

Echelle GUI User's Manual

CTIO 60 inches Echelle

ECH60S-1.3



La Serena, May 2010

Contents

Introduction.....	3
Chapter 1: Getting Things Running.....	4
1.1 Hardware Setup.....	4
1.2 Software Startup.....	4
1.3 Software Shutdown.....	4
Chapter 2: Main GUI Reference.....	5
2.1 MAIN_GUI > Exposure Progress (read-only).....	6
2.2 MAIN_GUI > Temperatures (read-only).....	7
2.3 MAIN_GUI > TCS (read-only).....	8
2.4 MAIN_GUI > ASYNC / ERROR box (read-only).....	9
2.5 MAIN_GUI > Exposure Setup (write).....	10
2.6 MAIN_GUI > Exposure Control (write).....	14
2.7 MAIN_GUI > PlugIns (write).....	15
Chapter 3: Plug-Ins Reference.....	17
3.1 Plug-Ins > DISPLAY.....	18
3.2 Plug-Ins > GEOMETRY.....	20
3.3 Plug-Ins > LAMPS.....	22
3.4 Plug-Ins > IODCEL.....	24
Appendix A: Header.....	25

Introduction

The following document is a reference to the CTIO 60 inches ECHELLE Software. It provides a way of understanding both handling and capabilities.

The software provides a Graphic Interface (GUI) for an easy manual operation; since the core of the software is ASCII-commands driven, using scripting is a natural way of handling the system. So, the user can use the GUI only or, if he/she prefers, can run the software fully based on scripts of any type (python, csh, tcl , cl, pearl or whatever the user prefer). See the scripting reference manual for scripting details (*ECH60S-3.X*)

The document describes how to startup and shutdown the software (*chapter 1*), and also how to operate the main and auxiliary GUIs (called Plug-Ins, *chapter 2* and *3*)).

As complimentary information there is a description of the fits headers on *Appendix A*

Chapter 1: Getting Things Running

1.1 Hardware Setup

Before firing up the ECHELLE software there are two hardware parts that should be powered:

- a) Orange monsoon controller
- b) Comparison Lamps control box. The Lamps control box can be turned on after the software has started; it just should be powered when the observation is in progress!

1.2 Software Startup

The user should be logged into the observer's account. From the regular X window, there are three ways of starting up the software:

- a) **Icon:** click on the desktop icon called “ECHELLE”
- b) type in any terminal or console: “start_ECHELLE”
- c) type, in any terminal or console: “start_application ECHELLE”

The startup scripts (located on \$HOME/bin) will startup all the software components (detector controller, TCS and Lamps modules, for detail see document *ECH60S-2.X* (Software architecture)). The startup process consists of a series of xterm windows (green for the application, gray for panview, the detector controller server) showing the startup progress. During this process the Main GUI will remain with a busy cursor; once all is done the GUI will become active and the cursor will come to normal.

For details on the startup scripts please see document *ECH60S-2.X* on architecture and maintenance

1.3 Software Shutdown

To shutdown the software there are four ways:

- a) **GUI:** click on the red “Exit” button on the main GUI
- b) **icon:** click on the desktop icon called “shutdown ECHELLE”
- c) type, on any terminal or console, “shutdown_ECHELLE”
- d) type, in any terminal or console, “shutdown_application ECHELLE”

In all cases a red xterm will appear showing the shutdown process.

Chapter 2: Main GUI Reference

When the application has started and it is ready, the user is presented with a single window. When the startup process started with no failures, then the three LEDs on the upper right corner of the GUI should appear GREEN, stating that the detector controller, TCS connection and temperature controller are initialized and ready. Figure 2.1 shows the main GUI and its LEDs

