# GS Server

a simpler way to clear skies

ASCOM Telescope simulator and SkyWatcher driver for EQ8 and HDX110 mounts.

Rob Morgan

*Contents*

## Overview

GS Server (GSS) is an ASCOM compliant middleware know as a local server or hub.  It's designed to allow other applications to access your telescope's mount in a practical and systematic way.

GSS can access the Sky Watcher EQ8 and Orion's HDX110 mounts.  It's also comes with a built in simulator so you can practice sessions without being connected to the mount.

Your mount should already be connected to the computer using the preferred method which is the EQ Direct approach.  Specific cable required for this approach can be found on the internet.  One popular site is [http://www.store.shoestringastronomy.com](http://www.store.shoestringastronomy.com).

# 1. Quick Start

1. Download and Install GS Server.
2. Open your planetarium program and select an ASCOM connection.
3. In the ASCOM chooser select ASCOM.GS.Sky.Telescope
4. In the ASCOM chooser click Configure and adjust any needed settings.  Configure Com Port, Baud Rate, Mount, and Observatory Location.  If on a permanent mount check 'Home is a Park" checkbox.  Mobile setups leave this unchecked.  Click Save and Close
5. Click Connect from your planetarium program.
6. GS Server will start and connect to the mount.  If you do not connect check your settings and click the 'Connect' button.
7. Connect any other applications needed.

## 2. Requirements

- Computer with Windows 7 SP1 or newer
- Microsoft .NET Framework 4.6.1 installed
- ASCOM Platform 6.3 or later
- EQ8 or HDX110 telescope mount
- EQ Direct cable to attach mount to the computer

## 3. Installation

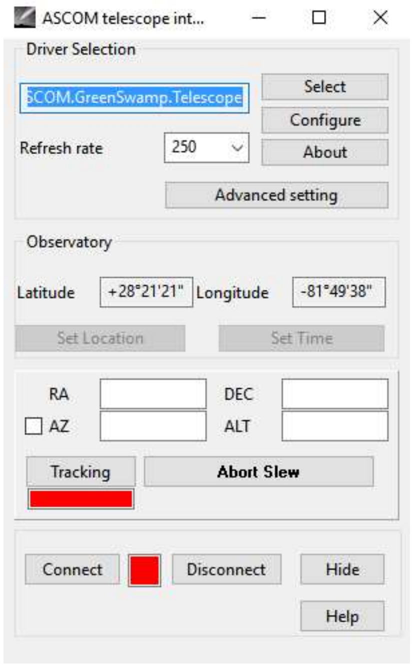
After downloading GSS run the .exe file and follow the prompts. The default location of files and settings can be found in the following locations on Windows systems.

- GSS Program Files- C:\Program Files (x86)\Common Files\ASCOM\GSServer
- Log files – My Documents/GSServer
- User Config – C:\Users\*User Name*\AppData\Local\GS

## 4. Running GSS

There are two ways to start GS Server; the preferred method is to allow other applications to start GSS as they connect using the ASCOM interface. Each application that supports ASCOM will have their way to configure and connect.

Here is the ASCOM interface from SkyChart (CDC)



ASCOM telescope int...

