# Goodman Spectrograph Control System Scripting Software Manual

DRAFT 10/24/2018

Stephen Fanale

stephenfanale@hotmail.com

## Contents

# 0. Introduction

The goal of this document is to provide detailed information of the Goodman Spectrograph Control System (GSCS) Scripting Project. This project was designed to both improve the abilities of local Goodman observers by leveraging software and to prepare Goodman and SOAR for integration with the LCOGT Network. As a result of the two different primary objectives of this project, there are two distinct use cases that involve different elements of the GSCS Scripting Software Suite. A description of both is provided in the next section.

Most generally, this project provides the tools necessary for a user to control the GSCS without having to interact with that software directly. Rather than a user commanding each step of their observing process directly, “script” files will be created and parsed by an “observation manager.” This “observation manager” will then govern the observing process for the user by organizing the information into commands for the telescope, calibration equipment, the Goodman Spectrograph, and the imaging software. All of these commands are communicated through a TCP/IP connection to the GSCS Scripting Server, which runs locally on one of the spectrographs operating computers and handles the distribution and execution of these commands.

# 1. Description of Use Cases

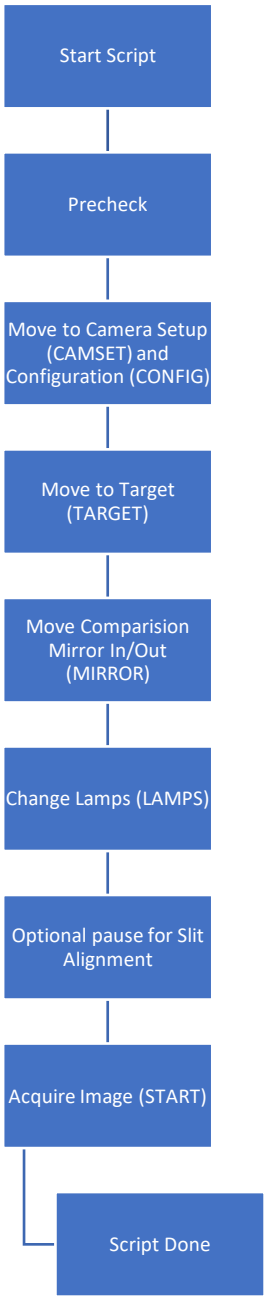
## a. Introduction to Using the GSS

The Goodman Spectrograph Control System Scripting Software suite is designed to extend the capabilities of the Goodman Spectrograph Control System. It does not replace GSCS and GSCS needs to be running and systems need to be homed before attempting to use the GSS. The GSCS front panel can be used to monitor Goodman systems and setup throughout the observing process. While the GSS is actively scripting, users should not interact with the GSCS main panel. If a user needs to make changes to GSCS then they should pause the GSS before doing so.

The GSS has two main use cases which will be described in detail in the next two sections. The tools are designed to be as flexible as possible so that different users can create custom scripts that behave as they need them too. One of the largest places the scripting tools can help local users is afternoon calibrations.

When the GSS is being used in local mode, the Scripting.vi program governs the flow of the observing process. The figure below illustrates the flow of this process. The Scripting program will dynamically adjust what happens at each step depending on what is contained on the script line. For example, if the target is not changing, the slit is removed, and a lamp turned on, the observing flow will not change targets, it will pause to move the comparison mirror in, it will turn on lamps, and it will not pause for slit acquisition.

# The Goodman Imaging Sequence

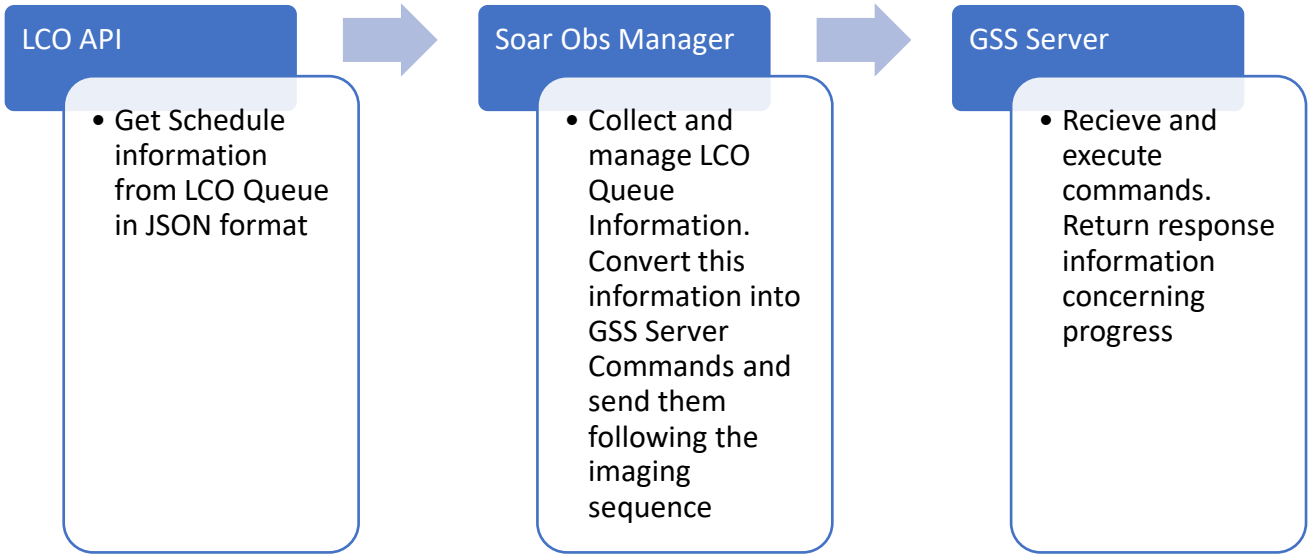


# b. LCOGT Network – Queue Based Observations

The primary long-term goal of this project is preparing for integration in the LCO Global Telescope Network and queue-based observations for rapid LSST follow-up. By the nature of queue-based observing, this is the easiest use case. That said, there are some complex steps.