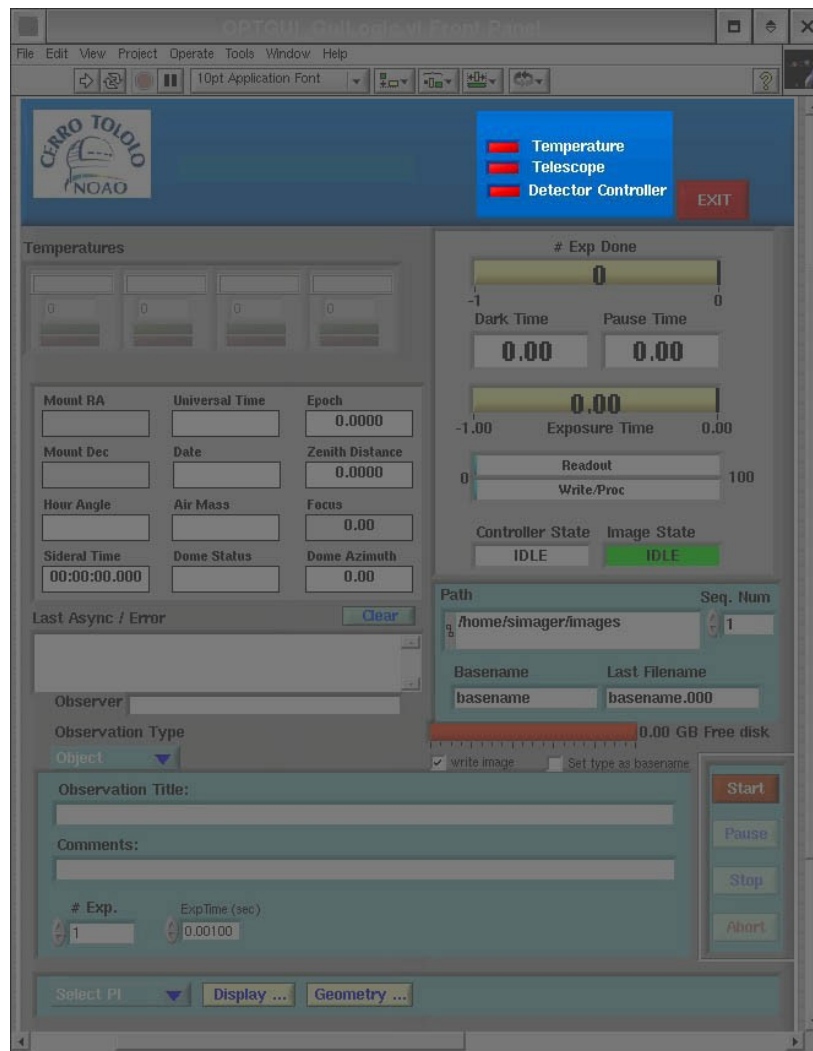


Figure 2.1: Main GUI and LEDs

2.1 MAIN_GUI > Exposure Progress (read-only)

The exposure progress area shows information regarding the image status when an observation is in progress.

Exp Done (upper yellow bar): shows the sequence (how many images has been done compared to the total amount of images requested)

Dark Time: shows the total amount of dark time in the current exposure (seconds)

Pause Time: shows the total amount of time elapsed since pause in the current exposure (seconds)

Exposure time (lower yellow bar): shows the total amount of exposure (integration time) elapsed (seconds)

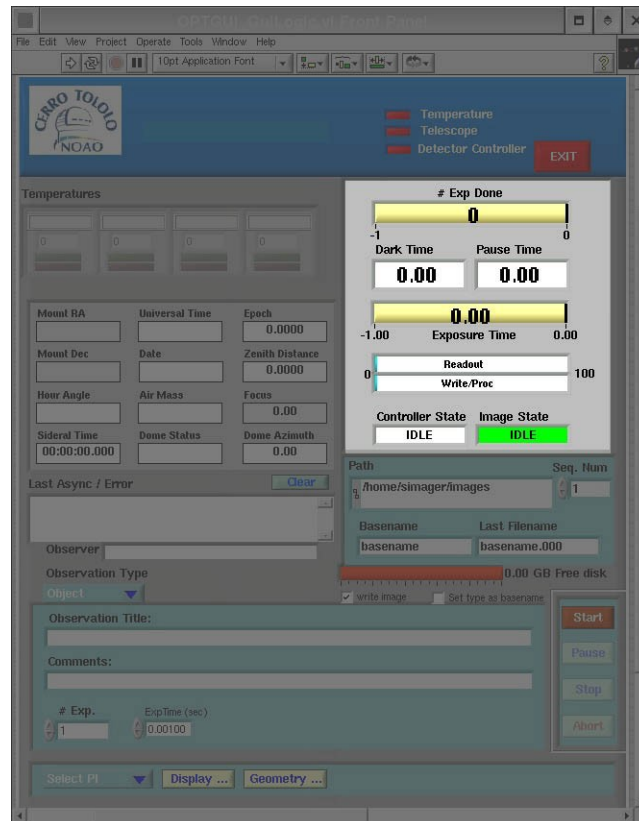


Figure 2.2: Exposure progress

Readout (upper cyan bar): shows the percentage of current image readout

Write progress (lower cyan bar): shows the percentage of current image write

Controller State: shows the detector state: **IDLE** – **EXPOSING** – **READING**

Image State: shows the image state: **IDLE** – **WAITING** – **WRITING** – **DONE**

2.2 MAIN_GUI > Temperatures (read-only)

The temperatures information area shows the current temperature of different points of interest in the system, Currently the only information corresponds to the detector (CCD) controller, but in the future there will be some others added (Echelle body, coude room temperature, etc).

The upper part shows the temperature name, then comes the actual temperature (in Celsius degrees) and finally an LED that remains green if the correspondent temperature is in reasonable ranges, and that turns red when the temperature goes out of the allowed limits (alarm). Under this condition an actual text warning message is displayed I the **async/error** dialog (see 2.4)

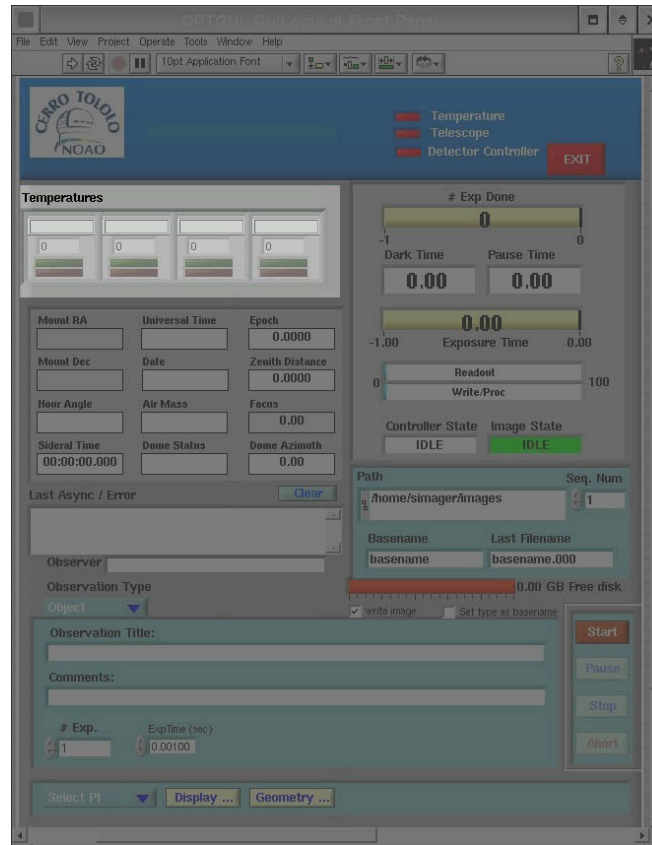


Figure 2.3: temperatures indicator

2.3 MAIN_GUI > TCS (read-only)

This part shows the Telescope information as returned by the Telescope Control System (TCS) by itself. Each field (box) name is self-explanatory. The information update rate is a parameter (see document *ECH60S-2.0* on administration), but by default it is 3 seconds (so the fields will be updated every 3 seconds)