Driver Selection

SCOM.GreenSwamp.Telescope

Refresh rate 250

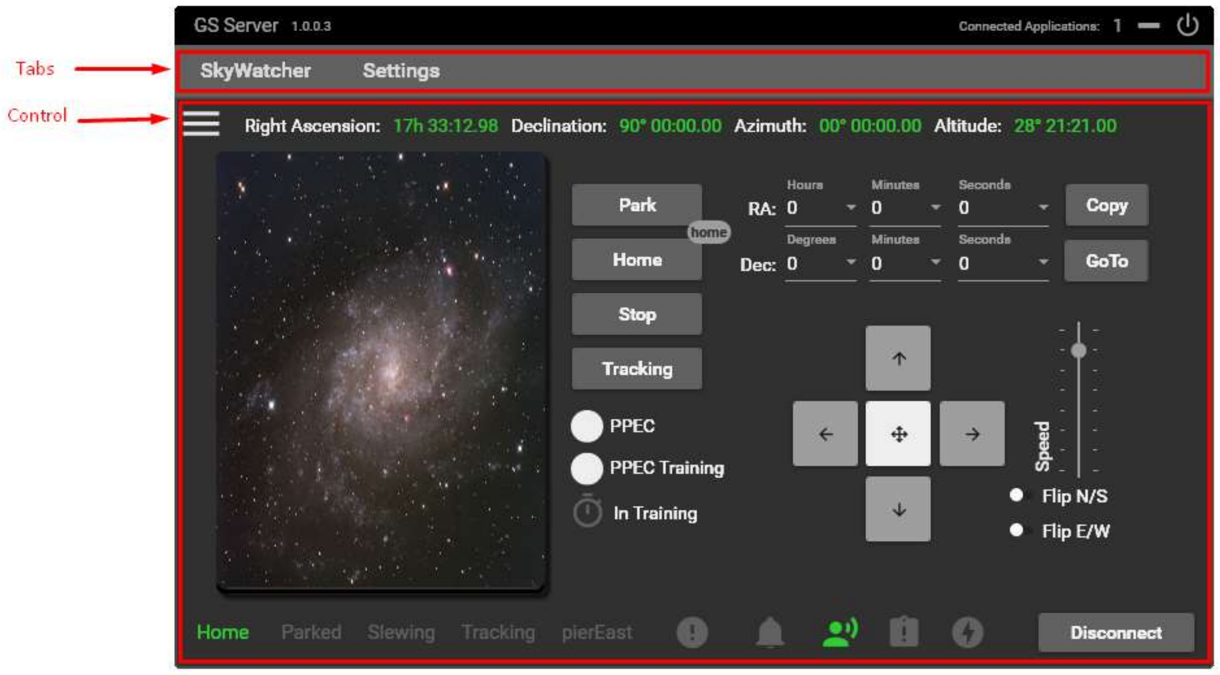
Observatory

Latitude +28°21'21" Longitude -81°49'38"

Use the Select button to select the ASCOM.GS.Sky.Telescope driver.
If this is the first time connecting to GS Server you will need to configure it before the applications will allow connections. See the configuration section. Once configured, click the connect button. If the configuration is correct GS Server will start within a few seconds. It's possible the connection may fail. If so GS Server will start but will not be connected to the mount. Reconfigure the driver and try connecting again.

The second method is to start GSS by running the .exe file located in the program directory. If starting for the first time configure the driver before hitting the connect button. Refer to the configuration section.

## 5. Main Window


Tabs
Control
GS Server 1.0.0.3
Connected Applications: 1
SkyWatcher
Settings
Right Ascension: 17h 33:12.98 Declination: 90° 00:00.00 Azimuth: 00° 00:00.00 Altitude: 28° 21:21.00
Park
home
Home
Stop
Tracking
PPEC
PPEC Training
In Training
RA: Hours 0 Minutes 0 Seconds 0
Dec: Degrees 0 Minutes 0 Seconds 0
Copy
GoTo
Speed
Flip N/S
Flip E/W
Home Parked Slewing Tracking pierEast
Disconnect

The layout of the main window contains two primary areas; Tabs and Control. Each selected tab will show it's Controls.

The top of the window shows the version of GS Server, Connected Applications, Minimize, an Close buttons.

Connected Applications – shows how many application are connected To GS Server. If you try to close GS Server when there are existing connections it will prompt you to confirm before closing. GS Server will exit on its own if the last calling application closes its connection.

## 6. SkyWatcher Controls

List of items in the SkyWatcher Controls

RA (Right Ascension) – the distance of a point east of the First Point of Aries, measured along the celestial equator and expressed in hours, minutes, and seconds.

Dec (Declination) – the angular distance of a point north or south of the celestial equator.

Azimuth – the direction of a celestial object from the observer, expressed as the angular distance from the north or south point of the horizon to the point at which a vertical circle passing through the object intersects the horizon.

Altitude – the distance measurement, usually in the vertical or "up" direction, between a reference datum and a point or object.

PPEC (Permeant Periodical Error Correction) - Checkbox to turn on or off PPEC.  See the PPEC section of this document.  The grayed out timer will turn yellow when training is in progress.

PPEC Train – Checkbox to start the training process for PPEC

Park Button – move the mount to the defined park position defined within the setup settings.

Hand Controller Buttons – for mount movements based on the speed slider.

Speed – 8 settings to control the speed of the hand controller buttons.

Tracking – turn on or off the selected tracking rate to use.  Will be automatically applied after a slew or goto.

Home Button– moves the mount to the initial home position

## Bottom Status Bar

– shows when the mount in the Home, Park, and Slewing positions or state.

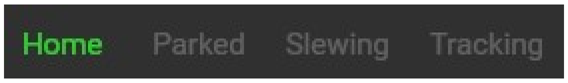
Connection – is lit green when the mount is connected with a serial connection.

Monitor – Status of the internal Monitor.   Will be list green when active.  See the Settings Control for more information about the Monitor.

Errors Alert – Will turn red if an error or warning is logged.   This can be reset by clicking on it.

Side of Pier - Normal (pierEast) where the mechanical Dec is in the range -90 deg to +90 deg.  Beyond the pole (pierWest) Where the mechanical Dec is in the range -180 deg to -90 deg or +90 deg to +180 deg.