This use case will be used only in preapproved time slots that will be governed by SOAR, CTIO, and LCO. This project is still under development and the following paragraph is an outline of what to be expected from this mode. Currently, the GSS Server is on port 6009, with a service name of GSS. It can accept local and remote connections. In order to connect from an outside machine, Windows Firewall settings will need to be modified on the Goodman Computers. It is also worth noting that the GSS server must be running on whichever computer's camera will be used. It is suggested to use the red camera.

In order to use the system in this way, (1) the GSCS will need to be running with all systems homed, (2) the GSS will need to running, and (3) the Soar Observation Manager (SOM) (This tool does not yet exist) will need to be running and connected to both the GSS and to the LCOGT through their provided REST API. The SOM will poll the LCOGT API for schedule information on targets and observations, organize this information into a script, and send commands over to the GSS. The GSS in turn will handle these commands and relay information back to the SOM, and the SOM will relay information back to LCOGT. The future SOM will replace the Scripting.vi program that is used in Local mode. In this case, the JSON script files are not relevant. Instead, the SOM will need to parse LCO JSON files and send commands as described in section 3c. The GSS Server intentionally accepts a flexible range of commands. The SOM can either send commands individually, or send grouped commands (CAMSET, SETUP,…) and allow the GSS Server system to only change what needs to be changed.



## c. Local User Scripting

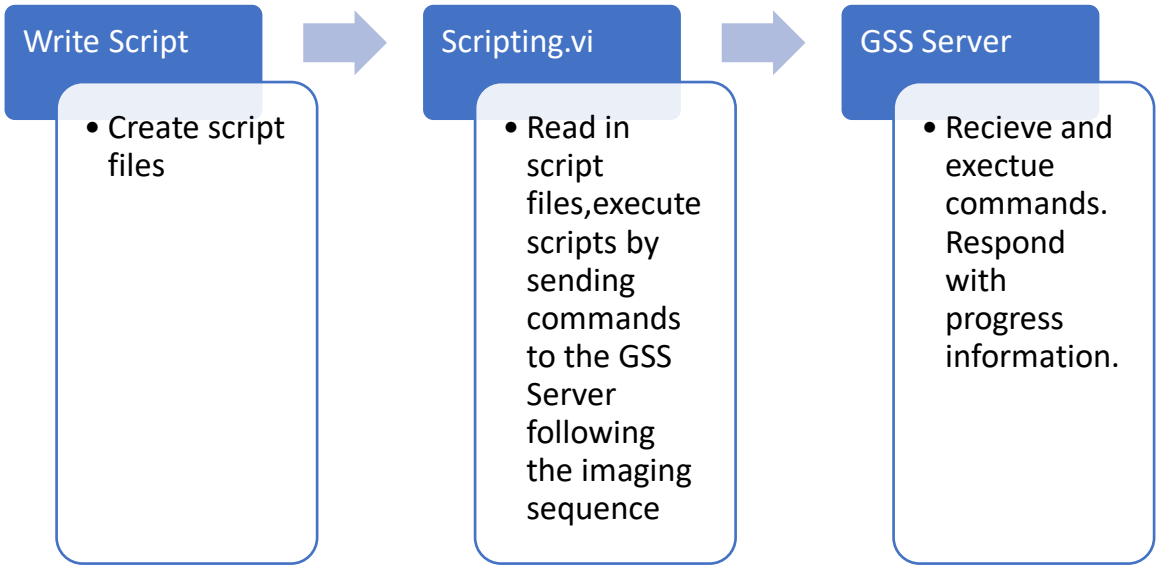
The local user scripting use case is the secondary goal of this project. It exists both as part of the development process for Use Case A, and to improve the observing experience of classical style observers. As this is for classical observing, the usage is much different and more complex. However, this allows the system to be very flexible to what a classical observers goals are for using the software.

The most straightforward use of the local system is for afternoon calibrations. This once tedious process can be made much faster, more reliable, and easier using the local scripting tools. The tools can also be prepared for either particular target observations for the night, particular styles of observation, or for a full night.

In order to use the local scripting capabilities, the first thing a user must do it prepare a script. GSCS does not need to be running for you to write a script, but you will want to make sure that the operators have already updated the files configuration files for the instrument. Then launch the WriteScript.vi either from the Desktop or from the Scripting Suite folder. This program will allow you to produce scripts that can be read and executed by the Scripting.vi. (1) In this program the first thing you want to do is give your script a name. (2) Next, you are faced with a screen that contains essentially all of the information on the GSCS front panel, but compressed down. You will want to be sure to select the camera color you want to use, name your file, name your setup, and then any of the parameters you planned to use for this image or sequence of images. A script line functions exactly the same as a single image or sequence of images in GSCS. It will go to the setup you ask and take the number of images you've asked for in the method you want them. Then it will move on to the next script line. (2b) The only exception to this is if you select the focus tab and include focus information. This allows you to access the latest focus test settings for GSCS, and you don't need to enter a new line for each of your camera focuses. Just enter in your setup, choose the focus tab, and enter in the information about your focus test. (3) Once you have finished entering in the information for your scripting line, just click add to script. Your script now includes this line. You can change which line of the script you are writing on using the script line arrow keys. You can also go to a line you have already filled out and modify it or clear it. If it is your first time using the system, it would be good to know that the Scripting program will automatically pause where ever it needs to, such as changing targets, or at the end of a focus test so that you can update the focus. Once you update your focus value for a setup name, the Scripting program will substitute your focus value for the one in the script until the Scripting program is stopped. This means feel free to ignore the camera focus for your setup while writing your script after the focus test and continue adding calibrations at this setup. (4) Once you have added in all of the lines you want in your script, remember to save your script.