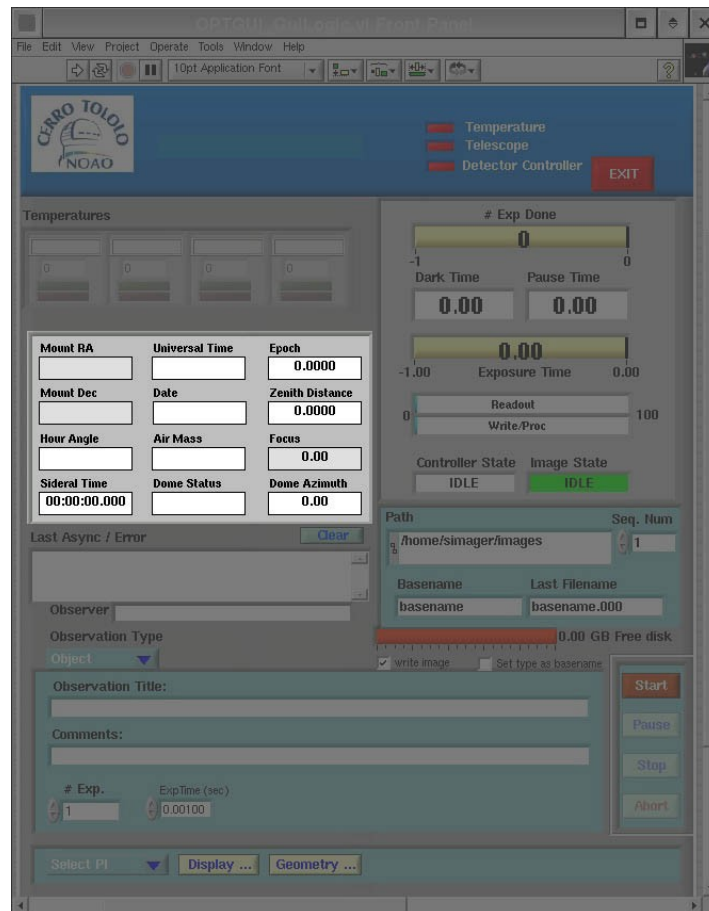


Figure 2.4: TCS info indicators

2.4 MAIN_GUI > ASYNC / ERROR box (read-only)

An asynchronous message is something not related to any specific command/request from the user, as a alarm, warning, etc (for example, if the CCD temperature is too high, etc). An error is a condition caused by a direct user command (for example, if the image could not be written because there is no enough disk space). This box displays both errors and asynchronous messages. If the asynchronous message is an alarm it may also have another associated action (like a beep)

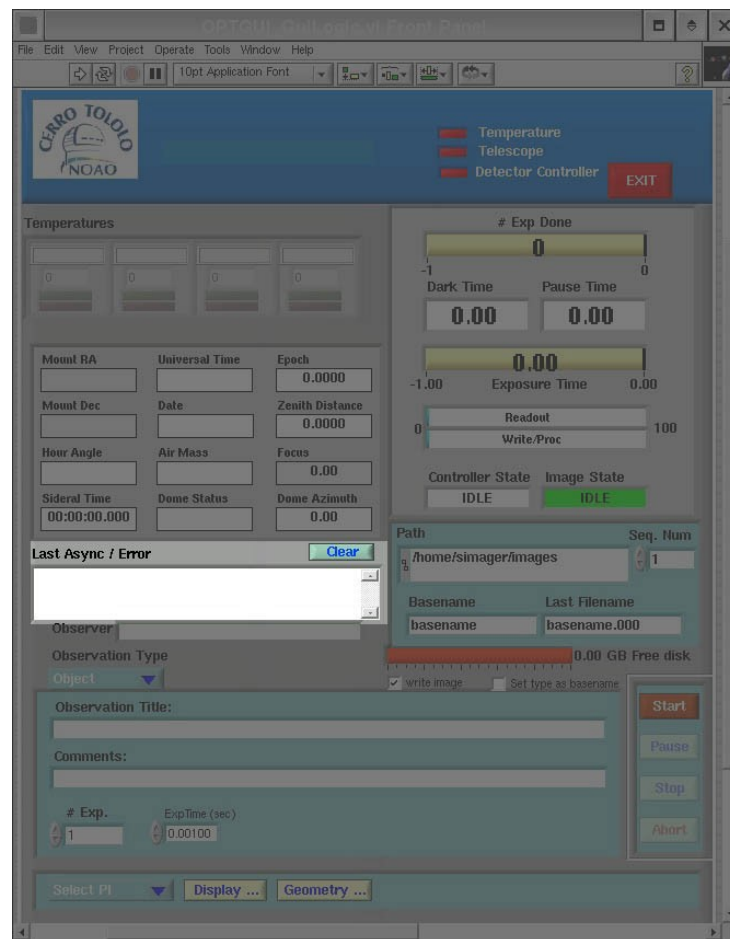


Figure 2.5: async / err box

2.5 MAIN_GUI > Exposure Setup (write)

This areas are for the exposure setup itself. These fields are all settable by the user. It allows to setup the exposure before it actually starts.