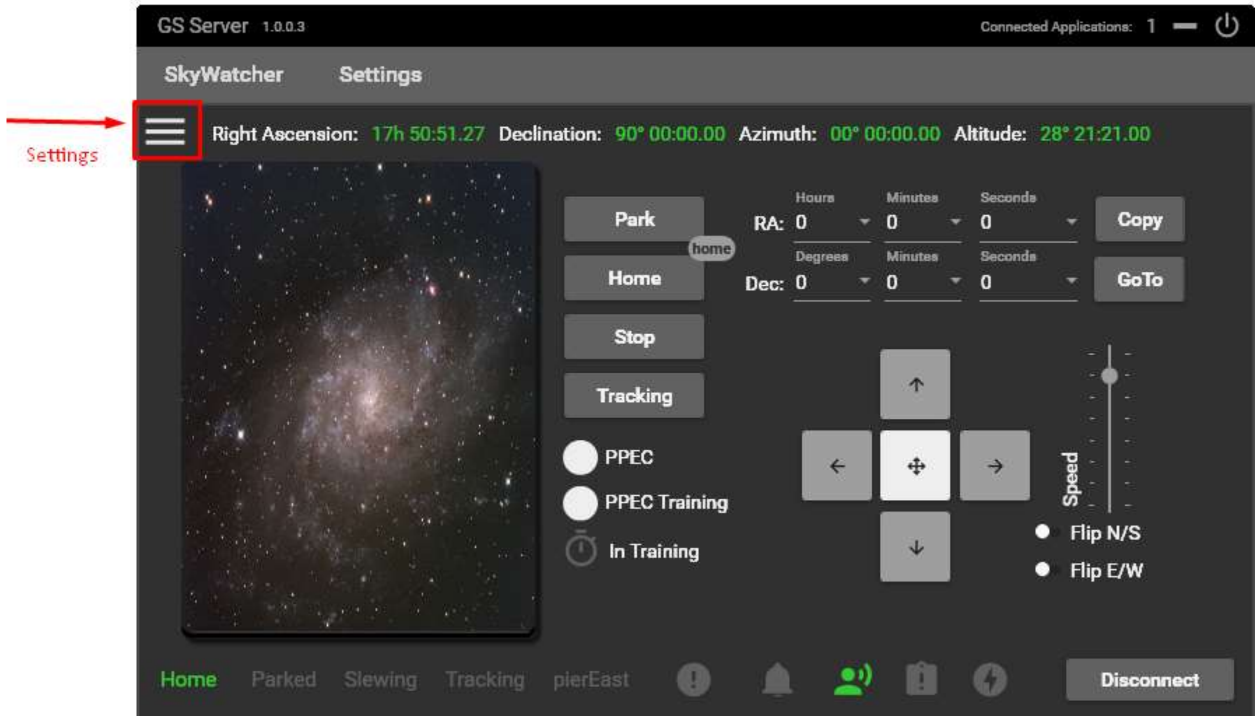
Axis Limits – is yellow when one or more of the axes reaches a limit such as passing too far past the meridian.



Status Lights – Parked, Home, Slewing, and Tracking are Items that are highlighted when active

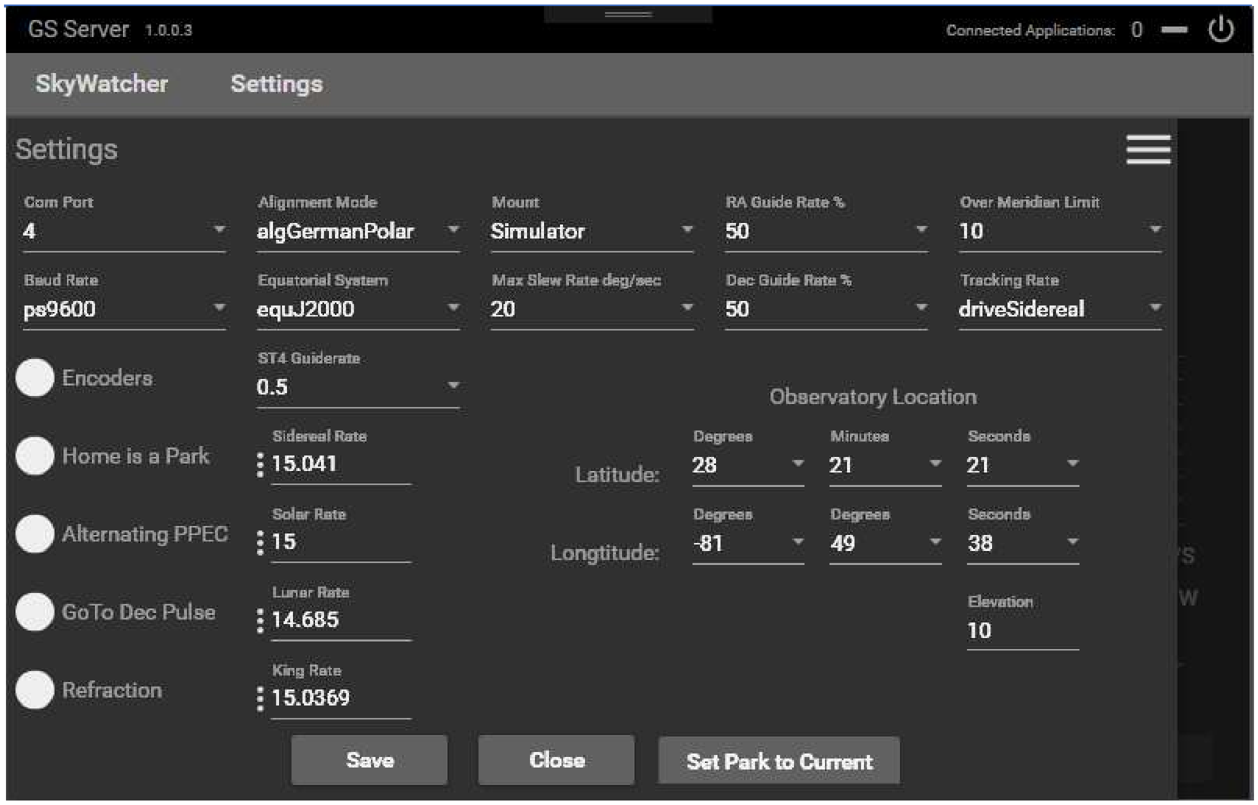
Mount connection button.  When connected Disconnect will be displayed.  When disconnected Connect will be displayed.  When connecting any errors will be logged and shown in a popup window.