When you are ready to execute your script, startup GSCS, log in, and home systems. Then launch the program called Scripting.vi on the desktop. This program acts as a local version of the SOAR Observation Manager. From this program you can load in your script, view some basic information about the setups you've included, start/stop/pause your script, and view feedback on the progress of your script. You are also welcome to watch the GSCS front panel to watch your scripts execute.

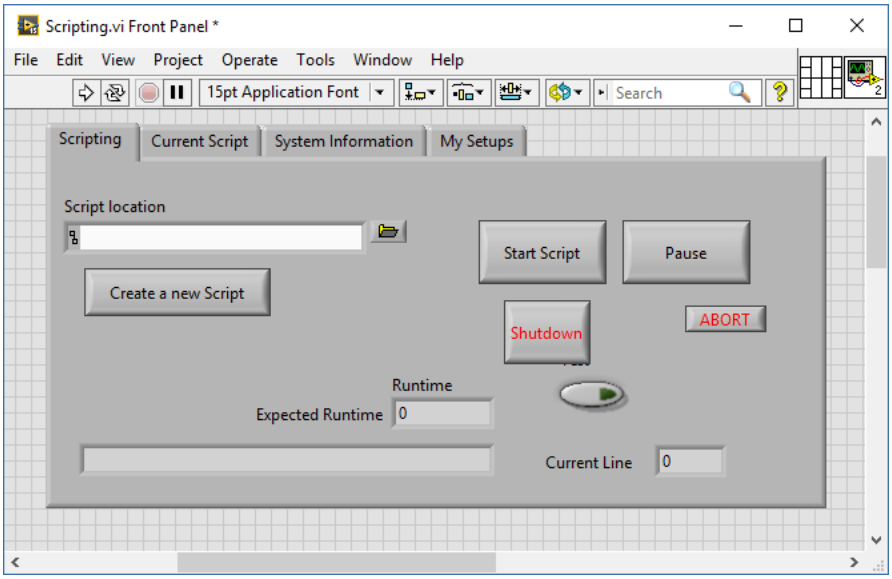
Though interaction is currently allowed, it is strongly recommended that you do not touch the GSCS front panel, or at least ensure the Scripting program is paused before you interact. (1) In order to run your script, you can select the open button and chose your script from the file explorer. (2) Once your script is read in, just click run. (3) Progress of your script is displayed in the dialogue bar at the bottom and the scripting tool will tell you when it's done.



# 2. Functional Description of Main Vi's

There are many LabVIEW vi's contained in the GSCS Scripting library. This section will provide a functional description of the key programs and will provide a working understanding of the nature of the scripting library.

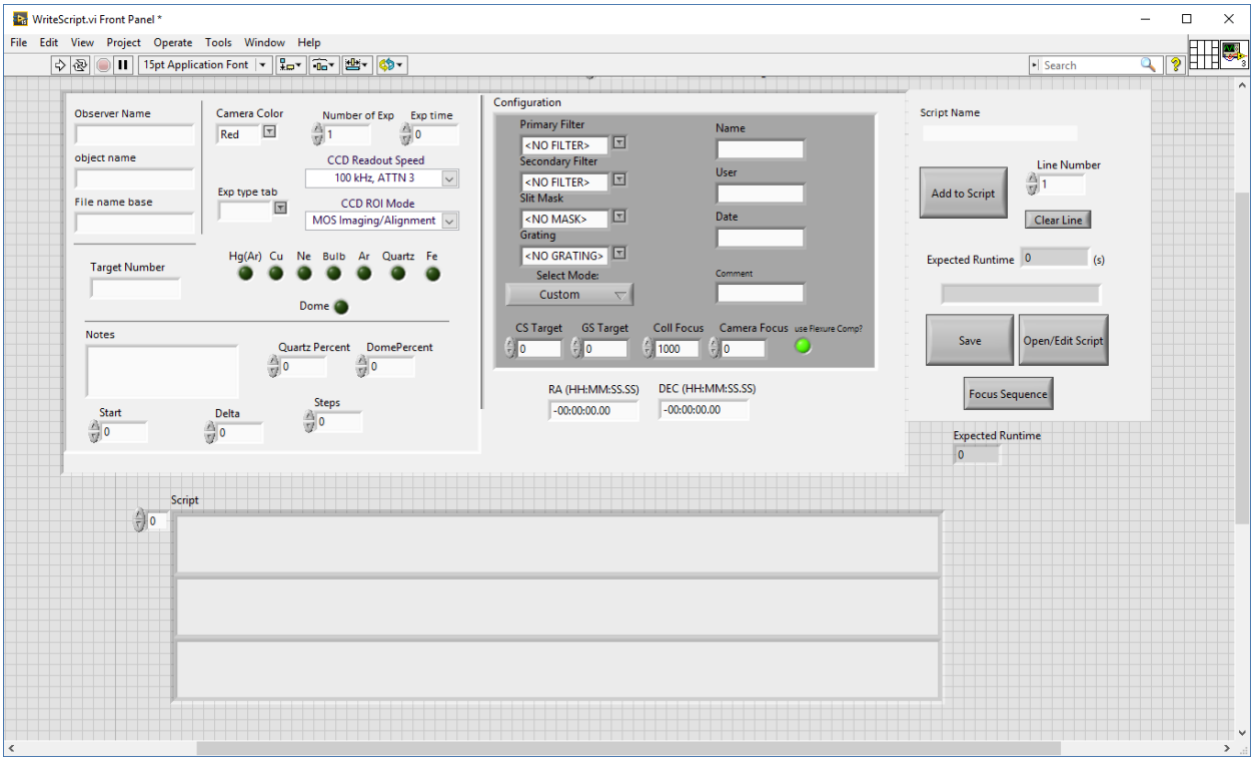
a. Scripting.vi




The main user interface of the GSCS Scripting library is the Scripting.vi. This program allows users to load and run scripts and provides feedback on progress. A full description of the features and operation of this vi are provided elsewhere.