Figure 2.6 a) shows the image setup information:

Path: directory where the image will be written

Seq. Num: initial image number. It will increment automatically for every new image

Basename: base name for images

These three variables define the image on disk as:

<Path>/<Basename><Seq.Num>.fits

The number is set with 4 places, and it is left-filled with zeros, so it always has 4 digits (%04d)

Example: if Basename is “test”, and Path is “/data”,and Seq.Num is “2”, the next image will be /data/test0002.fits

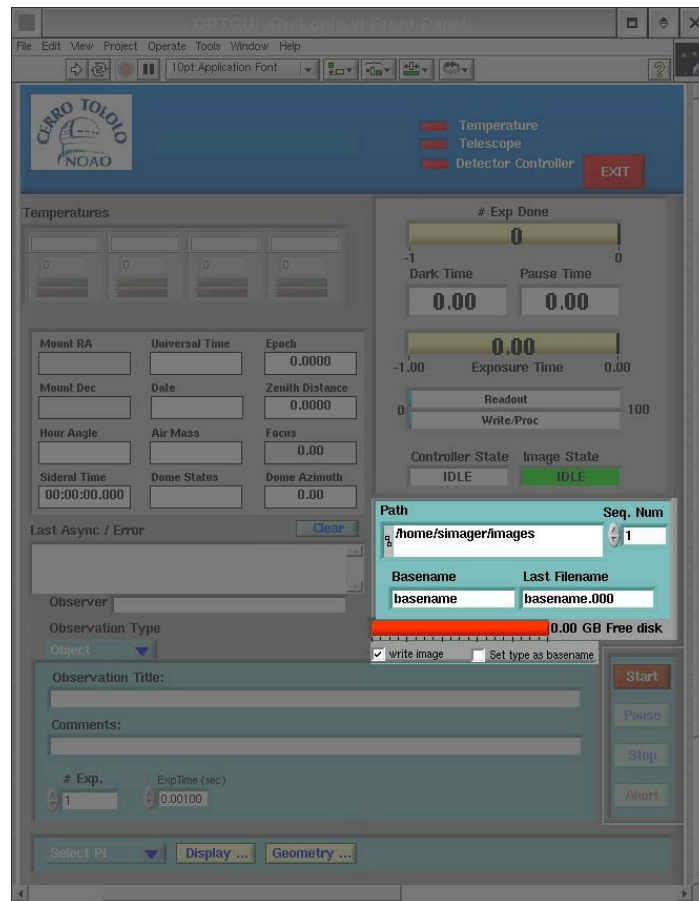


Figure 2.6 a): Image setup controls

The **red bar** shows the available space on the chosen file system (the disk where the chosen directory is. This is, of course, read-only).

The check box “**Set type as basename**”, if checked, will make that every time the image type (see below) changes, the base name changes accordingly (for example, if image type is “Dark”, the image base name will switch to “dark”, etc).

The check box “**write image**” enables or disables writing the image to disk. The normal state is TRUE, so every image taken is written to disk.

Figure 2.6 b) shows the exposure setup controls:

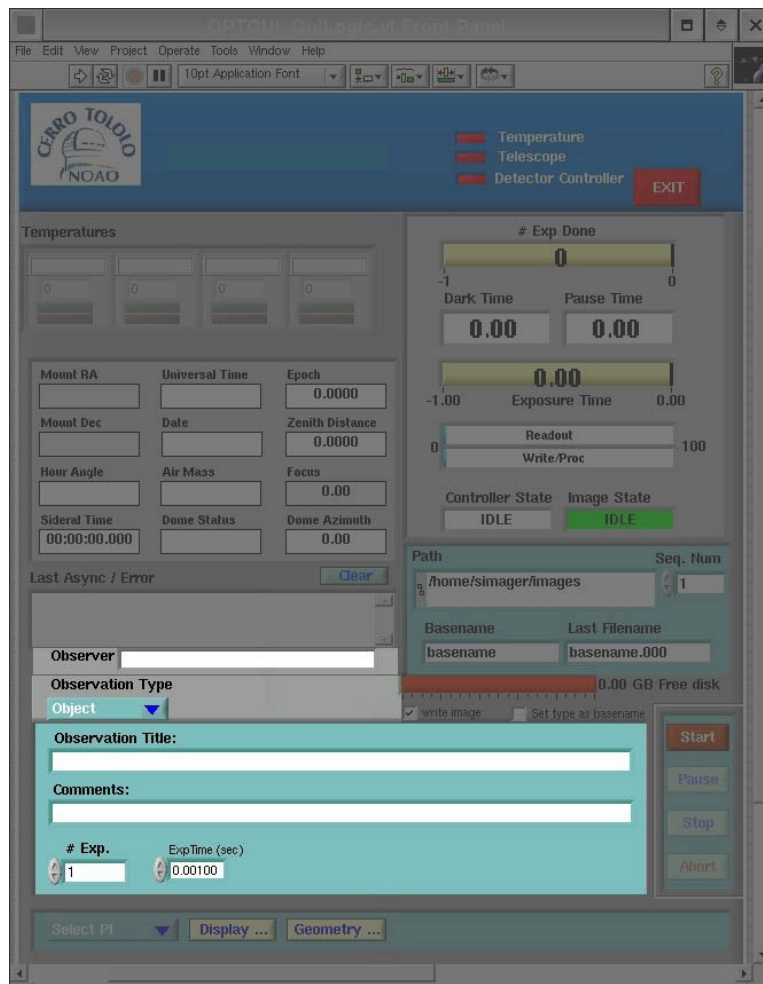


Figure 2.6 b): exposure setup

Observer: Observer (s) name. This information will appear on the “*observer*” fits key