Click the 3 bars to open the settings for the SkyWatcher Control.



Right Ascension: 17h 50:51.27 Declination: 90° 00:00.00 Azimuth: 00° 00:00.00 Altitude: 28° 21:21.00

| | | | |
|---|---|---|---|
| | Hours | Minutes | Seconds |
| RA: | 0 | 0 | 0 |
| | Degrees | Minutes | Seconds |
| Dec: | 0 | 0 | 0 |



| | | | | |
|---|---|---|---|---|
| Com Port: 4 | Alignment Mode: algGermanPolar | Mount: Simulator | RA Guide Rate %: 50 | Over Meridian Limit: 10 |
| Baud Rate: ps9600 | Equatorial System: equJ2000 | Max Slew Rate deg/sec: 20 | Dec Guide Rate %: 50 | Tracking Rate: driveSidereal |
| Encoders | ST4 Guiderate: 0.5 | | | |
| Home is a Park | Sidereal Rate: 15.041 | | | |
| Alternating PPEC | Solar Rate: 15 | | | |
| GoTo Dec Pulse | Lunar Rate: 14.685 | | | |
| Refraction | King Rate: 15.0369 | | | |

Observatory Location

| | Degrees | Minutes | Seconds |
|---|---|---|---|
| Latitude: | 28 | 21 | 21 |
| Longtitude: | -81 | 49 | 38 |

Elevation: 10

## Sky Watcher Settings

Com Port – Lists port 1 – 20. Ports are not checked before being listed. It is up to the user to select the correct port for the mount. If you don't know the mount port open or search for Device Manager and look in the 'Ports (COM & LPT)' section.

Baud Rate – Rates from 300 to 230400 are available. Most mounts work at the default 9600 rate.

Rates – Sidereal, Solar, Lunar, and King are present from the ascom standards. Clicking the 3 dots will confirm a reset to the default setting.

Alignment Mode – GermalPolar is the default. Others will work with the Mount set to simulator.

Equatorial System – J2000 is the default but. Topocentric is also known as JNOW. Other is set to apparent. The server will accept input and output to the display the coordinates selected.

Mount – The simulator will work in all alignment modes.  Which mount is selected will be used when an application attempts to connect to the server.

Max Slew Rate – speeds for the hand controller are a percent of the max rate.   If you want slower movements change this to a smaller number.

Guide Rates – A percentage of the selected rates to use for guiding.   The default 50% is a good starting point for guiding with applications like PHD2.

Over Meridian Limit – degrees passed the meridian the mount can travel before running into a limit alert.

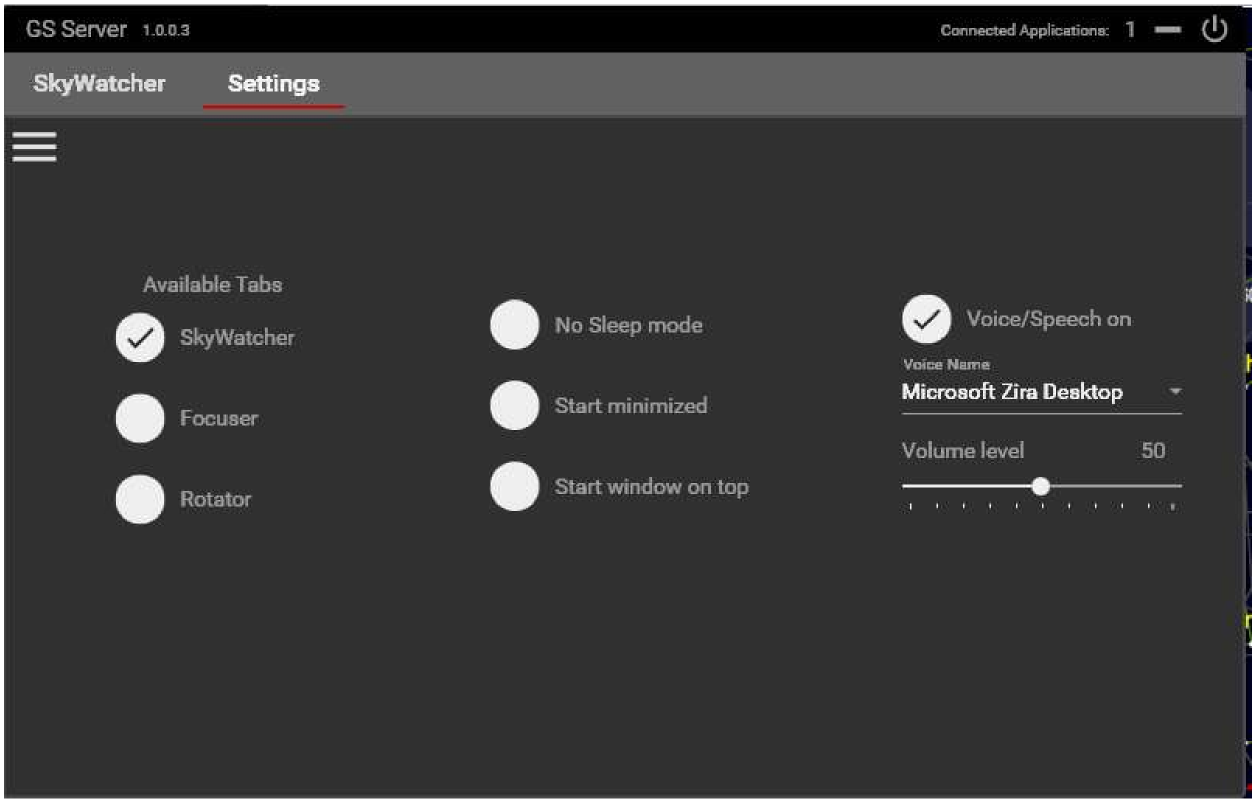
Encoders – Turns on and off the internal mount encoders.