Functionally, this vi is the topmost program in the system. It uses information about the script to govern the observing process. For example, if a script calls for lamps after an exposure, the Scripting.vi will recognize this and send the commands necessary to insert the comparison mirror and turn on the required lamps before beginning the next exposure. All such commands are sent via a TCP/IP connection the to GSCS_Server.vi (1.d). A detailed description of the nature of this communication system is provided elsewhere, and the functional overview will be captured in the following sections.

b. WriteScript.vi

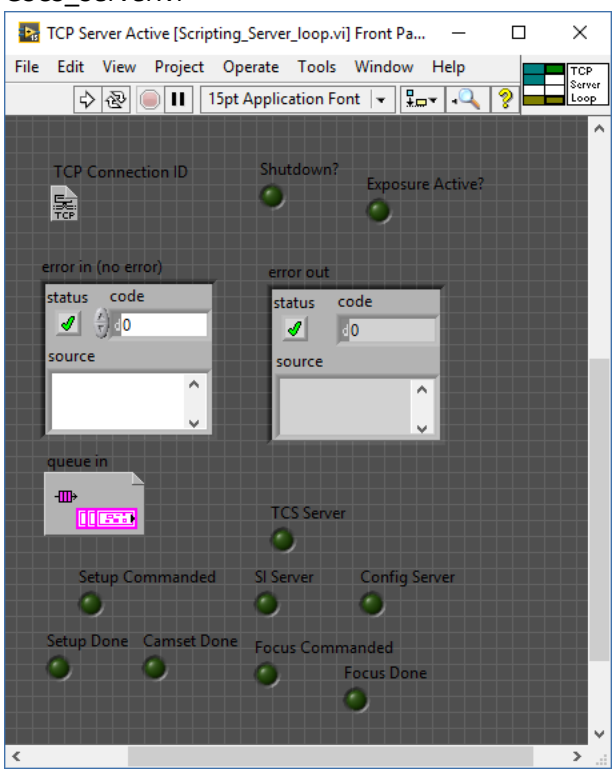



For local users, the WriteScript.vi can be used to generate scripts that will be accepted by the Scripting.vi. A full description of the features and operation of this vi are provided elsewhere. This program uses the current master configuration of the instrument to populate the same fields a user would find in GSCS_Main.vi. The information selected is written to a JSON file, in a specific format as required by the Scripting.vi

c. saveSetup.vi

The saveSetup.vi is a simple program that allows a local user to enter in a setup and save it. This can be helpful is there is a setup a user plans to use very often, because the WriteScript.vi will accept either a full setup description or a path to a setup file.

d. GSCS_Server.vi




The GSCS_Server.vi is the program that allows the Scripting.vi to connect to the instrument. The bulk of the work is done by a program within this program called “scripting_server_loop.” All communication between the scripting engine and the server is handled via a TCP/IP connection that is outlined elsewhere.

The server receives commands from the scripting engine and sends them to the necessary handler by adding commands with necessary parameters to their respective queues. TCS commands are added divided up between the ChangeSIServer.vi and ChangeTCSServer.vi programs. Commands concerning the camera and exposure setup are sent to the SI camera manager (GSCS) through the ChangeSIServer.vi. Commands are written to the queue with the command to be executed as the command, and the full cluster of Si setup details as the parameter. Commands concerning the physical setup of the instrument are sent to their respective motor drivers through the ChangeConfigServer.vi. Similar to ChangeSIServer.vi, the commands are written to the queue with the command to be executed as the command and the full setup cluster as the parameter. In both cases, this system allows for the GSCS_Server.vi to rapidly tell the handler programs which commands to execute and provide the information needed in a variable format.

Once commands are received and performed the GSCS_Server.vi responds to the Scripting.vi with command received and command results. While an exposure is active, the GSCS_Server.vi will not accept commands other than the “Abort” command until the exposure is processed as done by the ChangeSiServer.vi.

e. ChangeSIServer.vi

ChangeSIServer.vi is one of the two main handler programs under the GSCS_Server.vi. This program exists in a loop waiting for commands to be added to its queue by the GSCS_Server.vi. While exposing, this program also monitors the status of images to tell the GSCS_Server.vi when the exposure is done.

Commands are read from the queue with the required setup details provided in from the full setup cluster within the parameters. This follows the standard command/parameter cluster structure of the other GSCS LabVIEW programs. The only difference is that instead of the parameter being a string, it is the full cluster of Si camera manger setup values. This makes it so the required values for a command can be easily sent from GSCS_Server.vi to the ChangeSIServer.vi by writing to its queue. The required parameter values for any command can then be accessed by an unbundle from cluster using name command. These parameters are sent to the Si camera manager using one of the Set_Control_Values_by_Index_(Sgnl) functions. These commands are described later in this section.

f. ChangeConfigServer.vi

The ChangeConfigServer.vi receives commands from the GSCS_Server.vi in a similar fashion as the ChangeSIServer.vi. It exists in a loop waiting for commands to be written to its queue by the GSCS_Server.vi. Commands are read as the command item from the queue, and the full setup cluster is contained in the parameter item. When a command is read, the ChangeConfigServer.vi first compares the requested setup items to the current setup to make sure the new setup is possible (gratings/filters requested are available) and different from the current setup. Setup values that are different and possible are added to a script to be handled by the ChangeSetup.vi program. When commanded moves are done the ChangeConfigServer.vi tells the GSCS_Server.vi the results of the operation.

g. ChangeTCSServer.vi

The ChangeTCSServer.vi program receives commands from the GSCS_Server concerning TCS specific functions such as the comparison mirror and changing target coordinates. It writes its commands directly to the TCS command queue and receives answers from the response queue.

h. ChangeSetup.vi

The ChangeSetup.vi is the program that commands the setup changes to the each of the required drivers. All of these are commands are sent by commanding the motor driver, except for the wavelength mode command. The wavelength mode command is sent using the Set_Control_Values_by_Index_(Sgnl).vi function. Once all of the required motors have been commanded, the ChangeSetup.vi waits for the motor status lights to become green or red.