Observation Tile (Object): This sets the object name. This will appear as the key “*object*” in the image headers

Observation Type: This sets what kind of image will be taken. This goes on the “*imagetyp*” key header. The available image types are settable through the configuration file, but the “standard” ones are:

Object:

regular object observation :

clear -> shutter_open -> integration -> shutter_close -> readout

When this type is selected there will also appear a drop down menu to select the position of the IODINE cell (IN or OUT). Based on the selection the iodine cell will be moved IN or OUT of the light path (at the moment of the selection)

Dark:

dark image -no light- :

clear -> integration -> readout

Bias:

zero (bias) frame -no integration- :

clear -> readout

If this type is selected the “ExpTime” control will not be present (always 0 exposure time)

Flat / Calibration:

calibration (reference) image using comparison lamps.

Regular observation (as an object) but the selected comparison lamps is turned “on” :

lamp on -> clear -> shutter_open -> integration /->shutter_close -> readout -> lamp off

If this type is selected, there will appear a drop-down menu showing the available lamps, and there the user can choose what lamp will be used for the

calibration. The lamp will be turned off automatically after the sequence, so the the user does not need to worry about turning on/off the lamp.

When this type is selected there will also appear a drop down menu to select the position of the IODINE cell (IN or OUT). Based on the selection the iodine cell will be moved IN or OUT of the light path (at the moment of the selection)

ExpTime (sec): controls the length, in seconds, of the integration (goes into the “*exptime*” key header)

#Exp: controls how many exposures to do -using all the above settings

Comments: Any desired commentary. This goes into the fits header as a standard fits COMMENT

2.6 MAIN_GUI > Exposure Control (write)

This allows to start the actual exposure (s) and to control it once it is in progress

Start: once the exposure has been setup (see 2.5), by clicking this button the exposure will start

Pause: allows to pause an exposure when it is on the integration phase. To resume it, press the same button again (the label will change to “**resume**”)

Stop: Stops an exposure once it is on the integration phase. When stopped, it will stop integrating and will be readout immediately. The headers will show both the requested exposure and the actual exposure (see *Appendix A: headers*)

Abort: Aborts an exposure/readout. This button will cause the exposure to finish immediately, but the image will not be readout -data will be lost-

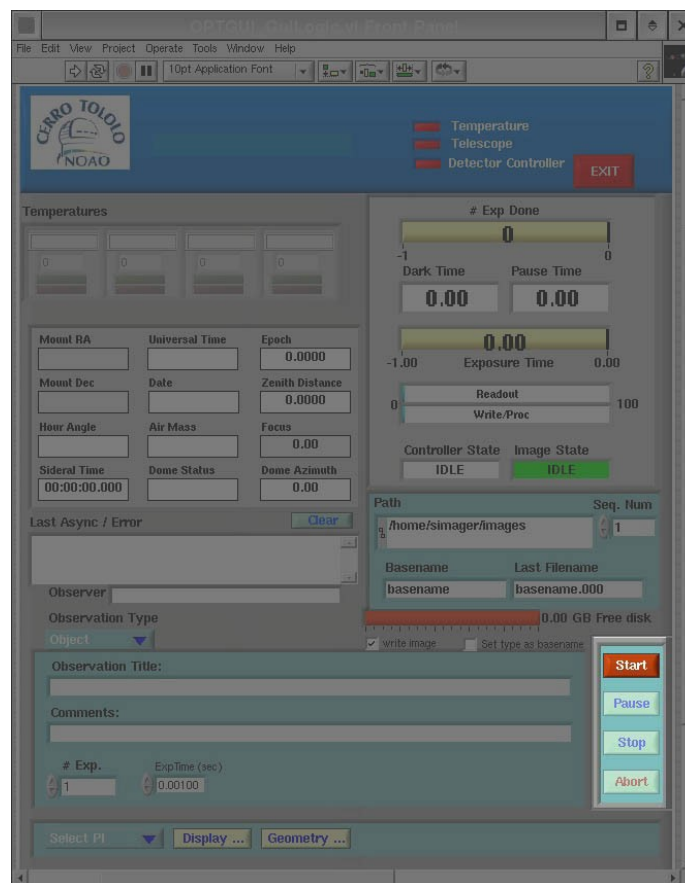


Figure 2.7: Exposure control

2.7 MAIN_GUI > PlugIns (write)

This area allows to access any Plug-In that may be available. A **Plug-In** is a sub-gui that allows to perform an specific task. It is an expansion to the functions of the main gui.

New Plug-Ins can be added at any time. The “**Select PI**” drop down menu will show any available Plug-In. The buttons “**Display ...**” and “**Geometry ...**” call some generic and useful Plug-Ins, described next
The currently available Plug-Ins are described in th following chapter