Home is a Park – If your mount is on a permeant pier then checking this option will allow the server to store the positions of home each time.  When the mount power is turned on and the server connects it will reset the axes position to the last home exact positions, same as a park.  Mobile users should uncheck this to do a polar alignment

Alternating PPEC – This allows pulse guides to not interfere with the PPEC.  When a pulse is received the server will turn off PPEC run the pulse and then turn back on PPEC.   This is an emulation of the SynScan hand controller.

Observatory Location - Latitude and Longitude of your observatory is used to calculate the coordinates used in the display.

# 7. Settings Controls


GS Server 1.0.0.3

Connected Applications: 1

SkyWatcher Settings

Available Tabs

✓ SkyWatcher

Focuser

Rotator

No Sleep mode

Start minimized

Start window on top

✓ Voice/Speech on

Voice Name

Microsoft Zira Desktop

Volume level 50

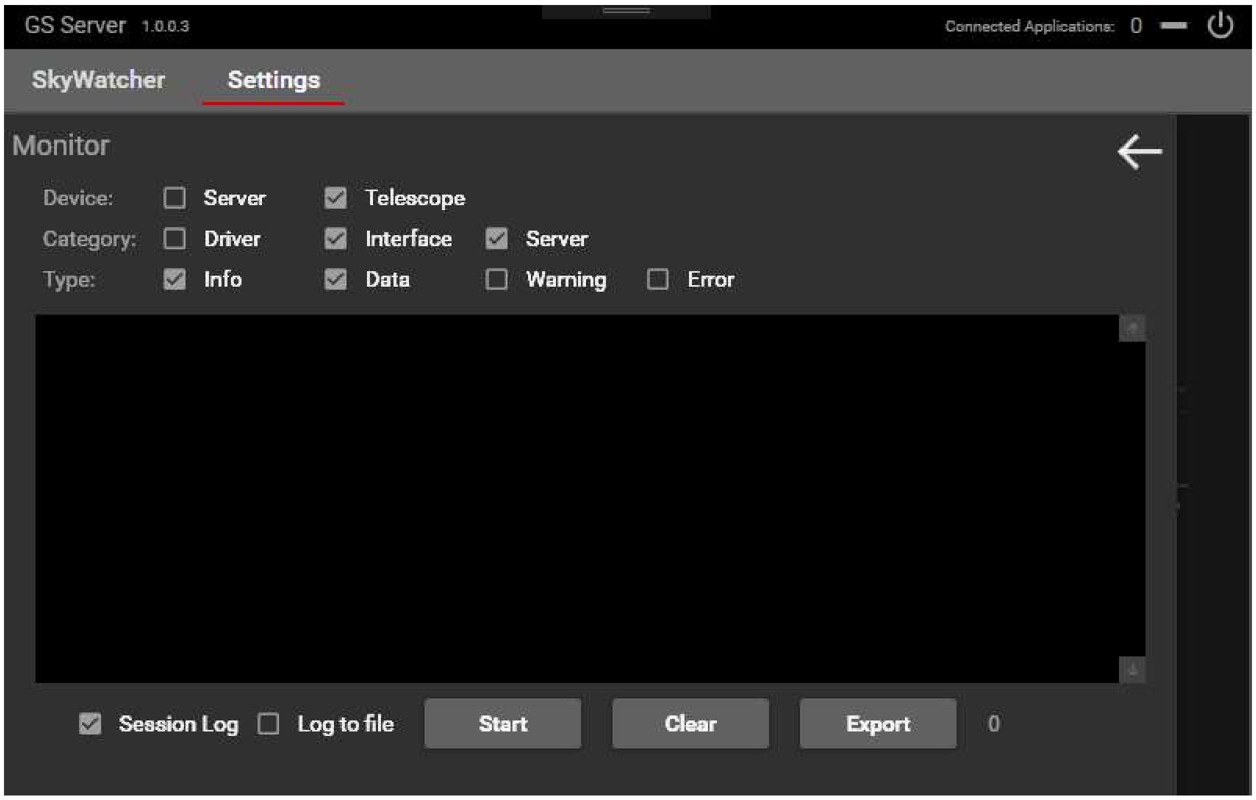
Available Tabs – Turns off or on selected tabs at top of the window. You cannot turn off the Settings tab it will always be available.  It's recommended to set what tabs you want before connected to any hardware.