When all lights are green, the commanded setup has been accomplished and the list of completed commands is returned to the ChangeConfigServer.vi program. If there is a red light, an error will be returned to the ChangeConfigServer.vi program.

i. Set_Control_Values_by_Index_(Sgnl).vi

This program is used to change front panel values on programs or values that do not work with the driver/program structure of the motor programs. Using the vi and control reference, the values are changed and then the change is signaled to the vi. This signal allows the vi to handle the changed value in the same way it would if a user had clicked on the value on the front panel. There are some variations on this program. The most common version of this program compares the current and requested values and only changes the values deemed different. The version used on the lamps does not do this, and always commands the lamps to be the values asked for it. This has to do with the nature of the lamp controls. Finally, a third version is used for the exposure type tab only. This version contains a short wait after its command. This wait gives the Si camera manager time to updated the control references before proceeding. If the commands visible on the screen ever change, such as using tabs, this version should be used because control references are not constant.

# 3. TCP Communication and the GSS

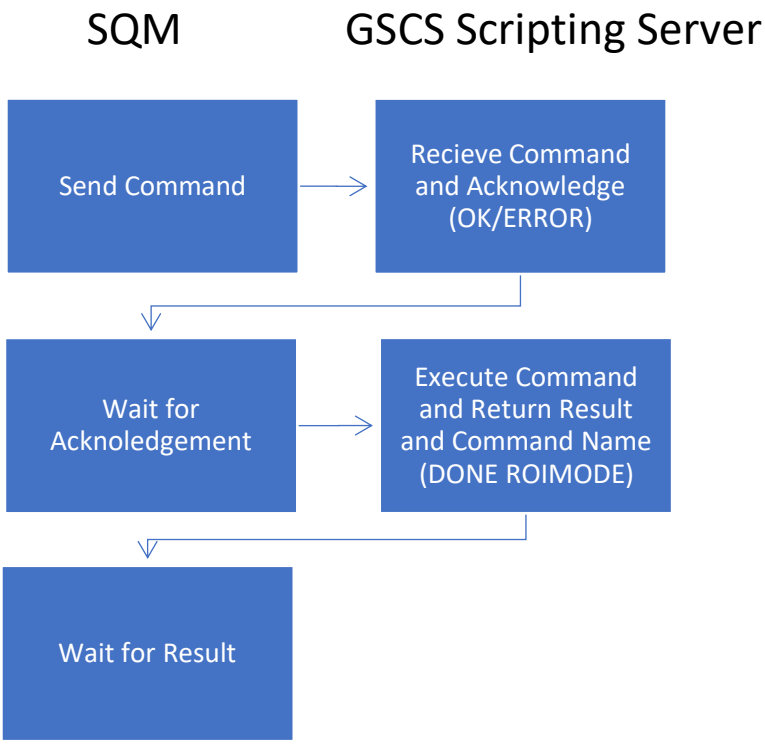
## a. Introduction and Explanation

The GSCS Scripting Server (GSS) is the local server that will run on both of the Goodman Spectrograph Control computers. The GSS radically changes the observing experience of the Goodman Spectrograph. Almost all commands and instrument monitoring are now remotely accessible.

The server will receive commands from the Soar Queue Manager (SQM), which will be governing the observing process. Communication between the SQM and GSS occur over a TCP/IP port using a communication system based on the SCLN. Commands will be sent as strings in the format of COMMAND PARAMTER1 PARAMETER… A full list of all commands with a description and example are listed in section 2. NOTE: commands and parameters are separated by tab constants NOT spaces. The full command packet is SIZE (4 bytes) COMMANDSTRING (Variable size).

The communication between the two will follow a basic pattern shown in the figure below. Commands from the SQM will be received by the GSS, which in turn will reply to acknowledge that the command was received without error. The GSS will then attempt to execute the command. Depending on the result of the execution, the GSS will send the SQM a command done or error packet. The command done and error packet consist of the status and the command that caused the status.

While the GSS is executing a command, it will also accept new commands from the SQM. The exception to this is if an exposure is active. In this case only the abort exposure command will be accepted. All other commands will return an error. There are two information commands available for the SQM to query the status of the Goodman Spectrograph. These commands, system status and system position, do not return a standard acknowledgment packet. Instead, they return the acknowledgment “OK” followed by a string including the information requested.


SQM
GSCS Scripting Server
Send Command
Recieve Command and Acknowledge (OK/ERROR)
Wait for Acknoledgement
Execute Command and Return Result and Command Name (DONE ROIMODE)
Wait for Result

## b. Example JSON Script

Below is an example of the JSON Script template accepted by Scripting.vi. Several of these values are not actively being used, but are available for future features to the Scripting.vi. Again, it is also important to note that if a new version of Scripting.vi is created for use in LCO mode, then this JSON script does not need to be use. In that case, please see the next section for all available commands that the GSS Server can accept. Commands to the GSS Server will still be needed in LCO mode.

```
4.
    gssTemplate = {"SI camera info":
                  {"Observer Name":"name","object name":"object_name","File name base":"file_name",
                   "Camera Color":"Red","Exp type tab":"Object","Number of Exp":1,"Exp time":1,"CCD Readout
    Speed":1,
                   "CCD ROI
    Mode":9,"Hg(Ar)":False,"Cu":False,"Ne":False,"Bulb":False,"Ar":False,"Quartz":False,
                   "Quartz Percent":0,"Fe":False,"Notes":"","Target Number":"","DomePercent":0,
                   "Dome":False,"Start":0,"Delta":0,"Steps":0,
                   "Custom Roi":
                     {"0":0,"1":4192,"2":1,"3":1800,"4":200,"5":2}},
                "Configuration":
                  {"Primary Filter":"<NO FILTER>","Secondary Filter":"<NO FILTER>","Slit Mask":"<NO SLIT>",
                   "Grating":"<NO GRATING>","CS Target":0,"GS Target":0,"Coll Focus":1000,"Camera Focus":0,
                   "Name":"setup_name","User":"","Date":"","Comment":"","use Flexure Comp?":True,"Select
    Mode:":34},
                "RA (HH:MM:SS.SS)":"-00:00:00.00","DEC (HH:MM:SS.SS)":"-00:00:00.00"}
```