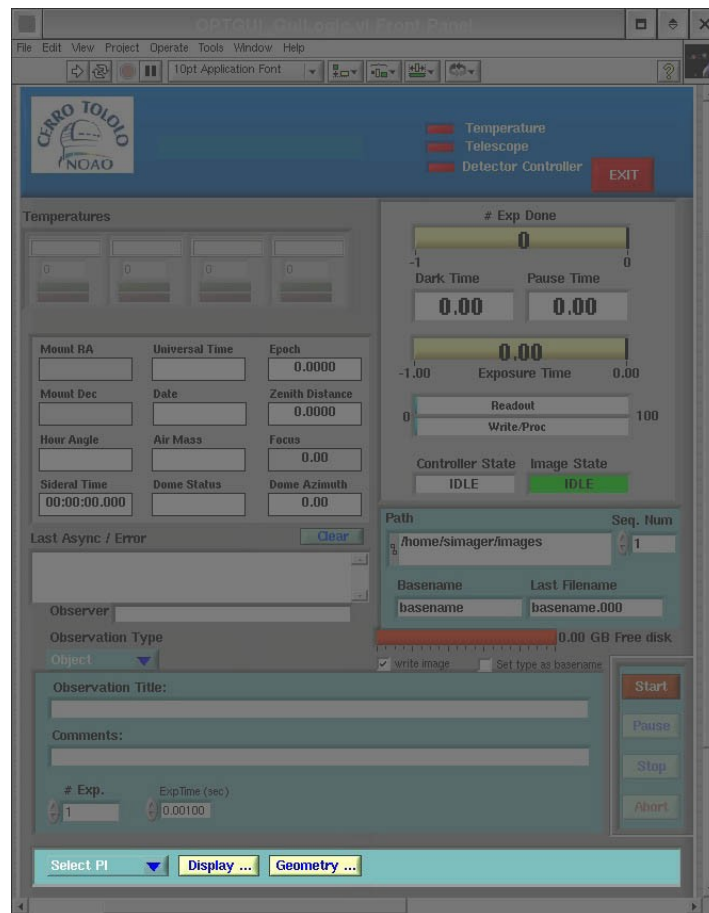


Figure 2.8: Plug-Ins drop-down

Chapter 3: Plug-Ins Reference

The Plug-Ins are “expansions” to the main GUI functionality. They are auxiliary guis that are specific to some specific tasks. The advantage of using Plug-Ins is that the main GUI does not get crowded with functionality that may not be used all the time; it allows to keep the main GUI as simple as possible. Also. It makes very easy to add functionality afterwards without affecting the main GUI. It provides modularity and the capability of natural expansion and compatibility between systems (you can have a common main gui for several different instruments).

3.1 Plug-Ins > DISPLAY

The **DISPLAY** Plug-In provides an interface to start and setup the real time display (**RTD**). The **RTD** is a graphic display that shows the last acquired image automatically. The **DISPLAY** feature can currently use **ds9** or **ximtool** as the actual displays. This can be set on the configuration files (see ECH60S-2.0 on system maintenance). To start the **DISPLAY** Plug-In the user can press the “**Display ...**” button at the bottom of the main GUI (see 2.7), or press in the **Plug-Ins** drop-down menu and select **DISPLAY** from the list

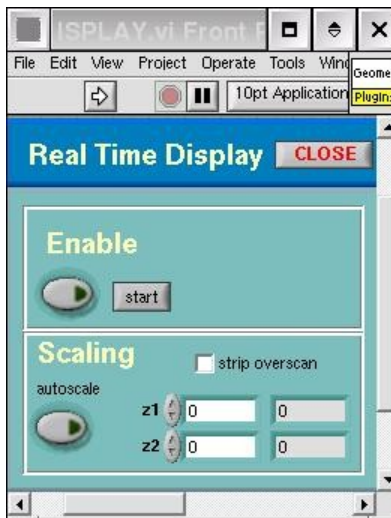


Figure 3.1: Real Time Display Plug-In

Enable:

The round button with the green LED is to enable/disable the RTD operation.

The “**start**” button allows to start the actual display device (**ds9** or **ximtool** as configured)

Scaling:

autoscale: if set, the RTD module will scale the gray levels as best as possible to have a good contrast. In automatic mode it will always represent all the pixel values in the image. If set, the **z1/z2** controls will be disabled (grayed out)

If **autoscale** is not set, then the **z1/z2** controls will be available (enabled). There the user can enter the minimum (**z1**) and maximum (**z2**) pixel value to be represented. This allows a better contrast in some desired area only. The indicators right to the **z1** and **z2** controls indicates the current **z1/z2** values (either manual or the last automatic ones if **autoscale** is selected)

strip_overscan: if this box is checked, the display will strip the overscan after the image has been read. Note that it does so for display purposes only; the actual image is not touched. This stripping is done

by invoking a python script (“autodisplay.py”) that uses pyraf to re-display the image. The script's description can be found in document ***ECH60-2.X*** (Software architecture)

3.2 Plug-Ins > GEOMETRY

This Plug-In allow to change image geometry/speed variables (parameters that can make the image smaller, or have a different aspect ratio, etc).

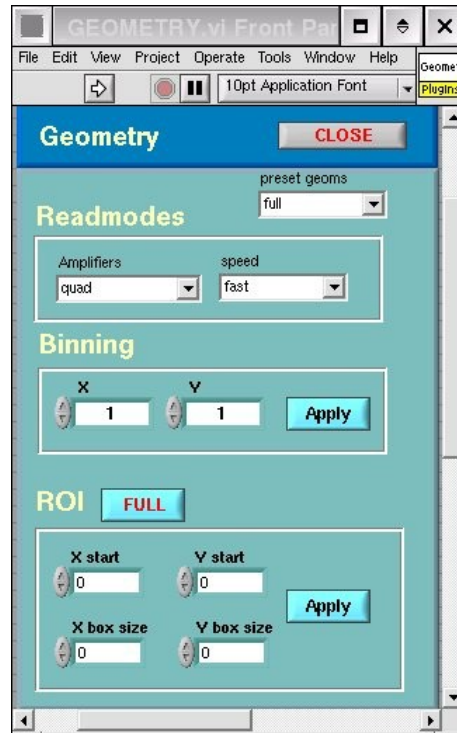


Figure 3.2: Geometry Plug-In

Readmodes:

Amplifiers: select what amplifer to use.

upperboth: use both upper-right and upper-left amplifiers.

Speed:select the readout speed to use: fast (noisier) or slow.