No Sleep mode – When GS server is running it will move the mouse a few pixels every 50 seconds to keep the windows screen saver from starting.  This may also help to keep USB ports from going into sleep mode.

Start Minimized – When GS server is started it will show only in the windows taskbar in a minimized state. Click the GS Server icon in the taskbar to open the window to normal state.

Start Window On Top - When GS Server is started the window will be forced to show and stay on top of all other windows.

Voice/Speech on – Turns on/off the selected Microsoft synthesized voice. Windows loaded voices will be shown in the selection box. Refer to Microsoft for loading or unload voices into Windows operating systems.



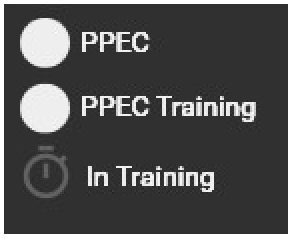
Device: [ ] Server [✓] Telescope
Category: [ ] Driver [✓] Interface [✓] Server
Type: [✓] Info [✓] Data [ ] Warning [ ] Error
[✓] Session Log [ ] Log to file

Monitor – The Monitor shows real-time logging by Device, Category, and Type. The Telescope selection refers to any control that is a type of Telescope like SkyWatcher. Category of Driver is for any incoming operations directly to the ascom driver. Interface would be the User Interface commands such as the hand controller items. Category Server would be the internal workings of GS Server. When the monitor is started it will not quit until turned off, even is the GS server is exited and started again. All files logged are kept in My Documents/GSServer.

Session Log – When checked turn on the log and keeps the last 5 rolling sessions logs.

# 8. PPEC

Permanent Periodic Error Correction (PPEC) allows the mount to correct the right ascension (RA) for manufacturing errors in the worm wheel.  PPEC Training your mount will take small movement corrections and store them within the mount.  These corrections are then played back as the RA moves when PPEC is turned on.



**How to PPEC train your mount** - Before starting a training session your mount should be able to track and guide at a stable rate.  When you are happy with the way in which the mount is guiding and you thinks its stable and consistent, start a guiding session near the meridian and let it stabilize.  Start the PPEC training session by clicking the PPEC Train checkbox.  When the mount starts collecting data the timer icon will turn yellow.   The mount will continue to collect data for several minutes.  When finished the timer will turn gray and the PPEC Train checkbox is unchecked.  You can now start and stop the PPEC replay using the PPEC checkbox.  It's recommended that you use PPEC for any future tracking or guiding sessions. If you're guiding session worsens turn off PPEC and validate your guiding is back to normal.  You can retrain the mount at any time as long as the tracking is turned on.

PPEC on/off checkbox – Turn on or off playback of corrections

PPEC Training Checkbox – Used to start a new collection session

PPEC Data collection icon – Indicates mount is collecting error correction data

When using PHD2 for guiding it's recommended to turn off PPEC during the PHD2 calibration process and turn it back when finished.

# 9. SkyWatcher Scripting

WARNING! – use the API at your own risk.  It's recommended that you do not leave the amount alone while running scripts.  Green Swamp is not responsible for any damage resulting from using the API interface.

Running scripts requires that GSS be running and connected to the mount.   You can start GSS on its own or by using other external programs that will load the GSS ASCOM driver such as CDC or SGP. Unless you're experienced with how GSS separates ASCOM commands from the API It's recommended that GSS be running on its own so no other external sources can interfere or interact with running scripts.

GSS exposes a number of class members available to external programs and scripting languages.  There are Powershell example scripts located in the GSS Program files scripting directory. Find the installation section of this manual for directory locations.

The examples scripts we're created using the windows 10 Powershell ISE.  To run Powershell, type in the Windows search box "Powershell ISE" and run either the 64 bit or X86 versions.  You may run into a problems executing scripts because of Windows security settings.  There are a number of ways to correct this and you should search google for Microsoft's recommended approach.  One way is to allow the current logged in user the ability to run script.  The following command can be run within PowerShell by placing it on the first line and clicking execute or the run button.

```
Set-ExecutionPolicy -Scope CurrentUser -ExecutionPolicy Bypass -Force;
```