# c. Command Library

## 1 SI Camera Manager Commands

### 1.1 Number of Exposures

- Number
- EXPNUM NUM
- EXPNUM 5

### 1.2 Exposure Time

- number
- EXP_TIME NUM
- EXP_TIME 5.5

### 1.3 Readout Speed

- Number to select option from preset list
- ROSPEED NUM
- ROSPEED 2

| Number | Red Camera | Blue Camera |
|---|---|---|
| 0 | 750 kHz, Attn 0 | 50 kHz, Attn 0 |
| 1 | 750 kHz, Attn 2 | 50 kHz, Attn 2 |
| 2 | 344 kHz, Attn 0 | 50 kHz, Attn 3 |
| 3 | 344 kHz, Attn 2 | 100 kHz, Attn 0 |
| 4 | 100 kHz, Attn 0 | 100 kHz, Attn 2 |
| 5 | 100 kHz, Attn 2 | 100 kHz, Attn 3 |
| 6 | | 200 kHz, Attn 0 |
| 7 | | 200 kHz, Attn 2 |
| 8 | | 400 kHz, Attn 0 |

### 1.4 ROI Mode

- Number to select option from preset list
- ROI NUM
- ROI 2

- Mode number maps to a mode name that is described by six numbers on the chip
- Parameters are Serial Origin, Length, Binning, Parallel Origin, Length, Binning

| Number | Mode | ROI Blue | ROI Red |
|---|---|---|---|
| 0 | Imaging, 1x1 | 516:3096:1<br>500:3096:1 | 530:3096:1<br>388:3096:1 |
| 1 | Imaging, 2x2 | 516:1548:2<br>500:1548:2 | 530:1548:2<br>388:1548:2 |
| 2 | Imaging, 3x3 | 516:1032:3<br>500:1032:3 | 530:1032:3<br>388:1032:3 |
| 3 | Spectroscopic, 1x1 | 0:4142:1<br>1100:1896:1 | 0:4142:1<br>980:1896:1 |
| 4 | Spectroscopic, 2x2 | 0:2071:2<br>1100:948:2 | 0:2071:2<br>980:948:2 |
| 5 | Spectroscopic, 3x3 | 0:1381:3<br>1100:632:3 | 0:1381:3<br>980:632:3 |
| 11 | MOS, 1x1 | 0:4142:1<br>740:2514:1 | 0:4142:1<br>650:2514:1 |
| 12 | MOS, 2x2 | 0:2071:2<br>740:1257:2 | 0:2071:2<br>650:1257:2 |
| 13 | MOS, 3x3 | 0:1381:3<br>740:838:3 | 0:1381:3<br>650:838:3 |
| 6 | Slicer | 0:2071:2<br>980:1048:2 | 0:2071:2<br>900:1048:2 |
| 14 | Spectroscopic, ZZ Ceti | 0:2071:2<br>1800:200:2 | 0:2071:2<br>1800:200:2 |
| 7 | Slit Imaging/<br>Alignment | 1850:400:1<br>1100:1896:1 | 1300:1500:1<br>1100:1896:1 |
| 8 | Set user-defined ROI… | | |
| 9 | Custom Mode | | |
| 21 | Custom Mode from Script | | |
| 10 | MOS Imaging/<br>Alignment | 1366:1370:1<br>790:2370:1 | 1300:1500:1<br>790:2370:1 |

1.5 Exposure Type

- Ring (number to select option from preset list)
- ROTYPE NUM
- ROTYPE 4

| Number | Type |
|---|---|
| 0 | Object |
| 1 | Flat |
| 2 | Comp |

| | |
|---|---|
| 3 | Dark |
| 4 | Zero |
| 5 | Focus |

1.6 File Name Base

- String
- FILENAME STR
- FILENAME MYFILE

1.7 Object Name

- String
- OBJNAME STR
- OBJNAME MYOBJECT

1.8 Observer Name

- String
- OBSNAME STR
- OBSNAME STEPHEN

1.9 Notes

- String
- This command only works for object type exposures
- NOTES STR
- NOTES THIS IS MY NOTE

1.10.1 Hg(Ar)

- ON/OFF
- Hg(Ar) ON/OFF
- Hg(Ar) OFF

1.10.2 Cu

- Cu ON/OFF
- Cu ON

1.10.3 Ne

- Ne ON/OFF
- Ne OFF

1.10.4 Bulb

- Bulb ON/OFF
- Bulb ON

#### 1.10.5 Ar

- Ar ON/OFF
- Ar ON

#### 1.10.6 Fe

- Fe ON/OFF
- Fe OFF

#### 1.11.1 Quartz

- Quartz ON/OFF PERCENTNUM
- Note: if turning lamp off, percent number is not needed (if it is included it will just be ignored and will not throw an error)
- Quartz ON 50
- Quartz OFF

### 1.12 TCS Offset

- Command Unavailable

### 1.13 Custom ROI Mode

- Use the custom mode number and include the six parameters
- Parameters are Serial Origin, Length, Binning, Parallel Origin, Length, Binning
- CUSTOMROI 21 SO SL SB PO SL SB