Binning

x: binning in columns

y: binning in rows

Apply: apply binning changes

ROI:

Select region of interest (sub-window) to read

Enable: enables the selected ROI

Disable: go back to full-frame mode

X start: starting column of interest

Y start: starting column of interest

X box size: number of columns of region, starting at “X start”

Y box size: number of rows of region, starting at “Y start”

Apply: apply the specified ROI

All the dimensions are in unbinned data pixels (so if binning is selected the ROI is still specified as for binning 1)

In the upper right corner there is a drop-down menu called “*pre-set geoms*”. By selecting something there, it will change the geometry (ROI, Binning, speed, etc) to some pre-set value. This allows speed in changing between common geometries. How to make a new geometry available for the drop-down is described in document *ECH60S-6.X* (panview configuration). When release the software, there were two available options: “full” (full frame, 1x1 binning) and “400_1600” (full in cols, 400 to 1600 in Y, 1x1 binning)

3.3 Plug-Ins > LAMPS

This Plug-In allow a direct and clear manipulation of the comparison lamps. This allows the user a “manual” operation of the comparison lamps, similar to the manual manipulation done through the actual, physical switches on the console

The logic of the lamps control is described in detail on document *ECH60S-5-X* on comparison lamps automation, and also on document *ECH60S-2.X* on software architecture, so here we will just describe its main aspects. The actual control signal for the each lamps is an **OR** between the “manual” switch and the software command. The motor control signal is also an **OR**, but now between the lamps control signals. So, if any lamp is “ON” the motor will be “ON”, and any lamp will be “ON” if either its manual switch is “ON” or the software commands an “ON”.

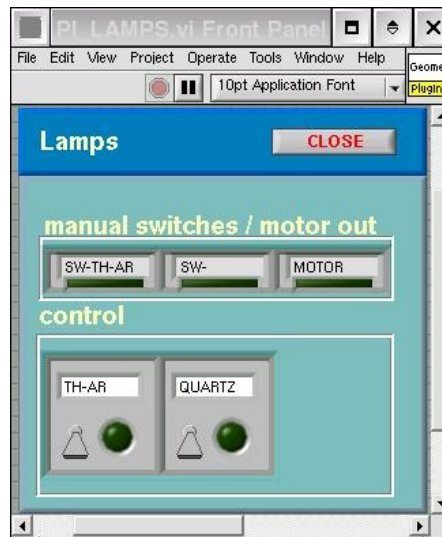


Figure 3.3: Lamps Plug-In

The upper boxes (**manual switches / motor out**) show the status of each manual switch (name of the switch on top, LED shows on/off condition). The last of the upper boxes shows the status of the motor (labeled as **MOTOR**)

The lower boxes show the name of the lamp, and the LED shows the status of the actual output control signal. The switch in the box indicates the software switch (status of software command)

If the user turns ON the software switch, the LED on the same box should also turn “on”, showing that the actual control signal is ON (this LED is actually showing a real feedback on the control signal).

The software provides some interlocks to avoid having two lamps turned “on” at the same time (if the user turns “on” a lamp when the other is “on”, he/she/ will see in the gui that the old lamp gets turned off before the new one is turned on). These interlocks are described in some details in document *ECH60S-2.X* on software architecture.

The software also provides a timer to set a maximum “on” time for the lamp. This time is a configurable parameter, but it is not available to the user from the GUI, but as a configuration file to be edited manually (see also *ECH60S-2.X*)

In general the user does not need to use this Plug-In, as the lamp will be automatically handled (including the lamp selection) when the observation type “*calibration*” is selected (see 2.5, under “*observation types -> calibration*”).

3.4 Plug-Ins > IODCEL

This Plug-In allow a direct and clear manipulation of the iodine cell position. This allows the user a “manual” operation of the comparison lamps.

The logic of the iodine cell operation is described in detail on document *ECH60HF-7.X* on RTD / Data I/O box description, and also on document *ECH60S-2.X* on software architecture, so here we will just describe its main aspects. The actual control signal for the motor is an **OR** between the “manual” switch and the software command.



Figure 3.4: Iodine cell Plug-In

The upper box (**manual switch**) show the status of the manual switch (name of the switch on top, LED shows on/off condition).

The lower box show the iodine cell position. The LED shows the status of the actual output control signal. The switch in the box indicates the software switch (status of software command)

If the user turns ON (IN) the software switch, the LED on the same box should also turn “on”, showing that the actual control signal is ON (IN) (this LED is actually showing a real feedback on the control signal).

Appendix A: Header

This appendix present some information in the standard header and its meaning. To consult how to edit/add headers dynamically please refer to document *ECH60S-3.0* on scripting.

Note that we describe only the keywords which are particular to this instrument, not the standard fits keywords (the keys that belong to the fits standard, as NAXIS, BZERO, etc), which are always present in any fits file

Effective integration time: shutter open time if object, dark time if dark

EXPTIME = 12. / Exposure time in secs

date/time of creation of the fits file

DATE = '2009-12-22T11:21:51' / file creation date (YYYY-MM-DDThh:mm:ss UT)

Title of the image, as filled in the “Observation title” main GUI control

OBJECT = 'none ' / Name of object observed

Name of the observer, as filled in the “Observer” main GUI control

OBSERVER= ' ' / observer

Type of observation, as chosen in the “observation type” drop down menu in the main GUI

IMAGETYP= 'object ' / Type of picture (object, dark, comp, etc)

Image binning in x (columns) and y (rows) as selected in the Geometry Plug-In

CCDSUM = '1 1 ' / On chip summation (X, Y)

UT at which the shutter was open. This is set internal, and has an accuracy of ~ 2 ms

