

Scripting utility for 10Micron mounts

The "mountsend" utility is a small command line application developed in order to make certain mount-related tasks available from various scripting engines. It can be used from Windows BAT-files, CMD-files, VBScript and any other scripting engine capable of issuing command lines.

In addition to being an essential tool in your scripting environment, it also serves as a really quick way of controlling your mount, especially if you compare to using the handset.

Special support has been added to automatically update refraction parameters (pressure and temperature) from a one-line file produced by standard weather stations. See the refraction command for details.

Mountsend requires .NET Framework 4.

Features/commands

A number of commands have been implemented and more are continuously added as the need arises. Some commands take arguments, some do not.

The general command format is:

```
Mountsend <ip-address> <command> [<option>] [<option>]...
```

The IP-address can be omitted if it is first saved by using the save command and the option "/a" is added to subsequent commands. The place of storage is in the logged in user's environment variable store, and the data is available from first save until it is manually removed from the user's space by means of the system properties dialog. It survives restarts as well.

All examples below use the IP-address 192.168.250.1. Don't forget to substitute your actual IP-address!

Running the application without any arguments at all will present a help text describing all commands in fair detail.

Save command

Saves the IP-address in the user's environment store under the name "MOUNT".

```
Mountsend 192.168.250.1 save
```

Park command

The park command will issue a park command to the currently selected park position and return immediately without waiting. There is no reply. If you are truly relaxed you may abbreviate the word "park" to "p".

```
mountsend 192.168.250.1 park
mountsend park /a
mountsend p /a
mountsend park /a
```

parkw command

This command has the same behavior as "park" but will wait until the park is complete before returning to the caller. This can be a good thing if your next script command needs the mount actually parked.

```
Mountsend parkw /a
mountsend 192.168.250.1 parkw
```

Unpark command

This is the exact opposite of park. The mount will enter the unparked state but will not start tracking. The command can be abbreviated to "up".

```
Mountsend 192.168.250.1 unpark
mountsend 192.168.250.1 up
mountsend unpark /a
mountsend up /a
```

Command command

The command command is used to send arbitrary information to the mount. The different commands that you can issue are described in the mount firmware manual. This command is rarely used and mainly serves as a test bed for upcoming enhancements to mountsend. It is unlikely that you will use it. Basically, the argument after the word "command" is sent to the mount as is and it is your responsibility to create something sensible that the mount can understand.

Stop command

The stop command will halt all movement of the mount immediately, including tracking.

```
Mountsend 192.168.250.1 stop
mountsend stop /a
```

Start command

Start initiates tracking at the currently selected rate.

```
Mountsend 192.168.250.1 start
mountsend start /a
```

Waitstationary command

This command will wait until all movements have ceased in the mount. It can be used to ensure that the mount really is not moving, but it can also bounce back and bite you if you, for instance have tracking on! Use with care!

Stat command

The stat command will return the current state of the mount as a text string. The defined states are:

- tracking
- stoppedOrHomed
- parking
- unparking
- slewinghome
- parked

- slewing
- stationary
- outsideTrackLimits
- needsOK
- mountError

You can also get the "no reply" answer if the mount doesn't respond within one second. Further information as to the meaning of the different states can be found in your mount firmware manual (:Gstat# command).

Maxslew command

The maxslew command will set the maximum slew rate of your mount in degrees per second. Valid range is 1-20°/s.

```
Mountsend 192.168.250.1 maxslew 9
mountsend maxslew 7 /a
```

Move command

The move command initiates telescope slew operations and immediately returns. There is no reply from the application. Coordinates are Azimuth and Altitude, both expressed in whole numbers. This command is not meant for precision slewing, rather for ensuring that the mount is pointing approximately where you want it before you do something else, like closing a roof.

```
Mountsend 192.168.250.1 move 180 10
mountsend 180 10 /a
```

The example will point the mount straight South at an elevation of 10°.

Movew command

This command does exactly the same as the move command but waits until the slew is complete. This is useful if you need the slew to be complete before proceeding to the next command in your script.

Gpsupdate command

This command will initiate a GPS update in the mount. Your site position and elevation as well as time and date will be correctly set. The mount spends quite an amount of time on this command, most likely because it wants to get the time really accurately, and is completely unresponsive during the update. This time delay may cause software such as FocusMax to report mount timeouts so you should make sure you know what you are doing. Needless to say, you need to have a GPS unit connected to the serial port of your mount and the mount needs to be set to use the port as a GPS port.

```
Mountsend 192.168.250.1 gpsupdate
mountsend gpsupdate /a
```

Refraction command

The refraction command will send pressure and temperature to the mount in order to update the refraction model parameters. Pressure is in hPa and temperature in °C. The command can be abbreviated to "refr" or just "r". The update is immediate so this may be a command that should not

be executed while imaging as that may cause the mount to slightly adjust its pointing and tracking models.

```
Mountsend 192.168.250.1 refraction 1013.3 -5.5
mountsend refr 1013.3 -5.5 /a
```

The decimal point character needs to be according to your local settings, i.e. a dot for the U.K. and a comma for countries that adhere to that standard (Sweden, Germany and many other European countries).