# 2 Instrument Setup Commands

### 2.1 Primary Filter

- The name of the primary filter
- FILTER1 *NAME*

### 2.2 Secondary Filter

- The name of the secondary filter
- FILTER2 *NAME*

### 2.3 Slit Mask

- The name of the Slit
- SLIT *NAME*

### 2.4 Grating

- The name of the grating
- GRATING *NAME*

2.5 Observing Mode

- Observing mode name
- WAVMODE MODESTR

2.6 Camera & Wavelength Target Angles

- doubles
- LOAD_DOUBLES CSNUM GSNUM

2.7 Collimator Focus

- Decimal string of number
- COLL_FOCUS COLLNUM

2.8 Camera Focus

- Decimal string of number
- CAMERA_FOCUS CAMNUM

2.9 Flexure Compensation

- true/flase
- FLEXURE ON/OFF

2.10 Toggle Slit hide show

- SLIT_TOG HIDE/SHOW

2.11 Rotate Slit

- Provide decimal string of slit number
- ROTATE_SLIT SLITNUM

# 3 General Commands

3.1 Start Exposure

- START

3.2 Abort Exposure

- ABORT

3.3 System Status

- Returns the thread states of all motors
- SYSTEM_STATUS

### 3.4 System Position

- Returns the current setup as a string of the number value
- SYSTEM_POS

### 3.5 Setup

- Single Command to include all main setup parameters
- SETUP F1STR F2STR SLITSTR GRATINGSTR CSNUM GSNUM CAMNUM COLLNUM FLEXBOOL WAVMODESTR

### 3.6 Camera Setup

- Single Command for all main camera settings
- CAMSET OBSSTR OBJSTR FILESTR NOTESTR EXPTYPENUM EXPNUMNUM EXPTIMENUM ROSPEEDNUM ROIMODENUM SO SL SB PO SL SB

### 3.7 Focus setup

- If using the focus tab, additional information is required and should be sent right after the CAMSET command
- The three numbers needed are the starting focus value, the step size, and the number of steps
- FOCUS STARTNUM DELTANUM STEPSNUM

### 3.8 Lamps general command

- Controls all lamp settings in one command
- Format is ON/OFF for each lamp in the following order: Hg(Ar), Cu, Ne, Bulb, Ar, Fe, Quartz, Dome, Quartz percent, Dome percent
- Note: It is recommended to include percent numbers even if their corresponding lamps are off. The numbers will be ignored if the lamp is not on.
- LAMPS ON1 ON2 ON3 ON4 ON5 ON6 ON7 ON8 QUARTZPERCENT DOMEPERCENT
- LAMPS OFF OFF OFF ON OFF OFF OFF OFF 0 0

### 3.9 Quit Server

- QUIT

### 3.10 Test Command

- Does nothing and returns OK
- TEST

# 4 TCS Commands

### 4.1 Comparison Mirror

- MIRROR IN/OUT

4.2 Move Telescope

- Change the target of the telescope using RA and DEC in the format HH:MM:SS.ss
- TARGET RA DEC

# 4 Common Errors

a. -1073807339
   Serial communication error – not communicating with motors
b. 1000
   ????
c. 1144
   ????
d. 3008

   This error is a tcp/ip communication error and can safely be dismissed. If you see this error often, please report it.

e. 56
   TCP/IP timeout error
f. 5002

Silvermax – Motor Kill Condition

g. 5000
   Silvermax – Unknown Response from Silvermax
h. 5001
   Silvermax – Silvermax NAK
i. 5003
   Silvermax – Aborted Packet
j. 5004
   Silvermax – Receive Checksum Error
k. 5005
   Silvermax – Soft stop limit
l. 5006
   Silvermax – Framing error
m. 5007
   Silvermax – Message too long
n. 5008
   Silvermax – CKS Condition met
o. 5009
   Silvermax – Receiver overflow
p. 5010
   Silvermax – Moving Error
q. 5011
   Silvermax – Holding Error
r. 5012
   Silvermax – Low/Over Voltage
s. 5013
   Silvermax – Move Stopped on Input
t. 5014
   Silvermax – Command Error
u. 5015
   Silvermax – NVM Checksum Error

# 5 Goodman Options

## GRATINGS

[SYZY_2100] mu=2100 dlogn=0.1 d=2.5 Comment="Working numbers only"

[RALC_1200-RED] mu=1200 dlogn=0.080000 d=8.000000 Comment=""

[SYZY_930] mu=930 Comment="" [SYZY_600-RED] mu=600 dlogn=0.070000 d=8.000000 Comment=""

[SYZY_400] mu=400 dlogn=0.1 d=2.5 Comment="Working numbers only"

[RALC_300] mu=300 dlogn=0.080000 d=8.000000 Comment=""

[SYZY_1800] mu=1800 dlogn=0.040000 d=5.000000

[SYZY_2400] mu=2400 dlogn=0.064000 d=4.000000

[RALC_1200-BLUE] mu=1200  dlogn=0.080000 d=8.000000

[SYZY_600-BLUE] mu=600 dlogn=0.070000 d=8.000000 Comment=""

[SYZY_600-OLD] mu=600 dlogn=0.070000 d=8.000000 Comment=""

## FILTERS

[U] Comment="Standard Johnson U band-pass filter."

[B] Comment="Standard Johnson B band-pass filter."

[V] Comment="Standard Johnson V band-pass filter."

[Rc] Comment="Standard Cousins R band-pass filter."

[U-Bessel] Comment="Standard Bessel U band-pass filter."

[B-Bessel] Comment="Standard Bessel B band-pass filter."

[V-Bessel] Comment="Standard Bessel V band-pass filter."

[R-Bessel] Comment="Standard Bessel R band-pass filter."

[I-Bessel] Comment="Standard Bessel I band-pass filter."

[u-Stromgren] Comment="Standard Stromgren u band-pass filter."

[v-Stromgren] Comment="Standard Stromgren V band-pass filter."

[b-Stromgren] Comment="Standard Stromgren b band-pass filter."

[y-Stromgren] Comment="Standard Stromgren y band-pass filter."