UTSHUT = '2009-12-22T11:21:31.683' / UT of shutter open

Number and order of amplifiers in the y (vertical) and x (horizontal) directions. Set automatically depending on the amplifiers in use, selected on the “readmodes” drop-down menu in the Geometry Plug-In

NAMPSYX = '1 2 ' / Num amps in y and x (eg. '2 2=quad')

AMPLIST = '21 22 ' / Readout order in y,x

Image geometry. This are standard NOAO WCS keywords. They describe the are of the image that corresponds to amplifier (ASEC), Bias (BSEC), CCD (CSEC), Data (DSEC), trim (TSEC). Besides that, we have RSEC (Roi, or sub-window section) and AB / AD (amplifier bias and data, which corresponds to

those sections inside the amplifier). The suffix (21 or 22) corresponds to the physical amplifier (21: upper left, 22: upper right, 11: lower left, 12: lower right)

ASEC21 = '[1:1081,1:2048]' / amplifier section 21(UL) detID 1

BSEC21 = '[1032:1081,1:2048]' / bias section 21(UL) detID 1

CSEC21 = '[1:1031,1:2048]' / ccd section 21(UL) detID 1

DSEC21 = '[1:1031,1:2048]' / data section 21(UL) detID 1

TSEC21 = '[8:1031,1:2048]' / trim section 21(UL) detID 1

RSEC21 = '[1:1024,1:2048]' / roi section 21(UL) detID 1

ABSEC21 = '[1032:1081,1:2048]' / overscan inside amp

ADSEC21 = '[1:1024,1:2048]' / detector section only

ASEC22 = '[1082:2162,1:2048]' / amplifier section 22(UR) detID 1

BSEC22 = '[1082:1131,1:2048]' / bias section 22(UR) detID 1

CSEC22 = '[1082:2112,1:2048]' / ccd section 22(UR) detID 1

DSEC22 = '[1132:2162,1:2048]' / data section 22(UR) detID 1

TSEC22 = '[1132:2155,1:2048]' / trim section 22(UR) detID 1

RSEC22 = '[1082:2105,1:2048]' / roi section 22(UR) detID 1

ABSEC22 = '[1:50,1:2048]' / overscan inside amp

ADSEC22 = '[1025:2048,1:2048]' / detector section only

ROISEC00= '[1:2048,1:2048]' / roi section

Detector and Focal plane array (FPA) ID, and detector characteristics (Read out noise (RON) and gain in e-/adu) as measured in the lab / telescope

DETECTOR= 'Site424_6' / Detector Identifier

GAIN21 = 0.88 / Gain for Amp A (electrons/adu)

RON21 = 6.6 / Read Noise for Amp 21 (electrons)

GAIN22 = 0.8 / Gain for Amp B (electrons/adu)

RON22 = 7.2 / Read Noise for Amp 22 (electrons)

FPA = 'ECHELLE_CT60' / focal plan array

Observatory and telescope information

OBSERVAT= 'CTIO ' / Origin of data

TELESCOP= 'CTIO 1.5 meter telescope' / Specific system

Information of the TCS, coming directly from the TCS and appended. The comment fields are self-explanatory

DATE-OBS= '2009-12-22T11:21:42.0' / date of observation start

UT = '11:21:42.0' / UT of TCS coords

RA = '12:45:28' / ra

DEC = '64°36' / dec

EPOCH = 2000. / epoch

ALT = '260.0 ' / altitude

HA = '-00:30:16.1' / ha

ST = '05:04:01.1' / sidereal time

ZD = '12.5 ' / zenith distance

AIRMASS = '1.015 ' / airmass

CCD, NECK, Coude room and Echelle body temperatures, in Celsius degrees

CCDTEMP = -120 / ccd temperature

NECKTEMP = -165 / dewar neck temperature

TEMPAIR1 = 17.12 /coude room temperature

TEMECH = 15.20 /echelle body temperature

Comparison lamp status. States what lamp was in use (“none” if no lamps was ON)

COMPLAMP = ' QUARTZ' / comparison lamp

Iodine cell position in the light path (IN or OUT)

IODCELL = 'OUT ' /iodine cell position

Requested exposure time. This value usually is the same as “exptime”, however, it may differ is the image was “stopped” during integration. In that case, “exptime” will show the actual (effective) integration, while this filed will show the originally requested exposure time

REXPTIME= 12. / requested exposure time

Dark current time. This shows how long the detector was acquiring dark current. This is usually the same as “exptime”, but it may differ if the exposure was paused / resumed. In this case “exptime” will show a smaller value than the dark time (the detector was generating dark current all the time, including the paused time)

DARKTIME= 12. / dark current time in secs

This is engineering information that may be useful: software/firmware versions, and hardware configuration: slot and card name, firmware version and electronic ID

DHEINF = 'MNSN fermi hardware' / controller info

DHEFIRM = '/home/observer/panview/fpas/_echelle/config/DETECTOR/auto_mcb.ucd'

SLOT00 = 'MCB 0xDA3AE16 4.400000' / dhe board: <type> <serial> <firmware>

SLOT02 = 'CLK 0x0 4.200000' / dhe board: <type> <serial> <firmware>

SLOT05 = 'CCD 0xDA33CEE 4.110000' / dhe board: <type> <serial> <firmware>

Identification of the panview focal plane name and the Data Handling system, in this case the same panview

PANID = '_echelle' / PAN identification

DHSID = '_echelle' / DHS identification

Comment as filled in the “Comments” control in the main Gui

COMMENT none

Shows the pixel time and speed mode used

PIXTIME = 8.9/ pixel time

SPEEDMOD='normal' / speed mode used