Autorefr command

The autorefr command needs a special file in order to function. The file in question is usually called realtime.txt and contains a single line with all relevant weather parameters. Software such as Cumulus (used by many weather stations) can generate this file periodically for you. The important parameters in the file is the current air pressure and the current temperature. These should be available in columns 11 and 3 respectively. A typical realtime.txt looks like this:

```
13-02-13 09:32:53 -1.9 99 -2.0 0.0 0.0 0 0.0 0.0 1026.3 --- 0 m/s C hPa
mm 0.2 +0.8 3.3 13.8 0.0 11.7 25 -1.9 +0.5 -1.3 03:14 -3.6 07:35 0.7
03:10 2.0 08:45 1026.3 09:32 1019.9 00:00 1.9.2 1027 0.0 -1.9 -1.9 0.0
0.0 0 0 0.0 1 1 0 --- 52 ft -4.2 0.0 93 0
```

(no line breaks exist in the file, it has been wrapped in order to fit this document)

As can be seen in this example, the temperature is -1.9°C and the pressure is 1026.3 hPa.

```
Mountsend 192.168.250.1 autorefr c:\weather\realtime.txt
mountsend autorefr c:\weather\realtime.txt /a
```

Time command

The time command will update the date and the time in the mount to match your PC clock at the time of command issue. As with the refraction update commands, this can cause the pointing and tracking models to immediately shift the mount position to reflect the change, so use it in between images, not during exposure.

Setting the mount time accurately is tricky business, but mountsend does a pretty good job of getting reasonably correct. Basically, the time involved in transmitting the command to the mount will impact the results, as will the mount internal processing of the command. Therefore, mountsend will report the expected accuracy as well as it can.

Prior to issuing the time command, mountsend will wait until a "fresh new" second has occurred. It does this by sticking to a loop until the millisecond part of the time is less than 2. If your computer is slow, this may take a number of seconds to accomplish. In contrast, if you have a fast system, the time spent waiting for the right conditions will be less than a second.

The 10Micron mount firmware only takes whole seconds as argument to the time setting, so getting as close to a whole second as possible is important.

Once the conditions have been reached, mountsend will report what time it will use. Next, it will report the exact time at the point where all the bytes of the time command has been sent to the mount. Lastly, it will report the exact time of the reply from the mount.

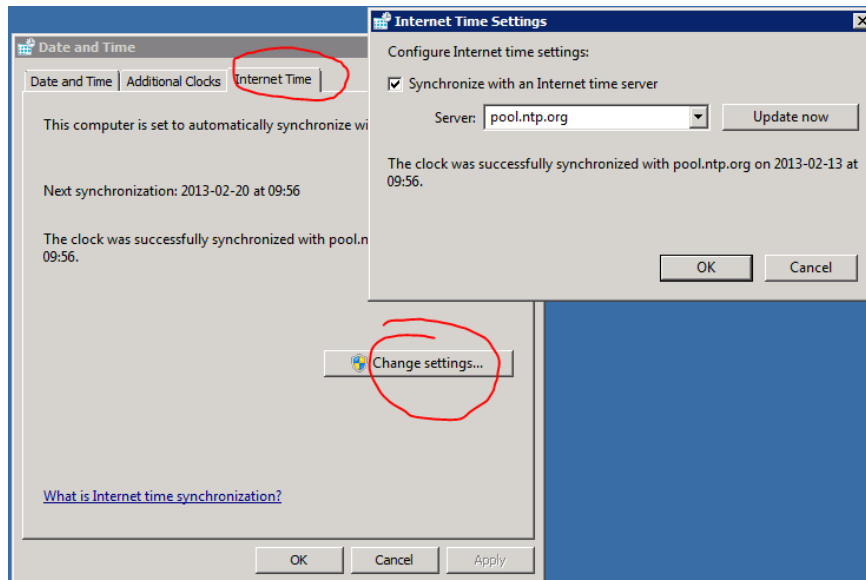
Assuming that the mount is fairly quick in setting the time and somewhat less quick at sending the reply, the time error in the mount should approximately match the middle value reported. It can, however, be as bad as the last time reported. On my rather slow observatory computer, the results of the command is the following:

```
mountsend time /a
Will use 09:46:05
Sent at 09:46:05.0176
Confirmed 09:46:05.1580
Updated
```

As can be seen from this, the accuracy on my system should be somewhere between 17.6 and 158 milliseconds, most likely closer to the 17.6 figure.

Remember, one second off in mount time means target missed by 15 arcseconds!

Obviously, having accurate computer time is essential. I suggest you use the "internet time" feature of Windows and modify the behavior of the time service. First, set up your computer like this:



My observatory computer is set to use pool.ntp.org, but this may not be the absolutely best one to use. Look for startum 1 time servers on the internet and choose one to test.

Windows will, by default, synchronize three times a day once accurate syncing has been confirmed. This should be adequate for most applications. Should you want to sync more frequently, you add a DWORD parameter with the name "Period" and a value corresponding to the number of times per day you want it synced to:

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\W32Time\Parameters

Using mountsend in your scripts

Mountsend is just as good a performer directly from a command line prompt as it is in various scripting engines. Below are some considerations that may apply to your environment.

Straight BAT and CMD files

Working with this kind of scripts, all you have to do is make sure that mountsend is in your path, or specifically point it out with a full path. The rest is very straight forward; just add the commands to the file, like this (content of test.bat):

```
Echo off
echo Starting slew
c:\mountsend\mountsend.exe movew 90 45
echo Slew complete, stopping mount
c:\mountsend\mountsend.exe stop
```

VBS Scripts

Windows scripting host supports an object model, so the first thing you have to do is create an object that will be used to talk to the system. After that, you just send the commands as usual. Example (content of test.vbs):

```
Set Shell = CreateObject("Wscript.Shell")
Shell.run "c:\mountsend\mountsend.exe park /a", 0, true
```