[u-SDSS] Comment="Standard Sloan u filter"

[g-SDSS] Comment="uprime Sloan g filter"

[r-SDSS] Comment="uprime Sloan r filter"
[i-SDSS] Comment="iprime Sloan i filter"
[z-SDSS] Comment="zprime Sloan z filter"

[GG385] Comment=""
[GG455] Comment=""
[GG495] Comment=""
[OG570] Comment=""
[S8612] Comment="red-blocking filter (similar to BG40)"

[Mask] Comment="4-Hole pupil mask"
[S8612] Comment="small red-blocking filter"
[H-alpha] Comment="H-alpha filter"
[SII] Comment=6738/50-4x4
[Hartmann Mask] Comment="4-hole Hartmann Mask"

[6600/75-4x4] Comment=6600/75-4x4
[5019/50-4x4] Comment=5019/50-4x4
[5007/55-4x4] Comment=5007/55-4x4
[GG385new] Comment=""
[GG495new] Comment=""
[OG570new] Comment=""
[SQUARE] Comment=""
[W012] Comment="WIYN Filter"
[C-Mask] Comment="Crotts mask"
[501.0/13] Comment="501.0/13.0 nm Fraga"
[521.5/13] Comment="521.5/13.0 nm Fraga"

[548.5/23] Comment="548.5/23.0 nm Fraga"

[Ha.6572/25] Comment="Winkler filters"

[SII.6734/48] Comment="Winkler filters"
[red continuum] Comment="Winkler filters"
[VR] Comment="Moskovitz filter"
[y PRECAM] Comment="y PRECAM DES 100mm"

[z PRECAM] Comment="z PRECAM DES 100mm"

[empty]
Comment="no filter"

## SLIT MASKS

[0.46" long slit] Observer=observer Field=Sky Width=0.46 Comment=""

[0.84" long slit] Observer=observer Field=Sky Width=0.84 Comment=""

[1.0" image slicer] Observer=observer Field=Sky Width=1.0 Comment="Image Slicer"

[1.0" 74" image slicer] Observer=observer Field=Sky Width=1.0 Comment="Image Slicer"

[1.0" 77" image slicer] Observer=observer Field=Sky Width=1.0 Comment="Image Slicer"

[1.03" long slit] Observer=observer Field=Sky Width=1.03 Comment=""

[1.35" long slit] Observer=observer Field=Sky Width=1.35 Comment=""

[1.5" image slicer] Observer=observer Field=Sky Width=1.5 Comment="Image Slicer"

[1.68" long slit] Observer=observer Field=Sky Width=1.68 Comment=""

[2.0" long slit] Observer=observer Field=Sky Width=2.0 Comment="in multi-slit holder"

[3.0" long slit] Observer=observer Field=Sky Width=3.0 Comment="in multi-slit holder"

[10.0" long slit] Observer=observer Field=Sky Width=10.0 Comment="in multi-slit holder"

[BIG SLIT] Observer=observer Field=Sky Width=40 Comment=""

[0.46" Pinholes] Observer=observer Field=Sky Width=0.46 Comment=""

[mask-A] Observer=observer Field=Sky Width=0.0 Comment="small dot-mask"

[mask-B] Observer=observer Field=Sky Width=0.0 Comment="large dot-mask"

[Out of service] Observer=observer Field=N/A Width=0 Comment="Indicates bad position in carousel"

[MOS Test1] Observer=observer Field=Sky Width=7 Comment=""

[MOS Test 2] Observer=observer Field=sky Width=7 Comment=""

[5" long slit] Observer=observer Field=Sky Width=5 Comment=""

[WASP69_2slits_20arsec] Observer=observer Field=Sky Width=20 Comment=""

[WASP69_2slits_1arsec] Observer=observer Field=Sky Width=1 Comment=""

## MODES

[Custom] ca=0 ga=0 Comment=""

[Imaging] ca=0 ga=0.661 Comment=""

[300 All] ca=11 ga=5 Comment=""

[600 UV] ca=15.25 ga=7.0 Comment=""

[600 Blue] ca=17 ga=7 Comment=""

[600 Mid] ca=20 ga=10 Comment=""

[600 Red] ca=27 ga=12 Comment=""

[1200 m0] ca=26 ga=16.3 Comment=""

[1200 m1] ca=29.5 ga=16.3 Comment=""

[1200 m2] ca=34.4 ga=18.7 Comment=""

[1200 m3] ca=39.4 ga=20.2 Comment=""

[1200 m4] ca=44.4 ga=22.2 Comment=""

[1200 m5] ca=49.6 ga=24.8 Comment=""

[1200 m6] ca=54.8 ga=27.4 Comment=""

[1200 m7] ca=60.2 ga=30.1 Comment=""

[1200 RESOLVE] ca=44.12 ga=24.7 Comment=""

[400 m1] ca=11.60 ga=5.8 Comment=""

[400 m2] ca=16.10 ga=7.50 Comment=""

[930 m1] ca=20.60 ga=10.30 Comment=""

[930 m2] ca=25.20 ga=12.60 Comment=""

[930 m3] ca=29.90 ga=15.00 Comment=""

[930 m4] ca=34.60 ga=18.3 Comment=""

[930 m5] ca=39.40 ga=19.70 Comment=""

[930 m6] ca=44.20 ga=22.10 Comment=""

[400WD] ca=11.71 ga=5.85 Comment=""

[930TP] ca=21.76 ga=10.88 Comment=""

[400Fermi] ca=13.8 ga=6.9 Comment=""

[400Padgett] ca=12.6 ga=6.3 Comment=""

[400Lewis] ca=15.4 ga=7.7 Comment=""

[400Crotts] ca=11.8 ga=5.8 Comment=""

[600BLee] ca=17.2 ga=8.6 Comment=""

[600RLee] ca=21.3 ga=10.6 Comment=""

[400WW] ca=14.94 ga=7.47 Comment=""

[ZZ Ceti Red] ca=35.2 ga=20.0 Comment=""

[ZZ Ceti Blue] ca=24.0 ga=13.0 Comment=""

## LAMPS

Hg(Ar) – On/off

Cu – On/off

Ne – On/off

Bulb – on/off

Ar – on/off

Fe – on/off

Quartz – on/off and percent

Dome – on/off and percent

## ADC

ADC can be In or out