# 10. SkyWatcher API

This list contains the definition for the available API commands.

```
public interface ISky
{
    /// <summary>
    /// Tells GSS not to process any ASCOM moment commands for external programs using the
        ASCOM driver.
```

```
/// </summary>
/// <returns></returns>
bool AscomOn { get; set; }
/// <summary>
/// Move axis number of microsteps, not marked as slewing
/// </summary>
/// <param name="axis">>axis number 1 or 2</param>
/// <param name="steps">number of microsteps</param>
void AxisMoveSteps(int axis, long steps);
/// <summary>
/// Send a pulse command
/// </summary>
/// <param name="axis">axis number 1 or 2</param>
/// <param name="guiderate">Guiderate with percentage applied to the tracking rate, example
    15.041/3600*.5</param>
/// <param name="duration">time in milliseconds</param>
void AxisPulse(int axis, double guiderate, int duration);
/// <summary>
/// Goto position in degrees
/// </summary>
/// <param name="axis">axis number 1 or 2</param>
/// <param name="targetPosition">position in degrees</param>
void AxisGoToTarget(int axis, double targetPosition);
/// <summary>
/// Slew axis based on a rate in degrees.  Use this for small movements
/// like pulseguiding, rate changes, guiding changes, not gotos
/// </summary>
/// <param name="axis">axis number 1 or 2</param>
/// <param name="rate">rate/sec in degrees</param>
void AxisSlew(int axis, double rate);
/// <summary>
/// K Slows to a stop movement of an Axis
/// </summary>
/// <param name="axis">axis number 1 or 2</param>
void AxisStop(int axis);
/// <summary>
/// L Abruptly stops movement of an Axis
/// </summary>
/// <param name="axis">axis number 1 or 2</param>
void AxisStopInstant(int axis);
/// <summary>
/// q Axes slews must start independently
/// </summary>
bool CanAxisSlewsIndependent { get; }
/// <summary>
/// q Does mount support AZ/EQ mode
/// </summary>
bool CanAzEq { get; }
```

```
/// <summary>
/// q Does mount support dual encoders
/// </summary>
bool CanDualEncoders { get; }
/// <summary>
/// q Does mount support half current tracking
/// </summary>
bool CanHalfTrack { get; }
/// <summary>
/// q Does mount support home sensors
/// </summary>
bool CanHomeSensors { get; }
/// <summary>
/// Test result if the mount can move Dec a single step in GoTo mode
/// </summary>
bool CanOneStepDec { get; }
/// <summary>
/// Test result if the mount can move Ra a single step in GoTo mode
/// </summary>
bool CanOneStepRa { get; }
/// <summary>
/// q Does mount support a polar LED
/// </summary>
bool CanPolarLed { get; }
/// <summary>
/// q Does mount support PPEC
/// </summary>
bool CanPpec { get; }
/// <summary>
/// q Does mount support WiFi
/// </summary>
bool CanWifi { get; }
/// <summary>
/// Gets the number of steps from the angle in rad
/// </summary>
/// <param name="axis"></param>
/// <param name="angleinrad"></param>
/// <returns></returns>
long GetAngleToStep(int axis, double angleinrad);
/// <summary>
/// e Gets versions of each axis in long format
/// </summary>
long[] GetAxisVersions();
/// <summary>
/// e Gets versions of each axis in string readable format
/// </summary>
string[] GetAxisStringVersions();
/// <summary>
/// j Gets current axis position in degrees
/// </summary>
/// <param name="axis">axis number 1 or 2</param>
/// <returns>position in degrees</returns>
double GetAxisPosition(int axis);
/// <summary>
/// j Gets axis poistion counter
/// </summary>
/// <param name="axis">axis number 1 or 2</param>
/// <returns>Cardinal encoder count</returns>
long GetAxisPositionCounter(int axis);
/// <summary>
/// d Gets Axis Current Encoder count
/// </summary>
/// <param name="axis">axis number 1 or 2</param>
double GetEncoderCount(int axis);
/// <summary>
/// Multiply the value of radians/second by this factor to get a 32-bit integer for the set speed used
    by the motor board.
/// </summary>
/// <returns>factor used to get the speed</returns>
double[] GetFactorRadRateToInt();
/// <summary>
/// Inquire motor high speed ratio
/// </summary>
/// <returns>Ratio used to determine high speed</returns>
long[] GetHighSpeedRatio();
/// <summary>
/// q Get Home position
/// </summary>
/// <param name="axis">axis number 1 or 2</param>
long GetHomePosition(int axis);
/// <summary>
/// h Get Current "goto" target
/// </summary>
/// <param name="axis">axis number 1 or 2</param>
double GetLastGoToTarget(int axis);
/// <summary>
/// i Get Current "slew" speed
/// </summary>
/// <param name="axis">axis number 1 or 2</param>
long GetLastSlewSpeed(int axis);
/// <summary>
/// Margin used to move from high speed to low speed
/// </summary>
/// <returns></returns>
long[] GetLowSpeedGotoMargin();
/// <summary>
/// e Gets the complete version string
/// </summary>
/// <returns></returns>
string GetMotorCardVersion(int axis);
/// <summary>
/// Runs a motor test to see of each axis can move one step in GoTo mode
/// </summary>
/// <returns>Result of each axis test, 0=axis1 1=axis2</returns>
bool[] GetOneStepIndicators();
/// <summary>
/// s Inquire PEC Period ":s(*1)", where *1: '1'= CH1, '2'= CH2, '3'= Both.
/// </summary>
/// <param name="axis">axis number 1 or 2</param>
double GetPecPeriod(int axis);
/// <summary>
/// c Microsteps from target where the rampdown process begins
/// </summary>
