these settings to make the appearance correct:

```
SET QT_SCALE_FACTOR=1
SET QT_FONT_DPI=96
```

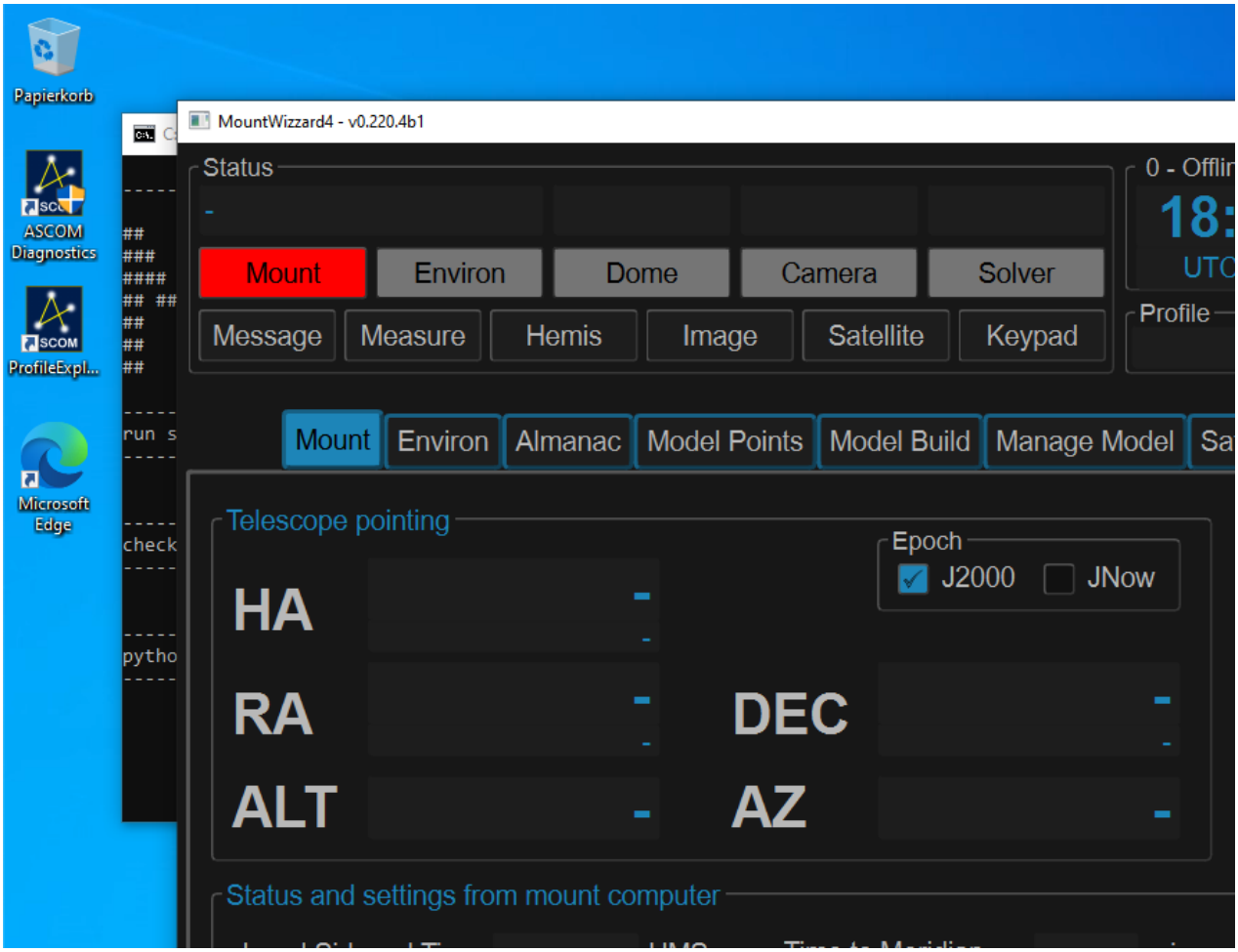
Here some examples of the settings: Normal scaling (scale = 1, dpi = 96)



Small fonts (scale = 1, dpi = 48)



Bigger scale (scale = 1.5, dpi = 96)



If you would like to have MountWizzard4 displayed bigger than 100%, please increase the QT_SCALE_FACTOR to the value desired. A value of 1 means 100%, so 2 means 200%. You will experience to set the font adequately.

DPI scaling on Ubuntu

This is quite similar to windows. You have to set the environment variables QT_SCALE_FACTOR and QT_FONT_DPI accordingly. They are already part of the MW4_Run.sh scripts.

Command line options

InstallerMW4 supports a number of command line options to allow for automated installation and updating of MountWizzard4. You could check them by running

```
python startup.pyz --help   # Windows
python3 startup.pyz --help  # Linux, MacOS
```

The following command line options are available:

‘-c’, ‘–clean’

Cleaning system packages from faulty installs

‘-d’, ‘–dpi’

Setting QT font DPI (+dpi = -fontsize, default=96)

‘-n’, ‘–no-start’

Running script without starting MountWizzard4

‘-s’, ‘–scale’

Setting Qt DPI scale factor (+scale = +size, default=1)

‘-u’, ‘–update’

Update MountWizzard4 to the actual release version

–update-beta’

Update MountWizzard4 to the actual beta version

–update-venv

Update the virtual environment directory to use this version of Python, assuming Python has been upgraded in-place.

‘-v’, ‘–version’

Update MountWizzard4 to the named version

Changelogs

The changelogs contains the user related function or environment updates. For a detailed changes list, please refer to the commit list on GitHub.

Unreleased versions of scripts

4.0.6

- MW4 v4 version is now python 3.10 - 3.13

4.0.5

- test version

4.0.4

- test version

4.0.3

- removing –break-system-packages

4.0.2

- adding support for python 3.12

4.0.1

- adding verbose option to get outputs during install
- refactor code to simplify

4.0

- rewrite an separation to app itself.
- adding version management
- better feedback during install

Released versions of scripts

3.3

- adding verbose option to get outputs during install

3.2

- remove support for win32

3.1

- enable back aarch64 support for arm

3.0

- enable all feature of scripts to one single startup.pyz file with parameters
- new script is limited to windows, mac and linux only. No support of Raspi or aarch64

2.4

- windows only: improved robustness against local installs
- adding a clean system utility

2.3

- windows only: improved robustness against local installs

2.2

- adding support for astroberry and stellar mate (based on astroberry?)
- adding workaround on windows for removed python2 support in setuptools

2.1

- fixes

2.0

- update to requirements v2.0beta release of MW4

1.1

- update to requirements v1.1 release of MW4

0.4

- support for python 3.7 - 3.9, removing support 3.6
- added prebuild wheels for ubuntu mate 18.04 aarch64 for python 3.7 - 3.9
- added prebuild wheels for ubuntu mate 20.04 aarch64 for python 3.7 - 3.9

0.3

- adding more error log support
- improved ui

0.2

- added support functions for better error handling while using

0.1

- adding support for raspberryPi3 with ubuntu mate
- adding support for raspberryPi4 with ubuntu mate server / kubuntu

0.0

- adding support for automatic installation on windows
- adding support for automatic installation on ubuntu
- adding support for automatic installation on osx