/// <param name="axis">axis number 1 or 2</param>
double GetRampDownRange(int axis);
/// <summary>
/// D Sidereal rate in axis speed
/// </summary>
/// <returns></returns>
/// <param name="axis">axis number 1 or 2</param>
long GetSiderealRate(int axis);
/// <summary>
/// Gets the angle in rad from amount of steps
/// </summary>
/// <param name="axis"></param>
/// <param name="steps"></param>
/// <returns></returns>
double GetStepToAngle(int axis, long steps);
/// <summary>
/// a Steps per revolution
/// </summary>
/// <returns>Step Count</returns>
long[] GetStepsPerRevolution();
/// <summary>
/// b Frequency of stepping timer
/// </summary>
/// <returns></returns>
long[] GetStepTimeFreq();
/// <summary>
/// F Initial the target axis
/// </summary>
void InitializeAxes();
/// <summary>
/// Is mount in a connected serial state
/// </summary>
bool IsConnected { get; }
/// <summary>
/// q Is the mount collecting PPEC data
/// </summary>
bool IsPpecInTrainingOn { get; }
/// <summary>
/// q Does the mount have PPEC turned on
/// </summary>
bool IsPpecOn { get; }
/// <summary>
/// j Is axis at full stop
/// </summary>
/// <param name="axis">axis number 1 or 2</param>
/// <returns></returns>
bool IsFullStop(int axis);
/// <summary>
/// j Is axis in highspeed mode
/// </summary>
/// <param name="axis">axis number 1 or 2</param>
/// <returns></returns>
bool IsHighSpeed(int axis);
/// <summary>
/// Is mount type set to SkyWatcher
/// </summary>
bool IsServerSkyWatcher { get; }
/// <summary>
/// f Is axis slewing normal mode
/// </summary>
/// <param name="axis">axis number 1 or 2</param>
/// <returns></returns>
bool IsSlewing(int axis);
/// <summary>
/// f Is axis slewing in a positive direction
/// </summary>
/// <param name="axis">axis number 1 or 2</param>
/// <returns></returns>
bool IsSlewingFoward(int axis);
/// <summary>
/// f Is axis slewing in goto mode
/// </summary>
/// <param name="axis">axis number 1 or 2</param>
/// <returns></returns>
bool IsSlewingTo(int axis);
/// <summary>
/// e Identify type of mount
/// </summary>
bool MountType { get; }
```

```
/// <summary>
/// e Identify board version
/// </summary>
bool MountVersion { get; }
/// <summary>
/// Turns PPEC off during movements and then back on for error correction moves
/// </summary>
/// <param name="on"></param>
void SkySetAlternatingPpec(bool on);
/// <summary>
/// E Reset the position of an axis
/// </summary>
/// <param name="axis">axis number 1 or 2</param>
/// <param name="position">degrees</param>
void SetAxisPosition(int axis, double position);
/// <summary>
/// M Set the break point increment
/// </summary>
/// <param name="axis">axis number 1 or 2</param>
/// <param name="stepsCount">The steps count.</param>
void SetBreakPointIncrement(int axis, long stepsCount);
/// <summary>
/// Turns on or off converting a Dec pulse guide into a Dec GoTo
/// </summary>
/// <param name="on"></param>
void SetDecPulseToGoTo(bool on);
/// <summary>
/// W 4-5 Turn on off encoders
/// </summary>
/// <param name="axis">axis number 1 or 2</param>
/// <param name="on"></param>
void SetEncoder(int axis, bool on);
/// <summary>
/// W 6 Enable or Disable Full Current Low speed
/// </summary>
/// <param name="axis">axis number 1 or 2</param>
/// <param name="on"></param>
void SetFullCurrentLowSpeed(int axis, bool on);
/// <summary>
///  H Set the goto target increment in steps
/// </summary>
/// <param name="axis">axis number 1 or 2</param>
/// <param name="stepsCount"></param>
void SetGotoTargetIncrement(int axis, long stepsCount);
/// <summary>
/// W 8 Reset the home position index
/// </summary>
/// <param name="axis">axis number 1 or 2</param>
```

```
    void SetHomePositionIndex(int axis);
    /// <summary>
    /// J Start motion based on previous settings
    /// </summary>
    /// <param name="axis">axis number 1 or 2</param>
    void StartMotion(int axis);
    /// <summary>
    /// G Set a different motion mode
    /// </summary>
    /// <param name="axis">axis number 1 or 2</param>
    /// <param name="func">'0' high speed GOTO slewing,'1' low speed slewing mode,'2' low speed
          GOTO mode,'3' High slewing mode</param>
    /// <param name="direction">0=forward/right, 1=backaward/left</param>
    void SetMotionMode(int axis, int func, int direction);
    /// <summary>
    /// W 2-3 Turn on off PPEC
    /// </summary>
    /// <param name="axis">axis number 1 or 2</param>
    /// <param name="on"></param>
    void SetPpec(int axis, bool on);
    /// <summary>
    /// W 0-1 Turn on off PPEC training
    /// </summary>
    /// <param name="axis">axis number 1 or 2</param>
    /// <param name="on"></param>
    void SetPpecTrain(int axis, bool on);
    /// <summary>
    /// I Set slewing rate, seems to relate to amount of skipped step counts.
    /// </summary>
    /// <param name="axis">axis number 1 or 2</param>
    /// <param name="stepSpeed">StepSpeed = 1 motor step movement, higher counts means slower
          movements</param>
    void SetStepSpeed(int axis, long stepSpeed);
    /// <summary>
    /// S Set absolute goto target
    /// </summary>
    /// <param name="axis">axis number 1 or 2</param>
    /// <param name="position"></param>
    void SetTargetPosition(int axis, double position);
}
```

# 11. Learn More